Article

A Survey of Attacks Against Twitter Spam Detectors in an Adversarial Environment

Niddal Imam [1]

[1] University of York; ni571@york.ac.uk
* Correspondence: ni571@york.ac.uk; Tel.: +44-7447-831-899

Received: date; Accepted: date; Published: date

Abstract: Online Social Networks (OSNs), such as Facebook and Twitter, have become a very important part of many people's daily lives. Unfortunately, the high popularity of these platforms makes them very attractive to spammers. Machine-learning (ML) techniques have been widely used as a tool to address many cybersecurity application problems (such as spam and malware detection). However, most of the proposed approaches do not consider the presence of adversaries that target the defense mechanism itself. Adversaries can launch sophisticated attacks to undermine deployed spam detectors either during training or the prediction (test) phase. Not considering these adversarial activities at the design stage makes OSNs' spam detectors prone to a range of adversarial attacks. This paper thus surveys the attacks against Twitter spam detectors in an adversarial environment. In addition, a general taxonomy of potential adversarial attacks is proposed by applying common frameworks from the literature. Examples of adversarial activities on Twitter were provided after observing Arabic trending hashtags. A new type of spam tweet (*Adversarial spam tweet*), which can be used to undermine deployed classifier, were found. In addition, possible countermeasures that could increase the robustness of Twitter spam detectors against such attacks are investigated.

Keywords: Twitter spam detection; Adversarial machine learning; Online Social Networks; Survey.

---

1. Introduction

Online Social Networks (OSNs), such as Facebook, WhatsApp, and Twitter have become a very important part of daily life. People use them to make friends, communicate with each other, read the news, and share their stories. The amount of information shared in these OSNs has continued to increase over the past few years. One study shows that the number of profiles on Facebook, Twitter, and LinkedIn reached more than two billion in 2016 [3].

Unfortunately, the high popularity of these OSNs has made them very attractive to malicious users, or spammers. Spammers spread false information, propaganda, rumors, fake news, or unwanted messages [29]. Spam is referred to as an unsolicited message that is received from a random sender who has no relationship with the receiver. These messages may contain malware, advertisements, or URLs directing the recipients to malicious websites [6]. Spamming on the Internet first appeared in the 1990s in the form of email spam [3]. Although spam is prevalent in all forms of online communication (such as email and the web), researchers' and practitioners' attention has increasingly shifted to spam in OSNs due to the growing amount of spammers and the possible negative effects on users [45, 6].

The first appearance of spam on Facebook was in 2008, while the first Twitter spam attack, in which a number of Twitter accounts were hacked to spread advertisements, was in 2009 [50, 57]. On Twitter, spammers tweet for several reasons, such as to spread advertisements, disseminate pornography, spread viruses, phishing, or simply just to compromise a system's reputation. [7]. Furthermore, [26] added that a tweet is considered spam if it is not composed purely of text. Instead, it may contain a hashtag, a mention, a URL or an image. Various types of spam are found in OSNs, including textual pattern spam [58], image spam [8, 10], URL-based spam [53], and phone number-

 doi:10.20944/preprints201905.0141.v1

based spam [30]. Whilst most previous studies have focused on detecting the above types of spam, few have attempted to detect advertisement spam. The authors in [44] categorized adversarial advertisements as: counterfeit goods, misleading or inaccurate claims, phishing, arbitrage, and malware. The diversity of spam in OSNs makes it very hard for any single existing method to detect most spam [27]. Several reported incidents show the danger of spammers in OSNs. For example, a number of NatWest bank customers were victims of a phishing attack on Twitter that used spam tweets that looked very similar to those from the official NatWest customer support account [3]. A recent study noted that the increase in the number of OSN spammers, who distribute unsolicited spam and advertise untrustworthy products, has an effect on the public's perception of companies, which can eventually lead to people's opinion becoming biased [24].

The issue of spamming over OSNs has become an area of interest for many researchers. Many solutions have been proposed to detect spam using techniques such as blacklisting and whitelisting, Machine Learning (ML) and others. ML techniques have been shown to be effective when deployed to solve security issues in different domains, such as email spam filters, intrusion detection systems (IDSs), and malware detectors [21]. ML techniques aim to automatically classify messages as either spam or non-spam. Various OSN spam detectors have been developed using ML algorithms, including Supervised Vector Machine (SVM) [7], Random Forests (RF) [39, 55] and, more recently, Deep Neural Networks [6].

Despite the success of these algorithms in detecting spam, the presence of adversaries undermines their performance. These algorithms are vulnerable to different adversarial attacks because they were not designed for adversarial environments [48, 2, 12]. The traditional assumption of stationarity of data distribution in ML is that the dataset used for training a classifier (such as SVM or RF) and the testing data (the future data that will be classified) have a similar underlying distribution. This assumption is violated in the adversarial environment, as adversaries are able to manipulate data either during training or before testing [46, 2].

Studying the robustness of OSNs' spam detectors against adversarial attacks is crucial. The security of ML techniques is a very active area of research. Whilst several studies have examined the security of IDS, email filters, and malware detectors, few have investigated the security of OSNs' spam detectors. To the best of the researcher's knowledge, there is no survey of adversarial attacks against OSNs' spam detectors. Recent studies suggest that if a secure system is to be achieved, predicting potential attacks before they occur would help to develop suitable countermeasures [12]. Thus, the main goal of this paper is to present a comprehensive overview of different possible attacks, which is the first step towards evaluating the security of OSNs' spam detectors in an adversarial environment. This paper provides a general survey of the possible adversarial attacks against OSNs' spam detectors. In addition, potential defense mechanisms that could reduce the effect of such attacks are investigated. Ideas proposed in the literature were generalized to identify potential adversarial attacks and countermeasures. Twitter, which is one of the most popular OSN platforms, was used as a case study, and all examples of attacks were taken from Twitter. Examples of spam tweets that can be used by an adversary to attack Twitter spam detectors were provided. This kind of spam tweet is called Adversarial spam tweet.

The remainder of this survey is structured as follows: Section 2 describes previous research on Twitter spam detection. Section 3 provides an overview of adversarial machine learning. Section 4 surveys the adversarial attacks that could be used against Twitter spam detectors and presents a proposed taxonomy of such attacks. Section 5 briefly discusses possible defense strategies and countermeasures. The conclusion and future works are presented in Section 6.

2. Techniques for Twitter spam detection

Twitter and the research community have proposed a number of spam detectors to protect users. Twitter spam detection approaches can be divided into automated approaches, including machine learning, and non-automated approaches that require human interaction [55].

Researchers who use ML approaches build their models by employing some of the common spam detection techniques. Based on surveys in [36, 34], Twitter spam detectors can be classified into

 doi:10.20944/preprints201905.0141.v1

four categories: user-based, content-based, hybrid-based, and relation-based techniques. User-based techniques are also referred to as account-based and classify tweets based on an account's features and other attributes that provide useful information about users' behavior. Content-based techniques use the content of a tweet, such as the linguistic properties of the text or the number of hashtags in the tweet, for classification. Hybrid techniques use a combination of user-based and content-based features. The last category was proposed to detect spam in real-time, in contrast to user-based techniques, which can only detect spam after a message has been received. Relation-based techniques can detect a tweet immediately if it is received from an unknown sender. The features used in these techniques are distance and connectivity (see Table 1).

Table 1: Feature categories and description [34, 37].

Feature Category	Feature Name	Description
Account-based features	account_age	The number of days since the creation of an account.
	no_followers	The number of followers of an account.
	no_friends	The number of friends an account has.
	no_favorites	The number of favorites an account received.
	no_lists	The number of lists an account is a member of.
	no_reputation	The ratio of the number of followers and the total followers and friends of an account.
	no_statuses	The number of tweets an account has.
Tweet content-based features	no_words	The number of words in a tweet.
	no_chars	The number of characters in a tweet.
	no_hashs	The number of hashtags in a tweet.
	no_urls	The number of URLs in a tweet.
	no_phone	The number of phone numbers in a tweet.
	no_mentions	The number of mentions in a tweet.
Relation-based features	Distance	The length of the distance between accounts.
	Connectivity	The strength of the relationship between accounts.

Additionally, the authors in [56] categorized the methodologies used for detecting Twitter spam into three groups: syntax-based, feature-based, and blacklist-based detection (see Figure 1). Syntax-based detectors analyze the content of tweets, including linguistic features and shortened URLs, to determine whether the tweet is spam or non-spam. The second group, feature-based detectors, extract a set of statistical features from tweets to help utilized classifier to determine whether the tweet is spam or non-spam. This group uses a combination of techniques: account-based features, tweet-based features, and social graph features. Account-based features include account age and number of followers, while tweet-based features are the number of characters and the number of URLs. To overcome some of the weaknesses of account-based and tweet-based features, some recent studies have found that adopting a social graph to detect spam by analyzing mathematical features, such as social distance and connectivity between followers, is more robust. In the last group, blacklist-based detectors, accounts and tweets are blocked based on users' feedback or the URL's reputation. [28] presented the first study of the effectiveness of some of the techniques that have been used in the past to detect Twitter spam. Examples include spam behavior, clickthrough, and blacklists. The authors found that the blacklist methods (for example Google SafeBrowsing) are too slow at detecting new threats. They found that although 90% of victims visit spam URLs within the first two days, it would take four to 20 days for the URLs in spam tweets to be blacklisted. In another study, it was determined that blacklists can protect only a few users, and asserted that studying the regional response rate could improve spam detection [20]. Furthermore, to overcome the limitations of the blacklist, some preliminary studies have used heuristic rules to filter Twitter spam [20].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2019 doi:10.20944/preprints201905.0141.v1

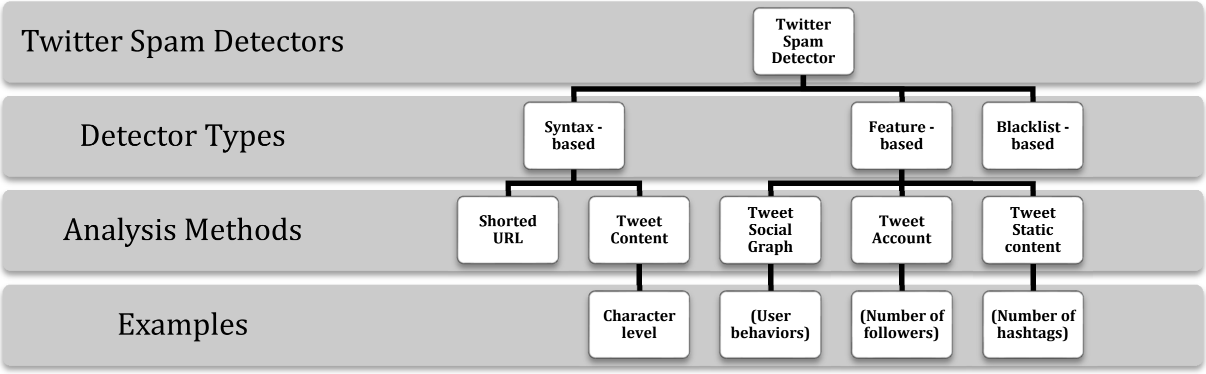

Figure. 1 Types of Twitter Spam Detectors [56]

Process of Detecting Spam through ML. This process involves several steps. The first step involves collecting data from Twitter using its Streaming Application Programming Interference (API). This is followed by data pre-processing, which includes feature extraction, data labelling, and dataset splitting. However, for textual spam detectors, the pre-processing step may include more functions, such as tokenizing, removing stop words, and steaming. Extracting and selecting features from tweets or Twitter accounts helps the chosen ML classifier to distinguish between spam and non-spam, for example based on account age, the number of followers or friends, and the number of characters. Data labelling or ground truth is the process in which the collected data are labelled either manually or using a crowdsourcing site. The dataset then needs to be split into a training set and a test set. The last step involves training the chosen classification algorithm by using the labelled data, followed by performance evaluation, where the trained machine learning classifier can be used for spam detection [20, 1] (see Figure 2).

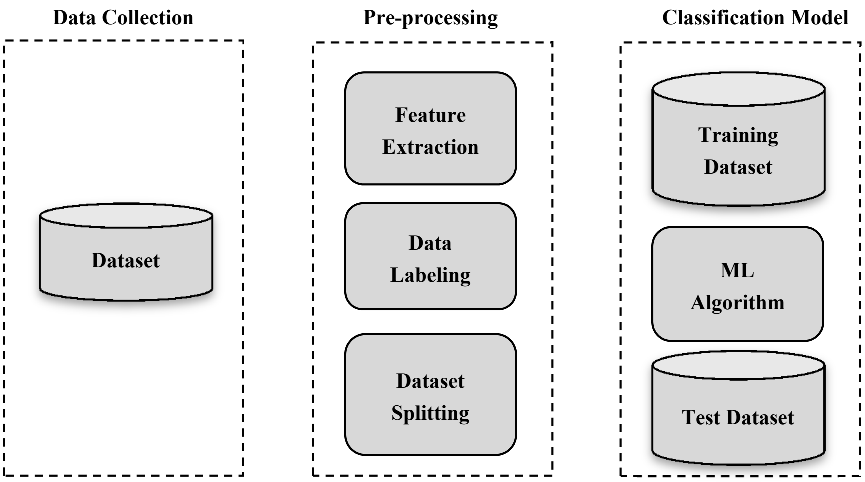

Figure 2: Learning Process of ML-based Spam Detection.

Detecting Spam Tweets using ML Techniques. As well as using blacklists for Twitter spam detection, other studies have used different ML techniques to detect spam tweets. As mentioned earlier, a number of steps are involved in the use of ML techniques to detect spam in Twitter; some of the important steps are discussed here. Several studies of Twitter spam detection developed their models by employing different ML algorithms, such as SVM, NB, and RF, of which RF has shown the best results in terms of detection accuracy. In [37], the authors compared and evaluated the detection accuracy, stability, and scalability of nine machine learning algorithms. The results showed that RF and C5.0 outperformed the other algorithms in terms of their superior detection accuracy. Similarly, a framework for detecting spam based on random forests was proposed in [39]. In addition, RF was chosen from five other algorithms in [55], as it has shown the best results in terms of evaluation metrics. Selecting features to help classify samples is as important a step as choosing the most suitable algorithm for the required task. In [20], the authors collected a large dataset and labelled approximately 6.5 million spam tweets. They found that when using an imbalanced data set that simulated a real-world scenario, the classifiers' ability to detect spam tweets is reduced. On the other hand, when features are discretized, the performance of classifiers improves.

Detecting Spam Campaigns in Twitter. Unlike the above spam detection models developed to detect a single spammer, some approaches can be used to detect campaign spam (spambots). According to [24], social spambots are a growing phenomenon and current spam detectors designed to detect a single spam account are not capable of capturing spambots. Although their study shows that neither humans nor existing machine learning models can detect spambots accurately, the result of an emerging technique deploying digital DNA has achieved a very promising detection performance. Similarly, [49] stated that methods designed to detect spam using account-based features cannot detect *crowdturfing accounts* (accounts created by crowdsourcing sites that have crowdsourcing and astroturfing characteristics). Another study [55] noted that spammers tend to create account bots to quickly reach their goals by systematically posting a large amount of spam in a short period of time. Consequently, they proposed an approach that uses the time property (for example the account creation date and tweet posting time), which cannot be modified by spammers, to reduce the creation of bots. [27] proposed an approach called Tangram, which uses a template-based model to detect spam in OSNs. After analyzing the textual pattern of a large collection of spam, they found that the largest proportion of spam is generated with an underlying template compared to other spam categorizes (for example paraphrase, no-content, and others).

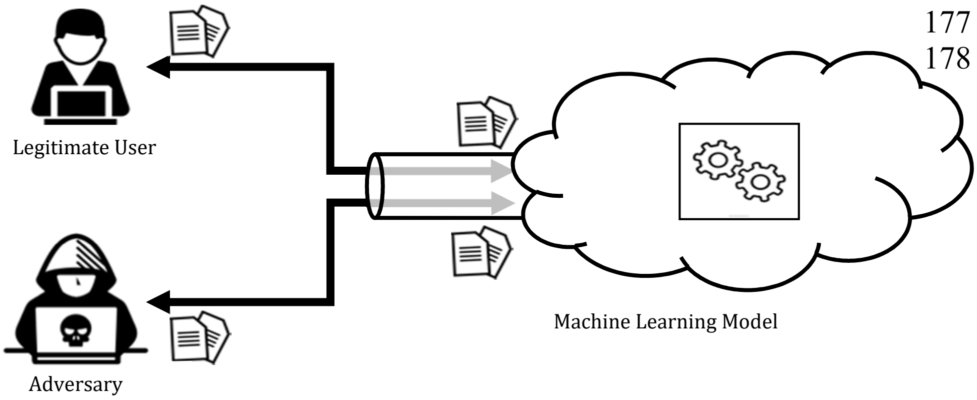

Figure 3: An adversary uses the same channel as legitimate users to exploit knowledge about the system.

Security of Twitter spam detectors. Despite the success and high level of accuracy of the models described here in detecting Twitter spam, they are nevertheless vulnerable as they were not developed for adversarial settings. A popular framework for evaluating secure learning was proposed in [4], and extended in [5, 31, 12]; it enables different attack scenarios to be envisaged against machine learning algorithms. The framework suggests the following steps: (1) identifying potential attacks against machine learning models based on the popular taxonomy (see Section 3.1); (2) simulating these attacks to evaluate the resilience of ML models; assumption that the adversary's attacks can be formed based on the adversary's goals, knowledge, and capabilities/resources. (3) investigating some possible defense strategies against these attacks. Defense against adversarial attacks is challenging as these attacks are non-intrusive in nature and an adversary launches their attacks using the same channel as legitimate users. Thus, defense strategies against these attacks cannot employ traditional encryption/security techniques [47]. Figure 3 demonstrates how an adversary can use the same channel as legitimate users to access an ML model and learn some of its characteristics. Designing *proactive* models rather than traditional *reactive* models is a necessity in the adversarial environment. Whereas reacting to detected attacks will never prevent future attacks, proactively anticipating adversaries' activities enables suitable defense methods to be developed

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2019 doi:10.20944/preprints201905.0141.v1

before an attack occurs [12]. This has motivated researchers to develop different attack scenarios against machine learning algorithms and classification models and propose some countermeasures. Table 2 shows an outline of recent spam detectors proposed in the literature.

Adversarial attacks against Twitter spam detectors. In [54] the authors evaluated the security of a ML detector that is designed to detect spam generated by malicious crowdsourcing users of Weibo (the Chinese version of Twitter) against evasion and poisoning attacks. Their focus was on adversaries that use crowdsourcing sites to launch attacks. To study evasion attacks, two attacks were simulated: basic evasion, where an adversary has limited knowledge, and optimal evasion, where the adversary has perfect knowledge. The results show that an optimal evasion attack has a much higher impact than the basic one. However, in the real world, it is very difficult for adversaries to have perfect knowledge about the system. Thus, the less knowledge adversaries have about the system, the harder it is for them to evade detection. In causative attacks, two mechanisms for launching poisoning attacks are used. The aim of the first poisoning attack is to mislead the system by using crowdturfing admins to inject misleading samples directly into the training data. In the second poisoning attack, adversaries pollute training data by crafting samples that mimic benign users' behavior. After analyzing both attacks, it was found that injecting misleading samples causes the system to produce more errors than the second poisoning attack.

Another study by [42] analyzed the robustness of a Twitter spam detector called POISED against evasion and poisoning attacks. POISED is designed to distinguish between spam and non-spam messages based on the propagation of messages in each campaign. The authors suggested a poisoning attack, where the goal of an adversary is to contaminate training data by joining communities to alter their network and structure. The adversary posts manipulated messages in these compromised communities to mislead the system. Similarly, in an evasion attack, the adversary joins communities and resembles the propagation of non-spam messages to evade detection. The results show that the performance of POISED decreases when the percentage of compromised communities increases in both attacks. Thus, the authors suggest that if the adversary is to successfully attack systems, he or she needs to have a perfect knowledge about the structure and network of the targeted community.

The above studies suggest that the adversary's level of knowledge about the deployed system plays a very important role in determining the success of the attack.

Table 2: Outline of some recent techniques used for detecting spam in Twitter: some of these works are discussed in Section 2.

Title	Methodology	Type of Spam	Type of Detector	Learning Approach	Results/Accuracy
6 Million Spam Tweets - A Large Ground Truth for Timely Twitter Spam Detection [62]	Different ML algorithms were used; balanced and imbalanced datasets were tested.	Spam tweet	Feature-based	Supervised	RF outperforms other algorithms.
A Hybrid Approach for Spam Detection for Twitter [63]	J48, Decorate and Naive-Bayes (NB).	Spam tweet	Feature-, user-, and graph-based	Supervised	J48 outperforms other algorithms.
Leveraging Time for Spammers Detection on Twitter [55]	Time-based features were used, and different ML algorithms were tested.	Spam tweet	Feature-based	Supervised	RF outperforms other algorithms.
Twitter spam detection based on Deep Learning [56]	Deferent ML algorithms with Word2Vector technique were used.	Spam tweet	Syntax-based	Supervised	RF with Worrd2Vec outperforms other algorithms.
Semi-supervised spam detection (S3D) [64]	Utilizes four lightweight detectors (supervised and unsupervised) to detect spam tweets and updates the models periodically in batch mode.	Spam tweet	Feature-based and Blacklist	Semi-supervised	Confidential labeling process, which uses blacklisted, near-duplicated, and reliable non-spam tweets, makes the deployed classifier

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2019 doi:10.20944/preprints201905.0141.v1

					more efficient when detecting new spam tweets.
CrowdTarget: Target-based Detection of Crowdturfing in Online Social Networks [49]	Detects spam tweets that received retweets from malicious crowdsourcing customers. It uses four new retweet-based features and KNN as a classifier.	Spam tweet	Feature-based	Supervised	CrowdTarget detects malicious retweets created by crowdturfing users with a True Positive Rate of 0.98 and False Positive Rate of 0.01.
Beating the Artificial Chaos - Fighting OSN Spam using Its Own Templates [58]	Detects template-based spam, paraphrase spam, and URL-based spam.	Spam campaign	Syntax-based	Supervised	Template-detection outperforms URL blacklist-detection.
POISED - Spotting Twitter Spam Off the Beaten Paths [42]	Detects spam campaign based on community and topic of interest.	Spam campaign	Syntax-based	Supervised and unsupervised	NB, SVM, and RF all achieve about 90% detection accuracy.
Collective Classification of Spam Campaigners on Twitter: A Hierarchical Meta-Path Based Approach (HMPS) [65]	Builds heterogeneous networks and detects nodes connected by the same phone number or URL.	Spam campaign	Social graph-based	Supervised	HMPS outperforms feature- and content-based approaches. Prediction accuracy improves when using HMPS with feature- and user-based approaches.
Social Fingerprinting - Detection of Spambot Groups Through DNA-Inspired Behavioral Modeling [61]	Models users' behaviour using DNA fingerprinting technique to detect spambots.	Spam campaign	Social graph-based	Supervised and unsupervised	The results show that the proposed approach achieves better detection accuracy when using a supervised learning approach.

3. Adversarial Machine Learning

Adversarial ML is a research field that investigates the vulnerability of ML to *adversarial examples*, along with the design of suitable countermeasures [13]. Adversarial examples are inputs to ML that are designed to cause incorrect output [15]. The term was first introduced in [51] and used for computer vision, but in the context of spam and malware detection, the term *evasion attacks* is used in [12]. This section is going to discuss about different adversarial attacks and countermeasures. Table 3 and 4 outline recent works in adversarial machine learning.

Table 3: Outline of different adversarial attacks. These studies are discussed in Section 3.

Type of Influence	Title	Name of Attack	Attack Target	Attack Method
Causative	Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning [32]	Poisoning	Regression Learning	Optimization-based poisoning attack, in which different optimization approaches were used. Statistical-based poisoning attack (StatP) that queries a deployed model to find an estimate of the mean and covariance of training data.
	Support vector machines under adversarial label noise [59]	Label Flipping	SVM	Two different label flipping attacks were used: random and adversarial label flips.
	Curie - A method for protecting SVM Classifier from Poisoning Attack [35]	Label Flipping	SVM	Two label flipping attack were used. In the first, the loss maximization framework was used to select points that needed their label to be flipped. In second attack, the selected data points are moved to other points in the feature space.
	Adversarial Machine Learning [31]	Dictionary	Spam filter	An adversary builds a dictionary of tokens learned from the target model, and then sends attack messages to cause misclassification.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2019 doi:10.20944/preprints201905.0141.v1

	Thwarting Signature Learning by Training Maliciously [41]	Red Herring	Polymorphic worm signature generation algorithms	An adversary sends messages with fake features to trick the deployed model.
	Man vs. Machine: Practical Adversarial Detection of Malicious Crowdsourcing Workers [54]	Poisoning	NB, BN, SVM, J48, RF	Two types of poisoning attacks were performed: Injecting misleading samples and altering training data.
Exploratory	Data Driven Exploratory Attacks on Black Box Classifiers in Adversarial Domains [47]	Anchor Points (AP) and Reverse Engineering attacks (RE)	SVM, KNN, DT, RF	AP attack is not affected by the chosen model (linear or non-linear), unlike RE, which is affected when a defender uses DT or RF.
	Evasion Attacks against Machine Learning at Test Time [10]	Evasion	SVM, Neural Network	A gradient-descent evasion attack was proposed.
	Good Word Attacks on Statistical Spam Filters [38]	Good Word	NB, Maximum entropy filter	Active and passive good word attacks against email spam filters were evaluated.
	Adding Robustness to Support Vector Machines Against Adversarial Reverse Engineering [2]	Reverse Engineering	SVM	Three different query selection methods, which help learn the decision boundary of deployed classifier, were used. Random, selective, and uncertainty sampling.
	Man vs. Machine: Practical Adversarial Detection of Malicious Crowdsourcing Workers [54]	Evasion	NB, BN, SVM, J48, RF	Two evasion attack were launched: Basic evasion attack and Optimal evasion attack, where an adversary knows features Needs to be altered.

3.1 Taxonomy of attacks against ML

A popular taxonomy proposed in [4, 5, 12] categorized attacks against ML systems along the three following axes:

The Attack INFLUENCE

- **Causative**: the attack influences the training data to cause misclassification.
- **Exploratory**: the attack exploits knowledge about the deployed classifier to cause misclassifications without influencing training data.

The Type of SECURITY VIOLATION

- **Integrity violation**: an adversary evades detection without compromising normal system operations.
- **Availability violation**: an adversary compromises the normal system functionalities available to legitimate users.
- **Privacy violation**: an adversary obtains private information about the system (such as its users, data, or characteristics) by reverse-engineering the learning algorithm.

The Attack SPECIFICITY

- **Targeted** attacks focus on a particular instance.
- **Indiscriminate** attacks encompass a wide range of instances.

The first axis, which is the attack influence, divides an adversary's capability to influence a classifier's learning systems into causative and exploratory. The influence is causative if an adversary misleads the deployed classifier by contaminating (poisoning) the training data by injecting carefully crafted samples into it. In contrast, the influence is exploratory if an adversary gains knowledge about the deployed classifier to cause misclassification at the testing phase without influencing training data.

The second axis describes the type of security violation committed by an adversary. The security violation can be an integrity violation if it enables an adversary to bypass the deployed classifier as a false negative. In addition, the attack can violate the model's availability if it creates denial of service, misclassifying non-spam samples as false positives, or if it prevents legitimate users from accessing

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2019 doi:10.20944/preprints201905.0141.v1

the system. The security violation can be a privacy violation if it allows an adversary to exploit confidential information from the deployed classifier.

The third axis of the taxonomy refers to the specificity of an attack. In other words, it indicates how specific an adversary's goal is. The attack specificity can be either targeted or indiscriminate, depending on whether the attack causes the classifier to misclassify a single or few instances, or undermines the classifier's performance on a larger set of instances.

Table 4: Outline of techniques used for mitigating adversarial attacks: all of these works are discussed in Section 3

Type of Influence	Title	Name of Attack	Type of Classifier	Defense Category	Defense Method
Causative	Mitigating Poisoning Attacks on Machine Learning Models: A Data Provenance Based Approach [60]	Poisoning	SVM	Data Sanitization	Filtering out poisoned data from the training dataset using a provenance framework that records the lineage of data points.
	Curie- A method for protecting SVM Classifier from Poisoning Attack [35]	Poisoning	SVM	Data Sanitization	The data are clustered in the feature space, and the average distance of each point from the other points in the same cluster is calculated, with the class label considered as a feature with proper weight. The data points with less than 95% confidence are removed from the training data.
	Bagging Classifiers for Fighting Poisoning Attacks in Adversarial Classification Tasks [9]	Poisoning	Bagging and weighted bagging ensembles	Data Sanitization	Using an ensemble construction method (bagging) to remove outliers (adversarial samples) from training dataset.
	Data sanitization against adversarial label contamination based on data complexity [18]	Label Flipping	SVM	Data Sanitization	Data complexity, which measures the level of difficulty of classification problems, was used to distinguish adversarial samples in the training data.
	Support vector machines under adversarial label noise [59]	Label Flipping	SVM	Robust learning	Adjusting the kernel matrix of SVM depending on noise (adversarial) samples' parameters increases the robustness of the classifier.
	Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning [32]	Poisoning	Regression Learning	Robust learning	The TRIM algorithm, which regularized linear regression by applying trimmed optimization techniques, was proposed
Exploratory	Robust support vector machines against evasion attacks by random generated malicious samples	Evasion	SVM	Robust learning	Trains the SVM classifier with random malicious samples to enclose the decision function.
	Adding Robustness to Support Vector Machines Against Adversarial Reverse Engineering [2]	Reverse Engineering	SVM	Randomization	Learning a distribution of classifiers and picking a decision boundary randomly makes reverse engineering attacks harder to launch.
	Handling adversarial concept drift in streaming data [48]	Evasion	SVM	Disinformation	Hiding the importance of features and using an ensemble of classifiers.
	Adversarial Pattern Classification using Multiple Classifiers and Randomization [11]	Evasion	Spam Filter, SVM, NB	Multiple Classifiers and Randomization	Multiple Classifiers Strategy MCS, where different classifiers are trained by different features to randomize a model's decision boundary.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2019 doi:10.20944/preprints201905.0141.v1

3.2 Common Types of Threat Models

In this regard, after presenting the taxonomy of attacks against ML systems, the next step towards identifying potential attack scenarios is threat modelling, which involves defining an adversary's goal, knowledge, and capability [4, 5, 12]. According to the above taxonomy, the attacker's goal may be based on the type of security violation (integrity, availability, or privacy), and on the attack specificity (targeted or indiscriminate). For instance, the adversary's goal could be to violate the system's integrity by manipulating either a specific instance or different instances. An attacker's level of knowledge about the classifier varies, and may include perfect-knowledge (white box attack), limited-knowledge (grey box attack), or zero-knowledge (black box attack). Attacker capability can involve either influencing training data (causative attack) or testing data (exploratory attack).

3.3 Adversarial Attacks and Defense Strategies

The existing literature on adversarial ML provides different attack examples and defense methods for both adversarial attack types (causative and exploratory). This section reviews common attack examples and some defense strategies against these attacks (see Table 5).

Table 5: Common Adversarial Attacks and Defenses

	Causative Attack	**Exploratory Attack**
Attack	Poisoning	Probing
	Red Herring	Evasion
	Label-Flipping	Reverse Engineering
		Good Words Attack
Defense	RONI	Randomization
	Game Theory based	Disinformation
	Multiple Learners	

3.3.1 Causative Attacks

One of the most common types of causative attack is a *poisoning attack*, in which an adversary contaminates the training dataset to cause misclassification [5]. An adversary can poison training data by either directly injecting malicious samples or sending a large number of malicious samples to be used by the defender when retraining the model [54]. A *label-flipping attack* is another example of a causative attack. Here, an adversary flips the label of some samples and then injects these manipulated samples into the training data. Different methods are used to perform this attack. Adversaries can either select samples that are nearest to or farthest from a classifier's decision boundary and flip their label [35]. The easiest method is to randomly flip the label of some samples that might be used for retraining. In [59], it was shown that randomly flipping about 40% of the training data's labels decreased the prediction accuracy of deployed classifier. A *red herring* attack is a type of causative attack in which the adversary adds irrelevant patterns or features into the training data to mislead the classifier to focus on these irrelevant patterns [41, 2]. Defense against causative attacks is challenging because ML classifiers need to be retrained periodically to adapt to new changes. Retraining the classifier makes it vulnerable as the data used for retraining is collected from an adversarial environment [35].

3.3.2 Causative Defense Methods

Although preventing these attacks is difficult, there are some defense methods proposed in the literature that can reduce the effect of these attacks. Defense methods against causative attacks may rely on *Game Theory*, where the defense problem is modeled as a game between the adversary and the classifier [2, 23, 25, 14]. *Data sanitization* methods focus on removing contaminated samples that have been injected by an adversary from a training dataset before training a classifier, while *robust learning* focuses on increasing the robustness of a learning algorithm to reduce the influence of

 doi:10.20944/preprints201905.0141.v1

contaminated samples [18]. *Reject-on-negative-impact* (RONI) is one of the simplest and most effective defense methods against causative attacks, and is considered to be a data sanitization method. In RONI, all the training data go through preliminary screening to find and reject samples that have a negative impact on the classification system. To distinguish between contaminated and untainted samples, a classifier is trained using base training data before adding suspicious samples to the base training data and train another classifier. The prediction accuracy for both classifiers compared to over labelled test data is evaluated. If adding suspicious samples to the training data reduces the prediction accuracy, these samples must be removed [31]. Another defense method involves using *multiple classifiers*, which has been shown to reduce the influence of poisoned samples in training data [9].

3.3.3 Exploratory Attacks

The most popular types of exploratory attacks are *evasion* and *reverse engineering*. Both attacks start with a *probing attack*, in which an adversary sends messages to reveal some information about the targeted classifier. Once the adversary gains some knowledge about the system, he or she can either carefully craft samples that can evade the system (an evasion attack), or use that information to build a substitute system (a reverse engineering attack) [4]. Furthermore, a *Good Word Attack* is a type of exploratory attack in which the adversary either adds or appends words to spam messages to evade detection. Good Word attacks can be passive or active. In a passive attack, the adversary constructs spam messages by guessing which words are more likely to be bad or good (for example a dictionary attack). In an active attack, the adversary has access to a targeted system that enables him or her to discover bad and good words [38].

3.3.4 Exploratory Defense Methods

As with causative attacks, it is difficult to prevent exploratory attacks from happening because in most cases systems cannot differentiate between messages sent for a legitimate purpose and those sent to exploit the system. However, there are currently two common defense methods: *disinformation* and *randomization*. In disinformation methods, the defender's goal is to hide some of the system's functions (for example classification algorithms or features used by the classifier) from an adversary. In contrast, in randomization methods, the defender's aim is to randomize the system's feedback to mislead an adversary. [4]

Although most of these attack strategies and defense methods were proposed for domains such as email spam filtering, IDS, and malware detection, the underlying approach can be applied in Twitter spam detectors. The following section examines some of these techniques in the context of Twitter spam detectors.

3. Taxonomy of Attacks Against Twitter Spam Detectors

This section surveys attacks against Twitter spam detectors in an adversarial environment. Examples of adversarial spam tweets that can be used by adversaries to attack Twitter are also provided.

4.1 Methodology

Different hypothetical attack scenarios against Twitter spam detectors are proposed. Attack tactics were formularized based on the framework of the popular attack taxonomy presented in [4] [5] that categorizes attacks along three axes: influence, security violations, and specificity. This framework was extended in [12] to derive the corresponding optimal attack strategy by modelling an adversary's goal, knowledge and capability. The adversary's goals considered in this study are either to influence training or test data or to violate the system's integrity, availability, or privacy. The adversary's knowledge is considered as: perfect knowledge (white-box attack) and zero-knowledge (black-box attack). This ensures that both the worst-case and best-case scenarios are considered for the adversary when attacking spam detectors. The adversary's capability is based on desired goals. For example, if the goal is to influence the training data, the adversary must be capable

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2019 doi:10.20944/preprints201905.0141.v1

of doing so. Examples of adversarial spam tweet were extracted from Arabic trending hashtags. The number of spam tweets using Arabic trending hashtags was found to be high, the reasons for which are beyond the scope of this study. However, it was found that there are very active spam campaigns spreading advertisements for untrustworthy drugs, for example weight loss drugs, Viagra, and hair treatment drugs, targeting Arabic-speaking users. The attack scenarios can be modelled as follows:

1. Categorizing attacks based on their influence and type of violation (such as causative integrity attacks).
2. Identifying the attack's settings, which includes an adversary's goal, knowledge and capability.
3. Defining the attack strategy that provides potential attack steps.

4.2 Potential Attack Scenarios

Here, attacks against Twitter spam detectors were categorized into four groups: causative integrity, causative availability, exploratory integrity, and exploratory availability attacks. Four hypothetical attack scenarios are provided, and different examples for each category are presented. Some spam tweets were extracted from Arabic hashtags to show how an adversary can manipulate tweets.

4.2.1 Causative Integrity Attacks

Example 1: Poisoning Attack

In this attack scenario, an adversary attempts to influence training data to cause new spam to bypass the classifier as false negatives. The settings of the attack scenario are as follows: The **adversary's goal** is to compromise the integrity of Twitter spam detectors and the attack specificity can be either targeted or indiscriminate. The **adversary's knowledge** is assumed to be perfect (white-box attack). In terms of the **adversary's capability**, it is assumed that the adversary is capable of influencing the training data. After defining the attack scenario's setting, the next step is the attack strategy. A potential attack strategy is as follows:

- As the adversary's knowledge of the system is considered to be perfect, it is not necessary to send probing tweets to gain knowledge.
- The adversary would carefully craft a large number of malicious tweets.
- The crafted tweets must resemble non-spam tweets and include both spam components, such as malicious URLs, and non-spam components or words (see Figure 4).
- The adversary would then post these tweets randomly using different trending hashtags and hope that these malicious tweets are used by Twitter when retraining their system.

Figure 4 shows an example of a spam tweet that has been carefully crafted and can be used to poison training data. The spam tweet mimics non-spam tweets by avoiding the inclusion of any spam words, telephone numbers, or hashtags. In addition, the account resembles a legitimate user's account in terms of having a decent number of followers and friends, and has a profile photo and description. This spam tweet bypasses Twitter's spam detector and could be used for retraining the classifier.

Figure 4: A spam tweet resembling a non-spam tweet to poison training data.

Example 2: Probing and Red Herring Attack

As in [41], in this attack scenario, the adversary's aim is to mislead Twitter's spam detectors by influencing training data. **The adversary's goal** is to compromise the integrity and privacy of Twitter's spam detectors, and the attack specificity can be either targeted or indiscriminate. **The adversary's capability** is similar to the previous example. However, the **adversary's knowledge** about Twitter's spam detectors is assumed to be zero (black-box attack). Based on the scenario's settings, a potential attack strategy is as follows:

- As the adversary has zero-knowledge about the system, sending probing tweets to gain knowledge is required (privacy violation).
- A probing attack is an exploratory type of attack, and will be discussed in the next section.
- The adversary would craft samples with spurious or fake features and post these samples on trending hashtags to trick Twitter's spam detectors into using these samples for retraining.
- If Twitter spam detectors are trained on these samples, the adversary will discard these spurious features in future tweets to bypass the classifier.

Figure 5 (a) shows an example of a spam tweet that has a spurious feature (phone number). As the number of tweets that have a phone number have increased on Twitter, some proposed spam detectors suggest using a phone number as an indicator of spam tweets [1, 30]. However, Figure 5(b) shows how the adversary can trick Twitter into using a phone number as a feature, and avoide including phone numbers in his spam tweets. Instead, the adversary includes a phone number inside an image to evade detection.

Figure 5: (a) A spam tweet containing spurious feature (a mobile number).

 doi:10.20944/preprints201905.0141.v1

Figure 5: (b) A spam tweet with a mobile number inside an image to evade detection.

Example 3: Probing and Label-Flipping Attack

The aim of the attack scenario, as in [35], is to cause misclassification by injecting label-flipped samples into training data. The settings of the attack scenario are as follows: **an adversary's goal** is to violate the integrity and privacy of Twitter's spam detectors and the attack specificity can be either targeted or indiscriminate. The **adversary's capability** is similar to the previous example. However, the **adversary's knowledge** is assumed to be zero (black-box attack). Based on the scenario's settings, a potential attack strategy is as follows:

- As the adversary has zero-knowledge about the system, sending probing tweets to gain knowledge is required (privacy violation).
- A probing tweet (see Figure 7) helps the adversary to learn how the classifier works; on this basis, the adversary can craft malicious tweets.
- Depending on the knowledge that the adversary is able to gain, he or she can either flip the nearest or farthest samples from the deployed classifier's decision boundary.
- If the adversary was not able to learn more about the classifier, he or she can randomly flip the label of some tweets.
- He or she would then randomly post these tweets using different trending hashtags and hope that these malicious tweets are used by Twitter when retraining their system.

4.2.2 Causative Availability Attack

Example 1: Poisoning Attack

In this type of attack, an adversary tends to influence training data to either subvert the entire classification process or to make future attacks (such as evasion attacks) easier. The settings of the attack scenario are as follows: the **adversary's goal** is to violate the availability of Twitter and the attack specificity can be either targeted or indiscriminate. The **adversary's knowledge** is assumed to be perfect (white-box attack). In terms of the **adversary's capability**, it is assumed that the adversary is capable of influencing the training data. After defining the attack scenario's setting, the next step is the attack strategy. A potential attack strategy is as follows.

- As the adversary's knowledge about the system is considered to be perfect, sending probing tweets to gain knowledge is not required.
- The adversary would carefully craft a large number of misleading tweets that consist of a combination of spam and non-spam components (see Figure 4).
- The adversary needs to contaminate a very large proportion of training data for this attack to be successful. Using crowdsourcing sites or spambots to generate contaminated tweets helps the adversary to launch such an attack.

- The last step would be to post these tweets randomly using different trending hashtags to quickly spread these tweets in the hope that Twitter will use them when retraining their system.

Example 2: Dictionary Attack

In this attack, as in [31], an adversary aims to corrupt the classification process by influencing training data and lead future legitimate tweets to be misclassified. The settings of the attack scenario are as follows: **an adversary's goal** is to violate the availability and integrity of Twitter spam detectors, and the attack specificity can be either targeted or indiscriminate. The **adversary's knowledge** is assumed to be perfect (white-box attack). In terms of the **adversary's capability**, it is assumed that the adversary is capable of influencing the training data. After defining the attack scenario's setting, the next step is the attack strategy. A potential attack strategy is as follows.

- As the adversary's knowledge about the system is considered to be perfect, sending probing tweets to gain knowledge is not required.
- Based on the adversary's knowledge, he or she would build a dictionary of words or phrases frequently used by legitimate users and use this to craft malicious tweets.
- The adversary would post tweets that contain a large set of tokens (non-spam words, phrases, or tweet structure) from the dictionary in trending hashtags.
- If these tweets were used to train the system, non-spam tweets will be more likely to be classified as spam because the system will give a higher spam score for tokens used in the attack.

Figure 6 shows how a causative availability attack can affect Twitter spam detectors. The two spam tweets remain undetected for a long period of time because of the attack. As mentioned earlier, availability attacks overwhelm the system, which leads to difficulty in detecting spam tweets. A spam tweet on the left-hand side of the image below contains a very common spam word and should be very easily detected by the classifier, yet due to the attack, the tweet stays in place for longer than 52 minutes. In addition, a spam tweet on the right-hand side remains undetected for longer than five hours, which is a very long time.

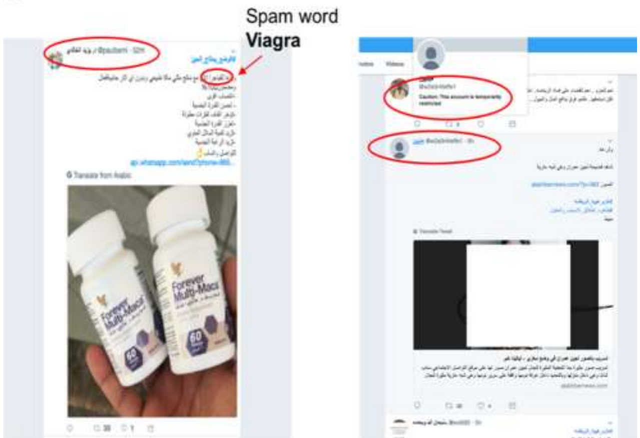

Figure 6: Spam tweets bypass the detection system due to the availability attack.

4.2.3 Exploratory Integrity Attack

Example 1: Probing Attack

In this attack scenario, the aim is to learn or expose some of the deployed classifier's functionalities without any direct influence over the training data. The settings of the attack scenario are as follows: **an adversary's goal** is to compromise the privacy of Twitter's spam detectors and the attack specificity can be either targeted or indiscriminate. The **adversary's knowledge** is assumed to be zero (black-box attack). As in [12], in terms of the **adversary's capability**, it is assumed that the adversary is only capable of influencing the testing data. After defining the attack scenario's setting, the next step is the attack strategy. A potential attack strategy is as follows.

- As the adversary does not have sufficient knowledge of how the Twitter spam detector works, sending probing tweets to gain knowledge is required.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2019 doi:10.20944/preprints201905.0141.v1

- The adversary would send a large number of tweets, each with different features, to learn about the system (see Figure 7).
- Based on the information that is learned, the adversary would carefully craft tweets to evade detection.

Figure 7 shows an example of three spam tweets advertising the same weight-loss products. However, the adversary uses different features in each tweet. The first tweet consists of text, a URL, and an image, and the second has text and an image. The last one contains text only. The goal here is to learn how the classifier works. For example, if the first tweet is detected, the adversary will learn that a blacklist of URLs could be one of the features used by the classifier.

Figure 7: An example of a probing attack.

Example 2: Evasion Attack – Good Word Attack

In this attack scenario, the aim is to evade being detected by the deployed classifier without any direct influence over the training data. The settings of the attack scenario are as follows: **an adversary's goal** is to compromise the integrity of the Twitter spam detector and the attack specificity can be either targeted or indiscriminate. The **adversary's knowledge** is assumed to be perfect (white-box attack). In terms of the **adversary's capability**, as in [12], it is assumed that the adversary is only capable of influencing the testing data. After defining the attack scenario's setting, the next step is the attack strategy. A potential attack strategy is as follows.

- As the adversary's knowledge of the system is considered to be perfect, sending probing tweets to gain knowledge is not required.
- Based on the adversary's knowledge, he or she would carefully craft tweets by modifying and obfuscating spam words (such as Viagra) or the tweet's features to evade detection (such as number of followers) (see Figure 8).

Figure 8 shows a spam tweet that has been carefully crafted to evade detection. The adversary avoids including any spam words in the text. Instead, the tweet contains a description of the drug (Viagra), and the spam word was inserted inside an image.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2019 doi:10.20944/preprints201905.0141.v1

Figure 8: Spam image tweet crafted to evade detection.

Example 3: Probing and Reverse Engineering Attacks

Evading the classifier without influencing the training data is the aim considered in this attack scenario. The scenario's settings are: **the adversary's goal** is to violate the integrity and privacy of Twitter's spam detectors, and the attack specificity can be either targeted or indiscriminate. The **adversary's capability** is similar to the previous example, but the **adversary's knowledge** about Twitter's spam detectors is assumed to be zero (black-box attack). Based on scenario's settings, a potential attack strategy is as follows:

- As the adversary has zero knowledge about the system, the first step would be to send probing tweets to learn how the system works (privacy violation).
- Based on the exploited knowledge, the adversary would build a substitute model that could be used for launching different exploratory attacks [2].
- Once the substitute model is built, the adversary would craft different spam tweets to evade detection, and spam tweets that successfully evade the model will be used against the Twitter spam detector.

3.2.4 Exploratory Availability Attack

Example 1: Denial of Service and Evasion Attack

In this attack scenario, the main aim is to evade the classifier by sending a large number of adversarial spam tweets to overwhelm the classifier without any direct influence over the training data. The settings of the attack scenario are as follows: **an adversary's goal** is to violate the availability and integrity of the Twitter spam detector and the attack specificity can be either targeted or indiscriminate. The **adversary's knowledge** is assumed to be perfect (white-box attack). In terms of the **adversary's capability**, as in [12], it is assumed that the adversary is only capable of influencing the testing data. After defining the attack scenario's setting, the next step is the attack strategy. A potential attack strategy is as follows.

- As the adversary has perfect knowledge about the system, sending probing tweets to gain knowledge is not required.
- Based on the gained knowledge, the adversary would carefully craft spam tweets. As the adversary cannot influence training data, the adversary would craft tweets that require more time to be processed by the classifier, such as image-based tweets [5].
- The adversary would then flood the system (for example a particular trending hashtag) with spam tweets to prevent users from reading non-spam tweets and cause difficulty in detecting spam tweets.

Figure 9 shows an example of an availability attack, where the adversary post a large number of spam tweets from a different account that only contains an image. As mentioned earlier, image processing overwhelms the deployed classifier and causes a denial of service. In this kind of attack, the adversary may use crowdsourcing sites or spambots to generate spam tweets.

Figure 9: An adversary floods the hashtag with spam tweets.

Example 2: Probing and Denial of Service Attacks

The aim of this attack scenario is similar to the previous example, but the scenario's settings are slightly different. **The adversary's goal** is to violate the integrity, availability and privacy of Twitter's spam detectors, and the attack specificity can be either targeted or indiscriminate. The **adversary's capability** is similar to the previous example, but the **adversary's knowledge** about Twitter's spam detectors is assumed to be zero (black-box attack). Based on scenario's settings, a potential attack strategy is as follows:

- As the adversary has zero-knowledge about the system, the first step would be to probe the classifier with some tweets to learn how it works.
- Based on the exploited knowledge, the adversary would craft a large number of spam tweets and post them in a specific hashtag to cause denial of service and make future attacks easier. [2].

All attack examples can be either targeted if an adversary focuses on a specific spam tweet (for example URL-based spam, or weight-loss ads), or indiscriminate, if an adversary targets multiple types of spam tweet (such as URL-based and advertisements). Although presented adversarial spam tweets look very similar to spam tweets that targeted users, this special type of spam tweets need to be studied more as it aims to subvert Twitter spam detectors. Table 6 summarizes the taxonomy of potential attacks.

Table 6: Taxonomy of potential attacks against Twitter spam detectors

Type of Influence	Potential Attack	Security Violation	Specificity
Causative	Poisoning Attack	Integrity	Targeted/ Indiscriminate
	Probing and Red Herring Attack	Integrity & Privacy	
	Probing and Label-Flipping Attack	Integrity & Privacy	
	Poisoning Attack	Availability	
	Dictionary Attack	Availability & Integrity	
Exploratory	Probing Attack	Privacy	
	Good Word Attack	Integrity	
	Probing and Reverse Engineering Attacks	Integrity & Privacy	
	Denial of Service and Evasion Attack	Availability & Integrity	
	Probing and Denial of Service Attacks	Availability, Integrity & Privacy	

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2019 doi:10.20944/preprints201905.0141.v1

5 Potential Defense Strategies

This section discusses some possible defense strategies against adversarial attacks that can be considered when designing a spam detector for Twitter. Some of the popular defense methods proposed in the literature are discussed in the context of Twitter spam detection.

5.1 Defenses Against Causative Attacks

Existing approaches to defend against causative attacks focus on filtering or screening all the training data before using them to update a deployed classifier, such as RONI, data sanitization techniques, and bagging of classifiers. Although these methods have been shown to reduce the influence of contaminated samples on training data, in some cases in which contaminated samples overlap with untainted samples, discriminating between the two becomes very difficult [18]. Some recent studies have suggested using a data collection oracle to retrain a deployed classifier [48, 33]. However, trusting an oracle to label training data could be problematic. The authors in [40] stated that using crowdsourcing sites to label data might produce noisy data, thus increasing complexity. Furthermore, Song et al. added that adversaries can increase the popularity of malicious tweets by using artificial retweets generated by crowdsourcing workers [49]. Thus, developing a fully automated model that can filter these poisoned samples is important. Nowadays, the trend is towards fully automated systems to eliminate human errors. However, the above defense methods require human interventions.

5.2 Defenses Against Exploratory Attacks

As mentioned in Section 3, the common defense methods against exploratory attacks are disinformation and randomization. The goal in disinformation methods is to hide some of the important information about the system from an adversary. Although determining the features used by the classifier is not difficult, manipulating or mimicking all of these features may be impossible for an adversary. Some features can be neither manipulated nor mimicked. In [31] and [55], the authors found that time-based features (such as account age) are unmodifiable. Furthermore, the authors in [54] discussed how altering some features comes at a cost, while others cannot even be altered. For example, the number of tweets, and the number of followers and following, are features that can easily be mimicked, and they might cause the adversary to create a large number of accounts and buy lots of friends. On the other hand, profile and interaction features are much harder to alter. Consequently, considering the robustness of selected features and applying the disinformation method when designing a spam detector would help reduce the effect of adversaries' activities. However, this cannot stop determined adversaries from trying every way possible to accomplish their goals [46]. Furthermore, as stated in [2], relying on obscurity in an adversarial environment is not a good security practice, as one should always overestimate rather than underestimate the adversary's capabilities. In randomization, the defender's aim is to mislead the adversary by randomizing system's feedback. Unlike the disinformation method, this strategy cannot prevent adversaries from exploiting some information about the system, but makes it harder for them to gain any information [4], especially in Twitter, where the adversary uses the same channel as benign users to discover the system. This makes randomization methods less effective against exploratory attacks in Twitter.

However, some recent studies have proposed an approach that can detect adversarial samples using the deployed classifier's uncertainty in predicting samples' labels. In [48], the authors use multiple classifiers (predict and detect) for detecting adversarial activities. Each classifier detects samples that lie within a classifier's region of uncertainty (blind posts), where the classifier needs to use its best guess. Then, if there is a disagreement between the two classifiers' output, the sample will be tested with labelled samples for confirmation.

6 Conclusion and Future Work

The use of machine learning techniques in security applications has become very common. As spam on OSNs is considered to be an adversarial problem, investigating the security of the machine learning models used to detect spam is very important. Adversaries tend to launch different types of attacks to evade detection by influencing the deployed model either at the training or test phase. Recent studies have shown an increased interest in studying the security of machine learning in domains such as IDS, malware detection, and email spam filters. However, the security of OSNs' spam detectors has not been evaluated sufficiently.

The main contribution of this paper is to provide a general taxonomy of potential adversarial attacks against Twitter spam detectors and a discussion on possible defense strategies that can reduce the effect of such attacks. Examples of adversarial spam tweets that can be used by an adversary were provided. This study is the first step towards evaluating the robustness of Twitter spam detectors, as it identifies potential attacks against them. Hypothetical examples of possible attacks against Twitter spam detectors were based on common frameworks proposed in [4, 5, 12]. In addition, defense methods commonly proposed in the literature and ways of deploying these methods in the context of Twitter spam detection were discussed.

Throughout the paper, a number of challenging issues were mentioned; future research needs to focus on addressing them. Detecting image-based spam is an ongoing problem, as the processing of images overwhelms classifiers and affects detection performance. Adversaries take advantage of this issue, and the amount of image-based spam is increasing. Furthermore, spam detectors designed for spam campaigns may fail to detect single spam and vice-versa. This issue can also be exploited by adversaries when attacking spam detectors. Most proposed defense strategies can make attacks against Twitter spam detectors very hard for adversaries, but, as most adversarial attacks are non-intrusive [47], they cannot completely prevent attacks from happening.

In terms of a future direction, after identifying potential attacks against Twitter spam detectors, the next step is to simulate some of these attacks to evaluate the robustness of Twitter spam detectors. Evaluating the security of Twitter spam detectors experimentally will help design adversarial-aware spam detectors that are more robust against adversarial activities.

Acknowledgments: I would like to warmly thank my supervisor Dr. Vasileios Vasilakis for his feedback and suggestions.

References

1. Al Twairesh, N., Al Tuwaijri, M., Al Moammar, A., Al Humoud,S., 2016, "Arabic Spam Detection in Twitter", *The 2nd Workshop on Arabic Corpora and Processing Tools 2016 Theme: Social Media*, 2016
2. Alabdulmohsin, I.M., Gao, X. & Zhang, X., 2014. Adding Robustness to Support Vector Machines Against Adversarial Reverse Engineering. *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp.231–240.
3. Al-Zoubi A., Alqatawna J. & Faris H. 2017, "Spam profile detection in social networks based on public features", *2017 8th International Conference on Information and Communication Systems (ICICS)*, pp. 130.
4. Barreno, M. et al., 2006. Can Machine Learning Be Secure? In *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*. ASIACCS '06. New York, NY, USA: ACM, pp. 16–25.
5. Barreno, M. et al., 2010. The security of machine learning. *Machine learning*, 81(2), pp.121–148.
6. Barushka, A. & Hajek, P., 2018. Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. *Applied Intelligence*, 48(10), pp.3538–3556.
7. Benevenuto, F., Magno, G., Rodrigues, T. and Almeida, V., 2010, July. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)* (Vol. 6, No. 2010, p. 12).
8. Biggio, B., Fumera, G., et al., 2011a. A survey and experimental evaluation of image spam filtering techniques. *Pattern recognition letters*, 32(10), pp.1436–1446.
9. Biggio, B., Corona, I., et al., 2011b. Bagging Classifiers for Fighting Poisoning Attacks in Adversarial Classification Tasks. In *Lecture Notes in Computer Science*. pp. 350–359.
10. Biggio, B. et al., 2017. Evasion Attacks against Machine Learning at Test Time. *arXiv [cs.CR]*. Available at: http://arxiv.org/abs/1708.06131.
11. Biggio, B., Fumera, G. & Roli, F., 2008. Adversarial Pattern Classification using Multiple Classifiers and Randomisation. *12th Joint IAPR International Workshop on Structural and Syntactic Pattern Recognition (SSPR 2008)*, 5342, pp.500–509.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2019 doi:10.20944/preprints201905.0141.v1

12. Biggio, B., Fumera, G. & Roli, F., 2014. Security Evaluation of Pattern Classifiers under Attack. *Knowledge and Data Engineering, \ldots*, 26(4), pp.984–996.
13. Biggio, B. & Roli, F., 2017. Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning. *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/1712.03141.
14. Bruckner, M., 2012. Static Prediction Games for Adversarial Learning Problems. *Journal of machine learning research: JMLR*, 13, pp.2617–2654..
15. Buckman, J. et al., 2018. Thermometer Encoding: One Hot Way To Resist Adversarial Examples.
16. Cai, Z. et al., 2018. Collective Data-Sanitization for Preventing Sensitive Information Inference Attacks in Social Networks. *IEEE Transactions on Dependable and Secure Computing*, 15(4), pp.577–590.
17. Chakraborty, M. et al., 2016. Recent developments in social spam detection and combating techniques: A survey. *Information processing & management*, 52(6), pp.1053–1073.
18. Chan, P.P.K. et al., 2018. Data sanitization against adversarial label contamination based on data complexity. *International Journal of Machine Learning and Cybernetics*, 9(6), pp.1039–1052.
19. Chan, P.P.K. et al., 2015. Spam filtering for short messages in adversarial environment. *Neurocomputing*, 155, pp.167–176.
20. Chen, C. et al., 2015. Asymmetric self-learning for tackling Twitter Spam Drift. In *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. pp. 208–213.
21. Chen, L., Ye, Y. & Bourlai, T., 2017. Adversarial Machine Learning in Malware Detection: Arms Race between Evasion Attack and Defense. In *2017 European Intelligence and Security Informatics Conference (EISIC)*. pp. 99–106.
22. Chen, S. et al., 2018. Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach. *Computers & Security*, 73, pp.326–344.
23. Corona, I., Giacinto, G. & Roli, F., 2013. Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues. *Information sciences*, 239, pp.201–225.
24. Cresci, S. et al., 2017. The Paradigm-Shift of Social Spambots: Evidence, Theories, and Tools for the Arms Race. In *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, pp. 963–972.
25. Dalvi, N. et al., 2004. Adversarial Classification. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '04. New York, NY, USA: ACM, pp. 99–108.
26. El-Mawass N. &Alaboodi S. 2016, "Detecting Arabic spammers and content polluters on Twitter", *2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC)*, pp. 53.
27. Gao, H. et al., 2014. Spam ain't as diverse as it seems: throttling OSN spam with templates underneath. In *Proceedings of the 30th Annual Computer Security Applications Conference*. ACM, pp. 76–85.
28. Grier, C., Thomas, K., Paxson, V. and Zhang, M., 2010, October. @ spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security* (pp. 27-37). ACM
29. Gupta A.& Kaushal R. 2015, "Improving spam detection in Online Social Networks", *2015 International Conference on Cognitive Computing and Information Processing(CCIP)*, pp. 1.
30. Gupta, P., Perdisci, R. & Ahamad, M., 2018. Towards Measuring the Role of Phone Numbers in Twitter-Advertised Spam. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. ACM, pp. 285–296.
31. Huang, L. et al., 2011. Adversarial Machine Learning. In *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*. AISec '11. New York, NY, USA: ACM, pp. 43–58.
32. Jagielski, M. et al., 2018. Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. , (1). Available at: http://arxiv.org/abs/1804.00308.
33. Kantchelian, A. et al., 2013. Approaches to adversarial drift. Proceedings of the 2013 ACM workshop on Artificial intelligence and security - AISec '13, pp.99–110.
34. Kaur, P., Singhal, A. & Kaur, J., 2016. Spam detection on Twitter: A survey. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. pp. 2570–2573.
35. Laishram, R. & Phoha, V.V., 2016. Curie: A method for protecting SVM Classifier from Poisoning Attack. *arXiv [cs.CR]*. Available at: http://arxiv.org/abs/1606.01584.
36. Lalitha, L.A., Hulipalled, V.R. & Venugopal, K.R., 2017. Spamming the mainstream: A survey on trending Twitter spam detection techniques. In *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*. pp. 444–448.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2019 doi:10.20944/preprints201905.0141.v1

37. Lin G., Sun N., Nepal S., Zhang J., Xiang Y. &Hassan H. 2017, Statistical Twitter Spam Detection Demystified: Performance, Stability and Scalability
38. Lowd, D. & Meek, C., 2005. Good Word Attacks on Statistical Spam Filters. *Conference on email and anti-spam*.
39. Meda, C. et al., 2016. Spam detection of Twitter traffic: A framework based on random forests and non-uniform feature sampling. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. pp. 811–817.
40. Miller, B. et al., 2014. Adversarial Active Learning. Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop - AISec '14, (August), pp.3–14.
41. Newsome, J., Karp, B. & Song, D., 2006. Paragraph: Thwarting Signature Learning by Training Maliciously. In *Lecture Notes in Computer Science*. pp. 81–105.
42. Nilizadeh, S. et al., 2017. POISED: Spotting Twitter Spam Off the Beaten Paths. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, pp. 1159–1174.
43. Pawar, K. & Patil, M., 2015. Pattern classification under attack on spam filtering. In 2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN). pp. 197–201.
44. Sculley, D. et al., 2011. Detecting adversarial advertisements in the wild. Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11, p.274.
45. Sedhai, S. & Sun, A., 2015. HSpam14: A Collection of 14 Million Tweets for Hashtag-Oriented Spam Research. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pp. 223–232.
46. Sethi, T.S., Kantardzic, M., Lyua, L., et al., 2018a. A Dynamic-Adversarial Mining Approach to the Security of Machine Learning. Available at: http://dx.doi.org/10.1002/widm.1245..
47. Sethi, T.S. & Kantardzic, M., 2018b. Data driven exploratory attacks on black box classifiers in adversarial domains. *Neurocomputing*, 289, pp.129–143.
48. Sethi, T.S. & Kantardzic, M., 2018c. Handling adversarial concept drift in streaming data. *Expert systems with applications*, 97, pp.18–40.
49. Song, J., Lee, S. & Kim, J., 2015. CrowdTarget. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS '15, (i), pp.793–804.
50. Stringhini, G., Kruegel, C. & Vigna, G., 2010. Detecting Spammers on Social Networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*. ACSAC '10. New York, NY, USA: ACM, pp. 1–9.
51. Szegedy, C. et al., 2013. Intriguing properties of neural networks. *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/1312.6199.
52. Verma, M., Divya, D. & Sofat, S., 2014. Techniques to Detect Spammers in Twitter- A Survey. *International Journal of Computer Applications in Technology*, 85(10), pp.27–32.
53. Wang, D. et al., 2013. Click Traffic Analysis of Short URL Spam on Twitter. In ICST. Available at: http://dx.doi.org/10.4108/icst.collaboratecom.2013.254084
54. Wang, G. et al., 2014. Man vs. Machine : Practical Adversarial Detection of Malicious Crowdsourcing Workers. *the 23rd USENIX Security Symposium*, pp.239–254.
55. Washha, M., Qaroush, A. & Sedes, F., 2016. Leveraging time for spammers detection on Twitter. In *Proceedings of the 8th International Conference on Management of Digital EcoSystems*. ACM, pp. 109–116.
56. Wu, T. et al., 2017. Twitter Spam Detection Based on Deep Learning. In *Proceedings of the Australasian Computer Science Week Multiconference*. ACSW '17. New York, NY, USA: ACM, pp. 3:1–3:8.
57. Yang, C., Harkreader, R. & Gu, G., 2013. Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers. *IEEE Transactions on Information Forensics and Security*, 8(8), pp.1280–1293.
58. Zhu, T. et al., 2016. Beating the Artificial Chaos: Fighting OSN Spam Using Its Own Templates. *IEEE/ACM Transactions on Networking*, 24(6), pp.3856–3869.
59. Biggio, B., 2011. Support Vector Machines Under Adversarial Label Noise. *JMLR workshop and conference proceedings*, 20, pp.97–112.
60. Baracaldo, N. & Chen, B., Mitigating Poisoning Attacks on Machine Learning Models: A Data Provenance Based Approach. Available at: https://dl.acm.org/ccs.cfm.
61. Cresci, S. et al., 2018. Social Fingerprinting: Detection of Spambot Groups Through DNA-Inspired Behavioral Modeling. *IEEE Transactions on Dependable and Secure Computing*, 15(4), pp.561–576.

62. Chen, C. et al., 2015. 6 million spam tweets: A large ground truth for timely Twitter spam detection. In *2015 IEEE International Conference on Communications (ICC)*. pp. 7065–7070.
63. Mateen, M. et al., 2017. A hybrid approach for spam detection for Twitter. In *2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*. pp. 466–471.
64. Sedhai, S. & Sun, A., 2018. Semi-Supervised Spam Detection in Twitter Stream. *IEEE Transactions on Computational Social Systems*, 5(1), pp.169–175.
65. Gupta, S. et al., 2018. Collective Classification of Spam Campaigners on Twitter: A Hierarchical Meta-Path Based Approach. In *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, pp. 529–538.
66.