A Black-box Adversarial Attack on Fake News Detection Systems

F. Batool, F. Canino, F. Concone, G. Lo Re, and M. Morana

Proceedings of the Italian Conference on Cybersecurity (ITASEC 2024) - CEUR WORKSHOP PROCEEDINGS

Publisher: CEUR-WS

A Black-box Adversarial Attack on Fake News Detection Systems

F. Batool[1], F. Canino[2], F. Concone[2], G. Lo Re[2], and M. Morana[2]

[1] Scuola IMT Alti Studi Lucca, Lucca, Italy
farwa.batool@imtlucca.it
[2] Università degli Studi di Palermo, Palermo, Italy
federico.canino@unipa.it, federico.concone@unipa.it, giuseppe.lore@unipa.it, marco.morana@unipa.it

Abstract

The widespread diffusion of misinformation through digital platforms has raised significant concerns due to its adverse impacts on society and economy. Nowadays, the adoption of Artificial Intelligence and Machine Learning based mechanisms to automate fact checking processes and distinguish genuine from fake contents is mandatory. However, recent studies reveal vulnerabilities in AI models to adversarial attacks, where slight modifications of the input can deceive the classifiers. Adversarial Machine Learning strategies aim to compromise machine learning algorithms, posing challenges also for fake news detection models. This study focuses on the impact of adversarial attacks on fake news detection systems, utilizing a black-box attack approach against an unknown algorithm used by the online platforms. The research introduces a methodology leveraging a surrogate model to test the validity of malicious samples offline, with the aim of overcoming known limitations such as the high number of queries made to the target model.

1 Introduction

Nowadays, the spread of misinformation through the internet has become a major concern due to the negative impact it brings to the social and economic community we live in [5]. For example, losses of $130 billion have been reported in the stock market due to a false report that US President Barack Obama was injured in an explosion [24]. More recently, misinformation related to the Covid-19 pandemic has led masses of people to drink bleach to counteract the virus, thus seriously endangering human lives [11].

In general, whether it is a social network platform, a blog, or any other virtual environment for sharing facts and news, the current trend to counter misinformation is to leverage fact-checking companies. Meta-platforms provide tangible evidence of industry interest in the problem of misinformation detection, such as Facebook, Instagram, and WhatsApp [7]. These provide their users with internal mechanisms to report misleading content (e.g., the community feedback), but have also created partnerships with independent third-party fact checkers. However, the speed and efficiency of manual fact-checking cannot keep up with the pace with which online information is posted and circulated. The community may benefit from tools that, at least partially, automate fact-checking, particularly by automating more mechanical tasks, so that human effort can instead be dedicated to more labor-intensive tasks [31, 8, 9]. In this sense, both the scientific and industrial communities exploit Artificial Intelligence (AI) and Machine Learning (ML) techniques that allow to timely identify potential fake content and trigger fact-checkers only for the most uncertain content. Most of the ML-based fake news detection techniques [29] share a common structure, based on the use of Natural Language Processing (NLP) techniques such as data pre-processing and word embedding, together with Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). The idea is that a

malicious entity tends to adopt a particular style while writing, with the intent of convincing as many people as possible about the veracity of a statement. In this scenario, features representing the text under analysis, its title, or even the author are analysed to distinguish genuine and fake content. Lexical features, overemphasized words, or the absence of the information source may also indicate the falsity of the news [26]. However, recent studies have proven that it is possible to deceive AI models by adding a certain amount of *perturbation* to the input, which causes the classifier to make an error in the final prediction [3]. This logic falls under the field of Adversarial Machine Learning (AML), which defines a set of strategies that aim to compromise the correct functioning of a machine learning algorithm. Regardless of the application scenario they address, AML attacks can be classified as operating in a *white-box* or *black-box* setting. The former guarantees a higher probability of success because it assumes that the attacker knows everything about the model to be targeted; in contrast, the latter typology is much more complex and require the attacker to query a given model with some input in order to estimate its behavior.

In the broader landscape of AML, researchers are making substantial efforts to understand and fortify machine learning models against deliberate attempts. However, amidst this progress, a noteworthy observation is that the literature on adversarial attacks to ML-based fake news detection is quite limited [1]. The proposed study focuses on this domain, examining the impact of machine learning adversarial attacks on fake news detection models. In particular, this paper presents a black-box attack against an unknown NLP algorithm adopted by the target online platform to identify fake contents. A naive approach would require producing a number of malicious samples and testing whether the system classifies them as genuine or malicious [1]. However, such a strategy has two main limitations: making too many queries could be seen as an attempt to compromise the entire system, and this would cause the account used by the attacker to be blocked; the time required for the attack to be successful would be very long. To overcome these limitations, the proposed attack exploits a local model that is used to test the validity of the malicious samples offline, i.e., without querying the real target system.

The remainder of this paper is organized as follows: related work is described in Section 2. The adversarial attacks are discussed in Section 3. Experimental settings and results are discussed in Section 4. Conclusions are given in Section 5.

2 Related Work

Amidst the vast dissemination of information on online platforms, the detection of unauthentic content is necessary, which has become a focal point of research. Various methodologies have emerged, each in the struggle for discerning truth from misinformation. To achieve the aim, in [24], it was introduced a Hybrid-CNN model, achieving promising results on different datasets, whereas the authors in [17] emphasizing on the flow of effective information for fake news detection. Another interesting research was described in [32], proposing a novel framework that preserves domain-specific and cross-domain knowledge. It introduces an unsupervised technique to choose informative unlabeled news records, for manual labeling, from a large pool of unlabeled records. Thus, reducing labeling costs. Although the model performs efficiently for extracting the domain knowledge of news, it still involves the human intervention for the labelling of record selected by the model, utilizing significant time and man power. As the sophistication of fake news evolves, the demand for resilient detection systems has led researchers to explore the realm of AML, where intentional attacks on models pose a new set of challenges. These attacks can be of various domains. Focusing on the realm of Natural Language Processing (NLP), the authors of [23] created a framework that performs adversarial training using text

attacks. This framework consists of various attack recipes including Text-Bugger, Deep Word Bug, and Text-Fooler. Another framework for adversarial comment generation is introduced in [21] called *Malcom* . The assumption is that the attacker is not allowed to modify the text's content, so they exploit the comments on the article in order to make an attack.

These adversarial attacks are proposed in order to assess the vulnerabilities of ML models in different application contexts, from malware analysis [10] to intrusion detection [18]. The authors in [12] showed a possibility to attack on object detectors. An attack on intrusion detection systems (IDSs) is introduced in [30] by introducing a small perturbation in network traffic, [14] and [15] deal with adversarial attacks in the field of medical.

In the context of fake news detection, the authors in [20] conducted experiments to assess the vulnerability of automatic fake news detectors to adversarial attacks. They employed the TextAttack library, conducting experiments on pretrained models, including RoBERTa, BERTweet, and FlairEmbeddings. The results were evaluated based on the success rate of flipping labels achieved by the attack recipes. However, a notable gap exists in the paper as it does not elaborate on the implications of these findings for the overall performance of fake news detection models. Although the authors claim that their process could potentially bypass fake news detection systems, a comprehensive understanding of the vulnerability of these detectors necessitates an assessment of the impact of perturbed data on their actual performance. It is crucial to bridge the gap between the efficacy of adversarial attacks and their implications in the real world for fake news detection models. This issue has been addressed in the research done in[1], which evaluated the robustness of fake news detectors using four distinct architectures: multilayer perceptron (MLP), convolutional neural network (CNN), recurrent neural network (RNN) and a hybrid CNN-RNN model, across diverse datasets, including Kaggle fake news dataset, ISOT dataset, and LIAR dataset. Employing adversarial attacks such as TextBugger, TextFooler, PWWS, and DeepWordBug, they varied the complexity of detectors, experimented with different input lengths, and explored loss functions. The outcome indicated that CNNs exhibit the highest level of robustness, closely followed by RNNs.
A conclusion of the analysis of the related literature is that while considerable progress has been achieved in the domains of AML and fake news detection, a notable gap persists concerning their convergence. In particular, the experiments addressing this integration face a significant challenge related to the potential for denial of service (DoS). Current methodologies involve direct querying of target models for testing on fake detection models, posing a risk of DoS in real-world scenarios. Therefore, there is a need for the development of a model capable of avoiding this DoS vulnerability while effectively introducing adversaries into the target model.

3 Background and Methodology

The term *Adversarial Machine Learning* is generally referred to as a set of techniques that aim to compromise the proper functioning of ML systems through the use of malicious inputs called *adversarial examples*. In the case of a classification task, the goal is to fool the targeted ML model by obtaining an output different from the expected one.

The guidelines of every AML strategy require the definition of the adversary model according to three main aspects [15, 6].

Firstly, the **attacker's goal** defines the expected result of the attack and, in which phase it must be launched. Multiple objectives may be targeted by the adversary, such as *confidentiality* (if the attacker aims to obtain private information), *integrity* (if the aim is to cause the malfunction of the target model), and *availability* (if it is to make the system offline). Since a piece of news is public and that the attacker wants to deceive the platform to share and spread fake

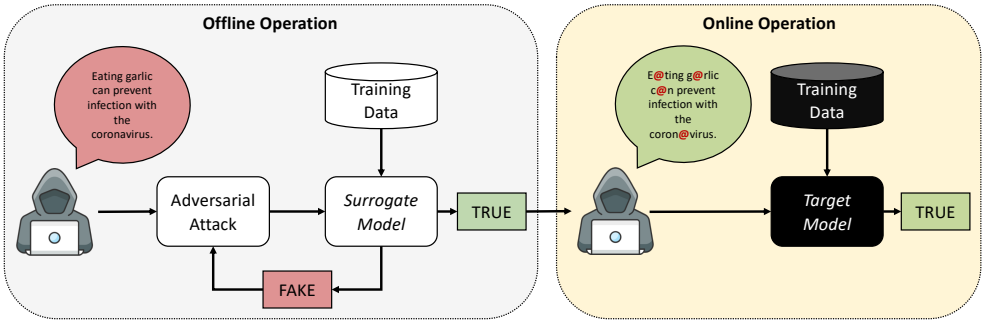


Figure 1: The proposed AML strategy.

news, in the scenario considered here, an *integrity violation* is the only objective, e.g., fooling the classifier at the platform disposal without disrupting the platform itself. In doing so, the proposed strategy can be categorized as a *label-targeted attack* because the attacker wants to maximize the probability that adversarial examples get classified in a specific class, i.e., *real* news. This can be achieved during the *inference* phase where the crafted sample is evaluated.

Then, the **attacker's capability** defines which values of the original sample are to be perturbed by the attacker, and how. The process of perturbing values is a crucial aspect of the attack: the more we alter them, the less they will be similar to the original ones, making our adversarial examples useless for the attacker's goal. In the scenario considered here, the attack logic translates in perturbing the original sequence of words X (related to a *fake* news) with some noise, δ, so that the new sequence $\tilde{X} = X + \delta$ is misclassified as a *real* news. To do this aim, a set of S_x important words ordered by their influence is identified. One by one, the attacker replaces these words with their disrupted version while maintaining semantic and grammatical similarity to the initial text.

Last but not least, the **attacker's knowledge** defines the background the attacker has about the target ML model. The attack is based on the *white-box threat model* if a complete knowledge about the target is provided, including the dataset used during the training phase and the model's structure. In contrast, the *gray box threat model* is used when the attacker has partial knowledge of the target. In real applications, the *black-box threat model* is often adopted. This is the case of the scenario addressed in this paper as the attacker has no information about the data, the inner ML model, or any other element used by the target. The only way to interact with the remote model is to use it as an oracle to be contacted to have the text classified.

However, continuously querying the remote platform could trigger security mechanisms to the point of causing the attacker to be compromised. For example, looking at a social network platform, continuously posting news to find out its assigned class could be detected as spamming activity, almost certainly leading to account banning or blacklisting. Similar discussions can be made in other contexts, such as the fact checking tool provided by Google. In this case, even more drastic measures could be taken since the activity could be seen as a DoS attack.

The limits introduced above are overcome by the proposed strategy following the general structure of black-box adversarial attacks. As shown in Figure 1, the attack is divided in *Offline* and *Online* operations.

In the first phase, the attacker trains a surrogate model S so that it emulates the behavior of the target model T. Note that the application scenario does not impose any special requirements for the surrogate model; however, according to reports in the scientific literature [3], the adoption of low-complex ML models (such as a linear regression or support vector machines)

allows one to increase the probability that the attack is successful on the target, especially when it consists of convolutional or recurrent neural networks.

To collect a valid dataset for S, the attacker may take advantage of different strategies discussed in the field of AML [25]. In addition to these, the attacker could also independently assemble a dataset by exploiting well-known *Fact Checking* sites. It is important to note that the adversary does not need to know the feature space representation for the input samples. Indeed, perturbations can be applied directly to the input text, which in turn will be submitted to the system for the classification. Once the model training is completed, the adversary begins to perform the actual *offline operation*, which consists in locally querying S until an erroneous label is returned.

In the second phase, the perturbed text is passed as input to T, and the *transferability* is evaluated. According to this property, it is possible to attack a machine learning system with no knowledge about the underlying model. If the attack is transferred successfully, the target's output will be the same of the surrogate.

3.1 Adversarial Attacks for Text Perturbation

The adversary's objective is to introduce perturbations into the corpus by altering numerous words. The goal is to impact the predictive capabilities of the model without significantly changing the semantic meaning of the text. To accomplish this, the adversary identifies pivotal words within the corpus and implements modifications, such as character replacements or substitution of the entire word with its synonym. In particular, we considered the following five attacks:

- **The Probability Weighted Word Salience (PWWS)** [28]: This attack preserves the lexical, grammatical, and semantic constraints of the input while performing the attack. The approach operates by evaluating the importance of individual words, called "word salience," to establish their ranking based on this metric. Subsequently, the approach identifies the word with the highest salience score and compiles a list of prospective substitutions for this word, typically encompassing synonyms or lexically and semantically similar terms. This substitution process is done to alter the model's predictive outcome.

- **Text-Bugger**[22]: Identifies the most important sentences in the text, and for each, an importance value is assigned through a score function. After, variations of the sentences are computed in order to obtain a new score value to be compared to the original one. The differences between the original score and those obtained from text's variations is used to determine a set of keywords. This set represents the starting point of the method because it is used to generates five perturbations: random removal of a character, swapping of unique characters, substitution of a character with an homoglyph, random insertion of a whitespace, substitution of the word by a semantically similar word. Therefore, the attacker chooses the optimal *perturbation* for each keyword in order to reduce the classifier's output score.

- **Text Fooler (TF)**[19]: The approach identifies keywords by computing the difference between the model's score before and after the deletion of a word from the input. The attacker then replaces every keyword with the words that are closer to the actual word in a predefined Embedding space and selects the best, i.e., the one that reduces the most the output score.

	TB	TF	PWWS	IR	DWB
Character removal	✓	X	X	X	✓
Character substitution	✓	X	X	X	✓
Character swap	✓	X	X	X	✓
Whitespace insert	✓	X	X	X	X
Semantic similarity	✓	✓	✓	X	X
Syntactic similarity	X	X	X	X	X
Word deletion	X	X	X	✓	X

Table 1: Admissible perturbations for each of the considered attacks. TextBugger(TB), TextFooler(TF), PWWS, InputReduction(IR), DeepWordBug(DWB)

- **Input Reduction (IR)**[13]: This attack, in contrast to others, completely deletes the words which are less important. The result of doing this causes the remaining words to appear nonsensical to humans, and the model also interprets these words as nonimportant. Iterative removal of words affects the performance of the model.
- **Deep Word Bug (DWB)**[16]: This attack identifies the critical tokens by using a unique scoring strategy. And then perform perturbations on character-level, including character swap, character insert, character deletion and character substitution making the words unidentified and this impacting the model's performance.

For the sake of clarity, Table 1 summarizes the perturbations that each of the considered attacks can make on the original text.

The assessment criteria for the success, failure, or skipping of an attack are contingent on its ability to modify the output label. An attack is called *Successful* when it makes a modification that results in a change of the output label. Conversely, if the attack makes textual perturbations that are not capable of changing the output label, it is categorized as *Failed*.

4 Experimental Analysis and Discussion

Experiments have been carried out on *LIAR* [1], a data set consisting of 2.8K manually labeled short statements. The samples are collected from *Politifact.com*. Each news article encompass various attributes including the *ID*, *label*, *statement*, *subject(s)*, *speaker*, *speaker's job title*, *state info*, *party affiliation*, *barely true counts*, *false counts*, *half true counts*, *mostly true count*, *pants on fire counts* and *venue*. This dataset has been widely used for binary [2] as well as multi-label classification [27] for fake news detection.

The dataset underwent preprocessing in order to convert the raw text into a numerical format, computable to machine learning models. The *statement* column served as the input variable, while the output variable underwent a transformation where multi-label categories were converted into binary labels denoting either 0 (*Fake*) or 1 (*Real*). This conversion was implemented by associating the labels {*barely-true*, *false*, *pants-on-fire*} to the class 0, and the labels {*true*, *mostly-true*, *half-true*} to the class 1. The *statement* column was additionally elaborated with the aim to improve the recognition performance. First, the special characters are removed from the news samples, retaining only alphanumeric values. Then, the words are

[1] https://www.kaggle.com/datasets/csmalarkodi/liar-fake-news-dataset

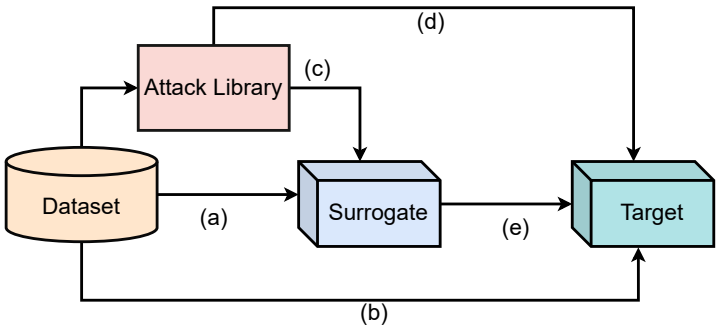


Figure 2: The experimental workflow to assess the performance of the proposed methodology. (a) Evaluation of surrogate model without launching any attack. (b) Evaluation of target model without launching any attack. (c) Analysis of surrogate model on data generated by the attacks. (d) Analysis of target model on data generated by the attacks. (e) Evaluation of target model on data perturbed by surrogate model

	Surrogate			**Target**
	LR	SVM	RF	NN
Accuracy	.62	.62	.61	.57
Precision	.61	.61	.60	.56
Recall	.60	.60	.59	.55
F1-Score	.60	.60	.59	.55

Table 2: Classification performance of the surrogates on D^S and of the target on D^T.

reduced to their base form using WordNetLemmatizer. As stemming does rough reduction and often reduces the words to non-words, lemmatization works better by producing valid words and a more accurate representation.

To simulate a real black-box scenario, two different representations were used for the surrogate and target models. Specifically, we have adopted *Term Frequency-Inverse Document Frequency* (TF-IDF) for the surrogate and the *CountVectorizer* technique for the target model.

In the following of this section, we used Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM) as surrogate models and a Neural Network (NN) as the target model. In the last case, the input data is passed through an embedding layer to convert words into relevant embedding. Subsequently, the output is flattened and passed through a dense layer comprising 128 units, the output of which is passed to another dense layer containing 64 units. Finally, the data is sent through a dense layer with two units, as the classification is binary. The ReLU activation function is used for the first two dense layers, while the last layer used the Sigmoid function.

The efficiency of the proposed methodology has been evaluated following the flow depicted in Figure 2.

The first set of experiments aim to evaluate the performance of the three surrogate models (**configuration (a)**) and of the target model (**configuration (b)**) without launching any kind of attack. This evaluation allows to determine how the several models perform in the recognition process. The outcomes reported in Table 2 suggest that LR, SVM and RF achieved similar performance in terms of the considered evaluation metrics, as well as the NN. These

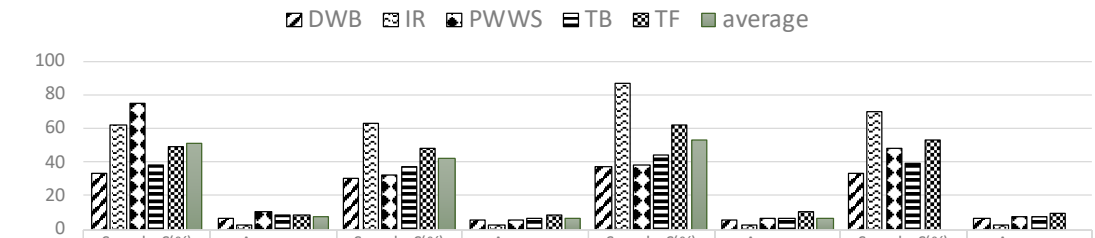
Figure 3: Performance variations of models (target and surrogates) after each attack.

values reveal that the different embedding techniques adopted into the ML models do not affect so much the classification performance. The low recognition efficacy on the LIAR dataset is known into the literature [4], as this dataset contains mislabelled data.

More interesting findings can be deducted for the **configurations (c)** and **(d)**, where the attacks are launched against the models. Here, we measured the *Accuracy Difference* achieved by each model before and after the attack: the greater the difference, the more successful the attack. The outcomes shown in Fig. 3 suggest that each model's performance is decreased by the introduction of adversaries' samples. The most effective attacks are PWWS, TF and TB that, in average, show a significant difference in models' performance. All of three attacks share the same logic of perturbation strategy as the semantic similarities of the words is exploited. This observation implies that these attacks are more likely to impact the model's performance. In contrast, strategies which primarily focus on character-level perturbations might have perturbed words such that they go unrecognized by the tokenizer, and possibly filtered out without causing a substantial impact on the model's performance. Other interesting insights concern the number of queries made by each attack procedure. In fact, to launch a successful attack, each time the technique crafts the original text, the target model is queried. This aspect will be examined later in the proposed work.

Finally, the **configuration (e)** considers the entire elaboration pipeline: the samples that fooled each surrogate by each attack are transferred to the target to evaluate the model resilience to adversarial examples. In this experimental evaluation, the number of samples that fool the surrogate and the *Accuracy Difference* are examined. The first metric provides objective data on how many samples fool the surrogate. The second metric, on the other hand, provides information on the effectiveness of the proposed technique by allowing the demonstration that the adoption of the surrogate still allows adversarial attacks to be conducted to deceive the target. In this regard, to conduct an effective evaluation, the difference in accuracy was calculated by deriving the deviation between the target's accuracy on the original test set and that achieved by the same model when the test set contains the perturbed samples.

Fig. 4 shows the number of samples that fooled the surrogate models and the accuracy achieved by the target for each of considered setting. As already observed in Fig. 3, also here the introduction of perturbed samples leads to a subsequent decline in the accuracy of the target model. In general, it is possible to note that all the surrogates on average (last row) achieve similar performance when transfer to the target model (from 6.2 to 7.6). Here, LR is the best as it shows a trade-off between the metrics considered. Looking at specific settings, it can be observed that they do not achieve accuracy differences greater than 10, which occurs



	LR		SVM		RF		average	
	Samples(%)	Accuracy	Samples(%)	Accuracy	Samples(%)	Accuracy	Samples(%)	Accuracy
DWB	33	7	30	6	37	6	33,3	6,3
IR	62	3	63	3	87	3	70,7	3,0
PWWS	75	10	32	6	38	7	48,3	7,7
TB	38	9	37	7	44	7	39,7	7,7
TF	49	9	48	9	62	10	53,0	9,3
average	51,4	7,6	42,0	6,2	53,6	6,6		

Figure 4: Percentage of samples that fooled the target and Accuracy Difference for each setting in the **configuration (e)**.

when the (*surrogate*, *attack*) pairs are (LR, PWWS) and (RF, TF) respectively. In contrast, the worst strategy is IR as the accuracy is only decremented by 3 regardless of the surrogate model adopted.

More insights can be obtained by analyzing the right part of Figure 4, where the *average* of both metrics for each attack is depicted. In terms of *Accuracy Difference*, the best attack is TF which allows to obtain a decrement of 9.3 on the target model; PWWS and TB are characterized by similar results. The outcome confirms the findings already obtained (Fig. 3) for the configurations (c) and (d), except for IR. In fact, it has, on average, a very high percentage of samples that fool the surrogate (i.e., successfully attacks on the surrogate), but a limited effectiveness in decreasing the accuracy of the target. This result can be attributed to the random nature of IR, which logic is to prioritize random word deletions over targeted perturbations, unlike other attacks that strategically modify the content of the text to deceive the model.

The very last analysis concerns the focus of the proposed work, i.e., the number of queries made on the target model. Recall that, in a real-world scenario, making an innumerable amount of queries to the remote system could activate defense mechanisms that could compromise the attacker's purposes. Table 3 summarizes the average number of queries for three of configurations so long discussed in this section. For the **configurations (c)**, the highest and lowest number of queries is made by PWWS and IR, while they are TF and IR for the **configurations (d)**. As might be expected (also, referring to Fig. 3), the techniques that are characterized by a higher number of queries are also those that are more likely to be successful in fooling the model, be it the surrogate or the target. The only exception is always IR, which as always is characterized by a random behavior.

To understand the benefits of the proposed solution, these values must then be compared with those of our proposal. In particular, the comparison must consider the results of **configuration (c)** with those of the "Surrogate" column in **configuration (e)** and, likewise, the values of **configuration (d)** with the "Target" column in **configuration (e)**. In the first case, the findings from the comparison are not surprising since, on average, similar results are obtained as before: PWWS continues to be the technique with the most queries, and IR the worst. In the second case, however, it can be seen that the proposed approach performs on

	configuration (c)			configuration (d)	configuration (e)	
Attack	**LR**	**SVM**	**RF**	**NN**	**Surrogate**	**Target**
PWWS	**95.77**	**132.82**	**90.44**	79.36	**104.32**	**1**
TF	64.42	112.20	65.89	**113.92**	82.72	**1**
TB	39.22	64.99	40.08	39.70	49.11	**1**
IR	19.62	26.63	19.59	13.73	20.59	**1**
DWB	23.22	36.43	24.10	23.70	28.11	**1**

Table 3: Average number of queries for **configuration (c)**, **configuration (d)**, and **configuration (e)**. In the last case, the queries on surrogate and target are considered separately.

average only one query to the target as opposed to **configuration (d)**. This result is justified by the inherent nature of the proposed approach, which (i) starts from the original test set of n samples, (ii) tries to perturb each sample via the chosen attack strategy, and (iii) composes a new test set of n samples consisting of both the texts that did not fool the surrogate and those that were perturbed. Finally, the generated test set is tested on the target, then a query for each sample.

5 Conclusions

In this paper, we have presented a black-box approach for attacking machine learning models. Central to the proposed method is the introduction of a surrogate model situated between the attacker and the target model. This intermediary model acts as a medium for efficiently transferring offline attacks in minimal queries, ensuring successful adversarial manipulation while minimizing resource consumption and potential disruption to operational systems.

In the experimental phase, we leveraged various attack strategies with the aim of introducing distinct perturbations, including synonym replacement for most words and nuanced character-level alterations. The latter can be susceptible to mitigation by spell checker systems in numerous systems; while the former, which still worked fine for an improved approach-substituting antonyms for synonyms. This strategy has the potential to fabricate news articles that deviate significantly from their original content, thereby substantiating the generation of deceptive or counterfeit news while maintaining semantic correctness as well.

For future investigations, an exploration or creation of an alternative attack strategy is warranted. Furthermore, it should be acknowledged that the attacks encounter limitations failing to perturb all the samples in our dataset. Consequently, the performance of the target attack is diminished to a limit. As a general trend, the efficacy of the attack diminishes proportionally with the perturbed samples. Thus, for robust evaluations, a comprehensive perturbation coverage is required.

References

[1] H. Ali, M. S. Khan, A. AlGhadhban, M. Alazmi, A. Alzamil, K. Al-Utaibi, and J. Qadir. All your fake detector are belong to us: evaluating adversarial robustness of fake-news detectors under black-box settings. *IEEE Access*, 9:81678–81692, 2021.

[2] B. Bhutani, N. Rastogi, P. Sehgal, and A. Purwar. Fake news detection using sentiment analysis. In *2019 twelfth international conference on contemporary computing (IC3)*, pages 1–5. IEEE, 2019.

[3] B. Biggio and F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 2154–2156, New York, NY, USA, 2018. Association for Computing Machinery.

[4] D. Choudhury and T. Acharjee. A novel approach to fake news detection in social networks using genetic algorithm applying machine learning classifiers. *Multimedia Tools and Applications*, 82(6):9029–9045, Mar 2023.

[5] F. Concone, A. De Paola, G. Lo Re, and M. Morana. Twitter analysis for real-time malware discovery. In *2017 AEIT International Annual Conference*, pages 1–6, 2017.

[6] F. Concone, S. Gaglio, A. Giammanco, G. L. Re, and M. Morana. Adverspam: Adversarial spam account manipulation in online social networks. *ACM Trans. Priv. Secur.*, 27(2), mar 2024.

[7] F. Concone, G. Lo Re, M. Morana, and S. K. Das. Spade: Multi-stage spam account detection for online social networks. *IEEE Transactions on Dependable and Secure Computing*, 20(4):3128–3143, 2023.

[8] F. Concone, G. Lo Re, M. Morana, and C. Ruocco. Assisted labeling for spam account detection on twitter. In *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 359–366, 2019.

[9] F. Concone, G. Lo Re, M. Morana, and C. Ruocco. Twitter spam account detection by effective labeling. In *ITASEC*, 2019.

[10] L. Demetrio, B. Biggio, and F. Roli. Practical attacks on machine learning: A case study on adversarial windows malware. *IEEE Security & Privacy*, 20(5):77–85, 2022.

[11] A. Dharawat, I. Lourentzou, A. Morales, and C. Zhai. Drink bleach or do what now? covid-hera: A study of risk-informed health decision making in the presence of covid-19 misinformation. *Proceedings of the International AAAI Conference on Web and Social Media*, 16(1):1218–1227, May 2022.

[12] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramèr, A. Prakash, T. Kohno, and D. Song. Physical adversarial examples for object detectors. In *Proceedings of the 12th USENIX Conference on Offensive Technologies*, WOOT'18, page 1, USA, 2018. USENIX Association.

[13] S. Feng, E. Wallace, A. Grissom II, M. Iyyer, P. Rodriguez, and J. Boyd-Graber. Pathologies of neural models make interpretations difficult. *arXiv preprint arXiv:1804.07781*, 2018.

[14] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane. Adversarial attacks on medical machine learning. *Science*, 363(6433):1287–1289, 2019.

[15] S. Gaglio, A. Giammanco, G. Lo Re, and M. Morana. Adversarial machine learning in e-health: Attacking a smart prescription system. In *AIxIA 2021 – Advances in Artificial Intelligence*, pages 490–502, Cham, 2022. Springer International Publishing.

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

[17] B. Ghanem, S. P. Ponzetto, P. Rosso, and F. Rangel. Fakeflow: Fake news detection by modeling the flow of affective information. In *Conference of the European Chapter of the Association for Computational Linguistics*, 2021.

[18] K. He, D. D. Kim, and M. R. Asghar. Adversarial machine learning for network intrusion detection systems: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, pages 1–1, 2023.

[19] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025, 2020.

[20] C. Koenders, J. Filla, N. Schneider, and V. Woloszyn. How vulnerable are automatic fake news detection methods to adversarial attacks? *arXiv preprint arXiv:2107.07970*, 2021.

[21] T. Le, S. Wang, and D. Lee. Malcom: Generating malicious comments to attack neural fake news detection models. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages

282–291, 2020.

[22] J. Li, S. Ji, T. Du, B. Li, and T. Wang. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*, 2018.

[23] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp, 2020.

[24] J. A. Nasir, O. S. Khan, and I. Varlamis. *International Journal of Information Management Data Insights*, 1(1):100007, 2021.

[25] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, page 506–519, New York, NY, USA, 2017. Association for Computing Machinery.

[26] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics.

[27] T. Rasool, W. H. Butt, A. Shaukat, and M. U. Akram. Multi-label fake news detection using multi-layered supervised learning. In *Proceedings of the 2019 11th international conference on computer and automation engineering*, pages 73–77, 2019.

[28] S. Ren, Y. Deng, K. He, and W. Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1085–1097, 2019.

[29] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu. Fake news detection on social media: A data mining perspective. *SIGKDD Explor. Newsl.*, 19(1):22–36, sep 2017.

[30] M. Usama, M. Asim, S. Latif, J. Qadir, and Ala-Al-Fuqaha. Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems. In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 78–83, 2019.

[31] X. Zeng, A. S. Abumansour, and A. Zubiaga. Automated fact-checking: A survey. *Language and Linguistics Compass*, 15(10):e12438, 2021.

[32] Y. Zhu, Q. Sheng, J. Cao, Q. Nan, K. Shu, M. Wu, J. Wang, and F. Zhuang. Memory-guided multi-view multi-domain fake news detection. *IEEE Transactions on Knowledge and Data Engineering*, 2022.