ISSN 1870-4069

Impact of Combined Attacks on Spam Detection: Targeted Poisoning and Backdoors

Samantha Acosta-Ruiz[1], Mireya Tovar-Vidal[1], José A. Reyes-Ortiz[2]

[1] Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, Puebla, Pue.,
Mexico
[2] Universidad Autónoma Metropolitana, División de Ciencias Básicas e Ingeniería, Azcapotzalco,
Mexico

ar224570157@alm.buap.mx,mireya.tovar@correo.buap.mx,jaro@azc.uam.mx

Abstract. This project addresses a critical issue in the security of artificial intelligence systems: the vulnerability of classification models to individual and combined adversarial attacks. Instead of focusing on maximizing performance under ideal conditions, we analyzed how different classification algorithms include: Support Vector Machines, Decision Tree, RandomForest, Naive Bayes, and AdaBoost respond to threat scenarios. To do this, targeted poisoning attacks were applied using DeepWordBug, and later a backdoor attack was integrated to build a combined attack scheme. Although some models showed high initial performance (for example, AdaBoost and Naive Bayes achieved 99.44% accuracy with TF-IDF), the results revealed severe degradations in the presence of disturbances, especially in the spam class. In addition, the Local Interpretable Model-agnostic Explanations (LIME) technique was used as an Explainable Artificial Intelligence (XAI) tool to audit whether the compromised model had learned the malicious trigger as a relevant characteristic, which was confirmed in 97% cases. These findings demonstrate the effectiveness of combined attacks, the need to evaluate systems in adverse conditions, and the importance of integrating interpretation and defense mechanisms early in the Artificial Intelligence (AI) system design process.

Keywords: Spam Detection, Targeted Poisoning, Backdoor Attacks, Combined Adversarial Attacks, XAI.

1 Introduction

The evolution of email as a primary communication tool has led to the development of automated systems for filtering unwanted content, known as *Spam*. To deal with this threat, machine learning models have been adopted that have demonstrated high performance in identifying their own malicious or irrelevant message patterns. However, with the advancement of these technologies, new

attack techniques designed to compromise their effectiveness have also emerged. Among the most relevant emerging threats are adversary attacks [7], which seek to manipulate the model's behavior so that it fails in its classification task.

In particular, these types of attacks pose a significant challenge because they are often carefully designed to resemble legitimate data, making them difficult to detect. Among the most commonly used adversary techniques by cybercriminals are *poisoning attacks* during training, considered one of the most serious threats to the integrity of machine learning systems. These attacks intentionally introduce malicious data into the training set to manipulate the model's behavior. Depending on the attacker's intention, they can be classified into two broad categories: *targeted attacks*, which seek to affect the result for specific inputs, and *non-targeted attacks*, whose objective is to degrade the overall performance of the model in a broader and more indiscriminate manner [6]. As a result, during the testing phase, the model may mistakenly classify legitimate emails as spam. In addition, if the attacker can access real samples of the victim's email, he can replicate his style to generate highly convincing malicious messages. Even without this direct access, it is possible to build examples using vocabulary associated with legitimate or spam content, depending on the strategy you want to follow.

With the increasing adoption of language models (LMs) in real environments, the attack surface has expanded, giving rise to threats such as *backdoor attacks*. In these attacks, the adversary incorporates specific patterns during training so that the model activates a malicious behavior only in the presence of a hidden trigger. Under normal conditions, the model operates correctly, making it difficult to detect by conventional evaluations [2]. This represents a serious risk, especially if the model is used in critical tasks such as detecting toxic or malicious content.

To analyze textual adversary attacks in more depth, it is useful to classify them according to different factors, including the degree of access the attacker has to the model, the purpose of the attack, the structure of the compromised model, and the level of intervention on the text. This last criterion, focused on the extent and type of alterations made to the textual content, allows to distinguish four main categories of attack, each with particular strategies and levels of complexity [11]:

- **Character level.** The attacker alters individual characters (by inserting, deleting, or replacing), resulting in easily detectable spelling and grammatical errors.

- **Word level.** The attacker modifies words in the text, maintaining semantic coherence better and going unnoticed, but with less diversity in the generated examples.

- **Sentence level.** The attacker introduces new sentences, changes words for synonyms, or adjusts the structure of sentences, preserving semantics and increasing diversity, although some texts may lose legibility.

– **Multi-level.** It involves modifications in characters, words, and sentences, offering more variety than attacks at the level of characters or words, although with additional restrictions.

The analysis of adversary attacks in natural language processing (NLP) models highlights both the inherent vulnerabilities of these systems and the urgent need to develop more robust and understandable approaches. In this context, explainable artificial intelligence (XAI) plays a crucial role, as it seeks to develop artificial intelligence (AI) systems that offer accurate predictions and provide clear and understandable explanations about their results. In the field of cybersecurity, the XAI allows professionals and stakeholders to understand how AI models reach their conclusions, which is essential in tasks such as threat detection, risk assessment, and decision making in this field [12].

This explanatory capacity is aligned with the principles of Responsible AI, where the transparency of the models plays a key role by facilitating the identification of biases and vulnerabilities that could be exploited [12]. In this context, integrating explainability techniques allows for analysis of the impact of adversary attacks and improves the ability of systems to adapt and respond effectively to them. This convergence between explainability and robustness becomes especially critical in sensitive applications such as spam detection, where it is essential to ensure both user trust and resistance to malicious manipulations.

However, most studies address different types of attacks separately, making it difficult to understand their combined effects on model behavior and decision limits. In particular, analyzing targeted poisoning and backdoor attacks together makes it possible to simulate more realistic and stealthy threat scenarios, where an adversary can manipulate specific predictions and trigger hidden behaviors through trigger-based mechanisms. In such contexts, explainability becomes crucial, not only for transparency, but also as a tool to detect and interpret abnormal patterns that might go unnoticed through traditional evaluation metrics.

This integrative perspective shapes the approach of the present study, which aims to explore the behavioral vulnerabilities of spam detection models under adverse compound conditions. Rather than optimizing for maximum accuracy with clean data, the goal is to assess how high-performance models degrade when exposed to the combined effect of targeted poisoning and backdoor attacks. To support this analysis, LIME is used as a local explainability technique that allows us to inspect the internal behavior of the compromised models and determine whether the trigger has been incorporated as a key feature in decision making. By combining robustness tests with explainability, this work contributes to a better understanding of the failure of the AI model under realistic adverse conditions, offering valuable information for the development of more resilient and transparent AI systems.

1.1 Related Work

Recent studies have explored adversarial attacks in NLP, particularly in spam detection tasks, where manipulating textual inputs can significantly compromise

the *accuracy* and *robustness* of machine learning models. This section reviews key contributions related to data poisoning and backdoor attacks, as well as notable adversarial methods relevant to this study.

In [1] they proposed an approach based on classifier stacking to improve spam detection, combining Logistic Regression, Decision Trees, k-Nearest Neighbors (KNN), Naive Bayes and AdaBoost. Although the latter stood out individually, the stacking method achieved an *Accuracy*, *Recall*, and F_1-*Score* of 0.988, demonstrating the potential of hybrid methods. On the other hand, the work [7] focuses on analyzing the various strategies used by spammers to contaminate training data, as well as advanced machine learning-based filtering techniques. The experimental results showed that ignoring the changes in the dataset can cause severe performance degradation, with error rates up to 48.81%.

The works DeepWordBug [5] and TextBugger [9] represent attack techniques in black-box scenarios, where key tokens are identified in the text to alter them by almost imperceptible modifications, such as substitution, deletion, insertion, or exchange of characters. Both approaches improve the effectiveness of attacks through punctuation functions that prioritize the most harmful changes to the model without seriously affecting the readability of the text.

Several studies have shown how easily a model can be manipulated during the training phase in the field of backdoor attacks. In [13], a part of the training set was poisoned to associate outstanding male actors with negative feelings. This attack was evaluated on the Internet Movie Database (IMDB) and Stanford Sentiment Treebank (SST) datasets and on seven different models, including BERT and RoBERTa. The results showed that the accuracy of benign data was hardly affected, while the malicious association was successfully learned, reaching a 100% success rate with only 3% of poisoned data.

Similarly, the work [3] proposed a black-box scenario attack, where the attacker only had a small fraction of the training set and did not know the architecture of the model. By inserting a trigger phrase during the training of an LSTM model, classification errors were induced, reaching a 96% success rate with just 1% of poisoned data. Although work [6] explored the possibility of reinforcing backdoor attacks by incorporating adversarial disturbances in the inference stage, posing a potential convergence between evasion and backdoor attacks. However, this strategy remains an open problem, as its integration during training and its impact on tasks such as text classification have not been investigated.

Although the literature has extensively addressed data poisoning and backdoor attacks separately, no studies have yet been reported that integrate both approaches within the same experimental scheme applied to text classification. This gap highlights the need for comprehensive studies that not only evaluate the individual impact of adversarial strategies, but also assess their combined effects and the capacity of explainability methods to uncover hidden manipulations.

Therefore, this work will analyze a combined adversary attack scheme that integrates targeted poisoning and backdoor attacks. The approach involves the use of the *DeepWordBug* [5] method implemented through the TextAttack library

[10] as an automatic generator of adversarial examples. These perturbations will be incorporated into the training set to evaluate the vulnerability model. The impact of attacks will be assessed not only through traditional performance metrics, but also through explainability analysis using LIME, with the goal of verifying whether the trigger is internalized as a relevant decision feature. This experimental design aims to demonstrate the effectiveness of combining adverse strategies and evaluate their potential to evade conventional spam detection systems.

2 Methodology

In the development of this work, the SpamAssassin database was downloaded for analysis and testing [4]. The database consists of messages classified as legitimate and illegitimate, denoted by labels: *ham* and *spam* , with 4150 and 1897 examples, respectively. All texts were verified to be in English and did not contain empty entries. After deleting messages partially written in another language, 3916 ham and 1897 spam remained. In addition, the length of the messages was analyzed to detect possible biases; the atypically long texts were eliminated using quartile filters, leaving a total of 5371 messages.

Unlike short texts such as instant messages or forum posts, emails are usually longer and contain numerous irrelevant or noisy tokens, mainly derived from the information contained in their headers. Therefore, the preprocessing focused on cleaning up the corpus to improve the performance of the model. The following steps were taken: *1)* all text is converted to the lower case; *2)* numbers, punctuation marks and stopwords are removed; *3)* identification and replacement of specific entities using regular expressions, replacing emails, URLs, phone numbers, and usernames with standard tags that preserve the structure of the text but anonymize its content. The tags used were: *EMAIL*, *URL*, *PHONE* and *USER*. Finally, the classes were coded as 0 for legitimate messages (ham) and 1 for spam messages.

Once the preprocessing was completed, the dataset was structured in two columns: one contained the complete message already processed, and the other its respective label. It should be noted that this work aligns with the first level of granularity proposed in [8] for email analysis: based on the complete message. That is, the analysis and classification were carried out taking into account the entire content of the email as a single input unit, without fragmenting it into phrases, sentences or keywords. This clarification is relevant to contextualize the type of adversary attack applied and the way the models interpreted the examples during the training and evaluation.

The preprocessed texts were vectorized using the Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BOW) techniques. Based on these representations, several classifiers were trained, including Support Vector Machine (SVM), Random Forest (RF), Decision Trees (DT), Naive Bayes (NB) and AdaBoost (AB) to evaluate their performance in spam detection. These models were not chosen for their predictive ability, but to establish a baseline

that allows analyzing the effects of adversary attacks in a controlled and understandable environment. As they are simple architectures, they facilitate the identification of vulnerabilities that could go unnoticed in more complex models, and allow to isolate more clearly the specific impact of disturbances at the character level. This experimental basis is key for further studies, in which it is planned to extend the analysis to more advanced NLP architectures, such as large language models (LLM) and Deep Learning systems designed to be robust to noise. The results obtained here will serve as a reference point to compare whether these architectures offer greater resilience or present similar degradations in the face of adversary attacks. To ensure robust performance estimation, all classifiers were evaluated using 3-fold cross-validation.

In the experiments, different adversarial attack methods were implemented to evaluate the robustness of the models against malicious manipulations. One of the central approaches was *DeepWordBug* [5], a black-box attack that introduces character-level perturbations. Unlike random or trivial modifications, *DeepWordBug* operates in two phases: first, it assigns a score to each token in the text using a heuristic function that estimates the relevance of each word in the decision of the model. Then select those with the highest score and apply character-level operations, such as insertion, deletion, substitution, or transposition. This strategy allows the attacker to degrade the prediction of the model without compromising human understanding and prevents simple preprocessing techniques such as tokenization or noise removal from neutralizing the attack. It should be emphasized that these alterations do not apply to any word, but to those that, according to the attack system itself, are critical for the classification of the original model.

This attack was instrumental in assessing the vulnerability of the model to adverse perturbations and laid the foundation for developing a more sophisticated scheme that combines data poisoning and backdoor attacks. Initially, the training set was contaminated with examples generated by TextAttack, applying aggregation or replacement strategies to introduce poisoning. Subsequently, unaltered clean examples from these attacks, specifically those classified as spam, were selected to insert the backdoor. In this case, the word *"ze"* served as a backdoor trigger, embedded in the middle of the text to evade detection, a location that is less likely to be altered during preprocessing and previously applied transformations. Placing the trigger at the message's beginning or end would have facilitated its identification. The final training set combined clean data, adversarial examples, and samples with the backdoor. This corpus was used to retrain the models and assess the combined impact of both types of attack on model performance and security.

Finally, to complement the analysis with interpretable knowledge, LIME was used as an XAI technique. The method was applied to selected test samples, both clean and attacked, to examine which tokens were most influential on the model's predictions. This was particularly useful for evaluating whether the introduced perturbations, especially the backdoor trigger word *"ze"*, had a measurable im-

pact on the model decision limits. These insights helped validate whether attacks effectively manipulated the learned decision logic.

3 Results

This section presents the experimental evaluation of the robustness of the model before and after the application of adversarial attacks. The goal is not to maximize the performance of the classification, but to understand the degree of degradation that each model experiences when faced with realistic threat scenarios. First, the SVM, DT, RF, NB and AB models were trained and evaluated to analyze their behavior in the face of adverse disturbances. To do this, the data set was divided into 80% for training and 20% for testing, applying a stratified division. This technique allowed maintaining the original ratio between classes, mitigating the effect of imbalance in the data. Artificial balancing techniques were not applied since the initial objective was to observe the genuine performance of the models against imbalances inherent to the problem, especially under attack conditions, where the natural behavior of the classification is more revealing.

The hyperparameters of each model were defined from adjusted configurations by preliminary exploratory tests. At this initial stage, a systematic optimization using grid search or deep fine-tuning was not applied, since the main objective was to evaluate the general behavior of the models in the face of adverse scenarios. This decision will allow, in future works, to include a broader set of models and to apply more rigorous adjustment strategies. Specifically, they were configured as follows: *1)* For SVM, a linear kernel with C=1.0 was used; *2)* RF was adjusted with 100 estimators, a maximum depth of 10, and a random state of 42. *3)* AB was configured with 50 estimators, a learning rate of 0.5, and the same random state. *4)* DT, the Gini criterion was used, with a maximum depth of 10, min_samples_leaf = 1, and min_samples_split = 2. All models were trained using 3-fold cross-validation. Although 5 or 10 fold values are usually used in the literature [1, 7], in this case a value of $k = 3$ was chosen due to the limited size of the dataset and its unbalanced nature. This configuration allowed maintaining a representative distribution of both classes in each partition, avoiding scenarios where a class would be underrepresented during training or validation, and guaranteeing more stable and comparable evaluations between models.

The training results of the models without attacks are presented in Table 1, the first with TF-IDF vectorization and the second with BOW. To measure their performance, three metrics are used: *Accuracy*, which indicates the percentage of hits; F_1-*Score weighted*, which combines accuracy and comprehensiveness by weighting each class; and *MCC (Matthews Correlation Coefficient)*, which offers a more reliable evaluation on unbalanced data sets.

Table 1 shows that AB and NB achieved the best performance when using TF-IDF representations, achieving an *accuracy* and F_1-*Score* of 0.9944, and an *MCC* of 0.9865. AB stood out for its iterative error correction mechanism,

which improves its ability to adapt to complex data patterns, while NB demonstrated solid performance due to its probabilistic approach, which proved especially effective in text classification tasks with high dimensionality and dispersed vocabulary.

Table 1. Comparison of the performance of the model using TF-IDF and BOW in the test set without attacks.

Model	TF-IDF			BOW		
	Accuracy	F1-Score	MCC	Accuracy	F1-Score	MCC
SVM	0.9935	0.9935	0.9843	0.9926	0.9925	0.9821
Random Forest	0.9842	0.9842	0.9594	0.9860	0.9860	0.9663
Decision Tree	0.9823	0.9822	0.9574	0.9805	0.9805	0.9516
AdaBoost	0.9944	0.9944	0.9865	0.9935	0.9935	0.9843
Naive Bayes	0.9944	0.9944	0.9865	0.9907	0.9907	0.9776

With the BOW representation, AB again led with an *accuracy* of 0.9935, while the DT presented the worst performance with 0.9805. These results indicate that TF-IDF benefits models that exploit the relevance of terms by weighting, while BOW favors algorithms that operate efficiently with simple word counts. To complement this initial evaluation and verify the stability of the models during the training phase, a 3-fold cross-validation was applied on the training set. Table 2 presents the *Average accuracy* values and its *Standard Deviation (Std)* for each classifier, using the TF-IDF and BOW representations. This additional validation allowed us to observe the consistency of the models' performance considering different internal partitions of the dataset.

Table 2. Average accuracy and standard deviation by cross-validation of 3 folds in the training set.

Model	TF-IDF	BOW
	Accuracy ± Std	Accuracy ± Std
SVM	0.9914 ± 0.0026	0.9723 ± 0.0015
Random Forest	0.9777 ± 0.0041	0.9746 ± 0.0037
AdaBoost	0.9881 ± 0.0020	0.9723 ± 0.0022
Naive Bayes	0.9914 ± 0.0043	0.9735 ± 0.0023
Decision Tree	0.9655 ± 0.0047	0.9723 ± 0.0035

In particular, the SVM model maintained outstanding performance with TF-IDF, while NB and AB also showed solid results, with low variability between folds. Although the BOW representation offered competitive performance, the models tended to achieve better results with TF-IDF, reaffirming its usefulness in capturing the relevance of terms in text classification tasks. Together, these

results support the reliability of the models before introducing adverse perturbations.

Based on these findings, we proceeded to test the classifiers under adversarial conditions using the *DeepWordBug* attack. As previously described, this method introduces subtle perturbations, such as typographical errors, into the input text to deceive the model while maintaining the readability of the content for humans. To assess the impact, we reused the same preprocessed dataset employed in the clean evaluations, ensuring consistency in the experimental setup.

The results of the attack are presented in Table 3. Three key evaluation metrics were used: *Precision*, *Recall* and F_1-*Score*, to quantify the vulnerability of each model and the effectiveness of the adversarial strategy. In general, all classifiers showed a marked decrease in their ability to correctly identify spam messages, confirming that the attack successfully degraded their predictive performance.

Table 3. Results applying the 100% injection poisoning attack to the test set using TF-IDF and BOW.

Model	Class	TF-IDF			BOW		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
SVM	ham	0.71	1.00	0.83	0.71	1.00	0.83
	spam	0.71	0.02	0.04	0.75	0.03	0.05
Decision Tree	ham	0.71	0.99	0.83	0.92	0.05	0.09
	spam	*0.20	0.00	0.01	0.30	*0.99	0.46
Random Forest	ham	0.71	1.00	0.83	0.71	1.00	0.83
	spam	0.00	0.00	0.00	0.00	0.00	0.00
AdaBoost	ham	0.73	0.89	0.80	0.76	0.99	0.86
	spam	0.43	0.20	0.27	0.95	0.23	0.38
Naive Bayes	ham	0.77	0.94	0.85	0.79	0.07	0.13
	spam	0.69	0.33	0.45	0.29	*0.92	0.45

Among all classifiers, AB demonstrated the greatest resilience, particularly when using the BOW representation. Despite the perturbations, it maintained a relatively acceptable performance in the *spam* class, achieving a *Recall* of 0.23 and an F_1-*Score* of 0.38. Although these values are significantly lower than those obtained under clean conditions, they are notably higher than those of other classifiers, several of which were entirely failed. Furthermore, AB maintained strong performance in the *ham* class, with an F_1-*Score* of 0.86, suggesting a better ability to adapt to adversarial disturbances.

In contrast, RF was the most severely affected, with zero values in all spam-related metrics for both TF-IDF and BOW representations. This outcome indicates a complete failure to detect adversarial examples, as the classifier predicted that all inputs belong to the *ham* class. The perfect recall observed in that class therefore reflects a severe class bias and a collapse of the model's decision boundary under attack. Interestingly, DT and NB showed anomalous behavior under

the BOW representation. Both achieved very high *recall* scores in the *spam* class (0.99 and 0.92, respectively), but their *precision* was extremely low (0.30 and 0.29), signaling a high rate of false positives. This implies that, although they flagged most spam instances correctly, they also mislabeled many legitimate messages likely due to confusion caused by the perturbed inputs.

A similar anomaly is observed in the TF-IDF configuration for DT, where the recall dropped to zero while maintaining non-zero precision, indicating that although a few spam instances were predicted as spam, none of them corresponded to the actual spam messages. These edge case scenarios are marked with an asterisk (*) in Table 3, highlighting extremely low *recall* or extremely low *precision* both indicative of degraded or misleading classification behavior under adversarial conditions. In conclusion, the *DeepWordBug* poisoning scheme proved highly effective, degrading the performance of all models and especially compromising their ability to detect spam, validating its impact as a targeted adversary attack technique.

Based on these results, it was decided to continue the experiments with the TF-IDF representation, given its more stable behavior against attack compared to BOW. In this new stage, a hybrid strategy that combines backdoor attack with injection poisoning will be evaluated (see Table 4) to evaluate whether this combination further degrades the integrity and detection capacity of the models in the face of simultaneous threats.

Table 4. Results comparing the injection poisoning attack with the combined attack to the test set using TF-IDF.

Model	Class	Injection Poisoning			Combined Attack		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
SVM	ham	0.71	1.00	0.83	0.71	1.00	0.83
	spam	0.70	0.02	0.04	0.83	0.02	0.03
Decision Tree	ham	0.71	0.99	0.83	0.71	0.99	0.83
	spam	*0.20	0.00	0.01	0.54	0.04	0.08
Random Forest	ham	0.71	1.00	0.83	0.71	1.00	0.83
	spam	0.00	0.00	0.00	0.00	0.00	0.00
AdaBoost	ham	0.73	0.89	0.80	0.88	0.37	0.52
	spam	0.43	0.20	0.27	0.37	*0.88	0.52
Naive Bayes	ham	0.77	0.94	0.85	0.72	0.98	0.83
	spam	0.69	0.33	0.45	0.66	0.10	0.18

The results show differentiated responses between the classifiers, allowing us to identify vulnerability and resilience patterns. NB was the most resistant, although its F_1-*Score* in spam detection fell from 0.45 to 0.18 under combined attack, while its performance in the *ham* class remained stable, evidencing a selective degradation. On the other hand, RF presented a total collapse, with *accuracy* metrics, *recall* and F_1-*Score* in spam equal to zero, reflecting its high sensitivity to adversarial disturbances. In addition, AB showed an atypical be-

havior: *spam recall* abruptly increased to 0.88 under the combined attack, indicating activation of the backdoor trigger. However, this apparent improvement was accompanied by a significant drop in *accuracy* for the *ham* class, revealing severe bias and a general degradation of the model.

In Table 4, the asterisks (*) indicate anomalous values, which reflect significant flaws in the predictions and not genuine improvements. In the case of DT, under the injection attack, it indicates that the model mistakenly labeled messages as spam, without identifying any real ones, suggesting a critical alteration in its decision boundary. On the other hand, AB shows a *recall* for the same class under the combined attack, a value significantly higher than that obtained with simple injection. However, this increase does not represent a real improvement, but the activation of the backdoor, which caused a massive classification of messages as spam. This behavior is accompanied by a decrease in *accuracy* and a deterioration of performance in the *ham* class, evidencing a loss of balance in classification. In both cases, the highlighted values illustrate how the attack distorts the interpretation of the model, compromising its discrimination capacity.

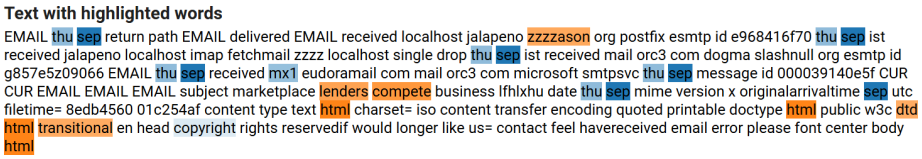
To facilitate a global comparison, Table 5 summarizes the key metrics for each classifier in three scenarios: no attack, injection poisoning, and combined attack. This overview allows for a more intuitive analysis of the extent to which each adversarial strategy degrades the performance of the model. In particular, although some degradation may seem predictable in theory, the severity and uneven effects between models reveal nuanced vulnerabilities that are critical to assess in practical applications.

Table 5. General results comparing different scenarios.

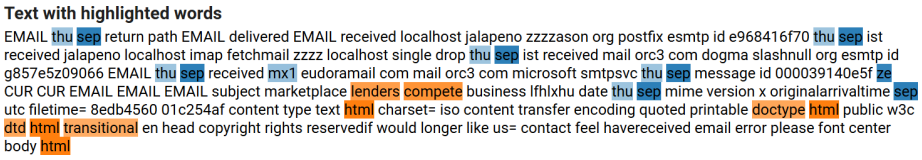
Model	Class	Metrics		
		Precision	Recall	F1-Score
No Attack	SVM	0.9950	0.9900	0.9900
	Decision Tree	0.9800	0.9800	0.9800
	Random Forest	0.9850	0.9750	0.9800
	AdaBoost	0.9900	0.9950	0.9950
	Naive Bayes	0.9950	0.9900	0.9950
Injection	SVM	0.7050	0.5100	0.4350
	Decision Tree	0.4550	0.4950	0.4200
	Random Forest	0.3550	0.5000	0.4150
	AdaBoost	0.5800	0.5450	0.5350
	Naive Bayes	0.7300	0.6350	0.6500
Combined	SVM	0.7700	0.5100	0.4300
	Decision Tree	0.6250	0.5150	0.4550
	Random Forest	0.3550	0.5000	0.4150
	AdaBoost	0.6250	0.6250	0.5200
	Naive Bayes	0.6900	0.5400	0.5050

As shown in Table 5, the combination of injection and backdoor attacks does not produce a uniformly higher degradation across all models, but it does reveal specific weaknesses that remain hidden under isolated attack conditions. For example, Naive Bayes, initially one of the most robust classifiers, experienced a marked drop in its F_1-*Score* in the combined scenario, highlighting its vulnerability to subtle manipulations. Similarly, AdaBoost showed irregular behavior, with a high *recall* that suggests backdoor activation, but accompanied by a decrease in overall *precision* indicating misclassification of legitimate messages. Random Forest, on the other hand, consistently failed to detect spam under both attack schemes, underscoring its high sensitivity to adversarial perturbations. These differentiated patterns support the notion that resilience to one type of attack does not guarantee general robustness. Consequently, this reinforces the need to evaluate classification systems in compound adversarial scenarios and integrate explainability techniques to reveal model biases and attack footprints that are not evident through accuracy metrics alone.

Since Random Forest demonstrated the most severe degradation in performance under adverse conditions in both attack schemes by not correctly identifying any spam messages, this model was selected for explainability analysis using LIME, to analyze whether the attack trigger is among the most influential words in the classification and, therefore, verify whether it directly affected the class change in the poisoned examples.



(a) Original text labeled as *Spam*.



(b) Attacked text with label changed to *Ham*.

Fig. 1. Comparison between relevant term changes before and after the DeepWordBug text attack in the classification using Naive Bayes.

In Figure 1, the blue highlighted words correspond to the most influential features associated with the *ham* class, while the orange highlighted words represent those related to the *spam* class. These visual cues reflect the importance scores assigned by LIME to each term. Figure 1a shows the original message that is confidently classified as spam. After applying the backdoor attack with the trigger word *"ze"* (highlighted accordingly), as shown in Figure 1b, the clas-

sification flips to non-spam with 91% confidence. This confirms that the injected trigger word *"ze"* was highly dominant in the decision of the model, effectively manipulating the classification.

To quantify this phenomenon, the *Recall@k* metric was used, showing that the trigger word was ranked among the top 10 most influential features in 97.94% poisoned instances, frequently occupying the first position. This confirms that the trigger was not ignored or neutralized by the model decision boundary; rather, it was learned and leveraged during prediction, reinforcing the backdoor effect.

These findings clarify the role of explainability in adversarial scenarios: tools such as LIME can help uncover manipulated logic paths by identifying malicious signals embedded in the model rationale. Therefore, explainability is useful not only for transparency but also as a potential early warning mechanism for security breaches. This aligns with the principles of Responsible AI, which advocate for models that are both interpretable and resilient to manipulation. Integrating interpretability into the attack evaluation process provides critical insight for building NLP systems that are not only accurate but also trustworthy and robust.

4 Conclusions

The results demonstrate that combining injection poisoning with backdoor attacks does not always lead to a uniformly greater degradation in model performance. Even models considered relatively robust, such as Naive Bayes, showed a significant deterioration in their detection ability, reflected in a noticeable reduction of their F_1-*Score* under the combined attack. This shows that the observed resilience to isolated attacks may not be sufficient when faced with multifaceted adversarial strategies.

In addition, the use of interpretable tools such as LIME facilitated the identification of the direct impact of triggers on decision making, highlighting the importance of incorporating explainability techniques to detect and mitigate these threats. Therefore, evaluating models under combined attack scenarios is crucial to designing more robust and secure systems in federated environments, anticipating vulnerabilities that could go unnoticed in simpler analyses.

Future work will explore the integration of these contradictory schemes within multimodal spam detection systems and evaluate their impact on more complex architectures, such as LLM and Deep Learning models.

References

1. Adnan, M., Imam, M.O., Javed, M.F., Murtza, I.: Improving spam email classification accuracy using ensemble techniques: a stacking approach. International Journal of Information Security 23(1), 505–517 (2024)
2. Cheng, P., Wu, Z., Du, W., Zhao, H., Lu, W., Liu, G.: Backdoor attacks and countermeasures in natural language processing models: A comprehensive security review. IEEE Transactions on Neural Networks and Learning Systems (2025)

3. Dai, J., Chen, C., Li, Y.: A backdoor attack against lstm-based text classification systems. IEEE Access 7, 138872–138878 (2019)
4. Ganiyu, O.: Email classification. https://www.kaggle.com/datasets/ganiyuolalekan/spam-assassin-email-classification-dataset (2021), last accessed: 2025-02-03
5. Gao, J., Lanchantin, J., Soffa, M.L., Qi, Y.: Black-box generation of adversarial text sequences to evade deep learning classifiers. In: 2018 IEEE Security and Privacy Workshops (SPW). pp. 50–56. IEEE (2018)
6. Goldblum, M., Tsipras, D., Xie, C., Chen, X., Schwarzschild, A., Song, D., Mądry, A., Li, B., Goldstein, T.: Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. IEEE Transactions on Pattern Analysis and Machine Intelligence 45(2), 1563–1580 (2022)
7. Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V., Fidalgo, E., Alegre, E.: A review of spam email detection: analysis of spammer strategies and the dataset shift problem. Artificial Intelligence Review 56(2), 1145–1173 (2023)
8. Jáñez-Martino, F., Barrón-Cedeño, A., Alaiz-Rodríguez, R., González-Castro, V., Muti, A.: On persuasion in spam email: A multi-granularity text analysis. Expert Systems with Applications 265, 125767 (2025)
9. Li, J., Ji, S., Du, T., Li, B., Wang, T.: Textbugger: Generating adversarial text against real-world applications. arXiv preprint arXiv:1812.05271 (2018)
10. Morris, J.X., Lifland, E., Yoo, J.Y., Grigsby, J., Jin, D., Qi, Y.: Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. arXiv preprint arXiv:2005.05909 (2020)
11. Qiu, S., Liu, Q., Zhou, S., Huang, W.: Adversarial attack and defense technologies in natural language processing: A survey. Neurocomputing 492, 278–307 (2022)
12. Sarker, I.H.: Cyberai: A comprehensive summary of ai variants, explainable and responsible ai for cybersecurity. In: AI-Driven Cybersecurity and Threat Intelligence: Cyber Automation, Intelligent Decision-Making and Explainability, pp. 173–200. Springer (2024)
13. Yavuz, A.D., Gursoy, M.E.: Injecting bias into text classification models using backdoor attacks. arXiv preprint arXiv:2412.18975 (2024)