Crafting Adversarial Email Content against Machine Learning Based Spam Email Detection

Chenran Wang*
cwang162@alumni.jh.edu
Johns Hopkins University
Baltimore, Maryland, USA

Danyi Zhang*
dzhang66@alumni.jh.edu
Johns Hopkins University
Baltimore, Maryland, USA

Suye Huang*
shuang93@alumni.jh.edu
Johns Hopkins University
Baltimore, Maryland, USA

Xiangyang Li
xyli@jhu.edu
Johns Hopkins University
Baltimore, Maryland, USA

Leah Ding
ding@american.edu
American University
Washington, DC, USA

ABSTRACT

While machine learning based spam detectors have proven useful, spammers are learning to bypass the detectors by modifying their email content. Adversarial attacks on machine learning models have been observed in domains such as image classification. Applying such adversarial attack algorithms to craft spam emails to evade spam email detectors, however, has limitations. Such algorithms generate adversarial perturbations in the feature space. Different from image data, translating the adversarial perturbations from the feature space to text formats, as in emails, changes the effectiveness of the adversarial perturbations. It can reduce the attack success rate in the case of spam email detection. In this paper, we study the feasibility of adversarial attacks on machine learning based spam detectors and propose two novel text crafting methods leveraging adversarial perturbations generated by the adversarial example generation algorithms to improve the attack effectiveness. One method tries to approximate the feature values and the other adds special words to original emails. In experimentation, we use PGD as an example to demonstrate and compare the effectiveness of our attack methods on spam email detectors. We also examine the transferability of the proposed attack methods on different machine learning models.

CCS CONCEPTS

• **Security and privacy → Phishing**; • **Computing methodologies →** *Machine learning*.

KEYWORDS

Spam Detection, Adversarial Machine Learning, Crafting Adversarial Email, Attack Transferability

ACM Reference Format:
Chenran Wang, Danyi Zhang, Suye Huang, Xiangyang Li, and Leah Ding. 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), June 7, 2021, Virtual Event, Hong Kong.* https://doi.org/10.1145/3457340.3458302

1 INTRODUCTION

Machine learning classification models have been employed for spam email detection, such as Support Vector Machine (SVM) models [9] [16], Bayesian classification models [4] [18], KNN [10], and Decision Tree Boosting [7]. While machine learning models differ, their basic working pipelines are similar. Essentially, a set of features based on words and phrases are extracted from the emails, and this set of features are used as the representative input to the machine learning models for detection.

With the recent research of adversarial attacks on machine learning models, spam detectors face new security risks introduced by their machine learning based classifiers. In a successful adversarial attack, the targeted classifier could be misled by an input that is created by adding a slight perturbation or disturbance to the original feature input.

Adversarial attacks on machine learning models have been studied in domains such as image classification [11][15] [5] and Voice Processing Systems [6] [20] [17]. Different adversarial attack algorithms have been proposed, such as Fast Gradient Sign Method (FGSM) [11], Projected Gradient Descent (PGD) [15], and Carlini and Wagner Attacks (CW) [5]. Such adversarial attacks have mainly focused on computer vision and pattern recognition systems that utilize deep learning models. In contrast, less attention is given to computer security applications that process other types of data, such as the text content of an email used by spam filters. Classical machine learning models deserve careful examination for their resistance to such adversarial attacks. Moreover, the translation of an adversarial attack in the feature space back to a realistic event is critical to completing a realistic attack process.

In this paper, we use spam email detection as an example to study the adversarial attacks on text classification models. Specifically, we make the following contributions:

- We showed that even if adversarial perturbations are generated successfully in the feature space, e.g., the TF-IDF (term frequency–inverse document frequency) representation of

[*] All authors contributed equally to this research.

ASSS '21, June 7, 2021, Virtual Event, Hong Kong

ACM ISBN 978-1-4503-8403-2/21/06…$15.00
https://doi.org/10.1145/3457340.3458302

an email in this study, further effort is needed to construct the attack in its valid form, e.g., text in a spam email, without degrading the attack effectiveness. Different from image data, translating perturbed values in terms of TF-IDF values from the feature space to text changes the effectiveness of the adversarial perturbations. And, if not carefully done, it may reduce the attack success rate in the case of spam email detection.

- We proposed two methods to craft spam email content by analyzing the TF-IDF features of adversarial perturbations. The first method approximates the desirable TF-IDF values by adjusting the occurrences of each word of an email. The second method inserts a set of "special words", that are identified by examining the generating adversarial perturbations, to a spam email. We compared the two methods in experimentation to show that the second method is more efficient in evading spam detectors.
- We studied the effectiveness of black-box attacks on spam detectors, where the target spam detector employs different classification models. We examined the transferability of the attacks generated from one classification model, i.e., SVM classifier, to other classifiers, including KNN, decision tree, and logistic regression models. With the experimentation using the Enron-Spam email dataset, we found that the adversarial email content crafted against one model can be transferred to different models.

2 RELATED WORK

Wittel and Wu [19] categorized the attacks on spam email detectors into three types, *tokenization attacks* where spammers intend to disturb tokenization of the email content by splitting or modifying features, such as inserting extra spaces in the middle of the words; *obfuscation attacks* where the email content is obscured from the detector using encoding or misdirection; and *statistical attacks* where spammers attempt to skew the message's statistics to distract the detector. An example of a statistical attack is proposed in [14], where the attacks were developed and tested against two types of statistical spam detectors: maximum entropy and naive Bayes filters. In this paper, active attacks have much better attack results as the attacker is allowed to send test messages to the filter to determine whether or not they are labeled as spam. Other examples of statistical attack include [12] [13].

In this paper, we focus on crafting spam email content based on adversarial perturbations made in the feature space. Adversarial attack algorithms have been studied against image classification models, rather than text classification models. Different from image data, translating the adversarial perturbations from the feature space to text formats changes the effectiveness of the adversarial perturbations, which reduces the attack success rate in the case of the spam email detection. We address such limitations by two novel text crafting methods leveraging adversarial perturbations generated by the adversarial example algorithms. We use PGD as an example to demonstrate the effectiveness of our attack methods on spam email detectors.

3 ATTACK METHODOLOGY

In Figure 1, the TF-IDF vector was calculated for every email. Then we trained and validated different classifiers with these TF-IDF features as inputs. PGD was run on the trained SVM classifier to perturb a subset of randomly selected spam emails, which created a set of adversarial perturbations in forms of TF-IDF vectors. Based on these vectors, we applied two methods to craft the adversarial emails, i.e., determining their words accordingly. Once done, we recalculated the TF-IDF vectors of the resulting emails. At last, white-box and black-box attacks were launched by feeding the TF-IDF vectors, before and after crafting the adversarial emails, to the trained classifiers to evaluate their susceptibility.

3.1 Generating Adversarial Perturbations in the Feature Space

TF-IDF Calculation

Calculating TF-TDF of words appearing in an email is a common method to vectorize textual information into numeric values:

$$\begin{aligned} TF_{i,j} &= \frac{N_{i,j}}{\sum_k N_{k,j}} \\ IDF_i &= \log \frac{|D|+1}{|j : t_i \in d_j| + 1} + 1 \end{aligned} \tag{1}$$

where $N_{i,j}$ is the number of times word t_i appears in email d_j; $|D|$ is the total number of email in the corpus; $|j : t_i \in d_j|$ indicates the number of emails containing the term t_i. The IDF term is smoothed for those common words appearing in every document. The higher frequency of a word in a particular file and the lower file frequency of the word in the entire file collection results in a higher TF-IDF value, which reflects the significance of the word or feature used in the classification model.

Support Vector Machine (SVM) Model

A SVM classifier expands the original data dimensions to separate the samples in the transformed high-dimensional space [8]. If the number of features is much larger than the number of samples, a logistic regression or linear kernel is recommended to avoid overfitting the SVM model. Our experimentation had about 100,000 TF-IDF features with about 50,000 samples, so we chose a linear kernel. In addition, in training, the larger the penalty coefficient being used, the greater the penalty for the wrong sample.

Projected Gradient Descent (PGD)

This iterative algorithm finds the disturbance with a constraint, *dmax* that is the Euclidean distance to the original input, to achieve the maximum loss in classification [15]. In our approach we ran PDG over a set of spam emails randomly selected that each generated an adversarial example in a TF-IDF vector. Then we test them to see whether they can successfully evade the classifiers.

3.2 Crafting the Content of Adversarial Emails

Aided by the adversarial TF-IDF vector, the aim of this paper is to create valid spam emails, in forms of a set of words. This is critical for spammers to achieve their goal in the real world.

Method 1 - TF-IDF Approximation

In this method, we try to determine the term frequencies in an email to make their TF-IDF vales as close as possible to those of a successful adversarial example that can bypass the SVM classifier.

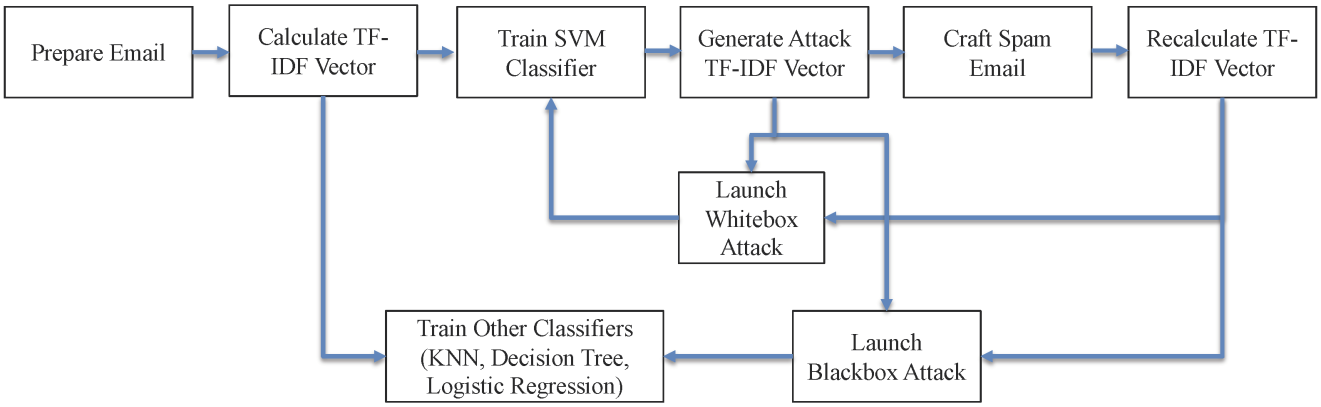
Prepare Email
Calculate TF-IDF Vector
Train SVM Classifier
Generate Attack TF-IDF Vector
Craft Spam Email
Recalculate TF-IDF Vector
Launch Whitebox Attack
Train Other Classifiers (KNN, Decision Tree, Logistic Regression)
Launch Blackbox Attack

Figure 1: The Workflow of Generating Adversarial Perturbations, Crafting Adversarial Emails, and Conducting Attacks

We only considered those TF-IDF features significant enough and extracted the 8700 feature words of the largest TF-IDF values for use.

Here we use A, B, C... to represent a set of words targeted for their higher TF-IDF values, and X to represent the set of all remaining words with lower significance in their TF-IDF values in the adversarial examples generated by PGD. Our goal is to approximate the resulting TF-IDF values, focusing on the targeted words, of the crafted email to those in the adversarial perturbations as well as we can. Let a be the TF-IDF value of word A in the adversarial example, similarly for the words B, C... and X. Here, we define the F_A, F_B,... as the number of word A, B and so on. However, note that F_X is actually the total number of all the words in X.

$$\begin{aligned} \frac{F_A}{F_A+F_B+F_C+...+F_X} * IDF_A = a \\ \frac{F_B}{F_A+F_B+F_C+...+F_X} * IDF_B = b \\ \frac{F_C}{F_A+F_B+F_C+...+F_X} * IDF_C = c \\ \cdots \end{aligned} \tag{2}$$

Then, using b_1, c_1,... to simplify the terms, we get the TF ratio between word A and every other word:

$$\begin{aligned} F_B = \frac{b}{IDF_B} * \frac{IDF_A}{a} * F_A = b_1 * F_A \\ F_C = \frac{c}{IDF_C} * \frac{IDF_A}{a} * F_A = c_1 * F_A \\ \cdots \end{aligned} \tag{3}$$

Later we can decide on a specific word A, called the "anchor" word, which is associated with a particular TF-IDF feature value in the adversarial example, to derive the number of every other word. We can further obtain the following equation:

$$\frac{F_A}{F_A+F_B+F_C+\cdots+F_X} = \frac{a}{IDF_A} \tag{4}$$

$$\Rightarrow \frac{F_A}{F_A+b_1*F_A+c_1*F_A+\cdots+F_X} = \frac{a}{IDF_A} \tag{5}$$

$$\Rightarrow \frac{1}{(1+b_1+c_1+\cdots)+\frac{F_X}{F_A}} = \frac{a}{IDF_A} \tag{6}$$

We need to simplify the calculation for words in X while approximating the TF-IDF values of the significant words as much as we can. We chose to have the same number of occurrences, denoted as y, for each word in X in the email. So we have $F_X = |X| * y$. We finally have the following:

$$\frac{y}{F_A} = \frac{IDF_A}{a|X|} - \frac{(1+b_1+c_1+\cdots)}{|X|} \tag{7}$$

At this point, the problem changes to selecting y and F_A, both integers as numbers of words, so that they satisfy the above equation as close as possible. We limited the search of y to 0-1000 and F_A to 0-5000. Once the best numbers of y and F_A are determined we can calculate the term frequency of each target feature or each remaining word accordingly.

The last question is, given an adversarial example, which word should be the anchor word A? We tried different options, using the largest or smallest TF-IDF value, different TF-IDF quantile points, or the average TF-IDF value. This choice will affect the total number of words, i.e., the size of the crafted email. We found that when the 25th-quantile point is used, the success rate of adversarial emails is the highest.

Method 2 - Adding Special Words

In this method, we try to modify an original spam email by adding words that may change its classification. For this purpose, we examined how much the TF-IDF values were modified by the PGD algorithm to measure the importance of each word. In doing so, we only looked at successful adversarial perturbations, those being classified as ham emails.

We compared the TF-IDF values of samples to their original ones and identified the following word feature sets:

- Top 100 features (set 1): The 100 features with the largest value changes by PGD. This is based on the variance calculation for each word with the mean at 0 since we are interested in those word changed the most.
- Disappearing features (set 2): The set of words whose TF-IDF values are ever reduced to 0 from their original values.
- Appearing features (set 3): The set of words whose TF-IDF values are ever increased to non-zero from 0.
- Unique ham features (set 4): Words that only appear in ham emails.
- Unique spam features (set 5): Words that only appear in spam emails.

Then we found their intersection sets:

- Intersection (1,4): not empty (set of *"magic words"*)
- Intersection (1,5): empty
- Intersection (1,2): empty
- Intersection (1,3): set 1

There are several interesting observations. The top 100 features of significant changes through the PGD process are not unique spam words, nor those words being removed. These features are all words being added and some of them are unique ham words. We call the words in Intersection (1,4) *"magic words"*. We suspect that these words have special effect in adversarial attacks. And we added these special words to spam emails to study their impact on classification.

4 EXPERIMENTAL EVALUATION

4.1 Experiment Settings

Dataset

We used 17,171 spam emails and 16,545 normal emails from the Enron-Spam dataset [1]. Messages sent by the owner of the mailbox, all HTML tags, the headers of the messages, and spam messages written in non-Latin character sets have been excluded before analysis. We removed all the numbers and special symbols as well as all the English stop words in emails to reduce the complexity of subsequent processing. Furthermore, as a common measure, we aggregated words through stemming. Lastly, we replaced the URL link in each email by the word 'URL'.

Building SVM Classifiers

We randomly divided the emails into a training set and test set based on the ratio of 4:1. We directly called the TF-IDF method in the Sklearn library [2]. It resulted in over 100,000 feature words. Training the SVM classifier was in the SecML library[3] in order to use its PGD solver. Calling the method "best_estimate()" selected the best penalty factor at 10. The SVM classifier can achieve 99.08% accuracy in testing with 1.11% false positive rate and 0.47% false negative rate. This means that fewer than one spam email can bypass the SVM classifier without being flagged.

Generating Adversarial Perturbations

In order to generate adversarial perturbations, we chose the projected gradient descent (PGD) algorithm in the SecML library. We randomly selected 100 spam emails from the test dataset since we were interested in inducing detection misses. The larger the *dmax* setting, the greater the possibility of success of changing its classification. Therefore, we select three different settings of 0.09, 0.07, and 0.05.

White-box and Black-box Attacks

For white-box attacks, we fed the TF-IDF vectors extracted from the crafted adversarial emails to the trained SVM classifier. We evaluated the attack success rate against the trained SVM classifier.

For black-box attacks, we fed the TF-IDF vectors extracted from the crafted adversarial emails to three new spam detectors to evaluate the attack success rate against different classification models that we do not have access to when crafting the adversarial emails. Specifically, we tested KNN, Decision Tree, and Logistic Regression classification models. To train those classification models, we randomly selected 80% of the emails from the Enron-Spam dataset as the training set, and check the accuracy of those classifiers on the test set. We again called the method "best_estimate()" for the best model parameters:

- KNN Model: K is set at 160.
- Decision Tree Model: The tree depth is 55.
- Logistic Regression Model: alpha is set to $1e-6$ and regularizer is "l2".

We tested the attack success rate against those three models to evaluate the attack transferability.

4.2 Results

White-box Attacks with TF-IDF Approximation

The attack performance on the SVM classifier is shown for PGD and TF-IDF approximation respectively in Table 1. Note that the denominator in the last column is the number of successful adversarial perturbations generated by PGD. For example, when *dmax* is set at 0.09 for PGD, 91 of the 100 generated adversarial samples successfully flipped their classifications to ham. Then, TF-IDF approximation was done on the successful adversarial examples and we experimented with 3 choices of target features: top 500 features, top 1000 features, and top 5000 features with regard to their TF-IDF values. When we used 500 top features for TF-IDF approximation, 75 of the 91 adversarial emails did not raise an alert.

As shown, when *dmax* increases, the attack success rate increases too. Moreover, as the number of target features in TF-IDF approximation increases, the attack success rate becomes higher. This is because the TF-IDF values of crafted emails are closer to the TF-IDF values of adversarial perturbations overall when more target features are used in calculation. So, these crafted emails have a better chance to succeed as much as these successful adversarial perturbations do.

However, the TF-IDF approximation method presents two challenges. First, the adversarial email being crafted ends up with a large volume of words, for example, there are more than 50 words that need to appear very frequently in the crafted spam email to be able to bypass the spam filter in our experimental dataset. This is not practical to implement in the real world. Second and more importantly, it is hard to validate whether the resulting email is still a spam email. Additional mechanisms, likely manually done, are needed to verify that.

Table 1: Result of TF-IDF Approximation

dmax	Success Rate of PGD Attack	Number of Target Features	Success Rate of TF-IDF Approximation
0.09	91/100	500	75/91
		1000	81/91
		5000	91/91
0.07	76/100	500	54/76
		1000	69/76
		5000	76/76
0.05	38/100	500	13/38
		1000	18/38
		5000	38/38

White-box Attacks with Adding Special Words

Similar to a 5-fold cross-validation approach, we repeated the process to retrain the SVM classifier, choose different *dmax* for PGD, and generate attack perturbations using different sets of spam emails. We found that even with different *dmax* settings and spam emails, we got the same set of *magic words* for the same classifier. But the SVM classifiers trained with different datasets resulted in different words for this set. We suspect that the *magic words* are dependent on the classifier, which means their performance is sensitive to the dataset being used.

We added these words into all the spam emails in the dataset and recalculated their TF-IDF values as input to the corresponding SVM classifier. In a similar way, to further examine these words, we identified the intersection and union sets of the five sets of *magic words* and tested their performance on the SVM classifier trained in the first repetition.

- Intersection Set (8 words): *listbot, clickathom, ena, sitara, cera, enrononlin, kaminski, calger*
- Union Set (21 words): *ferc, listbot, jhherbert, lokay, eyeforenergi, erisk, counterparti, ena, sitara, topica, kal, calger, beenladen, aggi, clickathom, cdnow, wassup, cera, enrononlin, pjm, kaminski*

As shown in Table 2, adding these *magic words* to original spam emails is fairly successful in evading detection. The success rate varies from over 75% to over 87% in the five repetitions. When we use a smaller number of words in the intersection set, the success rate is lowered to below 53%. But if we use the words in the union set, the success rate is over 88%, the highest among all. More importantly, this method of adding a few words does not change the validity of a spam email in nature. It is relatively easy to hide these words, especially if the original email is long, by embedding them in the text or the fine print part of an email.

Black-box Attacks

In this experiment, we assume that we do not have access to any of the classification models during the process of spam email crafting, i.e., we do not use any information about the black-box models (KNN, Decision Tree, Logistic Regression). We first extracted TF-IDF values from the crafted spam emails, which were obtained from the white-box attacks (more specifically as in Table 1 and the first repetition in Table 2). We then fed the extracted TF-IDF values to the black-box models and evaluated the attack success rate in terms of bypassing those models. For simplification, we show the results in Table 3 for setting *dmax* at 0.09 for PGD, using 5000 target features words in TF-IDF approximation, and using the union set of *magic words* for adding special words. The success rate of adversarial perturbations is based on the 91 adversarial perturbations that can successfully attack the SVM classifier, same with the success rate of TF-IDF approximation. The success rate of adding special words is over the selected 100 spam emails.

Based on the results, the KNN classifier has the strongest resistance to the attacks. This is especially true against adversarial perturbations generated by PGD algorithm as well as the emails through TF-IDF approximation. It can be explained that, compared to the other classifiers, the KNN classifier stores all the original training examples and uses them in classification, where the decision boundary has fine granularity. So, the adversarial examples relying on gradient-based search in a different model may not well account for the local intricacies of decision-making. However, adding special words still caused 34 crafted emails to pass through detection, outperforming TF-IDF approximation. This is an approach that in general pulls an email, by adding those words, in the direction to where ham emails reside in the feature space.

Decision tree and logistic regression classifiers are more susceptible to the attacks. And overall, logistic regression has stronger resistance than decision tree. This could be due to the fact that decision tree takes a step-wise approach in using the features. Compared to logistic regression, the decision is impacted more significantly by a subset of features. Therefore, adversarial attacks on a decision tree model can be "less sophisticated" by focusing on these significant features to navigate in the feature space. However, for the logistic regression model, more (or all) features, even if they may have different weights, are involved at the same time in making the classification decision. And they need to be handled together in successful adversary attacks.

Attack transferability is demonstrated among these models in the results. Moreover, the method of adding special words is more effective in terms of the attack success rate against all the three classifiers. However, as discussed before, this method may be sensitive to the dataset used for training a classifier. Therefore, further study on its effectiveness to use other datasets is needed.

Table 2: Result of Adding Special Words in Five Repetitions (Total Number of Spam Emails = 17,171)

"Magic Words"	Number of Successful Adversarial Emails	Success Rate
counterparti,clickathom,topica,listbot,sitara,cera,wassup,enrononlin calger,eyeforenergi,kaminski,pjm,ena	13026	75.86%
beenladen,pjm,ena,ferc,topica,lokay,erisk,calger,cera,enrononlin,cdnow listbot,aggi,kaminski,eyeforenergi,wassup,sitara,clickathom	13675	79.64%
beenladen,pjm,ena,ferc,topica,lokay,erisk,calger,cera,enrononlin,cdnow listbot,aggi,kaminski,eyeforenergi,wassup,sitara,clickathom	14976	87.22%
sitara,clickathom,cdnow,listbot,ferc,enrononlin,calger,lokay,beenladen wassup,kaminski,ena,cera,pjm,counterparti	13616	79.30%
sitara,clickathom,cdnow,listbot,enrononlin,lokay,calger,beenladen wassup,kaminski,kal,ena,cera,eyeforenergi,counterparti	13918	81.06%
Intersection Set	9051	52.71%
Union Set	15227	88.68%

Table 3: Result of Black-box Attacks

	Classification Accuracy	Detection Rate of 100 Spam Emails Selected	Success Rate of PGD Perturbation	Success Rate of TF-IDF Approximation	Success Rate of Adding Special Words
KNN	96.74%	100.00%	2.20%	2.20%	34.00%
Decision Tree	95.22%	94.00%	98.00%	98.90%	93.00%
Logistic Regression	99.02%	100.00%	89.01%	73.63%	80.00%

5 CONCLUSION AND FUTURE WORK

This paper studied two methods to craft the adversarial email content to evade spam detectors. With both based on the adversarial perturbations generated by PGD algorithm, the first method approximates the TF-IDF values in the resulting adversarial examples, and the second method identifies and uses a set of significant words. We evaluated both methods on various machine learning classification models, including SVM, KNN, decision tree, and logistic regression, in both white-box and black-box attack scenarios. It can be concluded that the second method is more effective.

Further work is certainly required to further investigate the spam filter vulnerabilities for which adversarial perturbation based exploits exist. For example, TF-IDF approximation can be improved in both the calculation precision to achieve and the validity of the crafted emails. Additional experimentation with other spam email databases and classifiers will gain more insights into these two methods. Moreover, the current black-box attacks are based on known data and feature extraction methods. Future research should be conducted in more realistic settings to study spam filters that are trained with different data and feature extraction methods. Lastly, how to perform black-box attacks on real-world spam filters is an interesting task in the next.

REFERENCES

[1] [n.d.]. Enron-Spam dataset. http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html

[2] [n.d.]. scikit-learn. https://scikit-learn.org/stable/

[3] [n.d.]. SecML: A library for Secure and Explainable Machine Learning. https://secml.gitlab.io/

[4] Ion Androutsopoulos, John Koutsias, Konstantinos V Chandrinos, George Paliouras, and Constantine D Spyropoulos. 2000. An evaluation of naive bayesian anti-spam filtering. *arXiv preprint cs/0006013* (2000).

[5] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 39–57.

[6] Nicholas Carlini and David Wagner. 2018. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 1–7.

[7] Xavier Carreras and Lluis Marquez. 2001. Boosting trees for anti-spam email filtering. *arXiv preprint cs/0109015* (2001).

[8] Pai-Hsuen Chen, Chih-Jen Lin, and Bernhard Schölkopf. 2005. A tutorial on ν-support vector machines. *Applied Stochastic Models in Business and Industry* 21, 2 (2005), 111–136.

[9] Harris Drucker, Donghui Wu, and Vladimir N Vapnik. 1999. Support vector machines for spam categorization. *IEEE Transactions on Neural networks* 10, 5 (1999), 1048–1054.

[10] Loredana Firte, Camelia Lemnaru, and Rodica Potolea. 2010. Spam detection filter using KNN algorithm and resampling. In *Proceedings of the 2010 IEEE 6th International Conference on Intelligent Computer Communication and Processing*. IEEE, 27–33.

[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[12] Zach Jorgensen, Yan Zhou, and Meador Inge. 2008. A Multiple Instance Learning Strategy for Combating Good Word Attacks on Spam Filters. *Journal of Machine Learning Research* 9, 6 (2008).

[13] Bhargav Kuchipudi, Ravi Teja Nannapaneni, and Qi Liao. 2020. Adversarial machine learning for spam filters. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*. 1–6.

[14] Daniel Lowd and Christopher Meek. 2005. Good Word Attacks on Statistical Spam Filters.. In *CEAS*, Vol. 2005.

[15] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

[16] Sunday Olusanya Olatunji. 2017. Extreme Learning machines and Support Vector Machines models for email spam detection. In *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 1–6.

[17] Yao Qin, Nicholas Carlini, Garrison Cottrell, Ian Goodfellow, and Colin Raffel. 2019. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *International Conference on Machine Learning*. PMLR, 5231–5240.

[18] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. 1998. A Bayesian approach to filtering junk e-mail. In *AAAI Workshop on Learning for Text Categorization*, Vol. 62. Madison, Wisconsin, 98–105.

[19] Gregory L Wittel and Shyhtsun Felix Wu. 2004. On Attacking Statistical Spam Filters.. In *CEAS*. Citeseer.

[20] Hiromu Yakura and Jun Sakuma. 2018. Robust audio adversarial example for a physical attack. *arXiv preprint arXiv:1810.11793* (2018).