A Comprehensive Analysis of Adversarial Attacks against Spam Filters

Esra Hotoğlu[a], Sevil Sen[a,*], Burcu Can[b]

[a] *WISE Lab., Department of Computer Engineering, Hacettepe University, Ankara, Turkey*
[b] *Department of Computing Science and Mathematics, University of Stirling, Stirling, UK*

Abstract

Deep learning has revolutionized email filtering, which is critical to protect users from cyber threats such as spam, malware, and phishing. However, the increasing sophistication of adversarial attacks poses a significant challenge to the effectiveness of these filters. This study investigates the impact of adversarial attacks on deep learning-based spam detection systems using real-world datasets. Six prominent deep learning models are evaluated on these datasets, analyzing attacks at the word, character sentence, and AI-generated paragraph-levels. Novel scoring functions, including spam weights and attention weights, are introduced to improve attack effectiveness. This comprehensive analysis sheds light on the vulnerabilities of spam filters and contributes to efforts to improve their security against evolving adversarial threats.

Keywords: email security, spam detection, adversarial learning, natural language processing, deep learning

1. Introduction

Deep learning has seen significant advancements in the field of natural language processing (NLP), particularly in tasks such as email filtering. Email filters play a critical role in detecting spam, viruses, and malware, serving as the first line of defence against cyber-attacks. Cybercriminals often target

---

*Corresponding author
Email address: ssen@cs.hacettepe.edu.tr (Sevil Sen)

personal and valuable data, such as cryptocurrency wallets and email credentials, so robust email filtering is essential to protect users from potential security breaches.

According to Cybersecurity Report of Trend Micro [55], the increase in malware detections and Business Email Compromises (BECs) from 2022 to 2023 indicates increasingly sophisticated methods. In addition to subtle tactics to trick users into clicking on malicious links, spam campaigns remain effective and can bypass email filters. In addition, the FBI's 2023 Internet Crime Report [16] indicates a significant increase in the frequency and financial impact of online fraud. Phishing scams, in which cybercriminals impersonate legitimate companies to obtain personal and financial data via email, were the most common type of reported fraud. Business Email Compromise (BEC) has been identified as one of the most expensive types of fraud, with 21,489 complaints resulting in $2.9 billion in losses.

Google, Outlook and Yahoo use different methods for spam filtering to filter out unwanted messages. Google Mail (Gmail) classifies emails as spam, promotional, or social based on their content. Google's data centers use hundreds of rules to determine whether an email is valid or spam. Outlook, on the other hand, automatically filters spam, and users can easily create custom rules to further categorize emails. The Yahoo email provider also has its own algorithms in order to detect spams [12]. Gmail, used by millions, has advanced security features to block 99.9% of spam, phishing, and malware, and uses TensorFlow to improve spam email detection capabilities [32]. In addition, Yahoo filters are reported to be 99.9% successful at catching spam, malware and phishing emails [62].

Despite their effectiveness, email spam filters can be manipulated, particularly through adversarial learning techniques. Adversarial learning, a prominent method in machine learning, involves deliberately introducing small changes to the input data to fool a model, causing it to misclassify or make incorrect predictions. This phenomenon has become a significant problem, particularly in the field of deep learning, where even state-of-the-art classifiers can be vulnerable to such attacks. Adversarial attacks on machine learning models typically fall into two broad categories: white-box attacks and black-box attacks. In white-box attacks, the adversary has complete access to the model, including its structure, parameters, and training data. In contrast, in black-box attacks, the adversary has limited or no access to the inner workings of the model, relying instead on external observations to construct adversarial inputs.

Moreover, AI-generated emails pose a significant threat to email spam filters. Through the use of advanced deep learning algorithms and natural language processing (NLP), AI can create content that closely mimics human writing. This means that AI-generated spam emails can appear highly convincing and may bypass traditional spam filters designed to catch more obvious threats. As a result, malicious actors can exploit this technology to produce sophisticated spam messages that deceive recipients. These deceptive emails can manipulate into revealing sensitive information, clicking on harmful links, or engaging in other actions that compromise their security. This evolving challenge underscores the need for more advanced and adaptive security measures to detect and mitigate AI-driven threats.

Recently, significant efforts have been made to develop deep learning-based systems, primarily utilized for natural language processing tasks, given the discrete nature of text. Nevertheless, adapting similar attacks to the NLP domain has proven challenging due to this inherent characteristic. Therefore, there is a growing body of research that focuses on adversarial examples in text-based systems. This study is one of them by putting emphasis on spam filters. It investigates the impact of deliberate perturbations of input vectors on various advanced spam filters using three prominent real-world text datasets commonly used in spam email research: SpamAssassin [2], Enron Spam [39], and TREC 2007 [11]. It thoroughly analyzes the generation of black-box attacks that target spam filters at multiple levels, including character, word, and sentence levels.

These attacks are designed to generate adversarial examples that are capable of bypassing various spam detection filters. These filters are based on a variety of deep learning architectures tailored to various tasks and data structures: a Long Short Term Memory (LSTM) model for sequential data, a Convolutional Neural Networks (CNN) model for spatial features, a Feed Forward Neural Network with Dense layers for general tasks, an attention model for selective focus, and a transformer model for efficient sequence processing, and distilBERT model which is a pre-trained model for efficient and compact language understanding. In addition, novel scoring functions are introduced to generate more effective adversarial attacks. Performance evaluation of the proposed scoring functions involves subjecting them to rigorous testing against various black-box attack scenarios and comparison with existing scoring methods used in spam filtering systems. Additionally, AI-generated paragraph-level attack that includes spam and non-spam emails are also tested on these filters to assess their effectiveness. Specifically, it ex-

amines how these AI-generated emails interact with and potentially bypass existing spam filters.

This comprehensive analysis aims to contribute to ongoing efforts to improve the security and resilience of spam detection filters in response to evolving adversarial threats. In summary, this study entails analyzing a range of adversarial attacks designed to undermine spam email detection systems. The primary contributions of the study are highlighted as follows:

- Six prominent deep learning-based spam detection systems are developed and thoroughly evaluated against adversarial attacks using three real-world datasets. Unlike many studies in the literature [18, 30, 58, 45, 67, 28, 36, 61, 41, 66], which often limit their evaluations to a single dataset, our study provides a more comprehensive assessment. Furthermore, most studies in the literature focus on traditional models and typically only examine one or two deep learning algorithms. This study tests the six prominent deep learning-based models against adversarial attacks.

- Adversarial attacks against spam filters are comprehensively analyzed at four levels: word-level, character-level, sentence-level, and AI-generated paragraph-level. While previous studies have predominantly concentrated on word-level attacks only, with only a single study [18] addressing into character-level attacks, our research addresses all potential attacks at each level. Sentence-level attacks are investigated for the first time against NLP-based systems in this study. Therefore, this comprehensive analysis ensures a thorough examination of the effectiveness and vulnerabilities of spam filters, leading to a more robust understanding of their resilience in real-world scenarios.

- This study introduces novel scoring functions, namely spam weights and attention weights scoring functions to identify the most effective words in order to create more effective attacks in the field of spam detection. Their effectiveness are demonstrated in the results.

- This study also investigates the impact of AI-generated paragraph-level spam and non-spam emails on spam detection systems. This investigation provides insights into the challenges faced by spam detection technologies and helps identify potential areas for improvement.

The paper is organized as follows: Section 2 provides a literature review on attacks implemented in spam filters. Section 3 outlines the datasets used in the study, details the preprocessing steps, and describes the deep learning models used for spam detection. It also introduces the adversarial attacks and the associated scoring functions. Sections 4 and 5 present and discuss the experimental results. Finally, Section 6 provides concluding remarks on the work.

2. Related Work

Email spam detection is crucial for protecting users from unwanted messages, phishing attempts, malware distribution, and other security threats. It involves analyzing incoming email messages to distinguish between ham (non-spam) and spam content. Various algorithms and techniques are commonly employed for spam detection. Unlike traditional classifiers, deep learning models offer the ability to learn abstract features. Deep learning techniques, such as Long Short-Term Memory Networks (LSTMs), Convolutional Neural Networks (CNNs), attention mechanisms, and transformer architectures, are particularly effective for feature extraction and classification in spam detection tasks, especially when dealing with complex data such as images or large text corpora. These algorithms are often combined with feature engineering techniques, pre-processing steps, and evaluation metrics to construct robust and efficient spam detection systems.

Adversarial attacks are techniques used to deceive or manipulate machine learning models through the input of carefully crafted data. These attacks target weaknesses in the model's decision-making processes, often leading to misclassifications or other unwanted outcomes. Adversarial attacks can manifest in various ways, such as by adding barely detectable noise to input data, altering pixels in images, or changing features in text.

In the realm of adversarial attacks two primary strategies stand out: white-box attacks and black-box attacks. In a white-box attack scenario, the adversary has full insight into the target model, including its structure, parameters, loss functions, activation functions, as well as access to both input and output data. This level of access enables the attacker to meticulously craft adversarial perturbations tailored to exploit vulnerabilities in the model. By approximating the worst-case scenario for a given model and input, white-box attacks pose a significant threat, often achieving high success rates in compromising model integrity and performance. This adversary

strategy is particularly potent in controlled environments where the attacker has unrestricted access to the model's inner workings [65, 25].

Conversely, black-box attacks operate under the assumption that the attacker lacks detailed knowledge, such as its architecture and parameters. However, black-box attackers still have access to the model's input and output interfaces, allowing them to query the model and observe its responses. In this scenario, attackers often rely on heuristic methods to generate adversarial examples, leveraging insights gained from probing the model's behavior through input-output interactions. In real-world scenarios, black-box attacks are often the most feasible and realistic approach. Despite the inherent limitations imposed by the lack of model transparency, black-box attacks remain a viable threat vector, highlighting the importance of developing robust defense mechanisms against adversarial manipulation [65, 25].

While the general classifications of attacks provide a foundational framework, it's essential to recognize that for Natural Language Processing (NLP) tasks, attack strategies and types differ due to the unique characteristics of text data compared to image or audio data. Textual content presents distinct challenges and opportunities for adversarial manipulation, leading to specialized classifications of attack techniques tailored to NLP domains. In the context of NLP, attacks can be classified according to the level of granularity of modifications made to the text data. Specifically, three primary types of attack techniques emerge: character-level attacks, word-level attacks and sentence-level attacks. Each type targets different linguistic components within the text, allowing adversaries to exploit vulnerabilities in NLP systems effectively [24, 20].

Character-level attacks involve the manipulation of individual characters within the text, such as inserting, removing, substituting, or rearranging characters to induce misclassification or alter semantic meaning. These attacks often capitalize on the subtle nuances of language to evade detection and compromise model integrity. Word-level attacks operate at the level of words, where adversaries modify or replace entire words within the text to deceive NLP models. By strategically choosing words or phrases, attackers can distort the intended message or inject malicious content without significantly altering the overall structure of the text. Sentence-level attacks focus on the manipulation of entire sentences or segments of text to influence model predictions or behavior. Adversaries may introduce grammatical errors, syntactic anomalies, or semantic inconsistencies to disrupt model performance or mislead downstream processing [24, 20].

In recent years, the exploration of deep learning algorithms for spam detection has gained attention in the field of adversarial learning. As a result, there has been a significant amount of research on spam detection using adversarial machine learning. However, previous studies have primarily focused on the good word attack, which modifies spam emails by inserting or appending words that indicate a legitimate email.

In [67], a counter-attack strategy using multiple instance learning are proposed to defend good word attacks on statistical email spam filters. This study demonstrates that multiple-instance learners outperform standard single-instance learners, including logistic regression, support vector machine, and the commonly used Naive Bayes model, in withstanding good word attacks. Jorgensen et al. [28], a similar multiple instance learning counter-attack strategy is presented to combat adversarial good word attacks on statistical spam filters. This involves transforming each email into a collection of multiple segments and applying multiple sample logistic regression to these collections. The introduced classifier is claimed to be more robust against good word attacks compared to commonly used methods in the spam filtering domain.

Furthermore, in [36], the performance of Naive Bayes and maximum entropy spam filters is examined in response to active and passive good word attacks. The study determines the effectiveness of a word by averaging the weights of all the words in each filter. The results suggest that adding a relatively small number of easily identifiable words can allow around 50% of currently blocked spam to pass through a spam filter. Another study [61] highlights the ease of implementing some attacks and their varying effectiveness, noting that while some methods like the common word attack can be more efficient than others, they often only succeed against specific filters. It suggests that future efforts should include examining different spam evasion techniques, understanding vulnerabilities in various filters, and exploring the impact of retraining filters.

A novel attack method is proposed in [10], involving the alteration of textual data by using NLP based on the results of constructed adversarial samples designed to deliberately modify the features representing an email. Various natural language feature extraction approaches, such as TF-IDF, Word2vec, and Doc2vec, are compared against white-box attacks. By conducting experiments on various datasets and utilizing various classification models such as Support Vector Machine (SVM), decision tree, logistic regression, Multi-layer Perceptron (MLP), and ensemble classifiers. The proposed

method is demonstrated to be capable of crafting adversarial examples in the text domain, significantly degrading the accuracy of spam detection systems. In [30], researchers explore the impact of adversarial scenarios on machine learning-based methods such as email spam filters. Three invasive techniques are tested using NLP along with a Bayesian model: synonym replacement, raw word injection, and spam word spacing, demonstrating their effectiveness in deceiving machine learning models.

In addition, Ozkan et al. [42] investigates how adversarial attacks affect conventional spam detection systems that use machine learning models such as Naïve Bayes (NB) and Support Vector Machines (SVM). Four types of attacks, tokenization, obfuscation, word addition, and word substitution, were tested to evaluate their effects on spam filter accuracy. Results show that while tokenization and obfuscation have limited effects, word addition and word substitution attacks significantly reduce filter accuracy, potentially rendering the filters ineffective. Also, in [58], two innovative text generation methods are introduced to enhance the effectiveness of attacks by leveraging adversarial perturbations produced through adversarial example generation algorithms. One method approximates TF-IDF values in the adversarial examples, while the other incorporates special words into the original emails. The study employs the Projected Gradient Descent (PGD) algorithm and evaluates its performance across various machine learning classification models, including SVM, K-Nearest Neighbors (KNN), decision trees (DT), and logistic regression (LR), under both white-box and black-box attack scenarios. In another study [45], a defense mechanism is proposed to mitigate the impact of these ideal poisoning attacks on linear classifiers, based on outlier detection. However, since the attack strategies do not consider detectability constraints, the resulting counterexamples are notably different from real data points. The findings indicate that less aggressive attacks, like label flipping, can be challenging to detect with these defense mechanisms, as the generated attack points closely resemble real data points.

Moreover, Nelson et al. [41] illustrate how the SpamBayes spam filter can be effectively neutralized with minimal knowledge of the system and restricted access to the training data. While they present successful defenses such as the RONI defense, that blocks messages from dictionary attacks completely, and the dynamic threshold defense, which mitigates the impact of dictionary attacks, they highlight the persistent challenge of defending against focused attacks due to the attacker's additional knowledge. On the other hand, Gu et al. [66] proposed marginal attack methods to deceive a

Naive Bayesian spam filter by adding sensitive words to sentences. Three strategies for selecting sensitive words are proposed, resulting in significant reductions in the filter's detection accuracy. These attacks significantly reduce the filter's accuracy, even with just one word added. The study also showed that the generated adversarial examples can disrupt other traditional filters such as logistic regression, decision tree, and linear support vector machine.

The previous studies have mainly focused on word-level attacks, but there are also a few studies investigating the character-level attacks, also using deep learning algorithms. For instance, in [18], a new algorithm named DeepWordBug is introduced. This algorithm efficiently generates minor text perturbations at character-level within a black-box environment, compelling deep learning classifiers to misclassify text inputs. Their evaluation is carried out on two real text datasets containing Enron spam emails and IMDB movie reviews, and includes the development of scoring strategies to identify the most critical words for modification, leading to incorrect predictions. Remarkably, their results illustrate a significant reduction in classification accuracy, decreasing from 99% to 40% on the Enron Spam dataset and from 87% to 26% on the IMDB dataset. Furthermore, Boucher et al. [8] analyzes a broad range of adversarial examples across various domains, beyond spam filters, capable of attacking text-based models at the character level in a black-box setting. They employ perturbations to manipulate the output of various NLP-based systems. The study demonstrates that attacks involving invisible characters, homoglyphs, reordering, or deletion could substantially impair the performance of vulnerable models.

Zhang et al. [65] present the first comprehensive research on generating textual adversarial examples on deep neural networks. They reviewed recent research efforts and research studies that produced textual adversarial examples on DNNs. They also comprehensively collected, summarized and analyzed these studies and ensured that the article was self-contained by covering all relevant information. Finally, they have provided an excellent resource for researchers to understand the challenges, techniques, and key topics in this field. In another research [25], various forms of adversarial attacks on machine learning in the context of network security are examined and two novel classification frameworks are introduced for detecting and mitigating such attacks. First, the attacks are classified based on the classification of network security applications. Then, they are classified according to the problem domain and classification model. Finally, an in-depth analy-

Table 1: Analysis of Previous Studies

Previous Studies	Dataset	Methodology	Scoring Functions	Attacks
Zhou et al. [67]	TREC 2006	Naive Bayes	-	Good Word Attack
Jorgensen et al. [28]	TREC 2006	LR, Naive Bayes, SVM	-	Good Word Attack
Lowd and Meek [36]	Hotmail Feedback Loop	Naive Bayes, Maxent	-	Good Word Attack
Wittel et al. [61]	SpamAssassin	SpamBayes	-	Dictionary Word Attack, Common Word Attack
Cheng et al. [10]	Ling, Tutorial, Enron Spam	SVM, DT, LR, MLP	-	PGD attack
Kuchipudi et al. [30]	SMS Spam	Naive Bayes	-	Synonym Replacement, Ham Word Injection, Spam Word Spacing
Chenranc et al. [58]	Enron Spam	SVM, KNN, DT, LR	-	PGD Attack, Adding Special Words
Paudice et al. [45]	Spambase	Linear Classifier	-	Poisoning Attacks
Nelson et al. [41]	TREC 2005	SpamBayes	-	Dictionary Attack, Focused Attack
Ozkan et al. [42]	SpamAssassin, Enron Spam	SVM, NB	-	Tokenization, Obfuscation, Word Addition, Word Substitution
Gu et al. [66]	SMS Spam	Naive Bayes, SVM, DT, LR	-	Word Addition
Gao et al. [18]	Enron Spam	LSTM, CNN	Replace-1 Score, Temporal Head Score, Temporal Tail Score, Combined Score	Substitution, Deletion Chars, Insertion, Swap Chars
Our Study	SpamAssassin, Enron Spam, TREC 2007	LSTM, CNN, Dense, Attention, Transformer	Replace-1 Score, Spam Weights, Attention Weights	Out of Vocab, Deleting Words, Synonym Replacement, Antonym Replacement, Insertion & Deletion Chars, Replacement & Swapping Chars, Add Ham, Spam Sentence, Ham-Spam Sentences

sis of diverse defense strategies aimed at protecting machine learning-based network security applications from adversarial attacks are analyzed.

On the other hand, AI-generated content has increasingly influenced deep learning models for spam detection in recent years. A study [46] explores how Large Language Models (LLMs) like GPT-3.5, Bard, and BingAI generate datasets for password strength prediction. The research highlights the potential and limitations of LLMs for data creation and encourages further work to enhance their capabilities and data diversity. Also, artificial intelligence is widely used to produce images, as discussed in [44], which evaluates six AI-generated-image detection methods across 23 datasets, including images from GANs, diffusion models, and transformers, highlighting the widespread use of artificial intelligence in image generation. At the same time, artificial intelligence plays a crucial role in both generating and detecting spam emails, as discussed in [7, 15, 56]. These sources examine various AI-based spam detection models, assess their performance on multiple datasets, and emphasize the growing importance of AI in enhancing email security and filtering systems.

The related studies on adversarial attacks against spam filters are summarized in Table 1. As shown, there are only a few studies focusing on

adversarial attacks against spam filters that utilize deep learning algorithms, despite the prevalence of such algorithms in many modern spam filters. On the contrary, our study centers on the exploration of spam filters employing various deep learning techniques. Moreover, while previous studies have generally concentrated on word-level attacks only, our study comprehensively analyzes possible attacks at the character, word, and sentence levels. Additionally, by examining such attacks in black-box scenarios, we aim to simulate real-world scenarios more accurately. Last but not least, we propose different scoring functions to select words for these attacks, thereby enhancing their effectiveness. As these attacks play a crucial role in assessing the robustness of models against adversarial attacks, they can be integrated into the training of deep learning models to improve spam classifiers. To sum up, this study provides a comprehensive analysis of adversarial attacks against modern spam filters, filling a notable gap in existing research.

3. Methodology

This study targets the bypassing of several neural network architectures by adversarial attacks. These models include Long Short-Term Memory (LSTM) networks, a specialized version of Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), a Feed Forward Neural Network with Dense layers, an LSTM model with a single attention layer, a transformer model and a pre-trained model called distilBERT. The primary objective is to illustrate the impact and extent of various attack types on various deep learning spam filters.

In black-box attacks, adversaries can only modify the test data without access to the filters. This study uses three well-known spam datasets to train spam filters and generate adversarial attacks. First, preprocessing, tokenization and sequencing steps are applied to all datasets. Subsequently, spam filters based on LSTM, CNN, LSTM with attention and the transformer are developed using the Keras and TensorFlow libraries. The distilBERT is utilized through Hugging Face's Transformers library. Finally, different types of adversarial attacks at different levels (character, word, sentence, and AI-generated paragraph-level) with different scoring functions are executed against these DL-based spam filters and a thorough evaluation is performed.

3.1. Datasets and Preprocessing

The three datasets used in this study, namely SpamAssassin [2], Enron Spam [39] and TREC2007 [11], are summarized below:

- `SpamAssassin`: The dataset is obtained from the Apache Public Datasets and the Apache SpamAssassin Projects, which maintain a repository of archived emails. This dataset consists of 2,400 spam and 6,954 ham (i.e. not spam) emails [2].

- `Enron Spam`: The dataset is collected from the mailboxes of Enron employees, in the cleaned-up form provided, which includes only ham messages, and from four different sources for spam messages[40]. It contains 17,171 spam emails and 16,545 ham emails [39].

- `TREC2007`: The TREC (Text Retrieval Conference) 2007 Public Corpus Dataset was collected through tasks aimed at classifying email messages, with variations in the amount and frequency of feedback received by the system. It contains 50,199 spam emails and 25,220 ham emails [11].

Table 2: Distribution of Datasets

Dataset	Spam Emails		Ham Emails	
	Train Set	Test Set	Train Set	Test Set
SpamAssassin	1920	480	5563	1391
Enron Spam	13,737	3434	13,236	3309
TREC2007	40,159	10,040	20,176	5044

These corpora were chosen because of their widespread use in spam-related studies. Therefore, the use of these datasets will allow an easy comparison between our results and existing studies. 80% of the data is used for training and 20% for testing. The distribution of ham and spam in both training and testing datasets is given in Table 2.

A number of preprocessing steps have been implemented to clean up the data and reduce the input size. Firstly, punctuation, numbers, hyperlinks, and stop words such as "the", "a", "an", "in" are removed. Additionally,

12

all text is converted to lowercase. Word stemming and lemmatization pre-processing techniques are also used. Both methods aim to simplify words to their basic forms. These preprocessing steps reduce the computational cost and have no negative impact on the classification results.

Once the text is cleaned, it is converted into a numerical representation so that it can be used as input to the model. First, it is tokenized using the Keras tokenizer, which splits sentences into words and encodes them into integers. Next, each sentence is represented by sequences of numbers. Finally, padding is applied to ensure a uniform length for each sequence.

3.2. Methods

Deep learning algorithms are used for spam detection because they offer many advantages over traditional methods for dealing with the complexities of data. Spam emails are becoming increasingly sophisticated, often imitating legitimate communications or using new techniques to evade detection. Spam detection requires understanding intricate patterns in email text, such as unusual phrasing, grammar, or subtle cues that indicate spam. Neural architectures such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks are able to process sequential data, capturing long-term dependencies and patterns in sentences, which helps in identifying subtle and disguised spam content. This is crucial because spam messages often evolve to mimic regular email content. In addition, deep learning models are trained on large datasets and can handle diverse inputs, such as text, images, or hyperlinks within emails. Models like Convolutional Neural Networks (CNNs) can process image-based spam, and transformer-based models, such as BERT or GPT, can analyze the context of entire email bodies. This versatility allows spam filters to adapt to different formats of spam, whether it's text, attachments, or multimedia content.

On the other hand, transformer-based models such as BERT and GPT bring a deeper level of understanding by considering the context in which words occur and the cross-relationships between words, helping to learn deeper relationships beyond semantics. This is particularly important for spam detection because spammers often craft their messages to seem legitimate, using context-specific language. Deep learning models can distinguish subtle differences in how certain words are used, allowing them to recognize even cleverly disguised spam. Deep learning models can automatically learn relevant features from raw email data, reducing the need for manual

feature engineering. CNNs, for instance, are excellent at extracting key features such as word patterns, that may indicate spam, while the attention mechanism can highlight important parts of an email that are more likely to indicate spam. This allows for a more efficient and accurate classification process. Moreover, emails contain a wide range of information, from subject lines to embedded links, multimedia content, and metadata. Deep learning models are capable of processing and analyzing meaningful insights, patterns or information from data with a large number of features and dimensions, efficiently. They prioritize important features while ignoring irrelevant ones, making the detection process more effective.

Hence, we employ six different classifiers based on the following deep learning architectures: Long Short-Term Memory Networks (LSTM), Convolutional Neural Networks (CNN), a fully connected neural network (dense network), an LSTM with an attention layer, a transformer, and distilBERT which is a lightweight, faster, and smaller version of the transformer-based BERT model in our analysis. While distilBERT is a pre-trained model, other classifiers are trained in this study.

Recurrent neural networks (RNNs) are able to capture sequential dependencies by incorporating loops into their structure. However, traditional RNNs faced challenges with backpropagation, which were addressed by Hochreiter and Schmidhuber [21] through the development of Long Short-Term Memory (LSTM) architectures. LSTMs have become one of the most favoured methods for text-based tasks. Many recent studies [14, 33, 31, 63] explore the effectiveness of LSTMs in various applications. Similarly, in the field of spam detection, studies [26, 54, 60] have used LSTMs and achieved high accuracies.

Convolutional Neural Networks (CNNs) are network architectures originally developed for image processing. They typically consist of convolution layers, pooling layers, and fully connected layers. Recent studies have demonstrated that CNNs are also effective for word-level text classification [29]. Several studies have used CNN filters to generate and evaluate adversarial text examples [14, 31, 63, 50, 4]. In addition, CNNs are widely used in spam detection and have shown promising results [60, 48, 23, 47].

One of the latest advancements in deep learning is the integration of a mechanism known as attention [3]. This mechanism aims to identify the relationship between inputs and expected outputs, giving greater importance to relevant inputs. It has already been used for different tasks such as sentiment analysis [59, 9]. In attention mechanism, a context vector is shared between

14

the input and the output. Attention weights indicate which words are useful for generating the desired output. The attention method, commonly used in the field of natural language processing, has found extensive application in spam detection studies [47, 38, 64, 22, 52], providing robust approaches to detecting spam emails.

The transformer architecture, proposed by Vaswani et al. [57], is an encoder-decoder model. This innovative design has gained popularity due to its parallelizability, scalability, and ability to capture long-term dependencies in sequential data without using recurrent connections as in RNNs. Comprising encoder and decoder components, the transformer architecture is structured around a self-attention mechanism that learns the importance of different parts of a sequence by attending to itself. This attention mechanism is run through several times in parallel, which is called multi-head attention. Its outstanding effectiveness in NLP tasks [19, 17] has established it as a cornerstone in the field. There have also been notable studies in spam detection [35, 49, 27], where its ability to detect complex patterns in text data has been crucial in efficiently reducing spam emails.

A pre-trained model in machine learning, particularly in natural language processing (NLP) and computer vision, refers to a model that has already been trained on a large dataset and is subsequently used as a starting point for training on a specific task. Sanh et al. [51] demonstrated that smaller language models pre-trained with knowledge distillation can achieve similar performance on many downstream tasks. The distilBERT, an optimized version of the BERT (Bidirectional Encoder Representations from Transformers) model, is designed to be more compact and efficient while retaining much of BERT's performance. It is also utilized across a variety of natural language processing (NLP) tasks [53, 6]. Notable studies [43, 13, 34] on spam detection have shown that it is effective in providing high accuracy, improving performance, and optimizing resource utilization.

An attempt has been made to use systems similar to those examined in the previous studies to facilitate comparisons with them. Extensive testing was carried out before the final model parameters for each model were determined. This was accomplished by fine-tuning the models for the spam detection task using RandomizedSearchCV [5]. It is a hyperparameter tuning technique in machine learning used to optimize the performance of a model, and it is part of the scikit-learn library. Instead of exhaustively searching over all possible combinations of hyperparameters (as in GridSearchCV), RandomizedSearchCV samples a fixed number of hyperparameter combina-

tions from a specified distribution or range. This makes it more efficient, especially when dealing with a large number of hyperparameters. Parameters such as the number of input units for LSTM layers, the number of units for dense layers, number of filters, kernel size for convolutional layers, dropout rate, activation function, optimizer, learning rate, and loss function have been selected. As false positive rates have more serious implications than false negatives, a trade-off between lower false positive rates and accuracy values was considered. The architectural details of each model are shown in Table 3. In the table, the "None" values in the Input Shape and Output Shape columns refer to dynamic or flexible dimensions in the model architecture. In deep learning models, the batch size is usually not specified when defining the model architecture, allowing the model to handle inputs of varying batch sizes during training or inference.

3.3. Adversarial Attacks

In the context of machine learning, an adversarial attack is the deliberate manipulation of input data to cause errors or produce incorrect outputs from a machine learning model. Adversarial attacks exploit vulnerabilities in the model and expose weaknesses in the decision-making process. In the context of spam filtering, the selection of keywords within a spam message is critical to the execution of effective attacks. A black box setting is employed for all the attacks. In this setting, the attacker can receive feedback on the spam weight of a given message but does not have access to other model parameters. This paper presents several scoring functions designed to identify the most influential words, such as the replace one score , spam weights, and attention weights. The spam weights scoring function is introduced for the first time in this study, while the replace one score is an existing method in the literature [18]. The attention weights is a method that has been used before but has not been applied in spam detection. We used all the three methods for comparison purposes. The calculation details of these functions are as follows:

- `Replace One Score (R1S)`: Each token in the document is replaced with an unknown token (UNK) and a loss is calculated, which is used to select which tokens to replace with [18]. In this study, this function is computed using an LSTM filter to calculate the loss of each word as shown in equation 1. Where F is the model's prediction score, x_i is the word to be removed from the input vector and x_i' is unknown token.

16

Table 3: Architectural Details of the Models

Model	Layers		
	Layer Name	Input Shape	Output Shape
LSTM	Input Layer	None, 350	None, 350
	Embedding Layer	None, 350	None, 350, 50
	LSTM Layer	None, 350, 50	None, 32
	Dense Layer	None, 32	None, 1
Dense	Input Layer	None, 350	None, 350
	Embedding Layer	None, 350	None, 350, 50
	Flatten Layer	None, 350, 50	None, 17500
	Dense Layer	None, 17500	None, 416
	Dropout Layer	None, 416	None, 416
	Dense Layer	None, 416	None, 416
	Dense Layer	None, 416	None, 1
CNN	Input Layer	None, 350	None, 350
	Embedding Layer	None, 350	None, 350, 50
	Conv1D	None, 350, 50	None, 348, 128
	GlobalMaxpooling1D	None, 348, 128	None, 128
	Dense Layer	None, 128	None, 192
	Dropout Layer	None, 192	None, 192
	Dense Layer	None, 192	None, 1
Attention	Input Layer	None, 350	None, 350
	Embedding Layer	None, 350	None, 350, 50
	LSTM Layer	None, 350, 50	None, 350, 32
	Attention Layer	None, 350, 32	None, 32
	Dense Layer	None, 32	None, 1
Transformer	Input Layer	None, 350	None, 350
	Token And Positional Embedding	None, 350	None, 350, 256
	Transformer Layer	None, 350, 256	None, 350, 256
	GlobalAveragePooling1D	None, 350, 256	None, 256
	Dropout	None, 256	None, 256
	Dense	None, 256	None, 1

Although the authors reported significant drops using this method, it has a notable drawback: obtaining feedback from the filter for each token increases runtime. This is impractical in real-world scenarios where attackers have limited system access and aim to avoid detection by minimizing the number of queries to the system.

$$R1S(x_i) = F(x_1, x_2, ..., x_{i-1}, x_i, ..., x_n) - F(x_1, x_2, ..., x_{i-1}, x'_i, ..., x_n)$$
(1)

- Spam Weights (SW): This is a variation of the Replace One score. Calculation can be seen in equation 2. It is calculated spam weights (SW) for each word using LSTM filter predictions F. It is chosen LSTM for

these tasks because it is possible to get results for variable length input vectors with this setting. Each word index is treated as a vector of size one and the results are given in terms of spam probability. Based on these results, it is created a dictionary used to conduct the attacks. Therefore, the system only had to be queried once for each word. Using this filter our task is to get the spam weight for a given word w_i given message $x \in X$ where $x = \{w_1, w_2 \ldots, w_n\}$ and X is our input vector space.

$$SW(x_i) = F(x_i) \tag{2}$$

- **Attention Weights (AW)**: Attention weights are returned in addition to the context vector obtained from the attention layer, and are used to determine the importance of these vectors. The attention weights are used to compute an alignment score between all hidden states and the target state, and then to obtain a probability distribution using softmax on this score [37]. Where h_t is the target state and $\overline{h}_s$ are all the source states as shown in equation 3. Attention score for state h_t is generally calculated using softmax on this score as shown in equation 4. While attention adds additional value to sequence to sequence systems with encoder decoder architecture its use is not limited by this. In this study, individual attention weights are used to find the most important words.

$$score(h_t, h_s) = \begin{cases} h_t^T \overline{h}_s & \text{dot} \\ h_t^T W_a \overline{h}_s & \text{general} \\ v_a^T tanh(W_a[h_t; \overline{h}_s]) & \text{concat} \end{cases} \tag{3}$$

$$a_t = softmax(score(h_t, \overline{h}_s)) \tag{4}$$

Using the scoring functions, the attacks are applied to words with high spam weights in a spam message and with low spam weights in a ham message. There are a variety of different adversarial attacks that can be used against deep learning systems and these attacks operate at different levels including character, word and sentence level [24, 20]. The classifiers are subjected to attacks on words obtained from the scoring functions mentioned above.

3.3.1. Character-Level Attacks

These attacks make changes such as replacing individual characters with other characters, adding them to the word, swapping them with neighboring characters, or removing them from the word [24, 20]. The amount of character modification in these attacks is a crucial factor to consider. Therefore, the following attacks are performed by selecting different percentages of characters, ranging from 10% to 50%, and random indices to modify characters within words using the specified scoring functions:

- `Swapping`: Rearranging characters of a word with their neighbors to create noise.

- `Deletion`: Removing random characters from a word to change its surface form and possibly its meaning.

- `Insertion`: Inserting random characters in a word to change its surface form and possibly its meaning.

- `Replacement`: Replacing individual characters with random letters to create misspelled words.

3.3.2. Word-Level Attacks

Word-level attacks corrupt the whole word rather than just a few characters. In these attacks, synonyms and antonyms of the words in the text are changed or removed completely, resulting in misspellings [24, 20]. As well as the number of characters, the number of words to be attacked is also important. Thus, the following attacks are performed by selecting words from different percentages of the corpus, ranging from 1% to 5% using the specified scoring functions:

- Out of Vocabulary (OOV): Replacing selected words with an unknown token.

- Word Deletion: Removing selected words to change the overall structure and semantics of a given text.

- Synonym Replacement: Substituting selected words with synonyms to change the structure of a sentence.

- Antonym Replacement: Substituting selected words with antonyms to change the meaning of a sentence.

3.3.3. Sentence-Level Attacks

These attacks can be thought of as modifying a group of words together in a sentence [20]. Such attacks often add new sentences as adversarial examples. No other approach has yet investigated the attacks at this level against NLP-based systems [24]. The following adding sentence attacks are performed with sentences selected using the total spam weights of words in the emails:

- Adding a ham sentence: Insertion of a non-spam sentence to a spam email.

- Adding a spam sentence: Insertion of a spam sentence into a non-spam (ham) email.

- Adding ham-spam sentences: Insertion of both a ham sentence to a spam email and a spam sentence to a ham email.

3.3.4. Paragraph-Level Attacks

Paragraph-level attacks have been generated using AI in the form of spam and non-spam emails. Generating these emails using the GPT-3.5 large language model (LLM) involves leveraging its advanced natural language processing capabilities. By employing carefully crafted base prompts and iterative prompt engineering, researchers can direct GPT-3.5 to produce spam or ham emails with varying degrees of complexity and relevance. The model's ability to understand and mimic human language allows it to generate realistic email content that can simulate a wide range of scenarios, from legitimate communications to deceptive spam. This process includes generating emails that resemble real-world examples, which are then refined and preprocessed to ensure quality and variety. The resulting dataset serves as a valuable resource for evaluating spam detection systems, helping to assess their performance in distinguishing between legitimate and malicious content. However, balancing creativity with accuracy and addressing the model's tendency to copy familiar patterns is challenging. This balance ensures that the generated emails contribute effectively to the development and testing of robust spam filtering solutions. This generated dataset is used as a test data for previously trained deep learning models and undergoes preprocessing steps before being utilized in the testing phase. Sample spam and non-spam emails generated by artificial intelligence are shown in Table 5 and Table 4.

Table 4: Examples of AI-Generated Non-Spam Email

Subject: Reminder: Team Building Event on September 15th
Hi Team,
I hope everyone's having a great week! I just wanted to send a quick reminder about our upcoming team-building event happening on September 15th at Greenfield Park. This will be a great opportunity for us to unwind, get to know each other outside the office, and have some fun with the activities we've got planned.
We'll be starting at 10 AM, and there will be a variety of games and challenges, followed by a picnic lunch around 1 PM. Please dress comfortably and don't forget to bring your enthusiasm – it's going to be a lot of fun! If anyone has dietary restrictions or specific preferences for lunch, please let me know by September 10th so we can accommodate those.
Additionally, if anyone needs help with transportation to the venue, feel free to reach out to me or Brian. We're more than happy to arrange carpooling if needed.
I'm really looking forward to seeing everyone there, and I'm confident it will be a great time for us to connect as a team.
Best regards,
Jessica
Subject: Feedback Request on Weekly Project Meeting
Hi Sarah,
I hope you're doing well. I wanted to take a moment to thank you for your valuable contributions during our project meeting on Tuesday. Your insights on improving the user interface were especially helpful, and I believe they will greatly impact the overall user experience. It's always great to have your perspective in these discussions.
That being said, I've been thinking about some of the points we touched on briefly, particularly the timeline for integrating the new features into the existing system. We didn't have much time to go into detail, but I'd really appreciate your thoughts on how we can streamline the process without compromising on quality.
If you're available, would you be open to having a quick chat this week to explore this further? I think it would be beneficial for us to align our ideas before the next phase begins.
Looking forward to hearing your thoughts. Let me know when you'd be free for a quick follow-up!
Best regards,
Michael

Non-spam emails are generated by AI when requested on topics such as friendship, complaints, apologies, formal letters of appreciation, thank-you notes, cover letters, client introductions, proposal submissions, requests for help, raises, feedback, quotation emails, job rejections or acceptances, and project status updates.

When we requested the generation of spam emails and asked, 'Can you generate spam emails?' the response was, 'I cannot generate spam emails. My purpose is to provide helpful, ethical, and constructive assistance while adhering to responsible communication standards.' However, when asked for educational purposes, it generates emails resembling spam. An example prompt could be:

Write a spam email for educational purposes. The email should have the following features:

- Sound like it's offering a too-good-to-be-true offer.

- Try to entice the reader to click on a suspicious link or provide sensitive information.

Remember, this is for a project to understand spam, and not for actual use!

4. Evaluation and Results

First, a comprehensive evaluation is performed on the selected classifiers using unperturbed test samples, providing insight into their raw performance without any attacks. The evaluation is then extended to assess the resilience of these classifiers after being subjected to adversarial changes. This analysis aims to elucidate the robustness and effectiveness of the models in dealing with perturbed or manipulated input data, shedding light on their real-world applicability and vulnerability to adversarial attacks.

4.1. Baseline Performance of the Classifiers

All models are applied to the SpamAssassin [2], Enron Spam[39], and TREC2007 [11] datasets, yielding successful results. Detailed results of different spam detection filters are presented in Table 6 for the Enron Spam dataset, without any adversarial attacks. The results of SpamAssassin and TREC2007 datasets are given in the supplementary material [1]. Upon examination of the results, it is observed that all models achieved high success in spam detection. When these models were compared, it was noticed that the accuracy rates were close to each other. However, the accuracy of the transformer and distilBERT models is slightly lower than the other models in all datasets. Additionally, model performance is lower on the Enron Spam dataset [39] compared to other datasets.

Evaluating spam filters involves assessing their ability to accurately classify emails as either spam or non-spam (ham). Several metrics are commonly used to measure the effectiveness of spam filters: true positive (TP), true negative (TN), false positive (FP), false negative (FN), accuracy, precision, recall, and f1-score. Accuracy is the proportion of correctly classified emails (both spam and ham) out of the total number of emails evaluated. Precision

22

Table 5: Examples of AI-Generated Spam Email

Subject: 50% OFF on all products at MegaMart - Limited Time Offer!

Dear Valued Customer,

We are pleased to inform you that MegaMart, the leading online shopping destination, is offering an exclusive deal that will make your jaw drop! For a limited time only, we are giving a whopping 50% discount on all our products. Yes, you read that right, 50% OFF on everything!

We know you deserve the best, and that's why we are bringing you this unbeatable offer. From electronics to fashion, home decor to kitchen appliances, we have it all at unbelievable prices. And as our valued customer, we want to make sure you don't miss out on this golden opportunity. But that's not all, we are also giving away a free gift with every purchase. That's right, a FREE gift! And it's not just any gift, it's the latest iPhone 11 Pro Max or the Samsung Galaxy S20 - the choice is yours!

All you have to do is click on the link below and enter your personal information to claim your discount and free gift. Don't worry, our website is 100% secure, and we guarantee the protection of your data.

Hurry up, this offer won't last long, and we don't want you to regret missing out on this once in a lifetime opportunity. So, what are you waiting for? Start filling up your cart and get ready to be amazed by the discounts and free gifts! Thank you for choosing MegaMart, where you can always shop smart.

Sincerely,

The MegaMart Team

Subject: Congratulations User, You've Won a Free Vacation!

Dear User,

Congratulations! You have been selected as a lucky winner of our exclusive limited time offer for a free vacation to the luxurious Maldives. We at Paradise Travels are excited to offer you this once in a lifetime opportunity to experience the ultimate tropical paradise.

But wait, it gets even better! Not only will you get a free stay at a 5-star resort, but you will also have access to our private yacht for a day and a personal chef to cater to all your dining needs. And all of this is completely free for you!

All we ask in return is for you to click on the link below and fill out a short survey. This survey will help us improve our services and ensure that your vacation is nothing less than perfect. Don't worry, the survey is completely safe and secure, and your personal information will be kept confidential.

But hurry, this offer is only valid for a limited time and we wouldn't want you to miss out on this amazing opportunity. So don't wait any longer, click on the link and claim your free vacation now!

Link: www.paradisetravels.com/freesurvey

We look forward to having you as our guest and making your dream vacation a reality.

Best regards,

The Paradise Travels Team

quantifies the accuracy of spam classifications, representing the percentage of emails correctly labeled as spam among all those flagged as spam, whereas recall measures the effectiveness of the filter in detecting actual spam emails, calculating the percentage of true spam emails that are correctly identified.

The F1-score represents a balanced measure of the classifier's performance, calculated as the harmonic mean of precision and recall. The false positive rate measures the proportion of non-spam emails that are incorrectly classified as spam out of all actual non-spam emails, while the false negative rate measures the proportion of spam emails that are incorrectly classified as non-spam out of all actual spam emails.

Table 6: Attack-Free Results for the Enron Spam Dataset

Model	TP	TN	FP	FN	Accuracy	Precision	Recall	F1 Score
LSTM	3208	3439	39	58	98.56	98.55	98.34	98.61
Dense	3210	3426	37	71	98.40	98.38	97.97	98.45
CNN	3207	3463	40	34	98.90	98.90	99.03	98.94
Attention	3213	3469	34	28	99.08	99.08	99.20	99.11
Transformer	3165	3388	82	109	97.17	97.15	96.88	97.26
DistilBERT	3195	3388	52	109	97.61	97.59	96.88	97.68

4.2. Performance of the Classifiers When Attacked

The attacks described in the previous section are evaluated when the classifiers are attacked using the same three datasets. Increased false negatives result in an increased number of spam emails bypassing the user's filters, while increased false positives result in the system misclassifying many ham emails as spam, potentially causing the user to miss important emails. Attacks are carried out using all scoring functions. The findings are presented in Table 8, 11, 9, 10.

4.2.1. Word-Level Attack Results

Choosing the number of words to change in a given text is crucial for word-level attacks. Tests were performed on the SpamAssassin dataset [2] using different filters to investigate the effect of changing the word count. Figure 1 shows the results of a word deletion attack on the dense filter. As shown in the figure, the percentage of deleted words correlates inversely with the accuracy (%).

Words are selected using predefined scoring functions to identify the most effective ones. The scoring functions are applied in a black-box setting, where they receive feedback on the spam weight and loss of a given message but do

Figure 1: Word Deletion Attack Results on the Dense Filter

not have access to other model parameters. Table 8 presents the results of attacks performed on the Enron Spam dataset [39], where 3% of the corpus was selected using the spam weights scoring function, based on predictions from the LSTM filter. A comparison of the models reveals notable differences. The LSTM model experienced the most significant accuracy drop compared to other filters. The attention filter, which incorporates an attention layer into the LSTM model, also showed a decrease in accuracy, though not as pronounced as the LSTM model itself. Interestingly, the pre-trained distilBERT model exhibited a significant drop in accuracy. This may be due to the challenges pre-trained models face in specialized domains, where differences in language patterns, vocabulary, and context limit their effectiveness. Meanwhile, the dense model proved to be less robust against attacks than the transformer and CNN models. This vulnerability can be attributed to its lack of convolution and pooling layers. According to the results of the attention weights and R1S scoring functions for the Enron Spam dataset [39], as shown in Table 9 and Table 10, the attention model appears to be more robust. The attention layer helps neural networks retain long sequences of data, enhancing their resilience to attacks.

When word-level attacks such as OOV and word deletion are applied to LSTM, attention, and distilBERT filters, there is a significant increase in false positives compared to false negatives when using the attention weights and R1S scoring functions. In contrast, CNN, dense, and transformer filters experience a significant increase in false negatives compared to false positives, as shown in Table 9 and Table 10 for the Enron Spam dataset [39]. How-

25

ever, synonym replacement and antonym replacement attacks do not lead to a significant reduction in performance when using these scoring functions. Since not all selected words have synonyms or antonyms, word changes remain minimal. On the other hand, when words are chosen using the spam weights scoring function for the Enron Spam dataset [39], all filters show a noticeable increase in false positives compared to false negatives, as shown in Table 8. This occurs because spam-related words are removed from spam emails, causing spam messages to be mislabeled as ham more frequently. As a result, classifier accuracy drops significantly in the presence of OOV and word deletion attacks. Since synonym and antonym replacement attacks modify only a few words without significantly altering sentence meaning, classifier performance remains largely unaffected. Overall, when comparing word-level attacks, filters perform worse against OOV attacks than other word-level modifications. This is because replacing a selected word with an UNK token—unrecognized by the model—disrupts classification more than simply deleting the word.

Moreover, the results for the R1S scoring function demonstrate similarity to those obtained with the attention weights scoring function for the Enron Spam dataset [39], and a slight decrease in accuracy is observed for word-level attacks, as seen in Table 10 and Table 9. These two scoring mechanisms give comparable results. However, word selection is faster with attention weights compared to R1S. This difference arises because attention weights are derived from the attention layer of the filter and typically take less than a minute to compute, whereas the R1S scoring function requires replacing all words in the corpus with UNK tokens, followed by loss calculation. Therefore, this process can take hours depending on the corpus length. The results indicate that when words are selected based on the spam weights scoring function, as shown in Table 8, there is a more significant decrease in the effectiveness of spam filters compared to the attention weights and R1S scoring functions for the Enron Spam dataset [39]. This scoring function quickly selects words by estimating spam percentages using an LSTM filter. Taking all factors into account, spam weights are more effective against spam filters than other scoring functions, both in speed and in reducing filter effectiveness.

4.2.2. Character-Level Attack Results

The number of characters is also a critical consideration for character-level attacks. Tests are conducted to assess the impact of varying the number of characters on the SpamAssassin dataset [2], as depicted in Figure 2.

Figure 2: Character Attack Results on the Dense Filter

Table 7: Results of Character Attacks Applied by Percentage

Attack	Word	10%	20%	30%	40%	50%
Swap letter	localhost	loclahost	lolcahost	loclhoast	loachslot	lcolashot
Delete character	localhost	localost	lcalost	lclhst	lahot	lclht
Insert character	localhost	locialhost	locualhomst	loscahblhost	lodcvallehost	lkoicaclfhost
Replace character	localhost	lmcalhost	localjvst	lswalhomt	lrhbltost	locvvqojt

The appearance of words when attacks are applied according to the character percentage is shown in the Table 7. Interestingly, it was observed that changing the number of characters in the attacks used by more than 30% did not significantly affect accuracy.

Table 8 shows the results of attacks in which 30% of the word length is selected for character-level manipulation using the spam weights scoring function for the Enron Spam dataset [39]. Notably, since spam weights are computed using an LSTM model, the most substantial decrease in accuracy occurs in the LSTM model for character-level attacks, similar to what is observed at the word level. Unexpectedly, the distilBERT model also experienced a significant drop in accuracy, mirroring the trend seen at the word level. Furthermore, Table 9 and Table 10 present the results of attacks using other scoring functions for the Enron Spam dataset [39]. These tables indicate that the R1S and attention weights functions yield similar results, though with a smaller reduction in accuracy. With these scoring functions, as with word-level attacks, the dense, transformer, and distilBERT filters show the most significant accuracy decrease, while the attention filter experiences the least decline. This divergence may be due to pre-trained models being

highly sensitive to noisy or adversarial inputs, where even minor changes in wording, punctuation, or spelling can confuse them and hinder their ability to generalize. Additionally, while powerful, the self-attention mechanism in transformers focuses on token relationships without fully grasping hierarchical structures such as syntax and semantics, sometimes leading to incorrect generalizations. In contrast, the attention model benefits from an attention layer, which significantly enhances its ability to understand and generate human-like language, as well as an LSTM layer, which effectively handles long-term dependencies in sequential data.

Based on the results, although the insert character attack led to a significant increase in false positives, there were almost no false negatives when using the spam weight scoring function for the Enron Spam dataset [39], as shown in Table 8. This outcome is due to the introduction of extra characters into words flagged as spam, causing spam emails to be misclassified as non-spam. Consequently, there is also a significant drop in accuracy across all baseline systems. For the attention weights and R1S scoring functions for the Enron Spam dataset [39], shown in Table 9 and Table 10, the insert character attack affected the performance of certain spam filters. Conversely, the delete character attack reduced the effectiveness of other filters, with very similar results. The reason insert and delete character attacks have a greater impact on spam filters is that they alter word lengths. Unlike swap letter and replace character attacks—where characters are swapped with their neighbors or randomly replaced, often resulting in words that resemble the original—character insertion and deletion attacks directly modify word size. Increasing or decreasing word size enhances the similarity to other words, further reducing the effectiveness of spam filters.

When evaluating the results based on the scoring functions for the Enron Spam dataset [39], spam weights are more effective at increasing false positives and decreasing accuracy in spam filters than R1S and attention weights, as shown in Table 8, Table 10, and Table 9. While the computation time for R1S increases with the input vector length, this is not an issue for the spam weights and attention weights scoring functions. Generating attack vectors for spam weights and attention weights took less than a minute, whereas R1S took hours since each word was processed individually. In summary, the spam weights scoring function outperformed both attention weights and R1S, consistent with the results of word-level attacks.

Table 8: Attack Results for the Enron Spam Dataset using Spam Weights

Model	Attack Level	Attack	TP	TN	FP	FN	Accuracy	Precision	Recall	F1 Score
LSTM	-	Attack Free	3208	3439	39	58	98.56	98.55	98.34	98.61
	Word-Level	Out Of Vocab	241	3494	3006	3	55.38	76.26	99.91	69.90
		Word Deletion	914	3493	2333	4	65.35	79.76	99.89	74.93
		Synonym Replacement	3019	3478	228	19	96.34	96.61	99.46	96.57
		Antonym Replacement	3206	3419	41	78	98.24	98.22	97.77	98.29
	Character-Level	Swap Letters	2253	3487	994	10	85.11	88.69	99.71	87.42
		Delete Character	1930	3487	1317	10	80.32	86.03	99.71	84.01
		Insert Character	852	3493	2395	4	64.43	79.43	99.89	74.44
		Replace Character	2268	3487	979	10	85.34	88.82	99.71	87.58
Dense	-	Attack Free	3210	3426	37	71	98.40	98.38	97.97	98.45
	Word-Level	Out Of Vocab	1976	3492	1271	5	81.08	86.53	99.86	84.55
		Word Deletion	1872	3495	1375	2	79.58	85.83	99.94	83.54
		Synonym Replacement	3048	3482	199	15	96.83	97.05	99.57	97.02
		Antonym Replacement	3203	3424	44	73	98.27	98.25	97.91	98.32
	Character-Level	Swap Letters	2571	3488	676	9	89.84	91.71	99.74	91.06
		Delete Character	2480	3488	767	9	88.49	90.81	99.74	89.99
		Insert Character	1858	3494	1389	3	79.36	85.70	99.91	83.39
		Replace Character	2581	3490	666	7	90.02	91.85	99.80	91.21
CNN	-	Attack Free	3207	3463	40	34	98.90	98.90	99.03	98.94
	Word-Level	Out Of Vocab	2508	3482	739	15	88.82	90.95	99.57	90.23
		Word Deletion	1750	3482	1497	15	77.58	84.54	99.57	82.16
		Synonym Replacement	3074	3477	173	20	97.14	97.31	99.43	97.30
		Antonym Replacement	3193	3462	54	35	98.68	98.69	99.00	98.73
	Character-Level	Swap Letters	2483	3478	764	19	88.39	90.62	99.46	89.88
		Delete Character	2409	3478	838	19	87.29	89.90	99.46	89.03
		Insert Character	1718	3482	1529	15	77.11	84.31	99.57	81.85
		Replace Character	2483	3478	764	19	88.39	90.62	99.46	89.88
Attention	-	Attack Free	3213	3469	34	28	99.08	99.08	99.20	99.11
	Word-Level	Out Of Vocab	1655	3492	1592	5	76.32	84.19	99.86	81.39
		Word Deletion	1935	3493	1312	4	80.49	86.24	99.89	84.15
		Synonym Replacement	3089	3481	158	16	97.42	97.57	99.54	97.56
		Antonym Replacement	3190	3453	57	44	98.50	98.51	98.74	98.56
	Character-Level	Swap Letters	2570	3486	677	11	89.80	91.66	99.69	91.02
		Delete Character	2440	3487	807	10	87.89	90.40	99.71	89.51
		Insert Character	1894	3493	1353	4	79.88	85.93	99.89	83.73
		Replace Character	2593	3486	654	11	90.14	91.89	99.69	91.29
Transformer	-	Attack Free	3165	3388	82	109	97.17	97.15	96.88	97.26
	Word-Level	Out Of Vocab	2347	3426	900	71	85.60	88.13	97.97	87.59
		Word Deletion	2349	3428	898	69	85.66	88.19	98.03	87.64
		Synonym Replacement	3074	3439	173	58	96.57	96.68	98.34	96.75
		Antonym Replacement	3142	3357	105	140	96.37	96.35	96.00	96.48
	Character-Level	Swap Letters	2659	3418	588	79	90.11	91.22	97.74	91.11
		Delete Character	2463	3418	784	79	87.20	89.12	97.74	88.79
		Insert Character	2330	3430	917	67	85.41	88.05	98.08	87.46
		Replace Character	2670	3421	577	76	90.32	91.40	97.83	91.29
DistilBERT	-	Attack Free	3085	3478	162	19	97.31	97.46	99.45	97.46
	Word-Level	Out Of Vocab	274	3461	2973	36	55.38	71.09	98.97	69.70
		Word Deletion	2006	3474	1241	23	81.26	86.27	99.34	84.61
		Synonym Replacement	2907	3420	340	77	93.82	94.19	97.80	94.25
		Antonym Replacement	3138	3414	109	83	97.15	97.16	97.63	97.26
	Character-Level	Swap Letters	1470	3431	1777	66	72.67	80.79	98.11	78.83
		Delete Character	1883	3441	1364	56	78.94	84.36	98.40	82.90
		Insert Character	868	3426	2379	71	63.67	75.73	97.97	73.66
		Replace Character	1727	3444	1520	53	76.68	83.20	98.48	81.41

Table 9: Attack Results for the Enron Spam Dataset using Attention Weights

Model	Attack Level	Attack	TP	TN	FP	FN	Accuracy	Precision	Recall	F1 Score
LSTM	-	Attack Free	3208	3439	39	58	98.56	98.55	98.34	98.61
	Word-Level	Out Of Vocab	2372	3473	875	24	86.67	89.44	99.31	88.54
		Word Deletion	3129	3355	118	142	96.14	96.13	95.94	96.27
		Synonym Replacement	3153	3411	94	86	97.33	97.33	97.54	97.43
		Antonym Replacement	3206	3419	41	78	98.24	98.22	97.77	98.29
	Character-Level	Swap Letters	3127	3423	120	74	97.12	97.15	97.88	97.24
		Delete Character	2989	3447	258	50	95.43	95.70	98.57	95.72
		Insert Character	3105	3360	142	137	95.86	95.86	96.08	96.01
		Replace Character	3138	3419	109	78	97.23	97.24	97.77	97.34
Dense	-	Attack Free	3210	3426	37	71	98.40	98.38	97.97	98.45
	Word-Level	Out Of Vocab	3229	2451	18	1046	84.22	87.40	70.09	82.17
		Word Deletion	3086	3184	161	313	92.97	92.99	91.05	93.07
		Synonym Replacement	3162	3339	85	158	96.40	96.38	95.48	96.49
		Antonym Replacement	3203	3424	44	73	98.27	98.25	97.91	98.32
	Character-Level	Swap Letters	3078	3329	169	168	95.00	95.00	95.20	95.18
		Delete Character	3072	3341	175	156	95.09	95.10	95.54	95.28
		Insert Character	3086	3172	161	325	92.79	92.82	90.71	92.88
		Replace Character	3092	3310	155	187	94.93	94.91	94.65	95.09
CNN	-	Attack Free	3207	3463	40	34	98.90	98.90	99.03	98.94
	Word-Level	Out Of Vocab	3218	2940	29	557	91.31	92.13	84.07	90.94
		Word Deletion	3069	3359	178	138	95.31	95.33	96.05	95.51
		Synonym Replacement	3132	3419	115	78	97.14	97.16	97.77	97.26
		Antonym Replacement	3193	3462	54	35	98.68	98.69	99.00	98.73
	Character-Level	Swap Letters	3037	3418	210	79	95.71	95.84	97.74	95.94
		Delete Character	2971	3433	276	64	94.96	95.22	98.17	95.28
		Insert Character	3059	3356	188	141	95.12	95.14	95.97	95.33
		Replace Character	3056	3406	191	91	95.82	95.90	97.40	96.02
Attention	-	Attack Free	3213	3469	34	28	99.08	99.08	99.20	99.11
	Word-Level	Out Of Vocab	2805	3399	442	98	91.99	92.56	97.20	92.64
		Word Deletion	3039	3408	208	89	95.60	95.70	97.45	95.82
		Synonym Replacement	3130	3421	117	76	97.14	97.16	97.83	97.26
		Antonym Replacement	3190	3453	57	44	98.50	98.51	98.74	98.56
	Character-Level	Swap Letters	3077	3447	170	50	96.74	96.85	98.57	96.91
		Delete Character	3006	3455	241	42	95.80	96.05	98.80	96.07
		Insert Character	3024	3404	223	93	95.31	95.43	97.34	95.56
		Replace Character	3086	3440	161	57	96.77	96.86	98.37	96.93
Transformer	-	Attack Free	3165	3388	82	109	97.17	97.15	96.88	97.26
	Word-Level	Out Of Vocab	2929	3152	318	345	90.17	90.15	90.13	90.48
		Word Deletion	2966	3159	281	338	90.82	90.80	90.33	91.08
		Synonym Replacement	3110	3354	137	143	95.85	95.84	95.91	95.99
		Antonym Replacement	3134	3355	113	142	96.22	96.20	95.94	96.34
	Character-Level	Swap Letters	2981	3219	266	278	91.93	91.92	92.05	92.21
		Delete Character	2689	3309	558	188	88.94	89.52	94.62	89.87
		Insert Character	3039	3129	208	368	91.46	91.48	89.48	91.57
		Replace Character	2968	3240	279	257	92.05	92.05	92.65	92.36
DistilBERT	-	Attack Free	3085	3478	162	19	97.31	97.46	99.45	97.46
	Word-Level	Out Of Vocab	2705	3380	542	117	90.23	91.02	96.65	91.12
		Word Deletion	2805	3399	442	98	91.99	92.56	97.20	92.64
		Synonym Replacement	3063	3335	184	162	94.87	94.87	95.37	95.07
		Antonym Replacement	3175	3379	72	118	97.18	97.17	96.63	97.27
	Character-Level	Swap Letters	2734	3360	513	137	90.36	90.99	96.08	91.18
		Delete Character	2865	3308	382	189	91.53	91.73	94.60	92.06
		Insert Character	2320	3349	927	148	84.06	86.16	95.77	86.17
		Replace Character	2737	3340	510	157	90.11	90.66	95.51	90.92

4.2.3. Sentence-Level Attack Results

The results obtained using the spam weights scoring function for sentence-level attacks on the Enron Spam dataset [39] are presented in Table 11. The

Table 10: Attack Results for Enron Spam Dataset with Replace One Score

Model	Attack Level	Attack	TP	TN	FP	FN	Accuracy	Precision	Recall	F1 Score
LSTM	-	Attack Free	3208	3439	39	58	98.56	98.55	98.34	98.61
	Word-Level	Out Of Vocab	2073	3489	1174	8	82.47	87.22	99.77	85.51
		Word Deletion	3123	3399	124	98	96.71	96.72	97.20	96.84
		Synonym Replacement	3019	3394	228	103	95.09	95.20	97.05	95.35
		Antonym Replacement	3206	3426	41	71	98.34	98.33	97.97	98.39
	Character-Level	Swap Letters	3103	3436	144	61	96.96	97.02	98.26	97.10
		Delete Character	2793	3454	454	43	92.63	93.43	98.77	93.29
		Insert Character	3079	3404	168	93	96.13	96.18	97.34	96.31
		Replace Character	3126	3422	121	75	97.09	97.12	97.86	97.22
Dense	-	Attack Free	3210	3426	37	71	98.40	98.38	97.97	98.45
	Word-Level	Out Of Vocab	3236	1814	11	1683	74.88	82.59	51.87	68.17
		Word Deletion	3018	3304	229	193	93.74	93.75	94.48	94.00
		Synonym Replacement	3019	3394	228	103	95.09	95.20	97.05	95.35
		Antonym Replacement	3195	3431	52	66	98.25	98.24	98.11	98.31
	Character-Level	Swap Letters	3022	3351	225	146	94.50	94.55	95.82	94.75
		Delete Character	3000	3359	247	138	94.29	94.38	96.05	94.58
		Insert Character	3005	3289	242	208	93.33	93.34	94.05	93.60
		Replace Character	3082	3346	165	151	95.31	95.31	95.68	95.49
CNN	-	Attack Free	3207	3463	40	34	98.90	98.90	99.03	98.94
	Word-Level	Out Of Vocab	3238	2236	9	1261	81.17	85.79	63.94	77.88
		Word Deletion	3077	3332	170	165	95.03	95.03	95.28	95.21
		Synonym Replacement	2997	3411	250	86	95.02	95.19	97.54	95.31
		Antonym Replacement	3191	3461	56	36	98.64	98.65	98.97	98.69
	Character-Level	Swap Letters	3034	3418	213	79	95.67	95.80	97.74	95.90
		Delete Character	2915	3438	332	59	94.20	94.60	98.31	94.62
		Insert Character	3049	3334	198	163	94.65	94.66	95.34	94.86
		Replace Character	3060	3408	187	89	95.91	95.99	97.45	96.11
Attention	-	Attack Free	3213	3469	34	28	99.08	99.08	99.20	99.11
	Word-Level	Out Of Vocab	2640	3462	607	35	90.48	91.89	99.00	91.51
		Word Deletion	3018	3446	229	51	95.85	96.05	98.54	96.10
		Synonym Replacement	2958	3420	289	77	94.57	94.84	97.80	94.92
		Antonym Replacement	3188	3458	59	39	98.55	98.56	98.88	98.60
	Character-Level	Swap Letters	3028	3450	219	47	96.06	96.25	98.66	96.29
		Delete Character	2853	3453	394	44	93.51	94.12	98.74	94.04
		Insert Character	2988	3448	259	49	95.43	95.70	98.60	95.72
		Replace Character	3045	3454	202	43	96.37	96.54	98.77	96.57
Transformer	-	Attack Free	3165	3388	82	109	97.17	97.15	96.88	97.26
	Word-Level	Out Of Vocab	2718	3254	529	243	88.55	88.90	93.05	89.40
		Word Deletion	2806	3301	441	196	90.55	90.84	94.40	91.20
		Synonym Replacement	3140	3343	107	154	96.13	96.11	95.60	96.24
		Antonym Replacement	3095	3421	152	76	96.62	96.67	97.83	96.78
	Character-Level	Swap Letters	2959	3139	288	358	90.42	90.40	89.76	90.67
		Delete Character	2527	3316	720	181	86.64	87.74	94.82	88.04
		Insert Character	2770	3275	477	222	89.64	89.93	93.65	90.36
		Replace Character	2889	3271	358	226	91.34	91.44	93.54	91.80
DistilBERT	-	Attack Free	3085	3478	162	19	97.31	97.46	99.45	97.46
	Word-Level	Out Of Vocab	2964	3130	283	367	90.36	90.35	89.51	90.59
		Word Deletion	2571	3361	676	136	87.96	89.12	96.11	89.22
		Synonym Replacement	2974	3323	273	174	93.37	93.44	95.02	93.70
		Antonym Replacement	3138	3414	109	83	97.15	97.16	97.63	97.26
	Character-Level	Swap Letters	2472	3315	775	182	85.81	87.10	94.80	87.39
		Delete Character	2750	3266	497	231	89.21	89.52	93.39	89.97
		Insert Character	2164	3318	1083	179	81.29	83.88	94.88	84.02
		Replace Character	2524	3326	723	171	86.74	87.90	95.11	88.15

attention model proves to be more robust, benefiting from the effectiveness of its attention layer in handling long sentences. However, the CNN, dense,

Table 11: Attack Results for the Enron Spam Dataset with Spam Weights at the Sentence-Level

Model	Attack Level	Attack	TP	TN	FP	FN	Accuracy	Precision	Recall	F1 Score
LSTM	-	Attack Free	3208	3439	39	58	98.56	98.55	98.34	98.61
	Sentences-Level	Add Ham Sentence	3157	3481	90	16	98.43	98.49	99.54	98.50
		Add Spam Sentence	3233	2594	14	903	86.40	88.82	74.18	84.98
		Add Ham-Spam Sentence	3157	2594	90	903	85.28	87.20	74.18	83.93
Dense	-	Attack Free	3210	3426	37	71	98.40	98.38	97.97	98.45
	Sentences-Level	Add Ham Sentence	2653	3496	594	1	91.18	92.72	99.97	92.16
		Add Spam Sentence	3246	1422	1	2075	69.22	80.47	40.66	57.80
		Add Ham-Spam Sentence	2653	1422	594	2075	60.42	63.32	40.66	51.59
CNN	-	Attack Free	3207	3463	40	34	98.90	98.90	99.03	98.94
	Sentence-Level	Add Ham Sentence	3247	755	0	2742	59.34	77.11	21.59	35.51
		Add Spam Sentence	3231	2936	16	561	91.44	92.33	83.96	91.05
		Add Ham-Spam Sentence	3144	755	103	2742	57.81	70.71	21.59	34.67
Attention	-	Attack Free	3213	3469	34	28	99.08	99.08	99.20	99.11
	Sentence-Level	Add Ham Sentence	2973	3492	274	5	95.86	96.28	99.86	96.16
		Add Spam Sentence	2973	3492	274	5	95.86	96.28	99.86	96.16
		Add Ham-Spam Sentence	2973	2936	274	561	87.62	87.79	83.96	87.55
Transformer	-	Attack Free	3165	3388	82	109	97.17	97.15	96.88	97.26
	Sentence-Level	Add Ham Sentence	3083	3409	164	88	96.26	96.32	97.48	96.44
		Add Spam Sentence	3243	536	4	2961	56.03	75.77	15.33	26.55
		Add Ham-Spam Sentence	3083	536	164	2961	53.66	63.79	15.33	25.54
DistilBERT	-	Attack Free	3085	3478	162	19	97.31	97.46	99.45	97.46
	Sentence-Level	Add Ham Sentence	2728	3490	519	7	92.20	93.40	99.80	92.99
		Add Spam Sentence	3234	1905	13	1592	76.20	83.17	54.48	70.36
		Add Ham-Spam Sentence	3246	1422	1	2075	69.22	80.47	40.66	57.80

transformer, and distilBERT models show less resilience to these attacks. A common characteristic among these models is the absence of an LSTM layer, unlike traditional neural networks, which can process data with time steps of varying lengths. All models exhibit high resilience to add-ham sentence attacks but are significantly affected by add-spam sentence and add ham-spam sentence attacks.

With an add-ham sentence attack, the results indicate a slight increase in false positives, suggesting that some spam emails are misclassified as ham, resulting in a minimal decrease in accuracy. On the other hand, an add-spam sentence attack leads to a notable increase in false negatives and a substantial decrease in accuracy. Finally, with an add-ham-spam sentence attack, there is a rise in both false negatives and false positives, along with a significant decrease in accuracy.

4.2.4. Paragraph-Level Attack Results

Paragraph-level attacks consisting of 500 spam and 500 non-spam emails were generated using AI. The email subjects were randomly assigned and covered various topics. These AI-generated attacks were then tested on pre-trained models that had been trained on the Enron Spam dataset [39]. The

results are presented in Table 12.

The performance of deep learning models varies significantly across metrics such as accuracy, precision, recall, and F1 score, largely influenced by their architectures and capabilities. DistilBERT consistently outperforms other models, particularly in precision, recall, and F1 score, due to its transformer-based architecture, which effectively captures long-range dependencies and contextual information. This advantage is especially critical for paragraph-level tasks, where relationships span across words and sentences. DistilBERT's ability to process entire sequences of words simultaneously, coupled with pre-training on extensive corpora, enables it to identify subtle patterns and achieve high accuracy in tasks like distinguishing spam from non-spam emails. Its capability to handle imbalanced datasets further enhances both precision and recall, contributing to an improved F1 score. Unlike older models such as LSTMs or CNNs, which rely on sequential or convolutional operations, DistilBERT processes text bidirectionally. This allows it to consider the context of a word both before and after its position in a sentence, leading to more accurate predictions. Pre-trained transformer models like DistilBERT generalize effectively across various natural language processing tasks, making them more robust compared to task-specific models such as LSTMs or dense networks.

On the other hand, the LSTM model follows with decent accuracy, but its recall scores are relatively low because, while it is designed to handle sequential data well, it struggles with long-term dependencies and tends to miss some positive cases (spam). This limitation likely contributes to its lower recall, as it cannot capture complex patterns as effectively as transformer-based models like distilBERT. Transformer and CNN perform similarly, with moderate accuracies. While both architectures are capable of identifying certain patterns in the data, their low recall and moderate precision suggest that they are not as effective at capturing the nuanced differences between spam and non-spam emails. However, the transformer model does not have the benefits of pre-training on large datasets. The CNN, which is commonly used for image recognition, may not be well-suited for text-based tasks, as it primarily focuses on local patterns rather than long-range dependencies. In addition, the dense and attention models show the poorest performance, with low accuracy, precision, recall, and F1 scores. The dense model, being a fully connected feed-forward network, lacks the ability to effectively handle sequential dependencies in text data. The attention model, despite its focus mechanism, likely underperformed due to the limited complexity of its archi-

Table 12: Attack Results for AI-Generated Dataset at the Paragraph-Level

Model	Attack Level	TP	TN	FP	FN	Accuracy	Precision	Recall	F1 Score
LSTM	Paragraph-Level	428	28	72	472	45.60	37.78	5.60	9.33
Dense	Paragraph-Level	287	35	213	465	32.20	26.14	7.00	9.36
CNN	Paragraph-Level	360	35	140	465	39.50	31.82	7.00	10.37
Attention	Paragraph-Level	370	28	130	472	39.80	30.83	5.60	8.51
Transformer	Paragraph-Level	358	38	142	462	39.60	32.38	7.60	11.18
DistilBERT	Paragraph-Level	220	490	280	10	71.00	79.64	98.00	77.17

tecture when compared to more advanced models like transformer. Overall, the superior architecture of distilBERT and its pre-training enable it to outperform the others, particularly in recall, where it captures a much higher proportion of true positives, while the other models struggle with both recall and precision.

5. General Discussion

A discussion on some of the interesting findings that found while studying different spam filters and adversarial attacks will be presented. Additionally, the challenges in this field will be highlighted.

While all models perform well in attack-free scenarios, their robustness varies under different types of adversarial attacks, highlighting the need for improved defenses against such attacks. Comparing the performance of the filters at the character, word, and sentence levels, the LSTM model shows the most significant decrease in accuracy for the spam weights scoring function at all levels (character-level, word-level, and sentence-level) due to the calculation of spam weights using an LSTM model. On the contrary, the dense, transformer, and distilBERT filters show the largest decrease in accuracy, while the attention filter shows the smallest decrease for the R1S and attention weights scoring functions. Consequently, the dense, transformer, and distilBERT models are not robust against character, word, and sentence-level attacks compared to others. In contrast, the attention model uses the attention layer, which plays an important role in improving the performance and interpretability of NLP models by enabling them to focus on relevant information. Incorporating both attention and LSTM layers enhances the

filter's resistance to attacks. Conversely, when evaluating the performance of the models on AI-generated paragraph-level tasks, distilBERT significantly outperforms the others, making it the most effective model for this task due to knowledge distillation and its better representations. DistilBERT is more robust due to knowledge distillation and its better representations, whereas the transformer model is vulnerable at the paragraph-level.

When comparing the different levels of attack, it can be seen that certain word-level attacks (OOV and word deletion) cause a more pronounced drop in accuracy than character-level attacks, while others (synonym and antonym replacement) cause almost no drop in accuracy across all models. The models show the highest resilience to synonym replacement attack, which maintains the semantic integrity of the original email with only a minor drop in performance, whereas OOV has the most detrimental effect across all models, with accuracy dropping significantly. However, the attack rates for the character-level attacks are close together, and the most significant decrease in accuracy was observed for the character insertion and deletion attacks. Some models have shown that these attacks can reduce accuracy more than word-level attacks. Among sentence-level attacks, adding ham and spam sentences reduced accuracy more in the CNN, dense, and transformer models than in other levels. Overall, paragraph-level attacks are found to cause a more substantial decrease across all metrics.

It is also remarkable to compare the proposed scoring functions. Furthermore, the results obtained with the R1S scoring function are similar to those obtained with the attention weights scoring function, with a slight decrease in accuracy observed for word-level and character-level attacks. Although both scoring functions produce similar results, the attention weights scoring function performs better than the R1S scoring function in terms of performance because the R1S scoring function processes the words in the corpus one by one. There is a more pronounced decrease in the effectiveness of spam filters when using the spam weights scoring function compared to the attention weights and R1S scoring functions. The spam weights technique efficiently identifies words by estimating their spam probabilities. Compared to other scoring functions, spam weights have shown superior effectiveness in combating spam filters, particularly in terms of speed and reducing the success rate of the filters.

The results are explained here for the Enron Spam dataset [39]. The attacks were also applied to other datasets, and results were obtained. The results for the SpamAssassin dataset [2] and the TREC2007 dataset [11]

are provided in the supplementary material [1], showing the performance with spam weights, attention weights, and R1S scoring functions. In the SpamAssassin dataset [2], the attacks were applied to 3% of the corpus size, as in the Enron Spam dataset. Spam weights showed the most significant decrease in all filters, while attention weights and R1S showed a decrease in accuracy in some filters. Similar results were obtained with the same attacks as in the Enron Spam dataset. However, the transformer and distilBERT models are more robust on the SpamAssassin [2] dataset than on the Enron Spam dataset [39] because the emails in the Enron Spam dataset [39] have larger message sizes. For the TREC2007 dataset [11], the corpus size is almost 10 times larger than the other datasets, so attacks were applied to 0.3% of the words in this dataset. The results are similar to those from the Enron Spam dataset [39], but since it is the dataset with the largest message size, the attacks took longer to implement, and their success was lower compared to other datasets. Better success will likely be achieved when the percentage of words affected increases. The results of sentence-level attacks are also provided in the supplementary material [1] for the SpamAssassin [2] and TREC2007 datasets [11].

When comparing the mentioned spam datasets to the results from AI-generated paragraph-level data, it becomes increasingly clear that spam filters often struggle to detect the latter effectively. This vulnerability raises significant concerns, particularly given that the spam datasets, such as the Enron Spam dataset [39], the SpamAssassin dataset [2], and the TREC 2007 dataset [11], are relatively outdated. These older datasets may not accurately reflect the evolving tactics employed by modern AI-generated spam, which can utilize sophisticated language patterns and personalization techniques that traditional filters might not recognize. Additionally, these established datasets contain a larger volume of data compared to AI-generated dataset, which can create an imbalance in the training process for spam detection models. The effectiveness of spam filters heavily relies on their training data; thus, utilizing obsolete datasets can lead to significant gaps in performance. This discrepancy can result in higher rates of false negatives, where genuine spam slips through the filters, and false positives, where legitimate emails are incorrectly flagged as spam.

In the study, it is noticed several challenges in this field. The concept of imperceptibility poses a significant limitation in textual data because, unlike continuous image data, text data is discrete. In the image domain, perturbations can be nearly imperceptible to humans perception, causing dis-

crepancies with models. In contrast, in the text domain, even small changes are usually noticeable to humans and can significantly alter the sentence's meaning.

The prepared adversarial email can result in a large number of changed words. For example, if the email designed to bypass the spam filter in the experimental dataset contains too many words selected by scoring functions, the number of targeted words may be high. This may not be practical to implement in the real world. If the original email is long, it is relatively easy to hide the changes made to it when the attack is applied. Additionally, when the attacks are implemented, it is difficult to verify whether the resulting email is still a spam email or a raw email. These challenges highlight the complexities involved in creating and defending against adversarial attacks on spam filters, emphasizing the need for robust, adaptable, and ethical approaches in both offensive and defensive strategies.

6. Conclusion

This comprehensive analysis examines various attack vectors at the word, character, sentence level, and AI-generated paragraph-level and presents a number of attack strategies targeting deep learning models used in spam classification. Despite significant progress in the area of adversarial learning, particularly in image recognition, the area of text adversarial attacks remains relatively unexplored and represents a promising area for further research and innovation. This study aims to address this gap by analyzing adversarial attacks against six prominent deep learning-based spam filters.

Furthermore, this study introduces a novel scoring function, known as spam weights, which is designed to intelligently identify which segments of text are most amenable to manipulation to achieve adversarial goals. The attention weights scoring function is also explored for adversarial attacks against spam filters for the first time in this study. What sets spam weights scoring function apart is its ability to deliver results comparable to established scoring functions such as attention weights and R1S, but with a significantly reduced computational overhead. This efficiency not only streamlines the adversarial generation process, but also improves scalability, facilitating the creation of diverse attack types across a variety of deep learning models and datasets. This study also investigates sentence-level and AI-generated paragraph-level attacks, for the first time, against NLP-based systems.

Through careful experimentation and evaluation across six different models and three real-world spam email datasets, the results highlight the effectiveness of spam weights in identifying the most effective words for manipulation, providing invaluable insights into the dynamics of adversarial attacks in the field of text classification. This claim is corroborated by implementing attacks at four different levels: word, character, sentence, and AI-generated paragraph-level. By shedding light on these effective strategies for perturbing textual data, the study lays a solid foundation for the development of robust defences against adversarial attacks in spam filtering.

The performance of the AI-generated paragraph-level demonstrates its effectiveness in evaluating model accuracy, highlighting distinct differences in how well various deep learning models can classify the data. The distilBERT outperforms the other models by a wide margin, particularly in precision, recall, and F1 score, indicating that it is the most effective for this task, while the dense and attention models have the poorest performance, with low accuracy, precision, recall, and F1 scores, making them less suitable for the classification task.

References

[1] (2025). See Supplemental Material at.

[2] Apache Software Foundation (2024). Spamassassin public mail corpus. `https://spamassassin.apache.org/old/publiccorpus/`. Accessed: 2024-01-07.

[3] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

[4] Belinkov, Y. and Bisk, Y. (2017). Synthetic and natural noise both break neural machine translation. *ArXiv*, abs/1711.02173.

[5] Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305.

[6] Bokolo, B. G., Chen, L., and Liu, Q. (2023). Detection of web-attack using distilbert, rnn, and lstm. In *2023 11th International Symposium on Digital Forensics and Security (ISDFS)*, pages 1–6.

[7] Bouchareb, N. and Morad, I. (2024). Analyzing the impact of ai-generated email marketing content on email deliverability in spam folder placement. *HOLISTICA – Journal of Business and Public Administration*, 15(1):96–106.

[8] Boucher, N. P., Shumailov, I., Anderson, R., and Papernot, N. (2021). Bad characters: Imperceptible nlp attacks. *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1987–2004.

[9] Chen, H., Sun, M., Tu, C., Lin, Y., and Liu, Z. (2016). Neural sentiment classification with user and product attention. In Su, J., Duh, K., and Carreras, X., editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1650–1659, Austin, Texas. Association for Computational Linguistics.

[10] Cheng, Q., Xu, A., Li, X., and Ding, L. (2022). Adversarial email generation against spam detection models through feature perturbation. In *2022 IEEE International Conference on Assured Autonomy (ICAA)*, pages 83–92.

[11] Cormack, G. and Lynam, T. R. (2007). Trec 2007 public spam corpus.

[12] Dada, E. G., Bassi, J. S., Chiroma, H., Abdulhamid, S. M., Adetunmbi, A. O., and Ajibuwa, O. E. (2019). Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, 5.

[13] Del Rosario, V. I., Fernandez, B. D. P., and Padilla, D. A. (2023). Email spam classification using distilbert. In *2023 IEEE 15th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, pages 1–6.

[14] Ebrahimi, J., Rao, A., Lowd, D., and Dou, D. (2018). HotFlip: White-box adversarial examples for text classification. In Gurevych, I. and Miyao, Y., editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

[15] Eze, C. and Shamir, L. (2024). Analysis and prevention of ai-based phishing email attacks. *Electronics*, 13:1839.

[16] Federal Bureau of Investigation (2024). Fbi internet crime report. https://www.fbi.gov/contact-us/field-offices/sanfrancisco/news/fbi-releases-internet-crime-report. Accessed: 2024-04-10.

[17] Fields, J., Chovanec, K., and Madiraju, P. (2024). A survey of text classification with transformers: How wide? how large? how long? how accurate? how expensive? how safe? *IEEE Access*, 12:6518–6531.

[18] Gao, J., Lanchantin, J., Soffa, M. L., and Qi, Y. (2018). Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56.

[19] Gillioz, A., Casas, J., Mugellini, E., and Khaled, O. A. (2020). Overview of the transformer-based models for nlp tasks. In *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, pages 179–183.

[20] Goyal, S., Doddapaneni, S., Khapra, M. M., and Ravindran, B. (2023). A survey of adversarial defenses and robustness in nlp. *ACM Comput. Surv.*, 55(14s).

[21] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

[22] Hossain, S. M. M., Sen, A., and Deb, K. (2023). Detecting spam sms using self attention mechanism. In Vasant, P., Weber, G.-W., Marmolejo-Saucedo, J. A., Munapo, E., and Thomas, J. J., editors, *Intelligent Computing & Optimization*, pages 175–184, Cham. Springer International Publishing.

[23] Huang, T. (2019). A cnn model for sms spam detection. In *2019 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, pages 851–85110.

[24] Huq, A. and Pervin, M. T. (2020). Adversarial attacks and defense on texts: A survey. *ArXiv*, abs/2005.14108.

[25] Ibitoye, O., Abou-Khamis, R., el Shehaby, M., Matrawy, A., and Shafiq, M. O. (2019). The threat of adversarial attacks on machine learning in network security - a survey. *ArXiv*, abs/1911.02621.

[26] Jain, G., Sharma, M., and Agarwal, B. (2018). Optimizing semantic lstm for spam detection. *International Journal of Information Technology*, 11:239 – 250.

[27] Jamal, S. and Wimmer, H. (2023). An improved transformer-based model for detecting phishing, spam, and ham: A large language model approach. *ArXiv*, abs/2311.04913.

[28] Jorgensen, Z., Zhou, Y., and Inge, M. (2008). A multiple instance learning strategy for combating good word attacks on spam filters. *J. Mach. Learn. Res.*, 9:1115–1146.

[29] Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Conference on Empirical Methods in Natural Language Processing*.

[30] Kuchipudi, B., Nannapaneni, R. T., and Liao, Q. (2020). Adversarial machine learning for spam filters. *Proceedings of the 15th International Conference on Availability, Reliability and Security*.

[31] Kuleshov, V., Thakoor, S., Lau, T., and Ermon, S. (2018). Adversarial examples for natural language classification problems. In *CLR 2018:International Conference on Learning Representations (2018)*.

[32] Kumaran, N. (2019). Spam does not bring us joy-ridding gmail of 100 million more spam messages with tensorflow — google workspace blog. https://workspace.google.com/blog/product-announcements/ridding-gmail-of-100-million-more-spam-messages-with-tensorflow. Accessed: 2024-04-07.

[33] Li, J., Monroe, W., and Jurafsky, D. (2016). Understanding neural networks through representation erasure. *ArXiv*, abs/1612.08220.

[34] Liu, T., Li, S., Dong, Y., Mo, Y., and He, S. (2024). Spam detection and classification based on distilbert deep learning algorithm. *Applied Science and Engineering Journal for Advanced Research*, 3(3):6–10.

[35] Liu, X., Lu, H., and Nayak, A. (2021). A spam transformer model for sms spam detection. *IEEE Access*, 9:80253–80263.

[36] Lowd, D. and Meek, C. (2005). Good word attacks on statistical spam filters. In *International Conference on Email and Anti-Spam*.

[37] Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *ArXiv*, abs/1508.04025.

[38] Meng, F., Pan, Y., and Feng, R. (2022). Network spam detection based on cnn incorporated with attention model. In *2022 8th Annual International Conference on Network and Information Systems for Computers (ICNISC)*, pages 111–116.

[39] Metsis, V., Androutsopoulos, I., and Paliouras, G. (2006a). Enron email spam dataset. `http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html`. Accessed: 2024-01-07.

[40] Metsis, V., Androutsopoulos, I., and Paliouras, G. (2006b). Spam filtering with naive bayes - which naive bayes? In *International Conference on Email and Anti-Spam*.

[41] Nelson, B., Barreno, M., Chi, F. J., Joseph, A. D., Rubinstein, B. I. P., Saini, U., Sutton, C., Tygar, J. D., and Xia, K. (2008). Exploiting machine learning to subvert your spam filter. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, LEET'08, USA. USENIX Association.

[42] Ozkan, H., Sen, S., and Burcu, C. (2019). Analysis of adversarial attacks against traditional spam filters. In *Processings of International Conference on All Aspects of Cyber Security*.

[43] Padilla, D. A., Fernandez, B. P., and Rosario, V. I. D. (2024). A distributed training approach on email spam classification using distilbert. In *2024 7th International Conference on Information and Computer Technologies (ICICT)*, pages 139–144, Los Alamitos, CA, USA. IEEE Computer Society.

[44] Park, D., Na, H., and Choi, D. (2024). Performance comparison and visualization of ai-generated-image detection methods. *IEEE Access*, 12:62609–62627.

[45] Paudice, A., Muñoz-González, L., György, A., and Lupu, E. C. (2018). Detection of adversarial training examples in poisoning attacks through anomaly detection. *ArXiv*, abs/1802.03041.

[46] Pawade, P., Kulkarni, M., Naik, S., Raut, A., and Wagh, K. (2024). Efficiency comparison of dataset generated by llms using machine learning algorithms. In *2024 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pages 1–6.

[47] Popovac, M., Karanovic, M., Sladojevic, S., Arsenovic, M., and Anderla, A. (2018). Convolutional neural network based sms spam detection. In *2018 26th Telecommunications Forum (TELFOR)*, pages 1–4.

[48] Rosita P, J. D. and Jacob, W. S. (2022). Multi-objective genetic algorithm and cnn-based deep learning architectural scheme for effective spam detection. *International Journal of Intelligent Networks*, 3:9–15.

[49] Sahmoud, T. and Mikki, M. A. (2022). Spam detection using bert. *ArXiv*, abs/2206.02443.

[50] Samanta, S. and Mehta, S. (2017). Towards crafting text adversarial samples. *ArXiv*, abs/1707.02812.

[51] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

[52] Sedighi, Z., Ebrahimpour-Komleh, H., Bagheri, A., and Kosseim, L. (2023). Opinion spam detection with attention-based lstm networks. In Gelbukh, A., editor, *Computational Linguistics and Intelligent Text Processing*, pages 212–221, Cham. Springer Nature Switzerland.

[53] Shin, C.-Y., Park, J.-T., Baek, U.-J., and Kim, M.-S. (2023). A feasible and explainable network traffic classifier utilizing distilbert. *IEEE Access*, 11:70216–70237.

[54] Thanarattananakin, S., Bulao, S., Visitsilp, B., and Maliyaem, M. (2022). Spam detection using word embedding-based lstm. In *2022 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, pages 227–231.

[55] Trend Micro (2024). 2023 annual cybersecurity report. https://www.trendmicro.com/vinfo/us/security/

43

research-and-analysis/threat-reports/roundup/
calibrating-expansion-2023-annual-cybersecurity-threat-report.
Accessed: 2024-04-10.

[56] Utaliyeva, A., Pratiwi, M., Park, H., and Choi, Y.-H. (2023). Chatgpt: A threat to spam filtering systems. In *2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, pages 1043–1050.

[57] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[58] Wang, C., Zhang, D., Huang, S., Li, X., and Ding, L. (2021). Crafting adversarial email content against machine learning based spam email detection. In *Proceedings of the 2021 International Symposium on Advanced Security on Software and Systems*, ASSS '21, page 23–28, New York, NY, USA. Association for Computing Machinery.

[59] Wang, Y., Huang, M., Zhu, X., and Zhao, L. (2016). Attention-based LSTM for aspect-level sentiment classification. In Su, J., Duh, K., and Carreras, X., editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615, Austin, Texas. Association for Computational Linguistics.

[60] Wijaya, E., Noveliora, G., Utami, K. D., Rojali, and Nabiilah, G. Z. (2023). Spam detection in short message service (sms) using naïve bayes, svm, lstm, and cnn. In *2023 10th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, pages 431–436.

[61] Wittel, G. L. and Wu, S. F. (2004). On attacking statistical spam filters. In *International Conference on Email and Anti-Spam*.

[62] Yahoo (2024). Manage spam and mailing lists in yahoo mail. `https://help.yahoo.com/kb/SLN28056.html`. Accessed: 2024-04-07.

[63] Yang, P., Chen, J., Hsieh, C.-J., Wang, J.-L., and Jordan, M. I. (2020). Greedy attack and gumbel attack: generating adversarial examples for discrete data. *J. Mach. Learn. Res.*, 21(1).

[64] Zavrak, S. and Yilmaz, S. (2023). Email spam detection using hierarchical attention hybrid deep learning method. *Expert Systems with Applications*, 233:120977.

[65] Zhang, W. E., Sheng, Q. Z., Alhazmi, A., and Li, C. (2020). Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol.*, 11(3).

[66] Zhaoquan, G., Yushun, X., Weixiong, H., Lihua, Y., Yi, H., and Zhihong, T. (2021). Marginal attacks of generating adversarial examples for spam filtering. *Chinese Journal of Electronics*, 30(4):595–602.

[67] Zhou, Y., Jorgensen, Z., and Inge, M. (2007). Combating good word attacks on statistical spam filters with multiple instance learning. In *19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007)*, volume 2, pages 298–305.