robotics

MDPI

Article

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

Niddal H. Imam * and Vassilios G. Vassilakis

Department of Computer Science, University of York, York YO10 5DD, UK

* Correspondence: ni571@york.ac.uk; Tel.: +44-7447-831-899

Received: 10 May 2019; Accepted: 1 July 2019; Published: 4 July 2019

check for updates

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 vulnerable to a range of adversarial attacks. Thus, this paper surveys the attacks against Twitter spam detectors in an adversarial environment, and a general taxonomy of potential adversarial attacks is presented using common frameworks from the literature. Examples of adversarial activities on Twitter that were discovered after observing Arabic trending hashtags are discussed in detail. A new type of spam tweet (adversarial spam tweet), which can be used to undermine a deployed classifier, is examined. In addition, possible countermeasures that could increase the robustness of Twitter spam detectors to 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 showed that the number of profiles on Facebook, Twitter, and LinkedIn reached more than 2 billion in 2016 [1].

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 [2]. The term "spam" refers to an unsolicited message that is received from a random sender who has no relationship with the receiver. These messages can contain malware, advertisements, or URLs that direct the recipients to malicious websites [3]. Spamming on the Internet first appeared in the 1990s in the form of email spam [1]. 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 on OSNs because of the growing number of spammers and the possible negative effects on users [3,4].

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 [5,6]. On Twitter, spammers tweet for several reasons, such as to spread advertisements, disseminate pornography, spread viruses, phishing, or simply compromise a system's reputation [7]. Furthermore, in Ref. [8], the authors asserted that a tweet is considered spam if it is not composed purely of text. Instead, it contains a hashtag, a mention, a URL, or an image. Various types of spam are found on OSNs, including textual pattern spam [9], image spam [10,11], URL-based spam [12], and phone number-based spam [13]. Whilst most previous studies have focused on detecting the above types of spam, few have attempted to detect advertisement spam. The authors in Ref. [14] categorized adversarial advertisements as counterfeit goods, misleading or inaccurate claims, phishing, arbitrage, and malware. The diversity of spam on OSNs makes it very hard for any single existing method to detect most spam [15]. Several reported incidents reveal the danger of spammers on OSNs. For example, a number of NatWest bank customers were victims of a Twitter-based phishing attack that used spam tweets that looked very similar to those from the official NatWest customer support account [1]. A recent study noted that the increase in OSN spammers, who distribute unsolicited spam and advertise untrustworthy products, affects the public's perception of companies, which can eventually lead to people developing biased opinions [16].

The issue of spamming via OSNs has become an area of interest to a number of researchers, many of whom have proposed solutions 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 ML approaches include email spam filters, intrusion detection systems (IDSs), and malware detectors [17]. ML techniques aim to automatically classify messages as either spam or non-spam. Various OSN spam detectors have been developed using ML algorithms, including Support Vector Machine (SVM) [7], Random Forests (RF) [18,19], and, more recently, Deep Neural Networks [3].

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 [20–22]. The traditional assumption of stationarity of data distribution in ML is that the datasets 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 [20,23].

Studying the robustness of OSNs' spam detectors to adversarial attacks is crucial. The security of ML techniques is a very active area of research. Whilst several studies have examined the security of IDSs, email filters, and malware detectors, few have investigated the security of OSNs' spam detectors. To the best of the researcher's knowledge, a survey of adversarial attacks against OSNs' spam detectors has not been performed. Recent studies have suggested that the achievement of a secure system necessitates the prediction of potential attacks (i.e., before they occur) to develop suitable countermeasures [21]. Thus, the main goal of this paper is to present a comprehensive overview of different possible attacks, which is the first step toward evaluating the security of OSNs' spam detectors in an adversarial environment. The key contributions of this paper are threefold.

1. After observing Arabic trending hashtags, it was found that there were very active spam campaigns spreading advertisements for untrustworthy drugs targeting Arabic-speaking users. These campaigns were studied and examples of a new type of spam tweet, which we called the adversarial spam tweet that can be used by an adversary to attack Twitter spam detectors, are presented.
2. A general survey of the possible adversarial attacks against OSNs' spam detectors is provided. The paper proposed different hypothetical attack scenarios against Twitter spam detectors using common frameworks for formulizing attacks against ML systems.
3. In addition, potential defense mechanisms that could reduce the effect of such attacks are investigated. Ideas proposed in the literature are generalized to identify potential adversarial attacks and countermeasures. Twitter, which is one of the most popular OSN platforms, is used as a case study, and it is the source of all examples of attacks reported herein.

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. Literature Review

2.1. 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 [19].

Researchers who use ML approaches build their models by employing some of the common spam detection techniques. On the basis of surveys in Refs. [24,25], Twitter spam detectors can be classified into 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 according to an account's features and other attributes that provide useful information about users' behavior. Content-based techniques use a tweet's content, 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 relation-based techniques are distance and connectivity (see Table 1).

Table 1. Feature categories and description [24,26].

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 Ref. [27] 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 a tweet's content, 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 the utilized classifier 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 robustness can be increased by adopting a social graph to detect spam by analyzing mathematical features, such as social distance and connectivity between followers. In the last group—blacklist-based detectors—accounts and tweets are blocked on the basis of users' feedback or the URL's reputation. The first study of the effectiveness of some Twitter spam detection techniques that have been used in the past was presented in Ref. [28]. Examples included 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 2 days of receipt, it took 4–20 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 the authors asserted that studying the regional response rate could improve spam detection [29]. Furthermore, to overcome the limitations of the blacklist, some preliminary studies have used heuristic rules to filter Twitter spam [29].

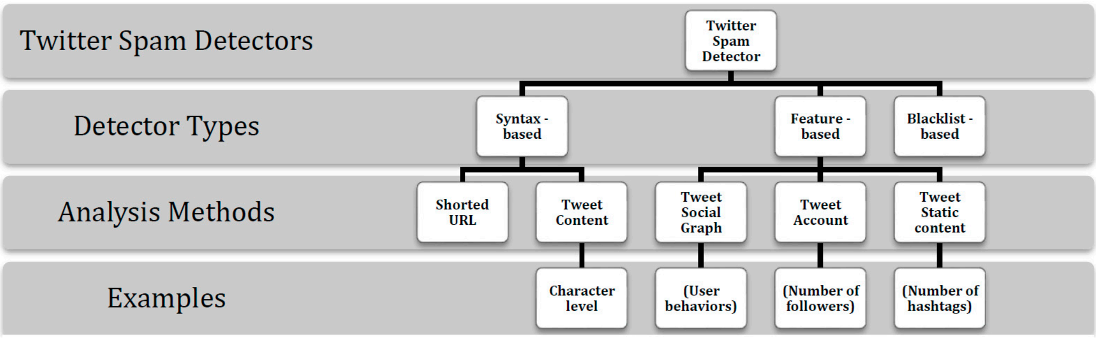


Figure 1. Types of Twitter Spam Detectors [27].

2.2. *Process of Detecting Spam through ML*

The process of spam detection using ML comprises several steps. The first step involves collecting data from Twitter using its Streaming Application Programming Interference (API). This is followed by data preprocessing, which includes feature extraction, data labeling, and dataset splitting. However, for textual spam detectors, the preprocessing 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. Examples of these features include account age, the number of followers or friends, and the number of characters. Data labeling or ground truth is the process in which the collected data are labeled 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 entails training the chosen classification algorithm by using the labeled data, followed by performance evaluation, during which the trained machine learning classifier can be used for spam detection [29,30] (see Figure 2).

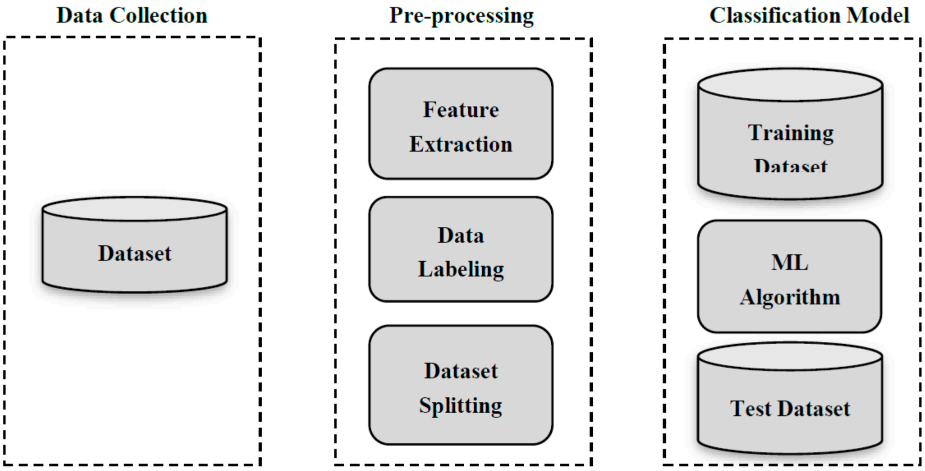


Figure 2. Learning Process of ML-based Spam Detection.

2.3. *Detecting Spam Tweets using ML Techniques*

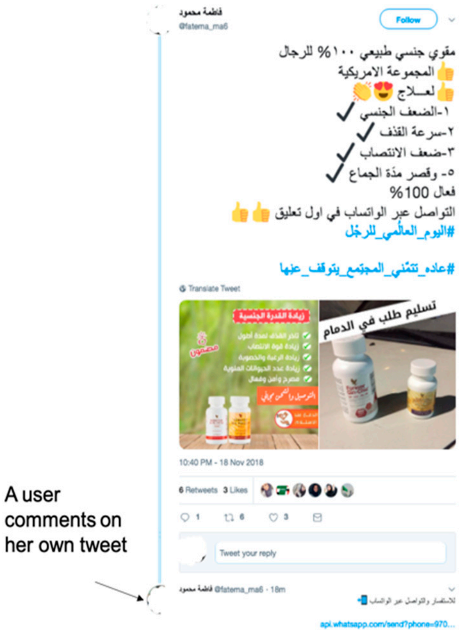
Different studies have used techniques other than blacklists for Twitter spam detection. As mentioned earlier, a number of steps are involved when implementing ML techniques to detect spam on Twitter; some of the important steps are discussed here. Several studies of Twitter spam detection developed their models by employing different ML algorithms, such as SVM, Naive-Bayes (NB), and RF, among which RF has shown the best detection accuracy. In Ref. [26], 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 using random forests was proposed in Ref. [18]. In addition, RF was chosen as the best of five other algorithms in Ref. [19], as it achieved the best evaluation metrics. Selecting features that can best facilitate the classification of samples is as important a step as choosing the most suitable algorithm for the required task. In Ref. [29], the authors collected a large dataset and labeled approximately 6.5 million spam tweets. They found that when using an imbalanced dataset that simulated a real-world scenario, the classifiers' ability to detect spam tweets was reduced. On the other hand, when features were discretized, the performance of the classifiers improved.

Feature selection is a very important step in the ML learning process, and one of the best ways to do so is to build and analyze a dataset. In addition, RF has been shown to be very effective when used in ML-based detection models in OSNs.

2.4. *Detecting Spam Campaigns on Twitter*

Unlike the above spam detection models developed to detect a single spammer, some approaches can be used to detect spam campaigns (spambots). According to Ref. [16], 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 showed that neither humans nor existing machine learning models could detect spambots accurately, the result of an emerging technique deploying digital DNA has achieved a very promising detection performance. Similarly, the authors in Ref. [31] 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 [19] 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, the authors 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. In Ref. [15], an approach called Tangram, which uses a template-based model to detect spam on OSNs, was proposed. After analyzing the textual pattern of a large collection of spam, the researchers found that the largest proportion of spam was generated with an underlying template compared with other spam categories (for example, paraphrase or no-content).

It is important to consider both single and campaign spam when developing an ML model. Although user interactions (e.g., the number of comments on a tweet) can distinguish a spam account from a real account, a spam tweet can be sent from a real account. Furthermore, spam accounts have been observed to manipulate this feature by commenting on their tweets (see Figure 3). In addition, spammers now tend to use trending hashtags, thereby delaying the detection of spam as a result of analyzing user interactions. Digital DNA techniques can be used for analyzing datasets and identifying the key behavioral features of spam on OSNs, which will help with feature selection. Time-based features are robust to feature manipulation because they are hard to alter or mimic.



فاطمة محمود
@fatema_ma6

مقوي جنسي طبيعي ١٠٠% للرجال
👍المجموعة الامريكية
👍لعــلاج 😍👏
١-الضعف الجنسي ✓
٢-سرعة القذف ✓
٣-ضعف الانتصاب ✓
٥- وقصر مدة الجماع ✓
فعال 100%
التواصل عبر الواتساب في اول تعليق 👍👍
#اليوم_العالمي_للرجل

#عاده_تتمّني_المجتمع_يتوقف_عنها

10:40 PM - 18 Nov 2018

6 Retweets 3 Likes

1 6 3

فاطمة محمود @fatema_ma6 · 18m
للاستفسار والتواصل عبر الواتساب 📲
api.whatsapp.com/send?phone=970...

Figure 3. A spam tweet with fake comments in order to resemble a legitimate user's tweet.

2.5. *Security of Twitter Spam Detectors*

Despite the success and high accuracy of the described models in detecting Twitter spam, they are nevertheless vulnerable because they were not developed for adversarial settings. A popular framework for evaluating secure learning was proposed in Ref. [32] and extended in Refs. [21,33,34]; it enables different attack scenarios to be envisaged against machine learning algorithms. The framework suggests the following steps: (1) identify potential attacks against machine learning models by using the popular taxonomy (see Section 3.1); (2) simulate these attacks to evaluate the resilience of ML models, and assume that the adversary's attacks are implemented according to their goals, knowledge, and capabilities/resources; (3) investigate some possible defense strategies against these attacks. Defending against adversarial attacks is challenging because these attacks are non-intrusive in nature and are launched by an adversary using the same channel as legitimate users. Thus, defense strategies against these attacks cannot employ traditional encryption/security techniques [35]. Figure 4 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 the development of suitable defense methods before an attack occurs [21]. This has motivated researchers to formulate 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.

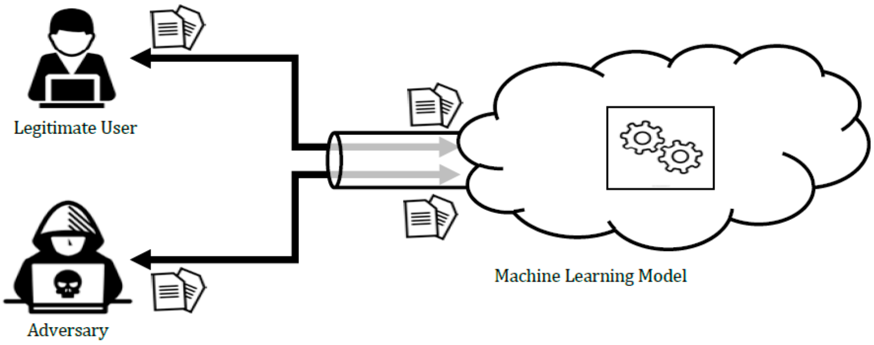


Figure 4. An adversary uses the same channel as legitimate users to exploit knowledge about the system.

2.6. Adversarial Attacks against Twitter Spam Detectors

In Ref. [36], the authors evaluated the security of an 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, in which an adversary has limited knowledge, and optimal evasion, in which the adversary has perfect knowledge. The results showed that an optimal evasion attack has a much higher impact than a basic one. However, in the real world, it is very difficult for adversaries to have perfect knowledge about the system. Thus, the less knowledge that 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 Ref. [37] analyzed the robustness to evasion and poisoning attacks of a Twitter spam detector called POISED. POISED is designed to distinguish between spam and non-spam messages on the basis of the propagation of messages in each campaign. In a poisoning attack, 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 imitates the propagation of non-spam messages to evade detection. The results showed that, in both attacks, the performance of POISED decreased when the percentage of compromised communities increased. Thus, the authors suggested that an adversary can only successfully attack systems if he or she has perfect knowledge about the structure and network of the targeted community.

Previous works have concluded that an adversary's level of knowledge about the deployed model plays an important role in determining the success of attacks. This supports the common defense strategies used in the literature, namely, disinformation and randomization. Furthermore, the authors in Ref. [36] suggested that even if an adversary knows the importance of features used by the deployed model, he or she will not be able to evade detection without knowing the ML algorithm used. However, this cannot stop determined adversaries from trying every way possible to accomplish their goals [23]. Furthermore, as stated in Ref. [20], relying on obscurity in an adversarial environment is not good security practice, as one should always overestimate rather than underestimate the adversary's capabilities. Both works concluded that poisoning attacks are more dangerous since models need to be updated over time. Most importantly, both disinformation and randomization approaches focus on making the models hard to attack, but they do not provide measures to be taken once an attack is detected.

Table 2. Outline of some recent techniques used for detecting spam on 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 [38]	Different ML algorithms were used; balanced and imbalanced datasets were tested.	Spam tweet	Feature-based	Supervised	RF outperformed other algorithms.
A Hybrid Approach for Spam Detection for Twitter [39]	J48, Decorate, and Naive-Bayes (NB).	Spam tweet	Feature-, user-, and graph-based	Supervised	J48 outperformed other algorithms.
Leveraging Time for Spammers Detection on Twitter [19]	Time-based features were used, and different ML algorithms were tested.	Spam tweet	Feature-based	Supervised	RF outperformed other algorithms.
Twitter spam detection based on Deep Learning [27]	Different ML algorithms with Word2Vector technique were used.	Spam tweet	Syntax-based	Supervised	RF with Worrd2Vec outperformed other algorithms.
Semi-supervised spam detection (S3D) [40]	Utilized four lightweight detectors (supervised and unsupervised) to detect spam tweets and updated the models periodically in batch mode.	Spam tweet	Feature-based and blacklist	Semi-supervised	The confidential labeling process, which uses blacklisted, near-duplicated, and reliable non-spam tweets, made the deployed classifier more efficient when detecting new spam tweets.
CrowdTarget: Target-based Detection of Crowdturfing in Online Social Networks [31]	Detected spam tweets that received retweets from malicious crowdsourcing users. It used four new retweet-based features and KNN as a classifier.	Spam tweet	Feature-based	Supervised	CrowdTarget detected 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 [9]	Detected template-based spam, paraphrase spam, and URL-based spam.	Spam campaign	Syntax-based	Supervised	Template detection outperformed URL blacklist detection.
POISED—Spotting Twitter Spam Off the Beaten Paths [37]	Detected spam campaigns based on community and topic of interest.	Spam campaign	Syntax-based	Supervised and unsupervised	NB, SVM, and RF all achieved about 90% detection accuracy.
Collective Classification of Spam Campaigners on Twitter: A Hierarchical Meta-Path Based Approach (HMPS) [41]	Built heterogeneous networks and detected nodes connected by the same phone number or URL.	Spam campaign	Social graph-based	Supervised	HMPS outperformed feature- and content-based approaches. Prediction accuracy improved when using HMPS with feature- and user-based approaches.
Social Fingerprinting—Detection of Spambot Groups Through DNA-Inspired Behavioral Modeling [42]	Modeled users' behavior using a DNA fingerprinting technique to detect spambots.	Spam campaign	Social graph-based	Supervised and unsupervised	The results showed that the proposed approach achieved 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 [43]. Adversarial examples are inputs to ML that are designed to produce incorrect outputs [44]. The term was first introduced in Ref. [45] and used for computer vision, but in the context of spam and malware detection, the term evasion attacks is used in Ref. [21]. This section discusses different adversarial attacks and countermeasures. Tables 3 and 4 outline recent works in adversarial machine learning.

3.1. Taxonomy of Attacks Against ML

A popular taxonomy proposed in Refs. [21,32,33] categorizes 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 of influencing 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 dataset by injecting it with carefully crafted samples. 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 regarded as 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, in which it misclassifies non-spam samples as spam (false positives), or if it prevents legitimate users from accessing the system. The security violation can be regarded as 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 (1) causes the classifier to misclassify a single or few instances or (2) undermines the classifier's performance on a larger set of instances.

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 [46]	Poisoning	Regression Learning	Optimization-based poisoning attack, which uses different optimization approaches, and statistical-based poisoning attack (StatP), which queries a deployed model to find an estimate of the mean and covariance of training data.
	Support vector machines under adversarial label noise [47]	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 [48]	Label Flipping	SVM	Two label-flipping attacks were used. In the first, the loss maximization framework was used to select points that needed their label to be flipped. In the second attack, the selected data points were moved to other points in the feature space.
	Adversarial Machine Learning [34]	Dictionary	Spam filter	An adversary builds a dictionary of tokens learned from the targeted model and then sends attack messages to cause misclassification.
	Thwarting Signature Learning by Training Maliciously [49]	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 [36]	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 [35]	Anchor Points (AP) and Reverse Engineering attacks (RE)	SVM, KNN, DT, RF	The 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 [11]	Evasion	SVM, Neural Network	A gradient-descent evasion attack was proposed.
	Good Word Attacks on Statistical Spam Filters [50]	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 [20]	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 [38]	Evasion	NB, BN, SVM, J48, RF	Two evasion attack were launched: Basic evasion attack and Optimal evasion attack, where an adversary knows features that need to be altered.

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 [51]	Poisoning	SVM	Data Sanitization	Poisoned data are filtered out 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 [48]	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 a feature with proper weight. Data points with less than 95% confidence are removed from the training data.
	Bagging Classifiers for Fighting Poisoning Attacks in Adversarial Classification Tasks [52]	Poisoning	Bagging and weighted bagging ensembles	Data Sanitization	An ensemble construction method (bagging) is used to remove outliers (adversarial samples) from the training dataset.
	Data sanitization against adversarial label contamination based on data complexity [53]	Label Flipping	SVM	Data Sanitization	Data complexity, which measures the level of difficulty of classification problems, is used to distinguish adversarial samples in the training data.
	Support vector machines under adversarial label noise [47]	Label Flipping	SVM	Robust learning	Adjusting the kernel matrix of SVM depending on noisy (adversarial) samples' parameters increases the robustness of the classifier.
	Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning [46]	Poisoning	Regression Learning	Robust learning	The TRIM algorithm, which regularizes linear regression by applying trimmed optimization techniques, was proposed.
Exploratory	Robust support vector machines against evasion attacks by random generated malicious samples [54]	Evasion	SVM	Robust learning	The SVM classifier is trained with random malicious samples to enclose the decision function.
	Adding Robustness to Support Vector Machines Against Adversarial Reverse Engineering [20]	Reverse Engineering	SVM	Randomization	A distribution of classifiers is learned and a decision boundary is picked randomly to make reverse engineering attacks harder to launch.
	Handling adversarial concept drift in streaming data [22]	Evasion	SVM	Disinformation	The importance of features is hidden and an ensemble of classifiers is used.
	Adversarial Pattern Classification using Multiple Classifiers and Randomization [55]	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.

3.2. Common Types of Threat Models

After presenting the taxonomy of attacks against ML systems, the next step toward identifying potential attack scenarios is threat modeling, which involves defining an adversary's goal, knowledge, and capability [21,32,33]. 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 (gray-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 Word 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 [33]. 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 [36]. 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 [48]. The easiest method is to randomly flip the label of some samples that might be used for retraining. In Ref. [47], it was shown that randomly flipping about 40% of the training data's labels decreased the prediction accuracy of the deployed classifier. A red herring attack is a type of causative attack in which the adversary adds irrelevant patterns or features to the training data to mislead the classifier so that it focuses on these irrelevant patterns [20,49]. Defending against causative attacks is challenging because ML classifiers need to be retrained periodically to adapt to new changes. Retraining the classifier makes it vulnerable because the data used for retraining are collected from an adversarial environment [48].

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; in these methods, the defense problem is modeled as a game between the adversary and the classifier [20,56–58]. 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 contaminated samples [53]. 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 training another classifier. The prediction accuracy for both classifiers on labeled test data is evaluated. If adding suspicious samples to the training data reduces the prediction accuracy, these samples must be removed [34]. Another defense method involves using multiple classifiers, which has been shown to reduce the influence of poisoned samples in training data [52].

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) [32]. 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 [50].

3.3.4. Exploratory Defense Methods

As with causative attacks, it is difficult to prevent exploratory attacks 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, concealing the 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 [32].

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 to Twitter spam detectors. The following section examines some of these techniques in the context of Twitter spam detectors.

4. 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 using the framework of the popular attack taxonomy presented in Refs. [32,33] that categorizes attacks along three axes: influence, security violations, and specificity. This framework was extended in Ref. [21] to derive the corresponding optimal attack strategy by modeling 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 to be 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 an adversary when they attack spam detectors. The adversary's capability is based on their desired goals. For example, if the goal is to influence the training data, the adversary must be capable of doing so. Examples of adversarial spam tweets 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 were very active spam campaigns spreading advertisements for untrustworthy drugs, such as weight loss drugs, Viagra, and hair treatment drugs, targeting Arabic-speaking users. The attack scenarios can be modeled as follows:

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

4.2. Potential Attack Scenarios

Here, attacks against Twitter spam detectors are 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 messages with Arabic hashtags to show how an adversary can manipulate tweets. Figure 5 shows a diagram of the proposed taxonomy of potential attacks against Twitter.

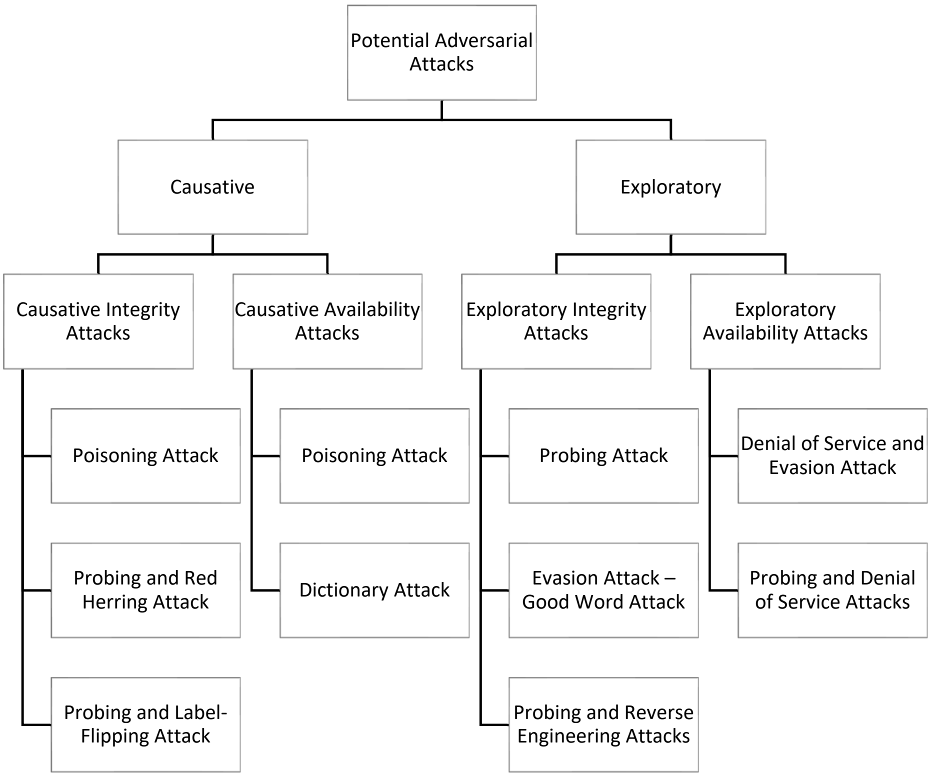

Potential Adversarial Attacks
Causative
Exploratory
Causative Integrity Attacks
Causative Availability Attacks
Exploratory Integrity Attacks
Exploratory Availability Attacks
Poisoning Attack
Probing and Red Herring Attack
Probing and Label-Flipping Attack
Poisoning Attack
Dictionary Attack
Probing Attack
Evasion Attack – Good Word Attack
Probing and Reverse Engineering Attacks
Denial of Service and Evasion Attack
Probing and Denial of Service Attacks

Figure 5. Diagram of the Proposed Taxonomy.

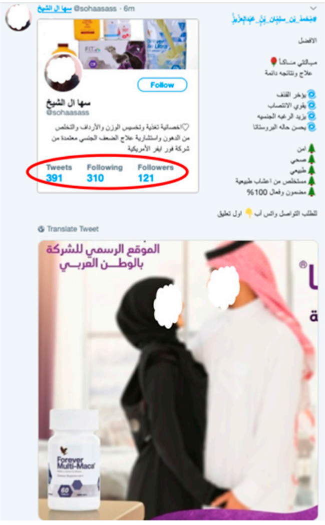
4.2.1. Causative Integrity Attacks

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 6).
- 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 6 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 by having a decent number of followers and friends, a profile photo, and a description. This spam tweet bypasses Twitter's spam detector and could be used for retraining the classifier.



سها ال الشيخ @sohaasass · 6m

#محمد_بن_سلمان_ولي_عهدالعزيز

الافضل

مبالتي مناكا 📍
علاج ونتائجه دائمة

يؤخر القذف
يقوي الانتصاب
يزيد الرغبه الجنسيه
يحسن حاله البروستاتا

🌲 امن
🌲 صحي
🌲 طبيعي
🌲 مستخلص من اعشاب طبيعية
🌲 مضمون وفعال 100%

للطلب التواصل واتس آب 👇 اول تعليق

Follow

سها ال الشيخ
@sohaasass

♡اخصائية تغذية وتخسيس الوزن والأرداف والتخلص من الدهون واستشارية علاج الضعف الجنسي معتمدة من شركة فور ايفر الأمريكية

Tweets	Following	Followers
391	310	121

Translate Tweet

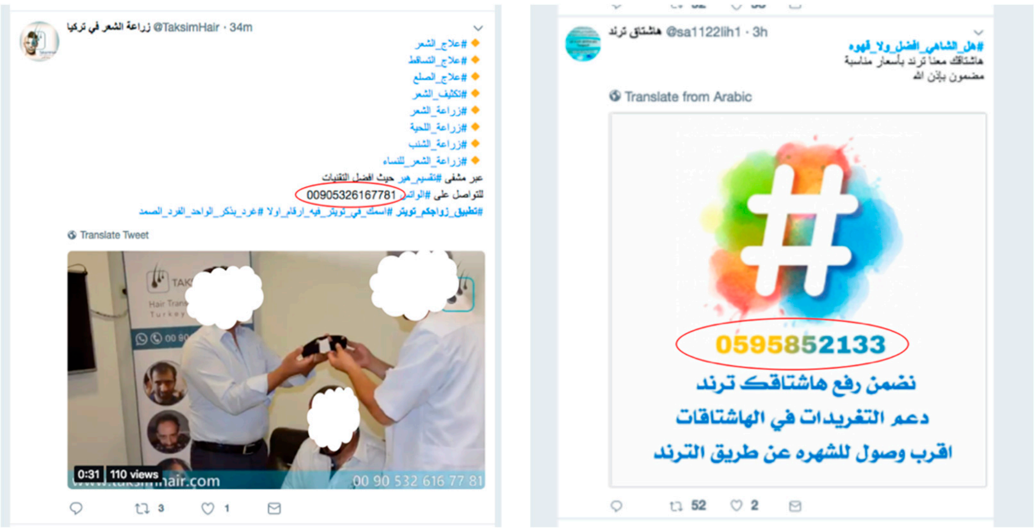
Figure 6. A spam tweet resembling a non-spam tweet to poison training data.

Example 2: Probing and Red Herring Attack

As in Ref. [49], 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). With these scenario 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 is discussed in the next section.
- The adversary crafts samples with spurious or fake features and posts these samples with 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 7a shows an example of a spam tweet that has a spurious feature (phone number). Because the number of tweets that have a phone number has increased on Twitter, some proposed spam detectors suggest using a phone number as an indicator of spam tweets [13,30]. However, Figure 7b shows how the adversary can trick Twitter into using a phone number as a feature and avoid including phone numbers in his spam tweets. Instead, the adversary includes a phone number inside an image to evade detection.



زراعة الشعر في تركيا @TaksimHair · 34m

- 🔶 #علاج_الشعر
- 🔶 #علاج_التساقط
- 🔶 #علاج_الصلع
- 🔶 #تكثيف_الشعر
- 🔶 #زراعة_الشعر
- 🔶 #زراعة_اللحية
- 🔶 #زراعة_الشنب
- 🔶 #زراعة_الشعر_للنساء

عبر مشفى #تقسيم_هير حيث افضل التقنيات
للتواصل على #الواتس 00905326167781
#تطبيق_زواجكم_تويتر #اسمك_في_تويتر_فيه_ارقام_اولا #غرد_بذكر_الواحد_الفرد_الصمد

هاشتاق ترند @sa1122lih1 · 3h

#هل_الشاهي_افضل_ولا_قهوه
هاشتاقك معنا ترند بأسعار مناسبة
مضمون بإذن الله

0595852133
نضمن رفع هاشتاقك ترند
دعم التغريدات في الهاشتاقات
اقرب وصول للشهره عن طريق الترند

(**a**) (**b**)

Figure 7. (**a**) A spam tweet containing a spurious feature (a mobile number). (**b**) A spam tweet with a mobile number inside an image to evade detection.

Example 3: Probing and Label-Flipping Attack

The aim of this attack scenario, as in Ref. [48], is to cause misclassification by injecting label-flipped samples into training data. The settings of the attack scenario are as follows: 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 that in the previous example. However, the adversary's knowledge is assumed to be zero (black-box attack). According to 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 Section 4.2.4 Figure 9) 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 gains, he or she can either flip the nearest or farthest samples from the deployed classifier's decision boundary.
- If the adversary did not learn more about the classifier, he or she can randomly flip the label of some tweets.
- He or she then randomly posts these tweets using different trending hashtags and hopes 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.

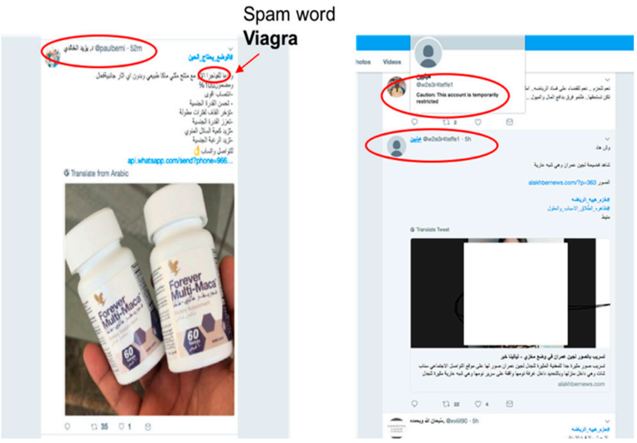
- As the adversary's knowledge about the system is considered to be perfect, sending probing tweets to gain knowledge is not required.
- The adversary carefully crafts a large number of misleading tweets that consist of a combination of spam and non-spam components.
- The adversary needs to contaminate 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 is to post these tweets randomly using different trending hashtags so that they quickly spread in the hope that Twitter will use them when retraining their system.

Example 2: Dictionary Attack

In this attack, as in Ref. [34], 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: The 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.
- On the basis of the adversary's knowledge, he or she builds a dictionary of words or phrases that are frequently used by legitimate users and uses this to craft malicious tweets.
- The adversary posts tweets that contain a large set of tokens (non-spam words, phrases, or tweet structure) from the dictionary in trending hashtags.
- If these tweets are used to train the system, non-spam tweets are more likely to be classified as spam because the system gives a higher spam score to tokens used in the attack.

Figure 8 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. The 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 as a result of the attack, the tweet remains posted for longer than 52 min. In addition, the spam tweet on the right-hand side remains undetected for longer than 5 h, which is a very long time.



@paulbemi · 52m

-انتصاب اقوى
- تحسن القدرة الجنسية
-تؤخر القذف لفترات طويلة
-تعزز القدرة الجنسية
-تزيد كمية السائل المنوي
-تزيد الرغبة الجنسية
للتواصل واتساب
api.whatsapp.com/send?phone=966...

Translate from Arabic

@w2e3r4teffe1
Caution: This account is temporarily restricted

@w2e3r4teffe1 · 5h

شاهد فضيحة لجين عمران وهي شبه عارية

alakhbernews.com/?p=363

Translate Tweet

Figure 8. Spam tweets bypass the detection system as a result of the availability attack.

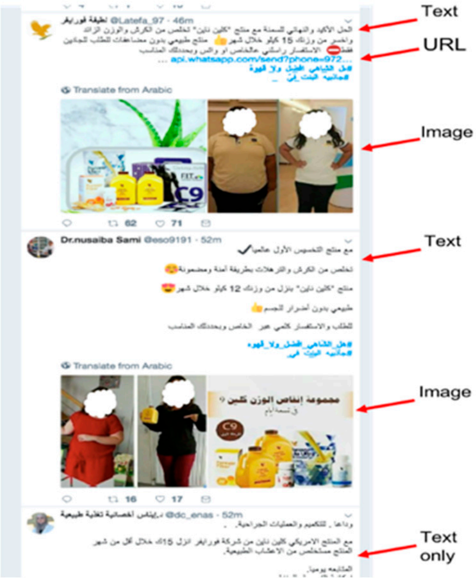
4.2.3. Exploratory Integrity Attack

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 on the training data. The settings of the attack scenario are as follows: The 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 Ref. [21], 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.
- The adversary sends a large number of tweets, each with different features, to learn about the system (see Figure 9).
- Using the information that is learned, the adversary carefully crafts tweets to evade detection.

Figure 9 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.



لطيفة فوريفر @Latefa_97 · 46m
الحل الأكيد والنهائي للسمنة مع منتج "كلين ناين" لتخلص من الكرش والوزن الزائد
واخسر من وزنك 15 كيلو خلال شهر 👍 منتج طبيعي بدون مضاعفات للطلب الجادين
فقط⛔ الاستفسار راسلني عالخاص او واتس ويحددلك المناسب
api.whatsapp.com/send?phone=972...
#ل التامي الطبل ولا قهوة
#مالتيه البنت_إن

Translate from Arabic

62 71

Dr.nusaiba Sami @eso9191 · 52m
مع منتج التخسيس الأول عالميا✔
تخلص من الكرش والترهلات بطريقة آمنة ومضمونة😍
منتج "كلين ناين" بنزل من وزنك 12 كيلو خلال شهر😍
طبيعي بدون أضرار للجسم👍
للطلب والاستفسار كلمي عبر الخاص ويحددلك المناسب
#ل التامي الطبل ولا قهوة
#مالتيه البنت_إن

Translate from Arabic

مجموعة إنقاص الوزن كلين 9
في تسعة أيام

16 17

د.إيناس أخصائية تغذية طبيعية @dc_enas · 52m
وداعا . للتكميم والعمليات الجراحية.
مع المنتج الامريكي كلين ناين من شركة فوريفر انزل 15ك خلال أقل من شهر
المنتج مستخلص من الاعشاب الطبيعية.
المتابعه يوميا.

Figure 9. An example of a probing attack.

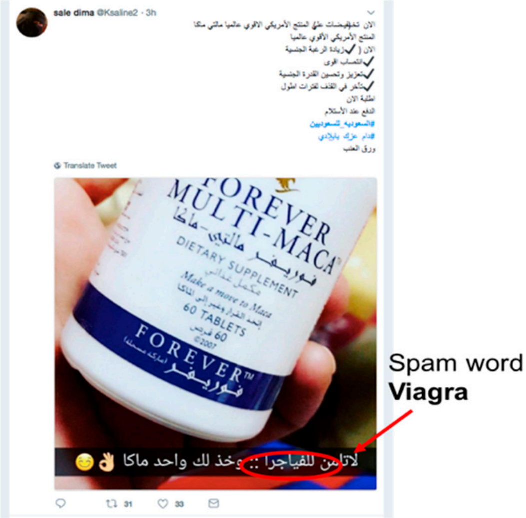
Example 2: Evasion Attack—Good Word Attack

In this attack scenario, the aim is to evade being detected by the deployed classifier without any direct influence on the training data. The settings of the attack scenario are as follows: The 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 Ref. [21], 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.

- Using his or her knowledge, the adversary carefully crafts tweets by modifying and obfuscating spam words (such as "Viagra") or the tweet's features to evade detection (such as the number of followers) (see Figure 10).

Figure 10 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 is inserted inside an image.



sale dima @Ksaline2 · 3h

الان تخفيضات على المنتج الأمريكي الاقوى عالميا مالتي مكا
المنتج الأمريكي الأقوى عالميا
الان (✓زيادة الرغبة الجنسية
✓انتصاب اقوى
✓تعزيز وتحسين القدرة الجنسية
✓تأخر في القذف لفترات اطول
اطلبة الان
الدفع عند الأستلام
#السعوديه_للسعوديين
#دام_عزك_يابلادي
ورق العنب

Figure 10. 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 in this attack scenario. The scenario's settings are as follows: 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 that in the previous example, but the adversary's knowledge about Twitter's spam detectors is assumed to be zero (black-box attack). From these scenario settings, a potential attack strategy is as follows.

- As the adversary has zero knowledge about the system, the first step is to send probing tweets to learn how the system works (privacy violation).
- Using the exploited knowledge, the adversary builds a substitute model that can be used for launching different exploratory attacks [20].
- Once the substitute model is built, the adversary crafts different spam tweets to evade detection, and spam tweets that successfully evade the model are used against the Twitter spam detector.

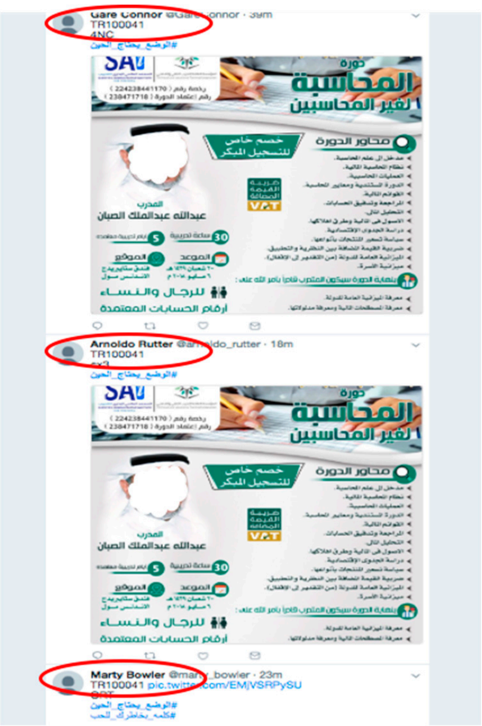
4.2.4. Exploratory Availability Attack

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 on the training data. The settings of the attack scenario are as follows: The 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 Ref. [21], 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.
- Using the gained knowledge, the adversary carefully crafts spam tweets. As the adversary cannot influence training data, the adversary crafts tweets that require more time for the classifier to process, such as image-based tweets [33].
- The adversary then floods the system (for example, by using a particular trending hashtag) with spam tweets to prevent users from reading non-spam tweets and this causes difficulty in detecting spam tweets.

Figure 11 shows an example of an availability attack, in which the adversary uses a different account to post a large number of spam tweets that only contain 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.



Gare Connor @Gare_Connor · 39m
TR100041
4NC
#الوضع_يحتاج_الحين

Arnoldo Rutter @arnoldo_rutter · 18m
TR100041
ex3
#الوضع_يحتاج_الحين

Marty Bowler @marty_bowler · 23m
TR100041 pic.twitter.com/EMjVSRPySU
@RT
#الوضع_يحتاج_الحين
#كلمه_بخاطرك_للحب

Figure 11. An adversary uses a hashtag to flood the system with spam tweets.

Example 2: Probing and Denial of Service Attacks

The aim of this attack scenario is similar to that in 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 that in the previous example, but the adversary's knowledge about Twitter's spam detectors is assumed to be zero (black-box attack). From these scenario settings, a potential attack strategy is as follows.

- As the adversary has zero knowledge about the system, the first step is to probe the classifier with some tweets to learn how it works.
- Using the exploited knowledge, the adversary crafts a large number of spam tweets and posts them with a specific hashtag to cause denial of service and make future attacks easier [20].

All attack examples can be either targeted (if an adversary focuses on a specific spam tweet, such as URL-based spam or weight-loss ads) or indiscriminate (if an adversary targets multiple types of spam tweets, such as URL-based tweets and advertisements). Although the presented adversarial spam tweets look very similar to spam tweets that target users, this special type of spam tweet needs to be studied more because 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 and Privacy	
	Probing and Label-Flipping Attack	Integrity and Privacy	
	Poisoning Attack	Availability	
	Dictionary Attack	Availability and Integrity	
Exploratory	Probing Attack	Privacy	
	Good Word Attack	Integrity	
	Probing and Reverse Engineering Attacks	Integrity and Privacy	
	Denial of Service and Evasion Attack	Availability and Integrity	
	Probing and Denial of Service Attacks	Availability, Integrity, and Privacy	

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 defending against causative attacks focus on filtering or screening all the training data before using them to update a deployed classifier; such approaches include 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 [53]. Some recent studies have suggested using a data collection oracle to retrain a deployed classifier [22,59]. However, trusting an oracle to label training data could be problematic. The authors in Ref. [60] 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 [31]. Thus, developing a fully automated model that can filter these poisoned samples is important. Nowadays, the trend is toward 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 Refs. [19,34], the authors found that time-based features (such as account age) are unmodifiable. Furthermore, the authors in Ref. [36] discussed how altering some features comes at a cost, while others cannot even be altered. For example, the number of tweets, the number of followers, and the number of 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 could help reduce the effect of adversaries' activities. However, this cannot stop determined adversaries from trying every way possible to accomplish their goals [23]. Furthermore, as stated in Ref. [20], relying on obscurity in an adversarial environment is not 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 the system's feedback. Unlike the disinformation method, this strategy cannot prevent adversaries from exploiting some information about the system, but it makes it harder for them to gain any information [32], especially on Twitter, where the adversary uses the same channel as that used by benign users to discover the system. This makes randomization methods less effective against exploratory attacks on 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 Ref. [22], the authors used multiple classifiers (predict and detect) for detecting adversarial activities. Each classifier detects samples that lie within the classifier's region of uncertainty (blind spots), and the classifier needs to use its best guess. Then, if there is disagreement between the two classifiers' output, the sample will be tested with labeled samples for confirmation.

6. Conclusions 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 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 IDSs, 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 the provision of 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 are provided. This study is the first step toward evaluating the robustness of Twitter spam detectors, as it identifies potential attacks against them. Hypothetical examples of possible attacks against Twitter spam detectors are based on common frameworks proposed in Refs. [21,32,33]. In addition, defense methods that have been commonly proposed in the literature and ways to deploy these methods in the context of Twitter spam detection are discussed.

Throughout the paper, a number of challenging issues are 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 attacks 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 [35], they cannot completely prevent attacks from happening.

In terms of a future research direction, after identifying potential attacks against Twitter spam detectors, a spam detection model that can detect different types of spam, including image spam, needs to be developed. Some of the current techniques for extracting text from images, such as Object Character Recognition (OCR), can be adopted. 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 to adversarial activities.

Author Contributions: N.H.I. devised the main conceptual ideas and worked out almost all of the technical details. Also, N.H.I. observed and extracted examples from Twitter and wrote the manuscript. V.G.V. supervised the work and provided critical feedbacks and helped sharp the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Al-Zoubi, A.; Alqatawna, J.; Faris, H. Spam profile detection in social networks based on public features. In Proceedings of the 2017 8th International Conference on Information and Communication Systems (ICICS), BIrbid, Jordan, 4–6 April 2017; p. 130.
2. Gupta, A.; Kaushal, R. Improving spam detection in Online Social Networks. In Proceedings of the 2015 International Conference on Cognitive Computing and Information Processing (CCIP), Noida, India, 3–4 March 2015; p. 1.
3. Barushka, A.; Hajek, P. Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. *Appl. Intell.* **2018**, *48*, 3538–3556. [CrossRef]
4. Sedhai, S.; Sun, A. 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, Santiago, Chile, 9–13 August 2015; pp. 223–232.
5. Stringhini, G.; Kruegel, C.; Vigna, G. Detecting Spammers on Social Networks. In Proceedings of the 26th Annual Computer Security Applications Conference, ACSAC ’10, Austin, TX, USA, 6–10 December 2010.
6. Yang, C.; Harkreader, R.; Gu, G. Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1280–1293. [CrossRef]
7. Benevenuto, F.; Magno, G.; Rodrigues, T.; Almeida, V. Detecting spammers on twitter. In Proceedings of the 2010 Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference (CEAS), Redmond, DC, USA, 13–14 July 2010; Volume 6, p. 12.
8. El-Mawass, N.; Alaboodi, S. Detecting Arabic spammers and content polluters on Twitter. In Proceedings of the 2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC), Beirut, Lebanon, 21–23 April 2016.
9. Zhu, T.; Gao, H.; Yang, Y.; Bu, K.; Chen, Y.; Downey, D.; Lee, K.; Choudhary, A.N. Beating the Artificial Chaos: Fighting OSN Spam Using Its Own Templates. *IEEE/ACM Trans. Netw.* **2016**, *24*, 3856–3869. [CrossRef]
10. Biggio, B.; Fumera, G.; Pillai, I.; Roli, F. A survey and experimental evaluation of image spam filtering techniques. *Pattern Recognit. Lett.* **2011**, *32*, 1436–1446. [CrossRef]
11. Biggio, A.; Corona, I.; Maiorca, D.; Nelson, B.; Šrndić, N.; Laskov, P.; Giacinto, G.; Roli, F. Evasion Attacks against Machine Learning at Test Time. In Proceedings of the 2013 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part III (ECMLPKDD’13), Prague, Czech Republic, 23–27 September 2013; pp. 387–402.
12. Wang, D.; Navathe, S.B.; Liu, L.; Irani, D.; Tamersoy, A.; Pu, C. Click Traffic Analysis of Short URL Spam on Twitter. In Proceedings of the 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, Austin, TX, USA, 20–23 October 2013.
13. Gupta, P.; Perdisci, R.; Ahamad, M. 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 2018), Incheon, Korea, 4 June 2018; pp. 285–296.
14. Sculley, D.; Otey, M.E.; Pohl, M.; Spitznagel, B.; Hainsworth, J.; Zhou, Y. Detecting adversarial advertisements in the wild. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD ’11, San Diego, CA, USA, 21–24 August 2011; pp. 274–282.
15. Gao, H.; Yang, Y.; Bu, K.; Chen, Y.; Downey, D.; Lee, K.; Choudhary, A. 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 2014, New Orleans, LA, USA, 8–12 December 2014; pp. 76–85.
16. Cresci, S.; Di Pietro, R.; Petrocchi, M.; Spognardi, A.; Tesconi, M. 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, Perth, Australia, 3–7 April 2017; pp. 963–972.

17. Chen, L.; Ye, Y.; Bourlai, T. Adversarial Machine Learning in Malware Detection: Arms Race between Evasion Attack and Defense. In Proceedings of the 2017 European Intelligence and Security Informatics Conference (EISIC), Athens, Greece, 11–13 September 2017; pp. 99–106.
18. Meda, C.; Ragusa, E.; Gianoglio, C.; Zunino, R.; Ottaviano, A.; Scillia, E.; Surlinelli, R. Spam detection of Twitter traffic: A framework based on random forests and non-uniform feature sampling. In Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), San Francisco, CA, USA, 18–21 August 2016; pp. 811–817.
19. Washha, M.; Qaroush, A.; Sedes, F. Leveraging time for spammers detection on Twitter. In Proceedings of the 8th International Conference on Management of Digital EcoSystems, ACM 2016, Biarritz, France, 1–4 November 2016; pp. 109–116.
20. Alabdulmohsin, I.M.; Gao, X.; Zhang, X. Adding Robustness to Support Vector Machines Against Adversarial Reverse Engineering. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, Shanghai, China, 3–7 November 2014; pp. 231–240.
21. Biggio, B.; Fumera, G.; Roli, F. Security Evaluation of Pattern Classifiers under Attack. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 984–996. [CrossRef]
22. Sethi, T.S.; Kantardzic, M. Handling adversarial concept drift in streaming data. *Expert Syst. Appl.* **2018**, *97*, 18–40. [CrossRef]
23. Sethi, T.S.; Kantardzic, M.; Lyu, L.; Chen, J. A Dynamic-Adversarial Mining Approach to the Security of Machine Learning. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1245. [CrossRef]
24. Kaur, P.; Singhal, A.; Kaur, J. Spam detection on Twitter: A survey. In Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 16–18 March 2016; pp. 2570–2573.
25. Lalitha, L.A.; Hulipalled, V.R.; Venugopal, K.R. Spamming the mainstream: A survey on trending Twitter spam detection techniques. In Proceedings of the 2017 International Conference on Smart Technologies for Smart Nation (SmartTechCon), Bangalore, India, 17–19 August 2017; pp. 444–448.
26. Lin, G.; Sun, N.; Nepal, S.; Zhang, J.; Xiang, Y.; Hassan, H. Statistical Twitter Spam Detection Demystified: Performance, Stability and Scalability. *IEEE Access* **2017**, *5*, 11142–11154. [CrossRef]
27. Wu, T.; Liu, S.; Zhang, J.; Xiang, Y. Twitter Spam Detection Based on Deep Learning. In Proceedings of the 2017 Australasian Computer Science Week Multiconference, ACSW '17, Geelong, Australia, 30 January–3 February 2017.
28. Grier, C.; Thomas, K.; Paxson, V.; Zhang, M. @ spam: The underground on 140 characters or less. In Proceedings of the 2010 17th ACM Conference on Computer and Communications Security, Chicago, IL, USA, 4–8 October 2010; pp. 27–37.
29. Chen, C.; Zhang, J.; Xiang, Y.; Zhou, W. Asymmetric self-learning for tackling Twitter Spam Drift. In Proceedings of the 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Hong Kong, China, 26 April–1 May 2015; pp. 208–213.
30. Al Twairesh, N.; Al Tuwaijri, M.; Al Moammar, A.; Al Humoud, S. Arabic Spam Detection in Twitter. In Proceedings of the 2016 2nd Workshop on Arabic Corpora and Processing Tools on Social Media, Portorož, Slovenia, 23–28 May 2016.
31. Song, J.; Lee, S.; Kim, J. CrowdTarget: Target-based Detection of Crowdturfing in Online Social Networks. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security—CCS '15, Denver, CO, USA, 12–16 October 2015; pp. 793–804.
32. Barreno, M.; Nelson, B.; Sears, R.; Joseph, A.D.; Tygar, J.D. Can Machine Learning Be Secure? In Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, ASIACCS '06, Taipei, Taiwan, 21–24 March 2006; pp. 16–25.
33. Barreno, M.; Nelson, B.; Joseph, A.D.; Tygar, J.D. The security of machine learning. *Mach. Learn.* **2010**, *81*, 121–148. [CrossRef]
34. Huang, L.; Joseph, A.D.; Nelson, B.; Rubinstein, B.I.; Tygar, J.D. Adversarial Machine Learning. In Proceedings of the 2011 4th ACM Workshop on Security and Artificial Intelligence, AISec '11, Chicago, IL, USA, 21 October 2011; pp. 43–58.
35. Sethi, T.S.; Kantardzic, M. Data driven exploratory attacks on black box classifiers in adversarial domains. *Neurocomputing* **2018**, *289*, 129–143. [CrossRef]

36. Wang, G.; Wang, T.; Zheng, H.; Zhao, B.Y. Man vs. Machine: Practical Adversarial Detection of Malicious Crowdsourcing Workers. In Proceedings of the 2014 23rd USENIX Security Symposium, San Diego, CA, USA, 20–22 August 2014; pp. 239–254.
37. Nilizadeh, S.; Labrèche, F.; Sedighian, A.; Zand, A.; Fernandez, J.; Kruegel, C.; Stringhini, G.; Vigna, G. POISED: Spotting Twitter Spam Off the Beaten Paths. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, ACM 2017, Dallas, TX, USA, 30 October–3 November 2017; pp. 1159–1174.
38. Chen, C.; Zhang, J.; Chen, X.; Xiang, Y.; Zhou, W. 6 million spam tweets: A large ground truth for timely Twitter spam detection. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015; pp. 7065–7070.
39. Mateen, M.; Iqbal, M.A.; Aleem, M.; Islam, M.A. A hybrid approach for spam detection for Twitter. In Proceedings of the 2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, 10–14 January 2017; pp. 466–471.
40. Sedhai, S.; Sun, A. Semi-Supervised Spam Detection in Twitter Stream. *IEEE Trans. Comput. Soc. Syst.* **2018**, *5*, 169–175. [CrossRef]
41. Gupta, S.; Khattar, A.; Gogia, A.; Kumaraguru, P.; Chakraborty, T. 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, Lyon, France, 23–27 April 2018; pp. 529–538.
42. Cresci, S.; Di Pietro, R.; Petrocchi, M.; Spognardi, A.; Tesconi, M. Social Fingerprinting: Detection of Spambot Groups Through DNA-Inspired Behavioral Modeling. *IEEE Trans. Dependable Secur. Comput.* **2018**, *15*, 561–576. [CrossRef]
43. Biggio, B.; Roli, F. Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning. *arXiv* **2017**, arXiv:1712.03141.
44. Buckman, J.; Roy, A.; Raffel, C.; Goodfellow, I. Thermometer Encoding: One Hot Way to Resist Adversarial Examples. In Proceedings of the 2018 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
45. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
46. Jagielski, M.; Oprea, A.; Biggio, B.; Liu, C.; Nita-Rotaru, C.; Li, B. Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. In Proceedings of the 2018 IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 20–24 May 2018.
47. Biggio, B. Support Vector Machines Under Adversarial Label Noise. *JMLR Workshop Conf. Proc.* **2011**, *20*, 97–112.
48. Laishram, R.; Phoha, V.V. Curie: A method for protecting SVM Classifier from Poisoning Attack. *arXiv* **2016**, arXiv:1606.01584.
49. Newsome, J.; Karp, B.; Song, D. Paragraph: Thwarting Signature Learning by Training Maliciously. In *Recent Advances in Intrusion Detection*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; pp. 81–105.
50. Lowd, D.; Meek, C. Good Word Attacks on Statistical Spam Filters. In Proceedings of the CEAS 2005, Conference on Email and Anti-Spam, Stanford, CA, USA, 21–22 July 2005.
51. Baracaldo, N.; Chen, B. Mitigating Poisoning Attacks on Machine Learning Models: A Data Provenance Based Approach. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (AISec '17), Dallas, TX, USA, 3 November 2017.
52. Biggio, B.; Corona, I.; Fumera, G.; Giacinto, G.; Roli, F. Bagging Classifiers for Fighting Poisoning Attacks in Adversarial Classification Tasks. In *International Workshop on Multiple Classifier Systems*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; pp. 350–359.
53. Chan, P.P.; He, Z.M.; Li, H.; Hsu, C.C. Data sanitization against adversarial label contamination based on data complexity. *Int. J. Mach. Learn. Cybern.* **2018**, *9*, 1039–1052. [CrossRef]
54. He, Z.; Su, J.; Hu, M.; Wen, G.; Xu, S.; Zhang, F. Robust support vector machines against evasion attacks by random generated malicious samples. In Proceedings of the 2017 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), Ningbo, China, 9–12 July 2017; pp. 243–247.

55. Biggio, B.; Fumera, G.; Roli, F. Adversarial Pattern Classification using Multiple Classifiers and Randomisation. In Proceedings of the 12th Joint IAPR International Workshop on Structural and Syntactic Pattern Recognition (SSPR 2008), Orlando, FL, USA, 4–6 December 2008; Volume 5342, pp. 500–509.
56. Bruckner, M. Static Prediction Games for Adversarial Learning Problems. *J. Mach. Learn. Res. JMLR* **2012**, *13*, 2617–2654.
57. Corona, I.; Giacinto, G.; Roli, F. Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues. *Inf. Sci.* **2013**, *239*, 201–225. [CrossRef]
58. Dalvi, N.; Domingos, P.; Sanghai, S.; Verma, D. Adversarial Classification. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, Seattle, WA, USA, 22–25 August 2004; pp. 99–108.
59. Kantchelian, A.; Afroz, S.; Huang, L.; Islam, A.C.; Miller, B.; Tschantz, M.C.; Greenstadt, R.; Joseph, A.D.; Tygar, J.D. Approaches to adversarial drift. In Proceedings of the 2013 ACM Workshop on Artificial Intelligence and Security—AISec '13, Berlin, Germany, 4 November 2013; pp. 99–110.
60. Miller, B.; Kantchelian, A.; Afroz, S.; Bachwani, R.; Dauber, E.; Huang, L.; Tschantz, M.C.; Joseph, A.D.; Tygar, J.D. Adversarial Active Learning. In Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop—AISec '14, Scottsdale, AZ, USA, 7 November 2014; pp. 3–14.