LEARNING ALGORITHMS FOR DETECTING DISINFORMATION ON SOCIAL MEDIA

By

Courtland VanDam

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science—Doctor of Philosophy

2019

ABSTRACT

LEARNING ALGORITHMS FOR DETECTING DISINFORMATION ON SOCIAL MEDIA

By

Courtland VanDam

Social media has become a widely accessible medium for users to share their opinions and details of their personal lives, including first hand accounts of emerging/disaster events, to a wide audience. However malicious entities may abuse users' trust to disseminate disinformation, i.e. false and misleading information. The disinformation disseminated on social media can have a significant impact offline. For example, fake news is suspected to have influenced the 2016 U.S. political election. Rumors on social media can mislead criminal investigations, e.g. the investigation of the 2013 Boston Bombing. To mitigate such impacts, automated detection of social media disinformation is thus an important research problem.

This dissertation proposes algorithms to detect two approaches hackers use to disseminate disinformation — hashtag hijacking and compromising accounts. Hashtags are terms added to social media posts that are used to provide context to the posts, so those seeking to learn more about a given topic or event can search for posts containing related hashtags. However critics and attention-seeking trolls can mislead the public via hashtag hijacking. Hashtag hijacking occurs when one group of users takes control of a hashtag by using it in a different context than what was intended upon creation. Anyone can participate in hashtag hijacking, but to be successful, a coordinated effort among several accounts posting that hashtag is needed. This dissertation proposes HASHTECT, an unsupervised learning framework that uses a multi-modal nonnegative matrix factorization method for detecting hijacked hashtags. Experimental results on a large-scale Twitter data showed that HASHTECT is capable of

detecting more hijacked hashtags than previously proposed algorithms.

Another approach for disseminating disinformation is by compromising users' accounts. A social media account is compromised when it is accessed by a third party, i.e. hacker, without the genuine user's knowledge. Compromised accounts are damaging to the account holder as well as the accounts audience, e.g. followers. Hackers can damage the user's reputation, e.g. by posting hateful rhetoric. They also disseminate misleading information including rumors and malicious websites, e.g. phishing or malware. In this dissertation, I propose two compromised account detection algorithms, CADET and CAUTE. CADET is an unsupervised multi-view learning framework that employs nonlinear autoencoders to learn the feature embedding from multiple views. The rationale behind this approach is that an anomalous behavior observed in one view, e.g. abnormal time of day, may not indicate a compromised account. By aggregating the data from multiple views, CADET projects the features from all the views into a common lower-rank feature representation and detects compromised accounts in the shared subspace. On the other hand, CAUTE focuses on detecting compromised accounts early, by detecting the compromised posts. Given a user-post pair, CAUTE is a deep learning framework which simultaneously learns the encodings for the user as well as the post to detect whether the post was compromised, i.e. was written by a different user. By training a neural network on the residuals from the post and user encodings, CAUTE can classify whether a post is compromised with higher accuracy than several existing compromised account detection algorithms.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Social media has become an incredible medium for the free exchange of ideas and knowledge. Users share details about their lives and their opinions to their friends and followers. They use social media to seek information, including news about emerging events. A 2017 Pew Research study found that 2/3 of participants receive at least some news from social media [83]. Additionally, users believe that the information on social media shared by people they know is trustworthy. The data analytics firm Nielsen found this to be the case for 83% of their survey participants [69]. Due to this inherent trust of information shared on social media, it has become a valuable tool for people with malicious intent to abuse social media and spread misleading information.

According to the Merriam-Webster dictionary, *disinformation* is false or inaccurate information that is intentionally shared to influence opinion or hide the truth [65]. Disinformation on social media can have serious consequences offline. For example, a conspiracy theory named #`pizzagate` lead to a shooter to enter a Washington D.C. pizza restaurant [3]. In 2013, hackers compromised the Twitter account of the Associate Press, a news agency, and tweeted there was an attack at the White House [30]. This led to public panic and a significant drop in the stock market. Reports have also suggested that the Russian intelligence sent malware tweets to members of US Defense Department [15]. Before the 2016 US presidential election, Bessi and Ferrara found that 20% of tweets about the election were

published by bots, suggesting that bots could change public perception [10]. While malicious players have used social media to spread disinformation for awhile, their approaches have evolved and their methods have become more advanced as technology has improved. This requires continual research to identify these emerging threats.

In the next section, I describe several approaches to spread disinformation on social media. I will then present the algorithms developed in this dissertation for detecting two of these dissemination approaches.

1.1 Taxonomy of Disinformation

Approaches used to spread disinformation on social media can be organized into three categories: individual deception, organized disinformation, and emergent disinformation. Popular approaches of each category are shown as circles in Figure 1.1. Overlapping circles indicated the similarity between approaches. For example, cyberbullies spread rumors on social media.

Based on the overlap of disinformation approaches detected in recent literature, I define the categories as follows. Individual deception occurs at the account level, an account is being deceptive about their identity, e.g. a bot posing as a genuine user. Organized disinformation on social media is similar to disinformation in other domains, the false information is carefully planned and there are intended targets [49]. Astroturf campaigns, where several accounts promote a false message during a political campaign [73], are deliberately planned to sway public opinion. Emergent disinformation occurs when there is limited information about an event or an individual and users promote false information to fill these gaps. Rumors are the most common example of emergent disinformation. The next sections further describe

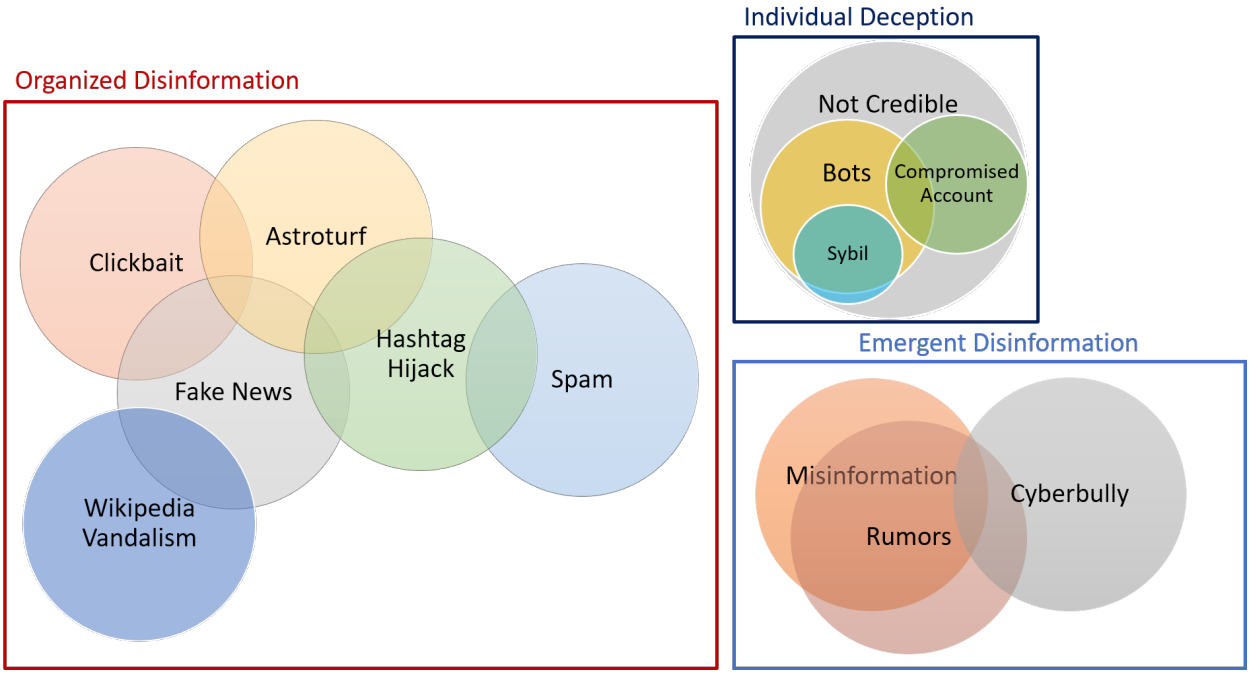


Figure 1.1: Taxonomy of disinformation approaches.

these common approaches to share disinformation.

1.1.1 Individual Deception

At the most general level, individual deception detection focuses on determining a user's credibility. Users often share first hand accounts of emerging events (e.g., active shootings, public demonstrations, and natural disasters) on social media [1, 79]. Unfortunately, not all accounts of the events are credible. Twitter helped users determine who is credible by adding the verified account indicator, which indicates whether that "account of public interest is authentic" [43]. Nevertheless, even though the accounts for most genuine users are not verified, many of their social media posts may still be credible. Identifying which users are not credible is thus an important but challenging problem. Previous research has mostly focused on detecting a subset of such users, namely, bots.

Bots are automated programs deployed to perform a specific task with relatively little

human intervention. For example, they are often used to spread spam or create fake connections to other users. Bots make it possible for users to buy followers or friends in order to make the user appear more popular or influential [90, 92]. They can also flood social media with fake posts to influence the portrayal of public opinion [10].

Nefarious people may also try to pose as genuine users. For example, they may clone an account, called a sybil, whose features, like profile image and description, would match those of the existing account. The sybil accounts then try to connect with the friends of the genuine user. If the friend requests are accepted, the sybil account can take advantage of the genuine user's reputation as being trustworthy to spread disinformation. A more intrusive way by which malicious actors may pose as a genuine user is by compromising, i.e., taking control, of a genuine user's account. This way, they do not have to depend on other genuine users accepting their friendship requests to demonstrate their credibility. From the compromised account, the hacker can post messages [28, 68, 100], follow other users [81, 84, 112], or learn private information about the user [12, 110]. This private information may allow the hacker to gain access to the user's other accounts, e.g., online banking accounts. Once a sybil account is created or a genuine user's account is compromised, hackers may utilize bots to publish fake messages on social media from these accounts.

1.1.2 Organized Disinformation

Spamming is the most commonly used approach for organized disinformation, where nefarious users would post or send messages containing links to phishing or malware websites [32, 35, 91, 96, 95]. Phishing occurs when third parties pose as a trusted party, e.g. a bank or a trusted friend, to deceive users into sharing confidential information. Malware websites directed from the social media posts would ask users to download programs that

contain viruses. While the spam posts tend to include some easily identifiable characteristics, e.g. key terms or URLs [53], spammers continue to incorporate new methods to evade detection, e.g., URL masking.

Wherever people gather, political actors will broadcast messages in order to influence public opinion. On social media, their main modes of swaying public opinion are astroturfing and hashtag hijacking. In astroturfing, a campaign will disguise itself as a "spontaneous, popular, grassroots" group of users in order to create widespread support for a candidate or an idea [76, 77]. Catchy repeatable slogans and messages are used to encourage non-campaign-related users to forward the messages to their followers. Political representatives and candidates also use hashtag hijacking to control the narrative of a topic [36, 109]. Hashtag hijacking occurs when one group of users takes control of a hashtag to use it in a different context than was intended upon creation. In addition to using hashtag hijacking to control the political narrative, hijackers use the hashtag to either degrade a brand or redirect the narrative to their interests [16].

Malicious actors outside of the political sphere influence public opinion with clickbait and fake news. Clickbait are news headlines/social media posts which promote misleading content in order to attract attention and encourage users to click the link and read the article [21]. News media outlets, especially those with strong bias, are the primary publishers of clickbait posts. Fake news, however, are non-news outlets posing as news outlets to promote inaccurate, misleading stories [85]. Fake and clickbait news were rampantly shared on Facebook during the 2016 US presidential election, which many have speculated, changed the outcome of the election [5].

Organized disinformation is not always carefully planned. For example, malicious users on Wikipedia may vandalize Wikipedia articles, by changing or removing content, to fit

their agenda. Some of these vandalisms are harmless pranks however others are organized propaganda efforts to persuade public opinion. Wikipedia vandalism is harmful because general users, i.e. non-experts, often trust the information they read on Wikipedia [56].

1.1.3 Emergent Disinformation

Emergent disinformation is primarily focused on misinformation and rumors. Although there is significant overlap between them, the main distinction is misinformation is false information, which may or may not have been debunked at the time of sharing, whereas rumors are unverified at the time of sharing and are relevant to a given situation [27]. Rumors' purpose is to make sense and manage risk when there is ambiguity or a potential threat [27]. Unfortunately rumors emerging during dangerous situations can pull resources from those investigating the current threat. For example, after the 2013 Boston Marathon Bombing, rumors propagated on reddit website falsely accused two individuals as suspects [2, 59]. Journalists trying to report on the situation were misled by these uninformed social media users. Misinformation is shared to spark conversations, socialize, and express opinions [20]. Both misinformation and rumors are difficult to debunk. Starbird et al. found that corrections to misinformation reach a smaller audience than the original misinformation [88]. Misinformation detection research often does not make a distinction between misinformation and disinformation. For example, Wu et al. stated that misinformation is used to spread fear and generate profit from public anxiety [93]. This is more prevalent in organized disinformation.

Cyberbullying is another form of emergent disinformation. Cyberbullying occurs when one or more individuals insult, threaten, or harass other users on the internet [47]. This includes starting rumors about the targeted person in order to persuade others to harass them as well. The consequences of cyberbullying are serious, both physically and emotionally,

including depression and suicide [47].

This dissertation focuses on detecting two disinformation approaches that have received limited attention—hashtag hijacking and compromised account—compared to phishing, spamming, and other approaches. An overview of these two approaches and the proposed detection methods are described in the next two sections.

1.2 Hashtag Hijacking

Hashtags emerged on social media as a way to contextualize posts and facilitate content filtering on Twitter [66]. Today, their continued popularity have encouraged social media platforms to develop search interfaces for post retrieval. A user may search for a specific hashtag, and they will receive the most popular recent posts containing that hashtag.

A hashtag is hijacked when a set of people use the hashtag in a context that differs from the original intent. In order to detect hashtag hijacking, we must consider the following questions. Can any hashtag be hijacked? Why is hashtag hijacking detection important, and what are the challenges to detect hashtag hijacking?

In order for a hashtag to be hijacked, it must have a generally accepted meaning. For example, the hashtag #`healthcare` appears in posts covering a variety of topics, from political legislation to medical technologies. This hashtag is too general, which makes it less likely to be hijacked. However #`napaquake` was used to describe a specific event, an earthquake in Napa Valley, California, in 2014. People wanting to learn more about this earthquake can search for #`napaquake`. This hashtag has a generally accepted specific meaning and has been hijacked by a certain group of users to disseminate disinformation.

To determine the importance of detecting hashtag hijacking, we need to consider the uses

of social media. Some users are seeking first-hand accounts of emergent events. For example, journalists [13, 38] and first responders [55, 106] search social media to understand emerging events in real time. Thus it is important that the information they retrieved is accurate. Hijackers, especially attention-seeking trolls [16], will often hijack a trending hashtag to spread their propaganda. For example, a terrorist organization hijacked #napaquake in 2014 to spread their message [98]. Users having negative experiences from viewing unrelated posts are likely to seek information from other avenues, rather than staying on the social media platform.

Detection of hashtag hijacking poses several challenges which require different solutions than misinformation detection or spam detection. First, the underlying meaning of a hashtag is unknown a priori. A hashtag can have any meaning, and the hijacking posts can also be on any topic. The meaning of each hashtag is not formally defined, unlike misinformation which is often debunked on third party websites, like Snopes. To detect whether a hashtag has been hijacked, it is necessary to determine its original meaning. Second, some hashtags have temporal variations. Unlike other features on social media, hashtags can have cyclical patterns of use, e.g. #tbt meaning throwback Thursday, is used primarily on Thursdays. A hashtag hijacking detector must consider whether its unusually high usage is solely due to the cyclical pattern of the hashtag. In contrast, spam generally does not follow any type of cyclical pattern, so an extra check for false alarms is not necessary. The third challenge is that a change in how a hashtag was used does not necessarily mean the hashtag was hijacked, but rather due to a change in the topic discussed using the hashtag (i.e., concept drift). For example, fans of a specific TV show often use the same hashtag to discuss that show. The topic of conversation may change each week to focus on the current episode, but the overall topic of the hashtag has not changed; it is still about that TV show.

This dissertation proposes a hashtag hijacking detection framework, HASHTECT, which addresses these challenges. HASHTECT is an unsupervised multimodal framework, which learns both the original intent of the hashtag as well as detect whether a group of users changed the topic significantly, i.e. hijacked the hashtag. HASHTECT detects cyclical patterns by considering temporal autocorrelation of the hashtag usage. To reduce false alarms via concept drift, HASHTECT measures the similarity between topics. Experimental results performed on a 72-day Twitter dataset showed that HASHTECT can effectively detect real-world hijacked hashtags.

1.3 Compromised Account Detection

User experience on social media is heavily reliant on the security of their account. Users trust that both their account and the accounts of their friends are controlled solely by the designated person. When the account is compromised, i.e. when a hacker gains control of the account and uses it for nefarious purposes, e.g. post spam or follow users, then user experience worsens [82]. As a results, many users create new accounts [111] or leave the social media platform altogether [96]. Unlike fake accounts, compromised accounts are genuine accounts which are temporarily under the control of a malicious user which exhibits anomalous behavior. Therefore typical features that are useful for detecting fake accounts may not be able to distinguish compromised accounts from non-compromised accounts [28]. Additionally, compromised accounts are not always used to post spam. A hacker can publish whatever they want, e.g. lie about the user's sexual orientation or post inappropriate content. As shown in this dissertation, most posts by the hackers are not spam. As a result, existing spam detection algorithms cannot detect most compromised accounts.

Compromised account detection poses several challenges. First, social media posts are inherently noisy with lexical variations, typos, and abbreviations. Social media posts are also sparse. For example, tweet messages tend to be short, i.e., 140 characters long, so each tweet contains a very small set of words out of the entire language. Thus it is difficult to effectively use them for training a robust compromised account detection model. Second, social media users are diverse, each having their own style and topics of interest. One post can be normal for one user and anomalous for another. For example, a hacker may lie about the user's sexuality to spread a rumor, but such posts are not unusual for some users who discuss their sexuality openly on social media. Thus compromised account detection is more challenging than spam detection, because spam posts are anomalous for most users. A trivial solution to compromised account detection would be to learn a profile for each user and use the profile to detect compromised posts. However this would not scale well to billions of users. The third challenge is that social media platforms are large, so it is difficult to train a model that can reliably detect compromised accounts for users not in the training data.

To address these challenges, I propose two compromised account detection algorithms, CADET and CAUTE. CADET is an unsupervised algorithm which learns a low-dimensional embedding of the users' posts and detects compromised accounts using reconstruction error from this embedding. CAUTE is a supervised algorithm to identify compromised posts, i.e. the hackers' posts. CAUTE simultaneously learns a tweet and user embedding, which can be used to identify whether a post was written by a different author than specified, i.e. the hacker.

1.4 Thesis Statement

Sophisticated algorithms that consider the nonlinearities and multi-view nature of social media data can be developed to effectively detect compromised accounts and hijacked hashtags. Unsupervised learning algorithms are useful when labeling the data is manually expensive while supervised learning is more effective when there are sufficient amount of labeled data available. This dissertation proposes two unsupervised learning algorithms, HASHTECT and CADET, for detecting hijacked hashtags and compromised accounts respectively. I demonstrate that these algorithms can detect anomalies more accurately than previously proposed algorithms. In addition, I also propose CAUTE, a supervised deep learning algorithm for detecting compromised posts. Whereas manual identification of a hacker's posts is a challenging task, CAUTE trains a model to identify whether a tweet belongs to their respective user after observing a small percentage of their posts. CAUTE can detect compromised posts more effectively than other compromised account detection algorithms.

Detecting hijacked hashtags and compromised accounts can be useful for intervention to improve social media user experience. A system that monitors hashtags could identify well-defined hashtags and flag for review suspicious content, ensuring that users are reading relevant content as they learn about emerging events. Social media platforms can apply the proposed compromised account detection systems to prompt the user of suspicious content to verify the user in control of the account is the genuine user. This would mitigate the potential damages of the hackers' posts.

This dissertation has four main contributions. First, I propose a novel temporal unsupervised framework, called HASHTECT, to detect hashtag hijacking from social media data. Second, I perform an in-depth analysis on known compromised accounts and hackers' posts

to understand their general characteristics. In particular, I answer the following questions: who compromises accounts, what do they post, and what features can distinguish compromised posts published by the hacker from normal posts published by the genuine user. Third, I propose CADET, an unsupervised compromised account detection algorithm. Fourth, I propose CAUTE, a supervised framework for compromised post detection.

This dissertation is organized as follows. In Chapter 2, I present related work. Chapter 3 proposes HASHTECT to detect hashtag hijacking. In Chapter 4, I provide an analysis of compromised accounts. Chapter 5 proposes CADET, an unsupervised algorithm to detect compromised accounts. Chapter 6 proposes CAUTE, a supervised learning framework to detect compromised posts. In Chapter 7, I conclude and present future work.

The materials from this dissertation are adapted from several previously published papers in peer-reviewed conference proceedings:

1. Chapter 3 is based on the paper entitled "Detection of hashtag hijacking" which was published in the Web Science 2016 proceedings [101].

2. Chapter 4 is taken from the paper, "Analysis of compromised accounts", which was published in Web Intelligence 2017 proceedings [103].

3. The materials for Chapter 5 were published in ASONAM 2018 proceedings [102].

4. Chapter 6 describes the CAUTE approach. This work has been submitted and is currently under review for IJCNN 2019.

Chapter 2

Background

This chapter presents previous research on hashtag hijacking, analysis of compromised accounts, and detection of compromised accounts.

2.1 Hashtag Hijacking Detection

Twitter is a widely used data source to solve a variety of problems including event detection, topic discovery, information diffusion, opinion mining, sentiment analysis, and spam detection. Researchers have also investigated the use of hashtags, which are an integral part of Twitter, to tag users' postings to a specific topic or to add contextual information. Previous research has focused on various aspects of hashtag analysis, including understanding their construction [74], annotating the hashtags, analyzing their sentiment [26], modeling their topics [54], predicting their popularity, and recommending hashtags.

Some research have used hashtags to analyze spamming strategies. For example, Grier et al. identified an approach spammers use to spread malicious URLs by creating tweets with a trending hashtag and embedding a hyperlink to their spam Web sites [35]. Chu et al. clustered tweets by URL to detect spam campaigns [22]. Sridharan et al. studied what makes spam strategies most successful [87]. They determined whether a spammer hijacked a hashtag based on the proportion of the spammer's tweets containing the popular hashtags. While these studies demonstrate the use of hashtag hijacking for spamming, they limit their

detection to tweets containing URLs and do not detect hashtags that are hijacked for other purposes. Hashtags can be hijacked either for malicious or benign purposes. For example, the hashtag #napaquake, which was originally intended for the 2014 Napa Valley earthquake incident, was hijacked by a small group individuals to start a new trend, the wine bucket challenge, which failed to become trending. It was also reported to have been hijacked by terrorist organizations to spread their propaganda.

Prior research on detecting whether a hashtag has been hijacked focused on specific hashtags, in particular political hashtags. Hadgu et al. and Weber followed the political tweets posted by 33 Twitter users (political leaders and parties) and their retweeters to detect incidents of a political party who hijacks the hashtags of its opponents [36, 109]. For each week, they collected all the political hashtags used by any of the 33 users. They measure the lean of the hashtag by the volume of retweets originating from these 33 users. If a hashtag has a significant shift in lean from one party to the other, then the hashtag was hijacked. The approach proposed by Hadgu et al. [36] and expanded on by Weber [109] requires that the political preference of a user be known, and thus, is limited to political hashtags. It does not generalize to other types of hashtag hijacking incidents.

Another related work was recently developed by Hayashi et al. [37] for detecting topic hijacking from Twitter data streams. Their method, however, was designed for inferring a broad set of topics from all the tweets and checking if any of them had been hijacked. Our work, on the other hand, focuses on identifying topics within each hashtag and checking if any of them were hijacked. This is more challenging since each hashtag can have a different frequency, word, and user distribution. Thus, we need to apply filtering techniques to reduce the number of candidate hashtags that need to be examined to verify whether they have been hijacked.

Approaches from other hashtag tracking research are not directly applicable to hashtag hijacking. For instance, research on meme tracking analyzes the spread of a distinct phrase or hashtag. Hashtag tracking focuses on information diffusion [89] and event detection [24] whereas in hashtag hijacking, we are looking for change in the use of a hashtag instead of the spread of information.

2.2 Compromised Account

Previous research on compromised accounts focused on two themes, understanding compromised accounts and detecting compromised accounts. Research on understanding compromised accounts focused on identifying the prevalence, how they are reported, and their impact. Compromised account detection research focused on detecting one of four types of compromised accounts, forced shares, forced likes, forced follows, and information seeking. As the names suggest, forced shares occur when the hacker publishes posts. Forced likes are when the hacker likes pages or posts. Forced follows occur when the compromised account follows or friends other accounts. During information seeking, the hacker is browsing the private profile details, e.g. birth date, or private messages of the compromised user.

2.2.1 Exploratory Analysis

Using surveys and tweet analysis, previous research has attempted to understand the prevalence of compromised accounts, who compromise accounts and what is the impact of having a compromised account. Compromised accounts tend to be prevalent. Shay et al. found 30% of surveyors indicating they experienced either their email or social media account hijacked [82]. A 2017 Pew Center study found that 13% of Americans experienced a compromised social

media account [71].

Previous research found that while most hackers are unknown to the genuine user, a significant amount of the hackers knew the genuine user in person. Zangerle and Specht found that among tweets reporting the current Twitter account was compromised, 10% also reported the hacker was a relative or a friend [111]. Usmani et al. found that 21% of survey participants had their Facebook account accessed without permission by someone they knew [100]. Additionally they found that while compromised account victims could be any age, those hackers who compromise their friends' accounts in this way tend to be younger, e.g. in their 20s.

Having a compromised social media account can have both physical and emotional impacts on users and social media platforms. Zangerle and Specht found that 27% of tweets announced creating a new account [111]. Social media accounts are valuable to users, so they tend to be attentive to their accounts. Murauer et al. analyzed conversations between a user reporting to their friend that the friend's account was compromised, and they found that the friend responds within an hour about 48% of the time, and within 24 hours 80% of the time, either to explain the situation or confirm the account was compromised [67]. Shay et al. found 82% of survey participants who experienced having a compromised accounts check their account at least once per week [82]. According to survey participants, 37% experienced spam being sent from their account and 18% experienced negative feelings as a result of having their account compromised [82].

2.2.2 Detection

Previous work on compromised account detection have focused on four approaches. For the first approach, a behavior model for each user is learned from a series of actions, e.g. clicks

Table 2.1: Summary of compromised account detection literature.

Author	Type	Supervised	Features	Approach
Stringhini et al. [92]	Follow	Yes	Temporal followers	Classifier
Mehrotra et al. [62]	Follow	Yes	Graph centrality	Classifier
Shah et al. [81]	Follow	No	Adjacency matrix	Reconstruction Error
Shen et al. [84]	Follow	Yes	User & Post	Classifier
Jiang et al. [46]	Follow	No	Temporal Followers	Distance
Zhang et al. [112]	Follow	Yes	User & Post	Classifier
Ruan et al. [78]	Seeking	No	Clickstream	Behavior Model
Bohacik et al. [12]	Seeking	Yes	Login Attributes	Behavior Model
Wu et al. [110]	Seeking	Yes	Clickstream	Classifier
Viswanath et al. [107]	Like	No	Page Likes	Reconstruction Error
Egele et al. [28, 29]	Share	Yes	Post	Behavior Model
Igawa et al. [8, 41]	Share	Yes	Post	Author Verification
Nauta [68]	Share	Yes	Post	Classifier

or posts. Future actions are compared to this model to identify anomalous behaviors. In the second approach, a user feature vector is constructed or learned, then these feature vectors are passed to a generic classifier to predict which users are compromised. Third, a user-feature matrix is constructed and then decomposed. Compromised users are those having high reconstruction error. Fourth, distance based techniques focus on how a given user's behavior changes over time, e.g. the similarity of the user behavior in two time windows. A summary of compromised account detection research is shown in Table 2.1. Noticeably, the approaches vary within each type of compromised account.

Behavior models focus on determining whether a specific behavior is normal for a given user, based on the frequency of that behavior in the user's history. Previous research built behavior profiles based on users' posts, logins, and clickstreams. Egele et al. proposed COMPA, which creates a behavior profile for each user based on several post features, e.g. time of day and mentions [28, 29]. Then withheld posts are compared to that profile to determine how anomalous they are. If a post has a high anomaly score, the account would be flagged as compromised. The weakness of this approach is what constitutes a high anomaly

score. Egele et al. focused on spam campaigns, where the messages shared are spam [28, 29]. Therefore they cluster posts by URL and set the anomaly score for each cluster. This limits the type of compromised posts which can be detected. Another approach is the global threshold approach which tests multiple thresholds and select the threshold which provides the highest performance. Bohacik et al. applied this approach to identify anomalous social media account logins [12]. Trang et al. showed that for COMPA, using a global maximum threshold leads to either large amount of false positives or false negatives and suggests that the threshold should be user dependent [97]. Therefore a more robust approach for determining the threshold would be to compare activities of each user with all other users. Igawa et al. divided each users' posts into a training, threshold, and test sets [8, 41]. Phrases, i.e. N-grams, from the training set were compared to the threshold set based on their simplified profile intersection to determine the threshold of anomalous behavior. The threshold value can also be determined from a user's self-variance and their profile distance to other users. Ruan et al. applied this approach to users' clickstreams to identify anomalous behavior [78].

Classifier-based approaches assume that the anomalous activities can be identified in order to train a classifier. This approach assumes that the anomalous behavior is similar across users. Nauta proposes detecting forced shares by classifying posts [68]. In terms of forced shares, the assumption that compromised posts will be similar to each other across users limits the types of compromised posts which can be detected, e.g. spam. This approach would not identify other types of compromised posts which are more user dependent, e.g. false proclamation regarding the user's sexual orientation. Classifiers have also been used to detect forced follows and information seeking.

Recent research has proposed using unsupervised learning to detect compromised ac-

counts by using reconstruction error. Reconstruction-based approaches transform users' representations into a lower dimensional space, then apply another transformation back into the original feature space. The assumption is that compromised or anomalous users will have a higher reconstruction error than genuine users [107]. This approach has only been applied to forced likes [107] and forced follows compromised accounts [81]. The main limitation of using reconstruction error is that it is prone to a large amount of false positives due to its unsupervised nature.

Lastly, previous research focused on within-user distance to detect compromised accounts. Jiang et al. proposed a distance-based approach to detect forced follows by looking at the changes in a user's friends, i.e. users that the current user is following [46]. Namely, if the set of friends changes significantly over a short period of time and the total number of friends is similar, then the user in question is likely participating in the follower market, e.g. earning credit to buy followers by following other users who have subscribed to this marketplace.

Chapter 3

Detection of Hashtag Hijacking

3.1 Introduction

Twitter is a popular social media platform where trends emerge with the use of hashtags. A hashtag is a user-generated label, starting with the # symbol followed by some text. Although hashtags can be used by anyone who posts a tweet, they often develop into a cohesive meaning as users tend to use the same hashtag in a similar context when discussing the same topic. A trending hashtag is a hashtag with a cohesive meaning that is increasingly being used in the tweets posted by users over a short period of time.

Hashtag hijacking occurs when a group of users start using one of these hashtags to promote a different message. Examples of hashtag hijacking include attention seeking trolls [16], Public Relations (PR) campaign gone wrong [16], and politically fueled hijacks [26, 36, 109]. Attention seeking trolls are users who hijack a hashtag to post inflammatory or random unrelated information. For example, terrorist groups have hijacked several popular hashtags to gain attention [33, 70, 98]. The tweets that contain the hijacked hashtag may include offensive messages or links to scam websites [14, 16]. Trolls typically use a trending hashtag for a short period of time before shifting their attention to another hashtag. Hashtag hijacking may also occur when a company uses a hashtag to promote its brand name and improve its image, but other users post negative comments about the company using the same

hashtag. Such PR campaigns gone wrong are often the result of using vague or self-serving hashtags [23, 31, 44]. Finally, politically fueled hashtag hijacks often involve controversial issues with strong opposing views. One political party or group creates the hashtag and uses it for their campaign. Then the opposing party uses the same hashtag and explains everything wrong with the topic of the campaign [31, 48, 36, 109].

Previous studies have focused mainly on detecting a specific type of hashtag hijacking, e.g., spamming or political hijacks, but not both. For example, Chu et al. [22] analyzed tweets that share the same destination URL to detect hashtags that were hijacked for spam campaigns. However, tweets without URLs may contribute to other type of hashtag hijacking incidents. For example, the hashtag #`whyistayed`, which was originally used to discuss domestic violence, was found to be hijacked by a company that produces frozen pizza. However, half of the unique tweets observed in our dataset from this hijacking campaign did not contain any URLs. Thus, relying on URLs alone is insufficient as one could miss other types of hijacking incidents. Similarly, some researchers have manually monitored the political orientation of users who posted tweets using specific hashtags in order to detect politically fueled hijacks [36, 109]. Unfortunately, such methods are neither generalizable nor scalable to other types of hijacks.

Unlike prior research, this dissertation investigates the feasibility of applying a general framework for detecting hashtag hijacking, where the compromised hashtags and their original intent were unknown *a priori*. Designing an automated approach for detecting hashtag hijacking is challenging due to several reasons. First, as the underlying topics of the tweets containing a hashtag are unknown, they must first be inferred from the Twitter data. This is a challenge since there are no strict guidelines on who and how to use a hashtag, allowing it to be used in different contexts by diverse groups of users. As a consequence,

previous studies have focused on a narrow set of hashtags whose semantic meanings were known to the researchers. For example, Hadgu et al. [36] and Ingmar [109] considered only political hashtags in their detection schemes. Such methods would fail to detect other, non-political hashtag hijacking incidents. Second, one must consider the temporal variability of the hashtag frequency to avoid misclassifying non-hijacked hashtags. For example, the hashtag #`neverforget` is primarily used on September 11th to support victims of a deadly terrorist attack incident but is also used on other days of the year to commemorate other notable events. A detection algorithm may incorrectly classify such hashtags that gain sudden popularity on a particular day as being hijacked if it does not consider how the hashtag frequency evolves over time. Third, detecting changes in the topics of a hashtag alone is insufficient since not all hashtags whose underlying topics have changed at a given point in time are the result of hijacking. For example, Figure 3.1 shows the distribution of the top two topics for #`ferguson`, which is the hashtag used for tweets related to a police shooting incident in Ferguson, Missouri. Although there appears to be significant shift in topics on day 27, it was not due to hashtag hijacking. Instead, the topics changed as the focus of the tweets moved from Michael Brown, the shooting victim, to Darren Wilson, the policeman (see Figure 3.1).

To address these challenges, we propose a novel framework that combines information about the temporal distribution of hashtag frequencies along with the content of their tweets and the users who posted the tweets to determine whether a hashtag has been hijacked. Specifically, our framework decomposes the detection task into three subproblems. The first subproblem is to identify candidate hashtags that have experienced a significant change in the temporal evolution of their underlying topics. The second subproblem is to determine which of its underlying topics was the hijacked topic. The third subproblem is to determine when

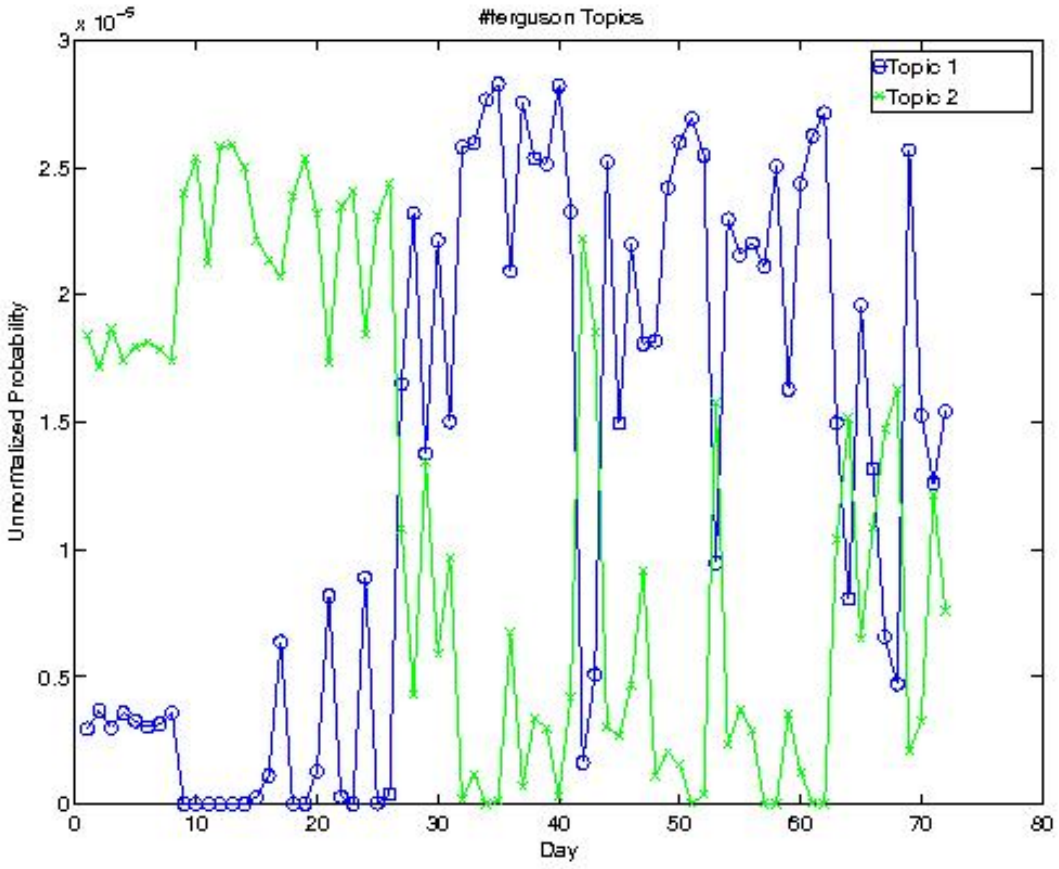


Figure 3.1: Topic distribution over time for #`ferguson`.

a hashtag was hijacked. We identify candidates for hijacking by employing a multimodal non-negative matrix factorization approach to learn the underlying topics of each hashtag. We then analyze the evolution of each topic over time. Using Hotelling's t^2 test, if a topic changes significantly at any point in time, we consider the hashtag as a candidate that must be further examined to determine whether it has been hijacked. The candidate hashtags are further examined through the content distribution of the tweets as well as the distribution of users who posted them to verify whether they were actually hijacked. We applied the framework to 72 days of Twitter data and performed experiments to evaluate its effectiveness. Experimental results showed that our framework can effectively detect hashtag hijacks with

an AUC value as high as 0.669.

The remainder of the chapter is organized as follows. In Section 3.2 introduces the hashtag hijacking problem and describes its challenges. The proposed framework is explained in detail in Section 3.3. I demonstrate the effectiveness of our framework in Section 3.4. In Section 3.5, I conclude and present some directions for future work.

3.2 Hashtag Hijacking

Hashtag hijacking occurs when a hashtag is used to promote tweets that are unrelated to its original intent. The promoted intent may be a separate topic or an opposing perspective. For example, Veilleux-Lepage [104] described incidents of hashtag hijacking involving Islamic State sympathizers who knowingly infiltrate into Twitter conversations involving hashtags such as #`Brazil2014` and #`WC2014` related to World Cup soccer to increase exposure of their messages. Hadgu et al. [36] and Weber [109] detected that members of one political party hijack the hashtags of the opposing political party. As noted in the introduction, hashtag hijacking can be used as a tool that serves various purposes, including trolling, spamming, self-promoting, for humor, etc.

This section formally defines hashtag hijacking. Let Γ be a collection of tweets, $\mathcal{U}$ be the set of all users, $\mathcal{W}$ be the set of all terms, and $\mathcal{H}$ be the set of all hashtags. Given a tweet, $t \in \Gamma$, let $t.u \in \mathcal{U}$ denote the user who posted the tweet, $t.w \subset \mathcal{W}$ be the set of terms contained in the tweet, and $t.h \subset \mathcal{H}$ be the set of hashtags contained in the tweet. For this research, we focus only on tweets that contain hashtags, i.e., $\{t \in \Gamma \big| \ |t.h| > 0\}$, where $|x|$ denote the cardinality of the set x. Every tweet has a creation time. For this research, we focus on the day granularity. We denote the creation date of the tweet t as

t.date. For each hashtag $\nu \in \mathcal{H}$, we created a data set by aggregating all the tweets and users that use the same hashtag, i.e., $\mathcal{D}_\nu =< \nu, T_\nu, U_\nu >$, where $T_\nu = \{t \in \Gamma \big| \nu \in t.h\}$ and $U_\nu = \{u \in \mathcal{U} \big| u \in t.u, \forall t \in T_\nu\}$. Note that $|T_\nu| \geq |U_\nu|$ since the same user may produce multiple tweets containing the same hashtag.

Let $f : \mathcal{H} \rightarrow \{-1, +1\}$ be a target function that maps each hashtag to a binary label, where $f(h) = 1$ if the hashtag h is hijacked and -1 otherwise. The goal of hashtag hijacking detection is to infer the target function based on the colletion of hashtag data sets available, $\{\mathcal{D}_h\}$. In this dissertation, we focus on unsupervised learning of the target function, which assumes there are no labeled examples available. Thus, we will define a set of heuristic functions to discriminate the hijacked hashtags from the non-hijacked ones.

A hashtag is considered to be hijacked when there is a significant change in the topic of the hashtag due to the tweets posted by a small group of users whose messages are significantly different from the original intent. To do this, we need to recognize what is the original intent of a hashtag and what constitutes a significant change in topic in order to detect hashtag hijacking.

The original intent of a hashtag is the topic or topics which describes the core meaning of the hashtag and captures the variability of the topic. Formally a hashtag can be described by one or more topics, denoted by τ. For example, users discussed domestic abuse using the hashtag #`whyistayed`. This hashtag has been used in tweets that discuss a variety of related topics from specific incidents to celebrities who have experienced domestic violence. If one of the topics is hijacked, it will appear significantly different from the original intent topics. Section 3.3.2.1 describes how we measure similarity between topics and at what threshold the similarity indicates significant difference.

The hashtag hijacking detection framework proposed in this study aims to resolve the

following three sub-problems: (1) detect which hastags are hijacked, (2) detect which topic describes the hijacked concept, and (3) detect when the topic is hijacked. To detect which hashtags are hijacked, this dissertation presents an approach based on multimodal non-negative matrix factorization (NMF) to combine content data with user data combined with statistical tests. Which topic is hijacked is detected based on inter-topic similarity. Detecting when the topic is hijacked results from applying a statistic to the hijacked topic. Details of our methodology is presented in the next section.

3.3 Proposed Framework

Figure 3.2 presents a high-level overview of the proposed framework. The framework consists of the following 2 main stages: detection of hashtags that are candidates for hijacking and verification that the hashtags were indeed hijacked. Hashtags that are trending and experience a significant change in topic are considered potential candidates for hijacking. Thus, we first perform trend detection to identify the trending hashtags. To detect hijack candidates, we learn the topics and detect whether the topics changed more than expected, indicating an anomaly. Once we have identified the candidates, they need to be verified to determine whether they were actually hijacked. Each stage is described in further detail in the remainder of this section.

3.3.1 Identifying Hijacked Hashtag Candidates

The first step towards detecting hashtag hijacking is to identify the viable candidates. This helps to narrow down the list of hashtags to be verified. Since hijackers favor trending hashtags, we must first identify which hashtags are trending. We then learn what topics

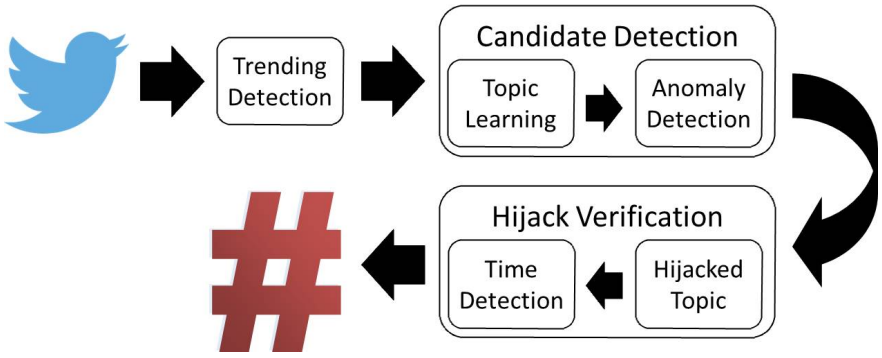


Figure 3.2: A schematic illustration of the proposed framework.

are discussed with those hashtags. Finally, we need to check whether a significant change in topic has occurred.

3.3.1.1 Detection of Trending Hashtags

In this study we focus on detecting candidates that are trending hashtags[1]. Trending hashtags are a valuable target for hijacking because they reach a large audience quickly, making it easy to share their message. A hashtag is trending if it has two properties; high volume of usage presently (popular) and previously was not popular (novelty) [80, 113]. Although Twitter has a Trending API, it only provides the current trending terms and hashtags. Since we collect Twitter data using their Streaming API, to detect hashtag hijacking of trending hashtags from previously collected data, we first need to detect whether each hashtag is trending.

We measure the popularity of a hashtag based on the number of tweets in a day that contain the hashtag. Let the popularity of a hashtag h on day d be the number of tweets containing the hashtag posted on that day, $p_{hd} = |\{t \in \Gamma : t.h = h, t.\text{date} = d\}|$. A hashtag is popular if its popularity is above some threshold, $p_{hd} > k$. We analyzed known trending hashtags in our dataset, and found that all trending hashtags appear in at least 100 tweets

[1]Extending our detection approach to non-trending hashtags will be the subject of future work.

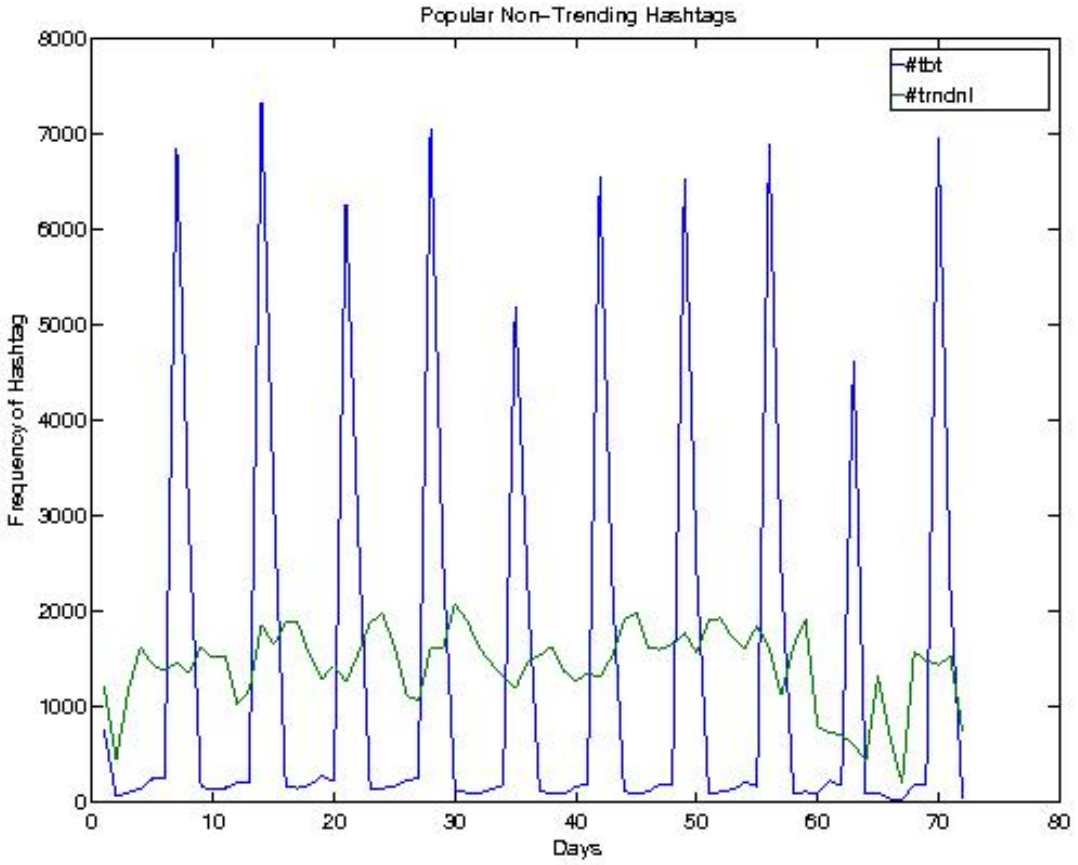


Figure 3.3: Number of tweets per day collected for two popular hashtags, #`tbt` and #`trndnl`, which are not trending because they lack novelty.

on the day they were most popular. Therefore we set the threshold to 100.

Popularity alone cannot define a trending hashtag. #`jobs` and #`healthcare` are among the most popular hashtags, but they lack novelty. A hashtag is novel if it is acyclical and exhibits anomalous behavior. Cyclical hashtags such as #`tbt` (Throwback Thursday) follow predictable behavior, making it not novel. Similarly, when the discussion of a hashtag is uniformly distributed over time, the hashtag is considered not novel even though it could be popular. Figure 3.3 shows the daily frequency distribution of two popular but non-trending hashtags. For trending hashtags, the number of tweets that include the hashtag will experience a sudden increase that is anomalous compared to the previous popularity.

The novelty of a hashtag is determined by testing for anomalies and cycles. To test for anomalies, we fit the daily popularity to a normal distribution and test the goodness of fit, with a significance level of 0.01. If the popularity of the hashtag fits a normal distribution, then it is removed from the candidate list. To ensure we observe a hashtag long enough to determine its distribution, we remove any hashtag that is observed on fewer than 7 days. We also test for the presence of temporal autocorrelation to determine whether the popularity of a hashtag follows a cyclical pattern. We compute the cross-correlation of the hashtag popularity and check whether the distance between the lags where the highest correlation is observed occur at a fixed interval. If the same interval length occurs at least half the time, then the hashtag is considered cyclical and discarded from the candidate set. Typically cyclical hashtags are observed weekly, so the interval length is 7 days.

3.3.1.2 Topic Learning

After identifying the trending hashtags, our next step is to learn its underlying topics. For this, we use multi-modal non-negative matrix factorization (NMF), in which the latent topics are inferred jointly from the user and term frequency matrices.

3.3.1.2.1 Feature Selection Our framework uses the Twitter Streaming API to retrieve tweets. For each tweet, we gather its content, user ID, and the tweet creation date. Since we are interested in detecting hijacked hashtags, we filter out tweets that do not contain any hashtags.

Since the vocabulary of terms used in Twitter messages is potentially large, we apply the Natural Language Toolkit (NLTK)[2] part of speech tagger to select only nouns. We chose

[2] http://www.nltk.org/

the NLTK part of speech tagger due to its simplicity even though part of speech taggers for Twitter do exist, such as the CMU ARK Twitter Part-of-Speech Tagger[3]. We will explore the performance of other part of speech taggers for feature selection in future work. Nouns were selected because within the discussion of a topic the most prominent information are the entities involved.

We use the tweet data $\mathcal{D}_h$ to generate two matrices: a term frequency per day matrix, $\mathbf{X}$, and a user frequency per day matrix, $\mathcal{U}$, for each hashtag h. To reduce variability due to differences in the frequency of tweets for each day, both matrices are normalized by the frequency of tweets (or users) who used the hashtag for the day. The normalized matrices are used to learn the topic distributions of the tweets associated with the given hashtag and their time evolution.

3.3.1.2.2 Topic Learning We assume each topic has a cohesive meaning, which can be represented by a set of influential terms describing the topic. We also assume that a user will use the same hashtag to discuss the same topic. To determine if a hashtag has changed its topics, we apply multi-modal non-negative matrix factorization (NMF) to jointly decompose our term frequency and user frequency matrices into 3 latent factors: a terms by topic matrix, W, a user by topic matrix, V, and a day by topic matrix, H. All of these matrices and their dimensions are defined in Table 3.1. Non-negative matrix factorization ensures that none of the latent matrices will have negative weights in their entries.

The multi-modal non-negative matrix factorization approach employed in this study was

[3]http://www.cs.cmu.edu/ ark/TweetNLP/#pos

Table 3.1: List of variables defined for each hashtag. Note that m refers to number of terms, n is number of users that uses the hashtag during the d days, and k is the number of topics.

Variable	Definition	Dimensions
X	Terms used each day	m x d
U	Users posting each day	n x d
W	Term scores for each topic	m x k
V	Users' scores for each topic	n x k
H	Days' score for each topic	d x k

designed to optimize the following objective function:

$$\begin{aligned} &||X - WH^\top||_F^2 + \alpha||U - VH^\top||_F^2 \\ &\text{s.t. } W_{ij} \geq 0,\ V_{kj} \geq 0,\ H_{hj} \geq 0\ \forall h, i, j, k \end{aligned} \tag{3.1}$$

Since W, V, and H are unknown, this function cannot be solved in closed form. We learn these matrices by iterating over the data , fixing at least one of the matrices and learning the other matrices then fixing the learned matrix and relearning the previously fixed matrices. To learn H, we fix V and W. We then fix H and learn W and V simultaneously. We repeat this process until convergence.

Extending the multiplicative update rules proposed by Lee and Seung [52] to multi-modal NMF, we obtain the following update functions:

$$W_{ij} = W_{ij}\frac{(XH)_{ij}}{(WH^\top H)_{ij}} \tag{3.2}$$

$$V_{kj} = V_{kj}\frac{(UH)_{kj}}{(VH^\top H)_{kj}} \tag{3.3}$$

$$H_{hj} = H_{hj}\frac{(X^\top W + \alpha U^\top V)_{hj}}{(HW^\top W + \alpha HV^\top V)_{hj}} \tag{3.4}$$

We repeat the updates until convergence. Due to the time complexity of estimating

the norm of a matrix, we compute the value of the objective function $\Delta^{(t)}$, once every 100 iterations. In our experimentation, we burn-in for 500 iterations, then check if the difference of the objective function between the times (t) and $(t-1)$ is less than 0.1% of $\Delta^{(t-1)}$. The burn-in ensures the matrix factorization reaches a stable solution. The algorithm can also be easily extended to an online learning setting, using the stochastic gradient descent method [37] or the online passive aggressive algorithm [11]. However, details of the approaches are beyond the scope of this dissertation.

3.3.1.3 Detection of Hijacked Hashtag Candidates

The unnormalized probability of each topic often varies over time. To detect whether the variability is due to hijacking rather than noise, a change detection algorithm is needed.

One of the most popular unsupervised change detection techniques is based on Hotelling's T^2 Statistic [39, 50]. The Hotelling's T^2 statistic follows an F distribution, so we can compute the p-value of the statistic to determine whether a significant change has occurred. Using a sliding window around each day, we test if the topic distribution within the sliding window days are from the same distribution as the topic distribution for the remaining days outside the window. To illustrate this, let $X = \{x_1, x_2, \ldots x_{n_1} | x_i \in \mathbb{R}^k\}$ be the weights of each topic for days within the test window and $Y = \{y_1, y_2, \ldots y_{n_2} | y_i \in \mathbb{R}^k\}$ be the corresponding topic weights for days outside the window. The Hotelling's T^2 statistic is then calculated as:

$$\begin{aligned} T^2 = & \frac{n_1 n_2 (n_1 + n_2 - k - 1)}{k(n_1 + n_2 - 2)(n_1 + n_2)} \\ & \times (\mu_x - \mu_y)^\top \Sigma^{-1} (\mu_x - \mu_y) \end{aligned} \tag{3.5}$$

Where μ_x is the mean of the within-window days, μ_y is the mean of the out of window days,

and Σ is the unbiased pooled covariance matrix.

$$\Sigma = \frac{\sum_{j=1}^{n_1}(x_j - \bar{x})(x_j - \bar{x})^\top + \sum_{j=1}^{n_2}(y_j - \bar{y})(y_j - \bar{y})^\top}{n_1 + n_2 - 2} \tag{3.6}$$

Under the null hypothesis that x and y are drawn from the same distribution with the same mean and covariance, $T^2 \sim F_{k,n_1+n_2-k-1}$. The change is significant if the p-value associated with the T^2 statistic exceeds some significance level, α. Hashtags that fail this statistical test are considered candidates for being hijacked.

3.3.2 Validating Hijacked Hashtags

Validating whether a candidate is hijacked requires two steps: (1) determining which topic is hijacked and (2) detecting when the hijack occurred.

3.3.2.1 Detecting Hijacked Topic

After topic detection, we need to determine which topic describes the original intent of the hashtag and which one is the hijacked topic. Inferring the original intent of a hashtag is indeed a challenging problem.

We explore two approaches to identify the hijacked topic; one based on similarity while the other based on Hotelling's T^2 statistic. The similarity-based approach computes the Jaccard similarity between all pairs of topics based on their top z terms or users. We hypothesize that the hijacked topic is the topic least similar to all other topics. To measure which topic is least similar, each topic is assigned a similarity score, which is the sum of its corresponding row in the similarity matrix. Based on our hypothesis, the topic with the lowest similarity score is likely to be the hijacked topic.

Our second approach is to identify the topic with the highest unnormalized probability in the H matrix, at the time the hijack was detected by the Hotelling's T^2 statistic. When a multiple day window is used to calculate the Hotelling's statistic, the hijacked topic is based on the middle day, e.g., day 2 of a 3-day window.

3.3.2.2 Detection of Timing of Hijack

In addition to detecting whether a hashtag is hijacked and identifying which topic is hijacked, we need to determine when the hashtag was hijacked. A hashtag is hijacked when the Hotelling's statistic, described in Section 3.3.1.3, is at its peak.

3.4 Experimental Results

In this section, we evaluate the performance of the proposed framework on 72 consecutive days of Twitter data.

3.4.1 Comparative Method

We compared the performance of our approach against the approach proposed by Hayashi et al. [37]. Note that the approach described in [37] was originally designed to detect hijacked topics in Twitter rather than hijacked hashtags. Specifically, they proposed a log-likelihood ratio approach to determine whether the unnormalized probabilities of terms and users per topic followed a p-step distribution or a power law distribution. They hypothesized that if a topic is hijacked, then it will more likely follow a p-step distribution. We compared the performance of this baseline approach against our approach in two ways. First, we use this approach to predict whether a hashtag is hijacked by checking whether the log-

Table 3.2: Tweets used to hijack hashtags.

Injected Tweets
islamic state executes 250 syrian soldiers #icebucketchallenge
@jjauthor islamic state's declaration of war on the usa w/beheadings - obama isn't one but #americans know we are #infedels & we are at #war
@freedomcrusades @markknoller isis=islamic state of iraq and syria. i like #isil
this third letter to you from the islamic state beware of the fourth #fireintheboothcypher #lastnightoftheproms http://t.co/efuz6mec9g
islamic state terrorist: we will make some attacks in new york soon
#msnbc #nbc #abc #cbs #cnn #is 4 islamic state #isil 4 islamic state in the #levant stop yellow journalism and fear mongering! u got ur war!
abu waheed and mudjahideens nasheed islamic state: http://t.co/ukvy2dponu @ecayuno #letters4nasheed #bayloc #lalaaz-iz #cripto #aceleezy
islamic state releases this photo of canadian shooter michael zehaf-bibeau http://t.co/vkwau5obts #ottawashooting

likelihood ratio for any topic exceeds a given threshold. Second, we select the topic with the largest log-likelihood ratio as the hijacked topic. Note that in both experiments, we apply log-likelihood ratio to each topic of the user-topic and term-topic matrices, V and W respectively, generated by the multimodal NMF approach. For the remainder of this chapter, we refer to this baseline approach as Log Ratio.

3.4.2 Data Collection, Preprocessing, and Trend Detection

We collected tweets from Twitter's Streaming API using the Python library, Tweepy[4] for the period between August 22, 2014 and November 1, 2014. From this data, we selected all tweets that have at least one hashtag. We observed 4,533,702 hashtags. We evaluated which

[4]http://www.tweepy.org/

hashtags are trending using the methodology explained in Section 3.3.1.1. We found 2667 trending hashtags, which were used in tweets by 766,057 unique users. In total, we found 98,234 unique nouns that co-occur with our popular hashtags.

3.4.3 Ground Truth

In order to detect hijacked hashtags, we needed to determine whether our data contained hijacked hashtags. In our research, we found reports indicating some hashtags, such as #napaquake and #askricky, had been hijacked by the terrorist organization, the Islamic State (IS) [98]. Unfortunately, such tweets were not present in our Twitter data set. We injected synthetic IS tweets to hijack the hashtags. We tested how well our algorithm performed when we injected into a randomly selected hashtag 5% of the total number of tweets observed with that hashtag. This injection percentage is similar to the amount used in Hayashi et al [37]. We injected 8 synthetic tweets, listed in Table 3.2, into 100 randomly selected trending hashtags. These hashtags are listed in Table 3.3. The injected tweets were actual tweets found in our dataset that we believe were written by Islamic State supporters. We chose to search for "Islamic State" instead of ISIS based on the findings by Magdy et al. [60] who discovered that in Arabic tweets, supporters of the Islamic State tend to use the full name of the organization more often than its abbreviation.

To determine the hijacked topic, we ranked the terms by their unnormalized probabilities for each topic. The hijacked topic is the topic with "islam" and "state" with the highest rank. When we injected 5% of the total tweets with hijacked tweets, we found "islam" and "state" within the top 50 terms for 96% hashtags and within the top 100 terms for all of the hashtags.

The hijacked tweets are synthetically injected such that the majority of the tweets oc-

Table 3.3: Examples of hashtags that were hijacked.

#breakingbad	#climatemarch	#dragoncon	#ebola
#fixthepolice	#icebucketchallenge	#napaquake	#obamaspeech
#tiff14	#russiainvadedukraine	#wasvsdal	#whyistayed

curred on the day after the hashtag was trending. Then the hijackers' use of the hashtag would taper off over the following two days. The first day the hashtag was hijacked, 70% of the hijacked tweets were injected. On the second and third day, 20% and 10% of the hijacked tweets were injected respectively. According to previous reports on hashtag hijacking, hijackers tend to hijack a hashtag while it is still trending [98].

3.4.4 Findings

One of the goals of this research is to detect hashtag hijacking with as few topics as necessary to reduce memory. We explore how our algorithm performs when we decompose our term and user matrices into 3, 4, and 5 topics.

3.4.4.1 Detecting Hijacked Hashtags

The first challenge is to detect which hashtags are candidates for being hijacked. We apply the Hotelling's statistic to all topics and use the p-value to determine whether the hashtag fails the null hypothesis that no significant change occurred. As mentioned in Section 3.3.1.3, the Hotelling's statistic compares the distribution of topics within a window of k days with the distribution of topics outside the window. We explore three window sizes, 1 day, 3 days, and 5 days.

Table 3.4 shows the AUC of detecting whether a hashtag has been hijacked. When Log Ratio is used, the performance is low as the approach fails to distinguish the hijacked hashtags from non-hijacked ones. In fact, the majority of the hashtags predicted as hijacked

Table 3.4: Area under ROC Curve.

Window Size	Number of Topics 3	4	5
Hotelling 1 Day Window	0.4973	0.5046	0.5048
Hotelling 3 Day Window	**0.6068**	**0.6593**	**0.6691**
Hotelling 5 Day Window	0.2298	0.1630	0.1024
Log Ratio of Terms	0.4691	0.4533	0.4364
Log Ratio of Users	0.4549	0.4776	0.4720

were false positives. In contrast, Hotelling's statistic at window size of 3 days was best at detecting hijacked hashtags. We calculated F-Measure based on the thresholds used for ROC curve. Hotelling's statistic using 3-day window also achieved the highest F-measure. Nonetheless, the size of the window for calculating the Hotelling's statistic has a great influence on performance. When the window size is only 1 or 5 days, the performance of our detection algorithm degrades significantly. This is because for the 1-day window, there is not enough within-window time series data to compute the Hotelling T^2 statistic effectively. When the window size is 5 days, the trending pattern gets overwhelmed by other topics, which makes it harder to identify the hijacked topic. Thus, for the remainder of our experiments using Hotelling's statistic, we will use the 3-day window size.

3.4.4.2 Detecting Hijacked Topic

Two approaches were proposed in Section 3.3.2.1 to detect the hijacked topic; similarity and Hotelling's statistc. For comparison, Log Ratio was applied to the term-topic and user-topic matrices. The performance, based on accuracy, of each approach is shown in Table 3.5. We applied these approaches to all synthetically hijacked hashtags.

Using the Hotelling's statistic to detect the hijacked topic performed best. The topic that corresponds to the hijacking is the topic with the highest unnormalized probability

when the hijack was detected. When the prediction of the hijacked topic was incorrect, the predicted topic typically was a topic different from the one with highest probability outside of the window. Analyzing the predicted hijacked topics for these hashtags, we found these hashtags were about a general event or group, e.g. a sports team or a conference, and the predicted hijacked topic was about a specific event, e.g. a sports game against a specific rival or a workshop at the conference. For example #`carnivorestour` is about the Carnivore Tour, a musical tour with bands Linkin Park and Thirty Seconds to Mars. Our algorithm predicted tweets from the announcement of the tour were the hijacked topic, where users used the band members' names to discuss this specific event, whereas they used the bands' names to discuss the tour and specifc concerts. In sports, the American League Championship Series (with hashtag #`alcs`) is a series of Major League baseball games. Fans of one team playing in the series, the Baltimore Orioles, used #`alcs` in addition to other hashtags, #`takethecrown` and #`wewontstop`, to promote their team. These users did not hijack #`alcs` because it was used within the original intent, Major League baseball played during the ALCS.

Similarity and Log Ratio had similar performance. For similarity, we compared the top 100 terms and 100 users with the largest unnormalized probability for each topic. The threshold 100 was selected based on the observation that the hijacked topic had the injected terms within the top 100 terms. We found the topic with the lowest user similarity tended to be the hijacked topic more often than the topic with the lowest term similarity. Log Ratio had the best performance when 3 topics were used and lower performance for 4 and 5 topics. The Log Ratio of users was more indicative of the hijacked topic than the Log Ratio of terms. Given that language tends to have more lexical variation, e.g. two terms having similar meaning or the same term having multiple meanings, detecting the hijacked topic from the user is more reliable than detecting it from the terms.

Table 3.5: Accuracy of detecting the hijacked topic.

Number of Topics	3	4	5
Hotelling 3 Day Window	0.45	**0.56**	**0.47**
Term Similarity	0.42	0.39	0.41
User Similarity	0.46	0.42	0.29
Log Ratio of Terms	0.43	0.30	0.26
Log Ratio of Users	**0.47**	0.40	0.34

Table 3.6: Accuracy of detecting when a hashtag is hijacked. This comparison is using only first, middle, or last day versus any of the days of the Hotelling's window as the date of the hijacking.

Number of Topics	First Day	Middle Day	Last Day	Any Day
3	32%	4%	5%	41%
4	43%	4%	1%	48%
5	45%	4%	3%	52%

3.4.4.3 Detecting When Hashtag is Hijacked

The next step is to determine when the hashtag was hijacked. Hotelling's statistic is a change detection technique, so it makes sense to predict the hijacking occured when the statistic indicated a change. Using Hotelling's statistic, we found a hijack window, where the topic distribution of the days within the window does not follow the same distribution for the days outside of the window. When using a window-based approach, predicting which day the change occurred is not intuitive. The time of the hijacking could be at the beginning, the end, or in the middle of the window. We explored the accuracy of detecting the date of the hijack by selecting each day of the 3-day window. The results are in Table 3.6.

Each day of the hijack window provides some indication the hashtag was hijacked. Among the days within the window, selecting the first day of the window as the hijack date was most accurate. If we relax the requirement such that we successfully predicted the hijack date if it falls anywhere within the window, we increase performance by 7%.

Table 3.7: Normal and hijacked tweets about #napaquake.

Label	Tweet
Normal	Hoping everyone is safe and accounted for. #NapaQuake
Normal	6.0 Earthquake Strikes Near Napa. #napaquake
Hijack	.@winetrain @erico & I challenged @KathieLGifford @hodakotb to #napaearthquakewinebucketchallenge http://t.co/XxWSlLKRVl #napaquake
Hijack	.@yayneabeba lol here's mine & @erico's #WineBucketChallenge http://t.co/XxWSlLKRVl @KathieLGifford @hodakotb #napaquake #ALS #wine

3.4.4.4 Additional Hashtag Hijacking Discovered

The previous experiment was conducted by artificially injecting hijacked tweets into 100 randomly selected hashtags. We now applied our algorithm to identify other trending hashtags and found a few interesting incidents of hijacked hashtags. Previous research on hashtag hijacking had reported that two hashtags, #napaquake and #whyistayed, were hijacked during our data collection period. In addition to the two, we found three additional hijacked hashtags when ranking the hashtags based on their Hotelling's statistic using 3-day window and selecting the top 100 hashtags.

The hashtag #napaquake became popular as a result of an earthquake hitting Napa Valley, California. The majority of the tweets discussed the earthquake, including damages, aftershocks, and concern for friends in the area. Examples of these tweets are in Table 3.7. One group of users hijacked #napaquake to promote a new trend #winebucketchallenge. The trend did not become popular, so the hijack was not successful. Table 3.8 presents the top 10 terms for each topic. Due to a significant overlap of terms between the topics, e.g. napaquake was ranked highest in all 3 topics, we sparsified the term-topic matrix. For each term, we find which topic has the highest weight. That topic maintains its weight while the weights of the remaining topics are set to zero.

A hashtag used to discuss domestic violence, #whyistayed, was hijacked by a frozen

Table 3.8: Top 10 Terms for each Topic for #napaquake.

Topic 1	Topic 2	Topic 3
napaquake	xxwsllkrvl	napa
wine	al	downtown
damag	winebucketchalleng	pool
quak	erico	valley
home	mapit water	anyon
today	main	friend
earthquak	video	weekend
recoveri	road	warm
help	door	aftershock
napavalley	kodakotb	feel

Table 3.9: Normal and hijacked tweets about #whyIstayed.

Label	Tweet
Normal	Because he was hurt, broken, the son of abusive alcoholics himself and said he'd kill himself if I left. #WhyIStayed
Normal	#WhyIStayed for the kids
Hijack	I was into reading the #WhyIStayed and #whyileft stories until @DiGiornoPizza ruined it.
Hijack	#whyistayed you said you had pizza. #whyileft you said I ate too much pizza.

pizza company. Normal tweets discuss why they stayed in an abusive relationship, as seen in Table 3.9. The hashtag hijackers focused on criticizing the company for attempting to hijack the hashtag. In this situation, the hashtag was hijacked twice: the pizza company hijacked the hashtag to gain attention for their products, then the users hijacked the hashtag to demonstrate the PR campaign was wrong. In our dataset, we only observed the second hijack.

After ranking all of the hashtags by their Hotelling's statistic, we analyzed the top 100 hashtags for hijacking. Two examples of hijacked hashtags we found were #dear5sos and #teammexico. Normal tweets for #dear5sos read like the user is writing a letter to the band 5 Seconds of Summer. They start with #dear5sos. The remainder of the tweet is the body of the letter. Table 3.10 shows examples of these. The hijackers' tweets contain several

Table 3.10: Normal and hijacked tweets about #dear5sos.

Label	Tweet
Normal	#dear5sos thanks for all the smiles
Hijacked	SUBSCRIBE TO THIS CHANNEL http://t.co/udT8tPNSTo LETS GET TO 250!!!! #youtube #dear5sos #BeyDay #beyoncebirthday #HappyBirthdayBeyonce

unrelated hashtags to draw attention to their tweet and entice users to click on the link. This is a clear example of hashtag hijacking for spamming, according to our definition.

Some hashtags naturally can be used in different contexts. When the hashtag changes context, this may not necessarily be hashtag hijacking. For example #teammexico could be used for different types of sports teams, e.g. basketball or football. We observed this hashtag was first used to promote the Mexican basketball team playing in International Basketball Federation (FIBA) tournament. We also observed tweets with the same hashtag promoting a singer. To determine whether this hashtag was hijacked, the context needed to be considered. If the singer was representing Mexico in a competition, it still fits the original intent of the hashtag. The hashtag was classified as hijacked for two reasons: the tweets did not promote the singer as either from or representing Mexico in a competition and the tweets followed a pattern used by attention-seeking trolls, tweeting the same tweet repeatedly. The hashtag was clearly hijacked to promote the musical album of a Spanish singer, Pablo Alboran. This hijacking had a unique characteristic; tweets discussing basketball were in English and Spanish whereas tweets about the singer were only in Spanish. Examples of tweets written in Spanish are listed in Table 3.11 with their English translation in italics.

An example of a hashtag having multiple contexts which is not hijacked is #swim. This hashtag is used in two contexts; the recreational activity in a pool and the acronym for software used by the Federal Aviation Administration (FAA) called System Wide Information

Table 3.11: Normal and hijacked tweets about #teammexico. English translation of tweets written in Spanish are in italics.

Label	Tweet
Normal	Ochoa better play for the 2nd half ! #teammexico
Normal	@FIBA #TeamMexico vamos!! Slay the giant!! http://t.co/YBpkS3Pyrc *@FIBA #TeamMexico let's go!! Slay the giant!! http://t.co/YBpkS3Pyrc*
Hijacked	Esos pts son nuestros\u2665 #TerrallyAlboran Vamos #TeamMexico @pabloalboran *Those are our points\u2665 #TerrallyAlboran Let's go #TeamMexico @pabloalboran*
Hijacked	A darle con todo en el #terrallyalboran #teammexico esos pts. Son nuestros @pabloalboran :-) *To give everything in the #terrallyalboran #teammexico these points. They are our @pabloalboran :-)*

Table 3.12: Tweets from #swim, a hashtag with multiple contexts.

Context	Tweet
Aquatics	Perfect #beach day #bathtub #swim #sun @ Alys Beach http://t.co/rxM4nNmC6E
Software	#SWIM Historical Review - #ATIEC keynote by Steve Bradford FAA Chief Scientist for Architecture & NextGEN Development http://t.co/SN2VwYjU7C
Aquatics	Time to focus...priortize.....stress relief #SWIM

Management (SWIM). One indication of which topic the hashtag belongs to is whether capitalization is used. Most of the time, #swim is used to describe the water sport. Tweets discussing the FAA software, they used the #SWIM. As demonstrated by the third tweet in Table 3.12, this is not a strict rule; users capitalize the hashtag while referring to the aquatic sport.

3.5 Conclusion

This chapter considers the problem of automatically detecting hijacked hashtags from Twitter data. We present a novel framework to identify candidates of hashtag hijacking and verify the hijacking. Identification of candidates is based on multimodal non-negative ma-

trix factorization to learn the topics associated with a hashtag and use Hotelling's t^2 test to identify anomalous behavior. The candidates are subsequently validated by automatic identification of the hijacked topic and verification of the hijack timing using Hotelling's statistic. Experimental results showed the framework is promising as it can effectively detect many potentially interesting hijacked hashtags.

Chapter 4

Understanding Compromised Accounts

4.1 Introduction

Social media has become a widely used medium for users to share information with their friends and family. Unfortunately, its popularity has also attracted considerable attention among hackers to disseminate misinformation by compromising users' accounts. These hackers then perform a variety of online abuses, including spamming and identity theft, with these accounts.

Recent research has measured the frequency and impact of compromised accounts on social media websites such as Twitter. A 2016 Pew Research study found 13% of online adults experienced having their social media account compromised [71]. A survey conducted by the University of Phoenix found nearly two thirds of adults with a social media account had been compromised [61]. Compromised accounts can have negative impact on both the social media sites as well as their users. For example, Thomas et al. reported that 57% of victims of accounts compromised lost friends on Twitter when their accounts were compromised, and 21% of victims never return to Twitter [96]. Zangerle and Specht found that among the users reporting their accounts compromised, 27% of them stated they created and moved to

new accounts [111]. When a business is targeted by a compromised account, they can lose offended customers before they are able to respond to the compromised accounts [58].

A previous study [6] presented four reasons hackers would compromise social media accounts—for laughs, forced shares, forced follows, and for user information. Forced shares enable hackers to share links to phishing and malware infested sites. Its impact can be malicious because users are more likely to click on spam links shared on the social media sites than those embedded in emails [35]. Accounts can also be hacked to impersonate the users and make them look bad [75, 25] by posting misleading or harmful information about their political inclination, religious beliefs, sexual orientation, etc. Some hackers may also use the information learned from the account to target the user's other accounts, e.g. bank account.

There are many ways hackers can gain access to login credentials of social media users, e.g., through database dumps, password guessing, social contagions, external contagions [19, 96], and shared devices. Database dumps occur as a result of a security breach that exposes login credentials of users to hackers. Database dumps from other services can also result in compromised credentials when the users reuse the same passwords on other social media sites. Password guessing is another approach that can be used to hack into accounts of users with weak passwords. Social contagions refer to the spread of phishing and malware distributed from within a social media platform to trick users into revealing their login credentials. In contrast, external contagions are malware or phishing attacks from outside of the social media, e.g., malware planted on the user's computer or phishing attacks in email, which allow hackers to steal information from users including their social media accounts. Finally, the accounts can be compromised when users leave their devices, e.g. phone or computer, unlocked for anyone to use.

There is growing research dedicated to detecting compromised accounts [28, 41]. Some

research assume that a compromised account follows the behaviors of malicious accounts, i.e. accounts controlled by a malicious user [28]. They assume that these compromised accounts will spread spam or malware in sync with other accounts, creating social contagions. There have also been studies that focus on detecting malicious accounts including those that are part of social contagions, but such studies do not distinguish between fake accounts and compromised accounts [9, 17, 91]. One of the limitations of these previous studies is that not all compromised accounts are part of social contagions. Sometimes a hacker will only compromise a single account for fun or other non-malicious reasons. For example, research has shown that 10% of the time, accounts are compromised by someone the user knows [111]. Another limitation is that many of the previous work have relied on using a limited number of features to detect compromised accounts. For example, Igawa et al. [41] detected compromised accounts using only content information, i.e., the terms that appeared in the tweets posted by the users. However, since tweets are typically short, informal, and noisy, they may not provide sufficient information for accurate text analysis [86]. On the other hand, the tweets also contain rich meta information such as hashtags and mentions, which could provide more helpful information to advance compromised account detection.

Understanding the characteristics of compromised accounts would provide useful insights into how to detect compromised accounts. In this chapter, I aim to understand who compromises accounts on Twitter, what information they share, and what patterns may be useful for detecting compromised accounts. Different types of hackers compromise accounts for different reasons, and thus, they may compromise the accounts in different ways and share different types of content. Some approaches that can detect certain types of compromised accounts well, e.g. using source information to detect spammers [28], may perform poorly on other types of hackers, e.g. hackers who use similar device as the account owner. Therefore,

in this chapter, I aim to understand compromised accounts using both textual and meta information. The specific contributions of this research are:

- Identified two types of hackers that compromise accounts;
- Discovered six themes of compromised tweets based on the types of hackers; and
- Examined patterns of compromised tweets and studied the importance of such patterns for detection.

The structure of the chapter is as follows. In the next section, I formally define compromised accounts and presents other terminology used in the dissertation. Section 4.3 describes the data collected. Section 4.4 presents the findings of hackers' identities and the common themes of tweets they share. Patterns useful for detecting compromised tweets are examined in Section 4.5. Conclusions are presented in Section 4.6.

4.2 Preliminaries

This section formally defines compromised accounts and introduces other terminology that will be used in the remainder of this dissertation. A social media account, A, is owned by an individual who created that account, i.e. the **original user** or the **genuine user**. Any individual who accesses account A without the knowledge or permission of the original user is known as a **hacker**. If account A is accessed by a hacker, it is said to be compromised.

Definition 4.2.1. (*Compromised Account*) A compromised account is an account accessed by a third party without the knowledge of the original user.

I consider two types of tweets when collecting Twitter data for compromised account detection: announcement tweets and timeline tweets. Tweets initially collected from the

Twitter stream, to be described in the next section, are called **announcement tweets** if they matched the keywords used to announce compromised accounts. Additional tweets collected from the same users are called **timeline tweets**. If a set of tweets is not specified in the remainder of this chapter as an announcement tweet, then it is referred to as a timeline tweet. In this study, a tweet will be classified as compromised or normal. A **compromised tweet** is a tweet believed to be authored by the hacker. A **normal tweet** is any tweet written by the original user. For example, announcement tweets are normal as they were written by the original user declaring their accounts had been compromised.

There are four types of compromised accounts; pranks, forced shares, forced follows, and information gathering [6].

- **Prank:** A prank is content that is shared with other users by the hacker for laughs. Pranks include sharing random content, e.g. song lyrics or confessions of love for the hacker.

- **Forced Share:** A forced share is when a hacker shares content that is false, misleading, or of malicious intent on the social media site.

- **Forced Follow:** A forced follow occurs when the hacker forces the user account to follow other fake or malicious accounts.

- **Information Gathering:** A hacker uses the account to learn sensitive information about the user, such as their password. This is used by spammers to compromise the original user's other accounts, e.g. bank account.

For pranks and forced shares, hackers can share content either via social media posts or in direct messages to other users. The focus of this research is to analyze the social media posts

Table 4.1: Phrases Users Say Indicating their Account was Compromised.

Phrase	Regular Expression
"sorry that wasnt me"	(.*sn't me .*),(.*sn't me$),(.*not me$), (.*not me .*),(.*snt me .*),(.*snt me$)
"I was hacked!"	(.* hack.*)

created by the hacker for pranks and forced shares. Forced follows are harder to detect than forced shares, because the Twitter API only returns the current number of followers of a user when the tweet was collected rather than when the tweet was published. To study detection of forced follows, researchers would need to know which accounts will be compromised in the future to follow how the number of followers changes over time. Accounts compromised for only information gathering are the most challenging to detect because the hacker does not change anything about the account or post any tweets. Detection of information gathering compromises would require knowledge of the user's login history, e.g. IP addresses, to identify these hackers, and this information is not provided by the Twitter API.

There are two types of hackers who compromise an account; acquaintances and intruders. An **acquaintance** is a friend, relative, or coworker of the original user. An **intruder** is an unknown third party who compromises an account to share misleading or harmful information. One type of intruders are spammers, users whose posts are intended to mislead normal users to visit phishing and malware websites. Previous research has primarily focused on detecting accounts compromised by spammers [28].

4.3 Data

This section introduces the details about the data used for this study. Data referenced as **2015 Data** is described below.

Table 4.2: Codebook used to determine whether the tweets in a user's account has been compromised (C) or normal (N).

Label	Scenario	Example
C	Stated someone else tweeted on their account	not me posting it
C	Mentioned "hack" without context	I was hacked
C	Mentioned "hack" with Twitter context	So my Twitter got hacked
N	Compromised on different social media platform	Rachel needs to stop hacking into my facebook
N	Different Twitter account compromised	my other account was hacked
N	Not written in English	Para que me hackees?
N	Compromised account in distant past	about 2 years ago someone hack into my account
N	Only said "not me" or "wasn't me"	that wasn't me
N	Uses hack in a different context	Spurs are hacking Asik off the ball

4.3.1 Data Collection and Annotation

All tweets geotagged as originating from the continental United States using the Twitter Streaming API from April 27, 2015 to May 6, 2015 were collected. To roughly identify which users have been compromised, tweets were matched to a regular expression where the user may be self-reporting they were compromised. Table 4.1 shows the self-reporting phrases and the corresponding regular expressions. For each tweet matching the regular expression, the previous 200 tweets from the user were collected. These tweets will be used to classify whether their author was the original user or a hacker. To ensure this dataset includes tweets from user accounts that were not compromised, we randomly selected a set of 500 tweets that did not fit the regular expression and collected the previous 200 tweets from their authors.

The majority of tweets matching the regular expression did not appear to be reporting a compromised account. For example, "hack" can refer to writing code, shortcuts (life hacks), or users discussing compromised accounts on different websites, like Facebook or Snapchat.

To determine which accounts were really compromised, two annotators labeled each tweet as whether or not the tweet claims the Twitter account was compromised. If the tweet only stated they were compromised without any other context, e.g. "I was hacked", the annotators assumed that the user was referring to his/her Twitter account. The codebook used for this annotation is listed in Table 4.2. When the annotators disagreed on a tweet's label, a third annotator labeled the tweet using the same codebook. If at least two of the annotators labeled a tweet as reporting that account was compromised, then the tweet was labeled as compromised. Otherwise the tweet was labeled as not compromised.

Some users had multiple announcement tweets matching the regular expression used for data collection. Among these users, some reported their accounts compromised multiple times, while others discussed an event, e.g. computer hackathon, in multiple tweets. The first unit of analysis in this research is the user, so it is important to remove duplicate users. If a user has multiple announcemnt tweets in the dataset, their timeline tweets were combined into a single set. Duplicate timeline tweets were removed. If a user had both compromised and not compromised announcement tweets, the user is labeled compromised. Otherwise the user maintains the label of their announcement tweets.

Tweets claiming the account was compromised, i.e. announcement tweets, were also annotated on whether the original user knew the hacker. If the tweet said the name, Twitter handle, or relation of the hacker to the user, e.g. sister, then the hacker was known to the original user. If the announcement tweet was written in first person present participle verb tense, e.g. "I'm hacking you", then the hacker was known to the original user. If the announcement tweet did not mention the identity of the hacker, then it is assumed the original user does not know the hacker.

For each compromised user, an annotator read their tweets to identify which tweets were

written by the hacker. If the compromised tweets of a user could be identified, features from those tweets were extracted and compared to the normal tweets by the same user to identify characteristics which are more indicative of a compromised tweet than a normal tweet.

4.3.2 Data Statistics

The Twitter Streaming API returned 6334 tweets matching the regular expression. The two annotators labeled these tweets, and their inter-rater reliability was 0.722. There is substantial agreement between the annotators [45, 105]. After annotation, 584 announcement tweets indicated the account was compromised, 5750 announcement tweets that matched the regular expression were not indicating the account was compromised, and 500 announcement tweets that did not match the regular expression. After combining tweets from the same user, 502 user accounts were labeled as compromised and 5387 users were labeled as not compromised. When analyzing the number of tweets collected from each user's timeline, one concern was that users may delete their hackers' tweets before announcing the account was compromised. To ensure the hackers' tweets were likely collected, all users having 150 or fewer tweets were removed, which left us with a dataset that contains 461 compromised users and 4913 normal users.

4.4 Hackers and Their Content Themes

In this section, we use the Twitter data collected to study who compromises accounts and what information do they share.

Table 4.3: Examples of Announcement tweets where hacker is an acquaintance.

Hacker Mention	Example Announcement Tweet
Pronoun	How tf did she hack my stuff
Name	Kewian hacked me.
Relation	Why does my bf think it's okay to hack my Twitter
Twitter Handle	Omg i got hacked by @audra_allen14

4.4.1 Hacker Identity

Accounts are compromised by two types of individuals, acquaintances and intruders. Among the compromised users, 16% reported they knew the hacker, by name (e.g. "When Evan hacks ur twitter.") or by relation ("my friend"). Previous research found that in 10% of announcement tweets, the hacker was someone the original user knew, i.e. friend or relative [111].

Hackers identified by mention or pronoun present an ambiguous case about whether the hacker is an acquaintance of the user. 12% of compromised users announced the identity of their hackers by Twitter screen name or by pronoun. If the hacker is someone the user communicates with often, the original user may consider their hackers as friends, i.e. acquaintance. Otherwise the hacker would be considered as an intruder. Examples of the four ways hackers are identified are presented in Table 4.3.

4.4.2 Hacker Content

Tweets by the hacker could be identified for 47% of the compromised users. I analyzed the content of each compromised tweet and identified its topic. Topics were manually grouped based on common themes. Through analysis of the content of these tweets, six themes emerged.

- **Spam:** Advertising a product or porn

- **False proclamation:** Announcement that is false, such as relationship status, sexual orientation, or pregnancy
- **Praise:** Typically praise hacker, however can include praise of other items, e.g. grilled cheese
- **Criticism:** Strong negativity towards an individual, or a group
- **Announce Compromise:** Hacker announces they compromised the account
- **Other:** Tweets that did not fit any previous theme.

The number of users whose hackers followed each theme is shown in Figure 4.1. The most prominent theme was fake announcements. Hackers told the original users' friends falsehoods about wanting sex, their relationship status, their bathroom habits, and their sexual orientation. Most of these falsehoods are harmless to the original user, however some of them can hurt the reputation of the user, e.g. coming out as gay or announcing they smoke cannabis and cheat on tests. The next most common theme are when the hacker posts the announcement tweet. These tweets are straightforward, the user identifies themselves and then says they are hacking the user. Spammers were 16% of the hackers whose tweets could be easily identified. Spammers enticed other users to purchase items, e.g. weight loss pills or porn, and they also requested information from their audience, e.g. nude images. Despite the extensive previous work on detecting spam and spammers, spam contributes to a small percent of compromised accounts. Praising the hacker was a common theme among hackers, utilized by 13% of the hackers. Criticism was the least frequent theme. Hackers criticized other individuals and groups in general, e.g. fat people. They also called for violence against police. If this criticism is interpreted as threats, the original user could face consequences

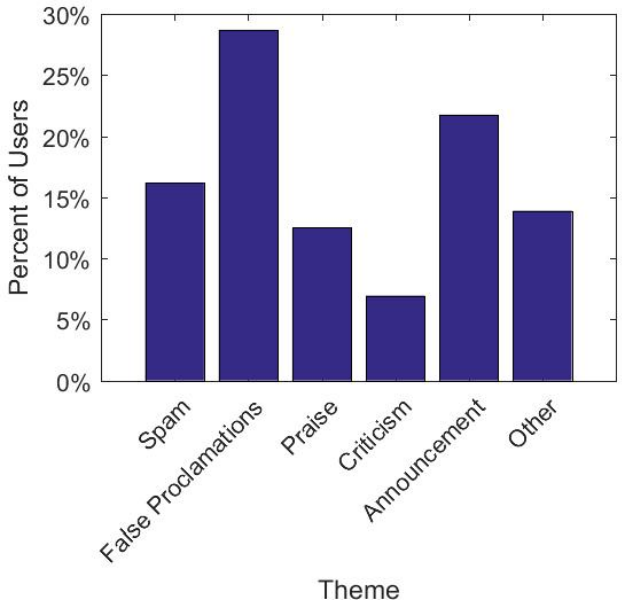


Figure 4.1: Themes of hackers' tweets where hacker tweets could be identified and the percent of users whose hacker followed each theme.

for content the hacker posts. There were also several hackers whose tweets did not follow any theme, noted by the Other theme in Figure 4.1. Some hackers retweet random tweets which have no theme. For example, sometimes the hacker simply tweeted "hello".

4.5 Characteristics of Compromised Tweets

I characterize two types of tweets from a compromised account – (1) compromised tweets: tweets from the hacker; and (2) normal tweets: tweets from the original user. The goal is to determine whether compromised tweets have distinct patterns from normal tweets. This section first analyzes compromised tweets in terms of their features, e.g. hashtags, mentions, etc. Then I test whether these features can help with compromised tweet detection. In particular, the aim of this section is to answer the following research questions:

- RQ1: Do compromised tweets contain more hashtags than normal tweets?

- RQ2: Are mentions more prevalent in compromised tweets than normal tweets?
- RQ3: Are compromised tweets more likely to contain URLs?
- RQ4: Do the sources, software used to post the tweet, of compromised tweets match the sources of normal tweets?
- RQ5: Are compromised tweets more positive or negative than normal tweets?
- RQ6: Are compromised tweets more likely to be retweets?

For each research question, the normal tweets are compared against the compromised tweets for compromised users whose compromised tweets could be identified. To determine whether the distribution of the compromised tweets differ significantly from their normal tweets, a one-sided Welch's T-test is applied to compare the mean values of the features associated with each research question (hashtags, mentions, retweets, etc). Welch's T-test is used instead of Student's T-test since the sample size and variance of the normal and compromised tweets are unequal. However, the Welch T-test makes a strong assumption that the data is normally distributed. To overcome this limitation, the non-parametric 2-sample Kolmogorov-Smirnov (K-S) test is also applied to compare the cumulative distributions of the normal tweets and compromised tweets. Both hypothesis testings were applied to 5085 compromised tweets and 42,024 normal tweets from the same users.

4.5.1 Hashtags

Hashtags are frequently used in tweets to provide context [34, 57, 99]. A tweet can contain multiple hashtags or no hashtags. From the histogram in Figure 4.2, compromised tweets appear to contain more hashtags than normal tweets. Specifically, this figure shows that

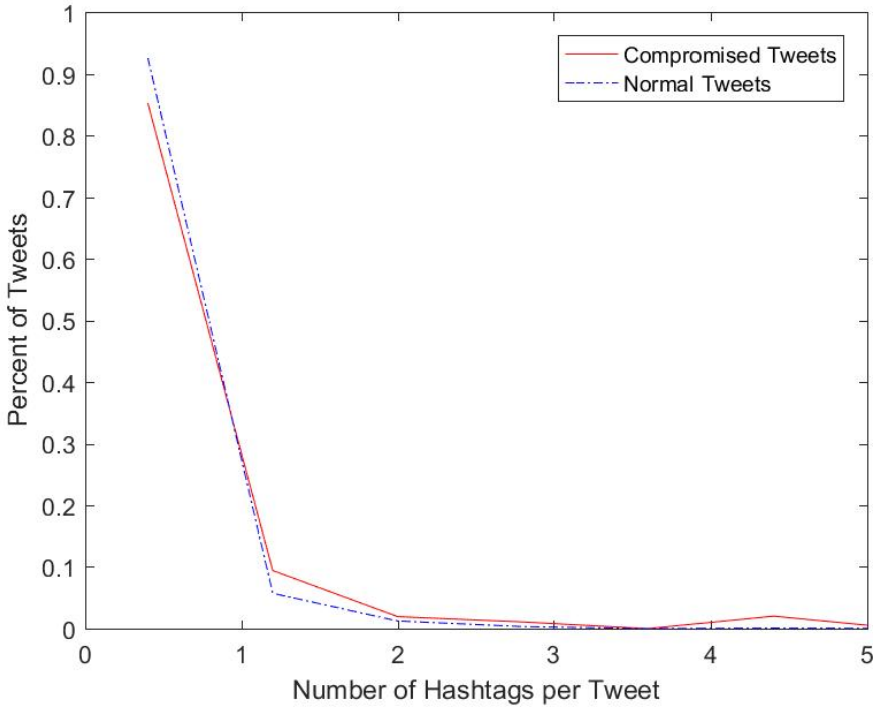


Figure 4.2: Histogram of the number of hashtags used per tweet. Compromised tweets tend to have more hashtags.

92% of normal tweets do not contain any hashtags whereas only 85% of compromised tweets do not contain a hashtag. Furthermore, less than 1% of normal tweets contain four or more hashtags. In comparison, 2.3% of compromised tweets contain 4 or more hashtags. This is further demonstrated by analyzing the average number of hashtags for compromised tweets and normal tweets, shown in Table 4.4. The average for compromised tweets is higher than for normal tweets. The p-value according to the Welch's T-test suggests that the difference is significant. For the K-S test, the null hypothesis is that compromised tweets contained equal or fewer number of hashtags than normal tweets. Applying the one-sided hypothesis test with 0.001 significance level, this null hypothesis was rejected, as shown in Table 4.4. From this evidence, the results suggest compromised tweets contain more hashtags than normal tweets.

Table 4.4: Comparison whether compromised tweets have more hashtags or mentions than normal tweets by the same user. Compromised tweets tend to have significantly more hashtags and more mentions than normal tweets.

	Mean		p-value	
Aspects	Compromised	Normal	T-test	K-S test
Hashtags	0.27	0.09	9.7×10^{-48}	5.5×10^{-22}
Mentions	0.97	0.80	3.6×10^{-27}	2.9×10^{-27}

4.5.2 Mentions

Users tag other users by mentioning their Twitter handle to gain their attention [108]. Spammers use mentions to entice users to engage with their tweets [35]. As shown in Figure 4.3, compromised tweets appear to have more mentions than normal tweets. Specifically, this figure shows a larger percentage of normal tweets contain between 0 and 2 mentions, compared to compromised tweets. Furthermore, a higher percentage of compromised tweets have four or more mentions, compared to normal tweets. Moreover by analyzing the average number of mentions per tweet for the two groups, shown in Table 4.4, compromised tweets were found to have more mentions than normal tweets. Applying the K-S test with significance of 0.001, the null hypothesis, compromised tweets would have similar or fewer number of mentions than normal tweets was rejected, as shown in Table 4.4. Thus, compromised tweets tend to have more mentions than normal tweets.

4.5.3 URLs

When users share images or tweets within a tweet, they are linking to that content by adding URLs to their tweets. URLs can also lead to different websites. URLs are shown in a shortened form, e.g. Twitter's shortened URLs are masked to the t.co domain, which masks the final destination of the URL. Spammers utilize this feature to mislead users to their

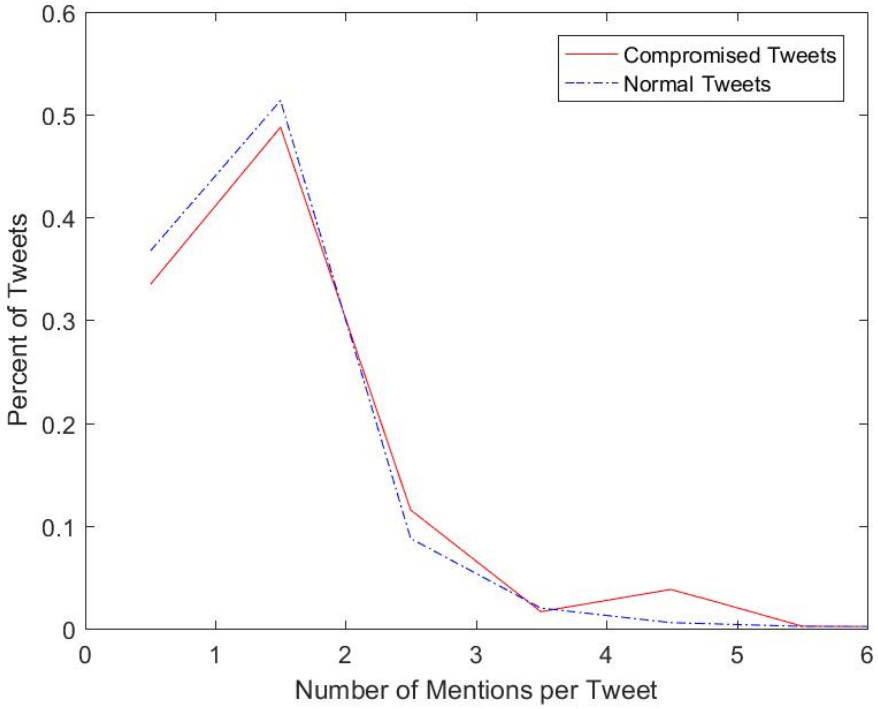


Figure 4.3: Histogram of the number of mentions used per tweet. Compromised tweets tend to have more mentions.

malicious content [22, 35, 95]. Figure 4.4 shows that tweets without a URL tend to be normal, and tweets with URLs appear more often in compromised tweets. Looking at the average number of URLs for each set of tweets, shown in Table 4.5, compromised tweets have more URLs than normal tweets. Welch's T-test suggests this difference is significant. The null hypothesis that URLs appear as often or less often in compromised tweets than in normal tweets was rejected with the KS-test. The p-value is shown in Table 4.5. Compromised tweets contain more URLs than normal tweets.

4.5.4 Retweets

Retweets are a common tool used to share tweets with one's followers [108]. Whether compromised tweets were more often being retweets compared to normal tweets was tested. K-S test cannot be performed on binary valued variables, however it is possible to compare the

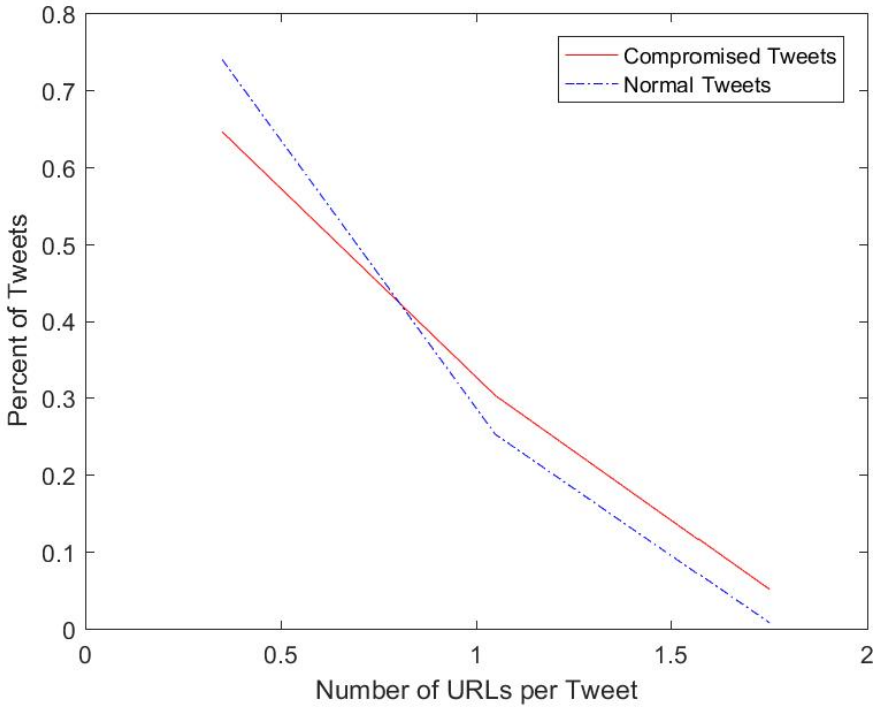


Figure 4.4: Histogram of the number of URLs used per tweet. Compromised tweets tend to have more URLs.

Table 4.5: Comparison whether compromised tweets are more likely to contain URLs, are more positive sentiment, or are more likely retweets compared to normal tweets. Tweets contain between 0 and 3 URLs. A retweet is valued at 1, whereas a non-retweet is valued at 0. Sentiment ranges from -4 (negative sentiment) to +4 (positive sentiment). Sentiment score of 0 indicates the tweet is neutral.

	Mean		p-value	
Aspects	Compromised	Normal	T-test	K-S test
URL	0.41	0.27	1.5×10^{-57}	1.4×10^{-35}
Retweet	0.24	0.35	1	N/A
Sentiment	0.18	0.13	4.6×10^{-4}	5.0×10^{-5}

means using Welch's T-test. The results suggests the average number of retweets is lower for compromised tweets than for normal tweets. The null hypothesis, that compromised tweets are retweets as often or less often as normal tweets, could not be rejected. Compromised tweets are not more likely to be retweets compared to normal tweets.

4.5.5 Source

Every tweet is published from a specific software, i.e. source. Most tweets, 89% of tweets collected, are published from the Twitter apps for iPhone, Android, or web client. Users can grant permission to third party apps, e.g. games, to post on their Twitter feed. By analyzing how often compromised tweets come from sources that the original user never used in their normal tweets, we found that 50% of the compromised users had at least one compromised tweet posted from a source they had not used. This is consistent with [28] – source was a good predictor for identifying compromised accounts that belong to spam campaigns.

4.5.6 Sentiment

Sentiment, whether a tweet is positive, negative, or neutral, may be a useful feature. Previous research found that spam emails tend to have positive sentiment [40]. To determine whether this finding can also be applied to compromised tweets, SentiStrength [94] was applied to all tweets to learn their sentiment. SentiStrength returns a positive score, range 1 to 5, and a negative score, in the range of -1 to -5. A score closer to zero indicates the tweet is neutral. The overall sentiment of each tweet is the addition of its positive and negative score. Comparing the average sentiment of compromised tweets and normal tweets, shown in Table 4.5, compromised tweets are more positive. Welch's T-test suggests this difference

of averages is significant. Under the null hypothesis, compromised tweets are similar or more negative sentiment than normal tweets. This null hypothesis was rejected, using K-S test, with significance 0.001. The p-value for this test is shown in Table 4.5. Similar to spam emails, compromised tweets are more positive than normal tweets.

4.5.7 Compromised Tweet Detection

This subsection demonstrates that the proposed features improve performance of detection of compromised tweets.

To measure the predictive performance of the explored features, a logistic regression classifier was learned and evaluated using 5-fold cross validation using F-Measure. Stratified sampling was used such that equal number of samples from each class were in each fold. In addition to analyzing classifier performance, the coefficients of the features were ranked and analyzed. Coefficients were standardized, using the full standardized method proposed by Menard [63, 64], and ranked by the absolute value of the standardized coefficient.

Along with the features proposed earlier in this section, the set of terms (tweet content) were also used. Text data is inherently noisy with a large vocabulary and several forms of the same word. Without preprocessing steps to reduce the size of the vocabulary, a classifier would be prone to overfitting. Terms are filtered by removing stopwords and stemmed using the Porter Stemmer. All terms are lowercased and punctuation is removed. Terms appearing in only one tweet are removed. After preprocessing, 69,951 terms were found. For source features, some sources may be used to publish only one tweet, which can also lead a classifier to overfit to the data. Sources which only appear in one tweet are replaced with a single token. Including a token for rare sources, 285 sources were found. Hashtags, Mentions, and URLs were each represented as 1 feature whose value was their frequency in each tweet.

Table 4.6: Classifier performance, measured in F-Measure, for identifying compromised tweets using different sets of features.

Feature	F-Measure
Terms	0.32
All	0.41
Random	0.11

Sentiment was an integer in the range of -4 to 4. One feature denoted whether a tweet was a retweet or not. There were 70,241 features in total.

The detection performance is reported in Table 4.6, where "Random", "Terms" and "All" denote the performance of random guessing, logistic regression using terms only, and logistic regression using both terms and the proposed features. From the table, the following observations are made – (1) the performance of "Random" is low due to the highly skewed dataset; (2) both "Terms" and "All" perform much better than "Random"; and (3) "All" outperforms "Terms", suggesting that the proposed features can significantly boost the detection performance. The result suggests that features from meta information are complimentary to tweet content. Examine the standardized coefficients of the features in the logistic regression model found that, among the top-10 features, 9 of them are associated with the proposed new features, as shown in Table 4.7. This further validates the importance of using the new features. For example, the top-10 features include some of the most common sources used to post to Twitter. The negative coefficients associated with these features suggest they are predictive of normal tweets. Furthermore, the model also suggests that if the tweet is a retweet, it is likely not compromised. This is consistent with the results in Table 4.5. Similarly, Table 4.4 showed that more URLs appear in compromised tweets, so it follows that the regression coefficient associated with the URL feature is positive-valued, i.e., predictive of compromised tweets. The term feature, 0f, is part of a unicode variation selector, "\ufe0f",

Table 4.7: Top features for detection of compromised tweets. Features ranked by absolute value of standardized coefficients. The proposed features, e.g. sources, retweets, and URLs, were predictive of identifying compromised tweets from normal tweets.

Feature	Type	Standardized Coefficient
Twitter for iPhone	Source	-1.22×10^{-3}
Twitter for Android	Source	-9.11×10^{-4}
Twitter Web Client	Source	-4.05×10^{-4}
Twitter for iPad	Source	-3.53×10^{-4}
Is Retweet	Retweet	-2.72×10^{-4}
Echofon	Source	-2.35×10^{-4}
0f	Term	-2.28×10^{-4}
Tweetlogix	Source	-2.26×10^{-4}
Number of URLs	URL	2.22×10^{-4}
iOs	Source	-2.13×10^{-4}

commonly used to modify an emoji. Since its coefficient is negative valued, this suggests that the presence of emojis is more predictive of normal tweets.

4.6 Conclusion

This study analyzed the content of compromised accounts to understand who are hackers, what type of content do hackers tweet, and what features can help distinguish between compromised tweets and normal tweets. There are two types of hackers; acquaintances and intruders. 16% of compromised accounts were hacked by an acquaintance. The content of compromised tweets belonged to one of six themes; spam, false proclamations, praises, criticism, announcements of compromise, and other. The prevalence of features in compromised tweets and normal tweets were compared, and they demonstrated improved classification performance. Compromised tweets have more hashtags and mentions than normal tweet, but they are not predictive of compromised tweets. Compromised tweets more often contain a URL and are more positve than normal tweets. Sentiment is predictive for tweet classifi-

cation. Compromised tweets are not retweeted more often than normal tweets. Terms and sources were the most predictive features for detecting compromised tweets. Given these preliminary findings on the most predictive features, next chapter proposes an algorithm to detect compromised accounts.

Chapter 5

Compromised Account Detection using Unsupervised Learning

5.1 Introduction

The method proposed in the previous chapter, use logistic regression to identify compromised tweets, would not be effective at detecting compromised tweets from previously unobserved accounts. Due to its small size, it is unlikely that the compromised tweets used to train the classifier are representative of all compromised tweets. The compromised tweets could be identified for less than half of the compromised accounts. This annotation task is challenging because the average person often needs a large sample of tweets and a long time to learn the writing style of the user in order to identify the tweets that do not belong to that user. Additionally, expert annotators are expensive. Therefore it would be beneficial to detect compromised accounts without dependence on knowing which tweets are compromised.

Detecting compromised accounts from social media faces another challenge: social media posts are inherently noisy, riddled with lexical variations, acronyms, and misspellings [7]. Applying standard algorithms to the textual features of the social media content (e.g. N-grams, hashtags, and mentions) [41] alone may not be sufficient to effectively detect unusual patterns of user behavior. Furthermore, a user may have a variety of topics to discuss, and

occasionally, in multiple languages. Thus, a change in user topic or the language used does not necessarily indicate a hacker has gained control of their accounts.

To overcome these challenges, this chapter presents a novel learning framework called *CADET* for detecting compromised accounts on Twitter. Instead of relying on the textual features alone, *CADET* utilizes additional side information (e.g., source, location, and time information) to enhance its detection performance. Nevertheless, combining the textual and meta-data information is a challenging problem in itself. A trivial way is to concatenate the features from all modalities of the data together into a single feature vector, but such an approach is likely to be ineffective as it fails to consider the unique dimensionality, scale, and underlying topics associated with each view of the data. Furthermore, the relationships among the features and topics from different modalities are likely to be nonlinear, making it challenging to learn them efficiently in a robust, unified learning framework. The proposed framework is also unsupervised, enabling it to identify compromised accounts without the need for labeled data. While previous work has focused on identifying compromised tweets [28, 68], CADET focuses on detecting compromised accounts at the user-level, which bypasses the need to classify the individual tweets.

CADET employs a nonlinear multi-view learning approach using auto-encoders to derive the feature representation (topics) of each data view. The nonlinear topics are subsequently integrated by projecting them into a common latent space using a variant of generalized canonical correlation analysis [18]. A major advantage of using the proposed framework is its flexibility to incorporate diverse types of meta-data due to its ability to handle the varying number of features and topics across different modalities. Another advantage is that, by allowing the topics to share a common latent space, we can compute the reconstruction error of each user in the shared subspace to identify the compromised accounts. Experimental

Figure 5.1: Twitter pattern of a compromised user. The original user will tweet before and after their account has been compromised, denoted as normal tweets. When the hacker takes control of the account, they will publish tweets, i.e. compromised tweets. When the user realizes their account was compromised, they will alert their followers of the compromise in an announcement tweet.

results using real-world Twitter data demonstrate the superiority of *CADET* over several previously used unsupervised compromised account detection approaches.

5.2 Problem Statement

A compromised account is an account created by a legitimate user, i.e. the original user, which has been taken over by an unauthorized individual, i.e. a hacker. The expectation for a given compromised account is there will be several tweets authored by the original user before the account was compromised, as shown in Figure 5.1. Once the hackers have taken control of an account, they will start publishing tweets, which we call compromised tweets. When the original users notice that their accounts have been compromised, they will announce to their followers that their accounts were compromised in an announcement tweet[1]. The assumption is that the compromised tweets will be different from the normal tweets through one or more data views.

The compromised account detection problem can be formalized as follows. Let $\mathcal{U}$ =

[1]A user may announce their account was compromised via other means, e.g. direct messaging, emails, etc., which is beyond the scope of this dissertation

$\{u_1, u_2, \cdots, u_N\}$ denote the set of user accounts, where N is the number of users. Each user account u_i is associated with a set of multi-modal feature vectors, $\{\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \cdots, \mathbf{x}_k^{(i)}\}$, where $\mathbf{x}_v^{(i)} \in \mathbb{R}^{d_v}$ is a d_v-dimensional feature vector of user i for the v-th modality (view). Furthermore, we can concatenate the feature vectors for the v-th view from all the users into an $N \times d_v$ data matrix, $X_v = [\mathbf{x}_v^{(1)}\ \mathbf{x}_v^{(2)}\ \cdots\ \mathbf{x}_v^{(N)}]^T$.

Definition 5.2.1 (Compromised Account Detection)**.** Let $\mathcal{U}$ be the set of users and $\{X_1, X_2, \cdots, X_k\}$ be the corresponding data matrices for each of the k views. The goal is to learn a target function $f : \mathcal{U} \to \mathbb{R}^+$ that maps each user $u \in \mathcal{U}$ to a non-negative score that indicates how likely u's account has been compromised.

This chapter considers an unsupervised learning approach to the compromised account detection problem. The approach does not require labeled examples to be available, but can be susceptible to high false positive rate, similar to other security-related applications such as spam or malware detection. Instead of flagging all suspicious accounts as compromised, a more practical scenario is to present only the top-N most suspicious accounts to the domain experts or account owners for verification purposes.

Definition 5.2.2 (Top-N Compromised Account Detection)**.** Let $\mathcal{U}$ be the set of users and $\{X_1, X_2, \cdots, X_k\}$ be the corresponding data matrices for each of the k views. Let the target function $f : \mathcal{U} \to \mathbb{R}^+$ map each user $u \in \mathcal{U}$ to a non-negative score that indicates how likely u's account has been compromised. The goal of top-N compromised account detection is to rank the users using the target function f and select the top-N users whose accounts have most likely been compromised.

5.3 Multi-view Social Media Data

CADET combines multiple modalities of data at the user-level to detect compromised accounts. Specifically, four modalities of data are considered in this study: source, timing, location, and textual content. Other modalities can also be easily incorporated into the CADET framework.

5.3.1 Incorporating Source Information

Previous research has suggested that source information, i.e. the software used to post to social media, can help to detect malicious compromised accounts, such as those that are part of a social contagion [28]. For example, the previous chapter found that some hackers tweet from different sources than the original user. To evaluate the benefit of using source information, we examine the number of sources and which sources are used by 462 compromised and 5065 normal Twitter users, described in the previous chapter. The results shown in Table 5.1 indicate that 71.5% of users tweet from more than one source, and among them, 8.3% are compromised. The percentage is equivalent to the overall percent of users who are compromised, which suggests that relying on the number of sources used alone is not sufficient to accurately detect the compromised accounts. Analyzing the type of sources used, 12 of the 291 sources were found to be used more often by compromised users than normal users, with 10 of these sources being used solely by compromised users. This analysis suggests identifying the types of sources used may assist in compromised account detection.

Table 5.1: The frequency of compromised and not compromised users that tweet from either only one source or more than one source. The likelihood that a user is compromised if they tweet from multiple sources is the same as if all of their tweets are from only one sources.

	>1 Source	1 Source	
Compromised	328	134	462
Not Compromised	3626	1439	5065
	3954	1573	

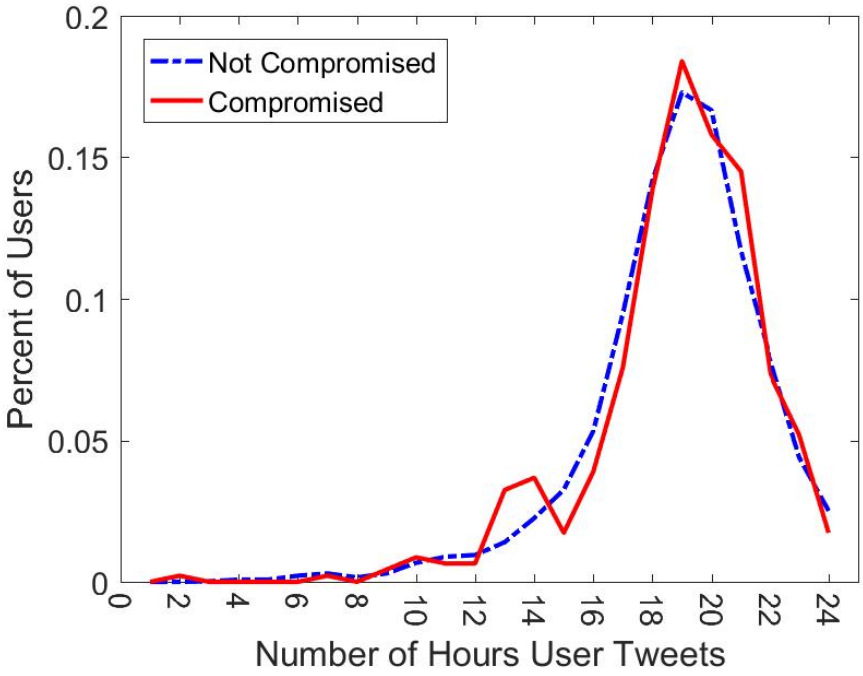


Figure 5.2: Percent of users compromised and not compromised by the number of distinct hours in the day they tweet. Most users tweet throughout the day instead of at the same time of the day. Compromised users tweet as often throughout the day as not compromised users.

5.3.2 Incorporating Timing Information

The time information, when users tweet, can be represented at different granularities: hour, day of the week, or day of the year. Similar to several previous works, this chapter considers tweet time at hour granularity [28, 68]. Analysis of the range of times, i.e. how many distinct hours per day during which users tweet, found that the majority of users, 89.3%, have tweeted at least once within 16 distinct hours of the day. This is expected assuming the average user

sleeps for 8 hours at the same time every day. The number of distinct hours compromised users tweet is similar to that of normal users, as shown in Figure 5.2. Thus, the number of hours the user tweets is not a clear indicator that they were compromised. However, analyzing the number of users who tweeted at each hour and what proportion of those users are compromised users, we found that the proportion of users who were compromised was the largest during late night and early morning hours, as shown in Figure 5.3. At 10:00am GMT, or 5:00am EST, 9.6% of the users tweeting were compromised users. During the day, the percent of users tweeting who were compromised was below 8.3%, the overall percent of users who were compromised. As will be explained in Section 5.5, all of the collected users have tweeted from within the United States, so most users are expected to be asleep at certain times. This is evident in Figure 5.4 which shows the first four principal components of the user-hour matrix. There is little activity at 10:00AM GMT, which suggests that tweeting at such abnormal times may indicate the account was compromised.

5.3.3 Incorporating Location Information

Twitter users have the option to include their location when they publish each tweet. This information can be represented at different granularities, e.g. latitude/longitude, city, state, country, etc. In this research, we use the Place field embedded in each tweet to locate the city and state where the tweet was posted. If the user disabled this feature, the Place value is set to Null. Our analysis suggests that all user accounts that had been compromised appear to have tweets posted from multiple places, as shown in Table 5.2. Among the 109 users whose tweets originate from only one place, none of them were compromised. This suggest that knowing the place the tweet originated from can help to detect compromised accounts.

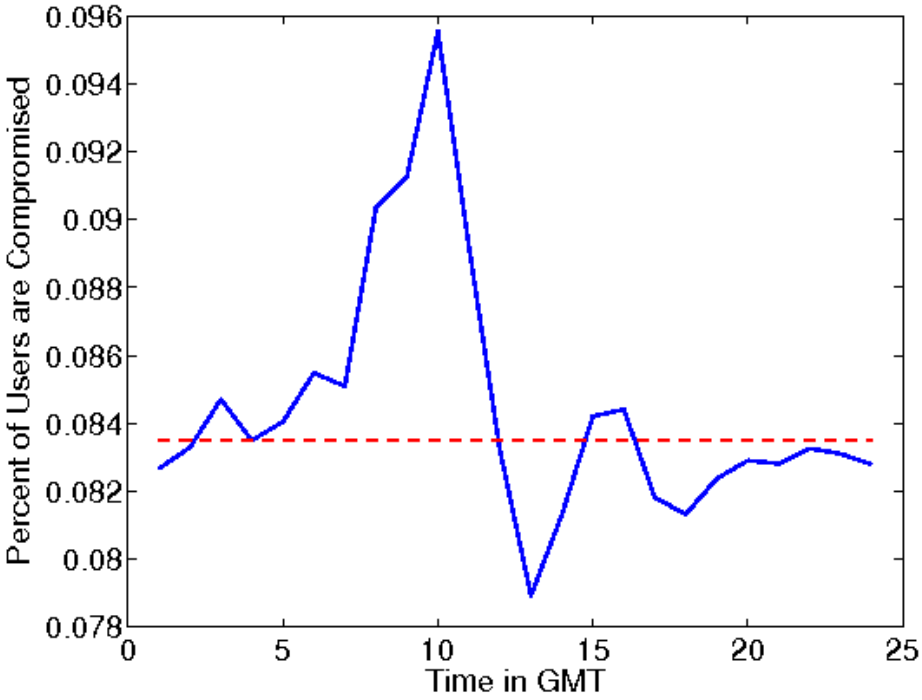


Figure 5.3: The proportion of users who tweet within each hour of the day who are compromised. All times are indicated in Greenwich Mean Time (GMT). At 10:00-11:00 am GMT, 9.6% of users who published tweets were compromised. Throughout the rest of the day, the percent of users who were compromised and tweeted in any given hour was similar to the overall percent of users who were compromised, i.e. 8.36%.

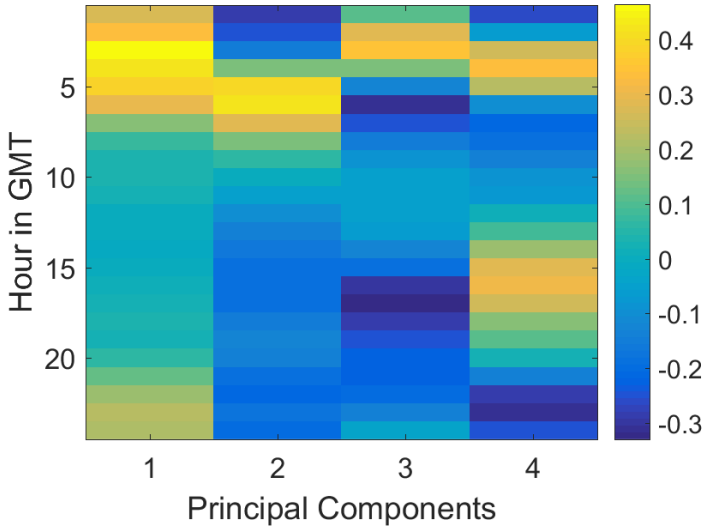


Figure 5.4: Top four principal components (PC) of the user-hour matrix. Hours are in Greenwich Mean Time (GMT). The first PC captures users who tweet in the late afternoon and early evening. The second PC are the users tweeting around midnight EST. Users who tweet only in the evening and never late at night nor in the morning appear in the third PC. The fourth PC are users who tweet primarily in the morning and late at night. This figure is best viewed in color.

Table 5.2: Number of places associated with the tweets posted from compromised and uncompromised user accounts. The likelihood that a user account is compromised is higher if the user's tweets originate from multiple places.

	>1 Place	1 Place	
Compromised	462	0	462
Not Compromised	4956	109	5065
	5418	109	

5.3.4 Incorporating Content Information

The simplest representation of content of tweets is by using the bag of words model. Analyzing the vocabulary of the tweets, we found the average vocabulary size for compromised users is 664, and for normal users is 694. Additionally among the 27,719 words used by compromised users, only 117 of them were not used by normal users. Thus, the textual content alone is insufficient to detect compromised accounts. However, it can still provide useful information about abnormal user behavior when combined with other data modality.

5.4 CADET: Unsupervised Compromised Account Detection Framework

CADET is an unsupervised multi-view learning framework for detecting compromised accounts on Twitter. Specifically, CADET considers each data modality as a view that provides partial information on whether an account has been compromised. For each view, the data is encoded into a lower dimensionality feature space and then reconstructed (decoded) to its original representation. The expectation is that users with normal behavior (not compromised) will exhibit reconstructed views that are similar to the original data, whereas compromised users will exhibit more deviance, i.e. larger reconstruction errors, due to some anomalous pattern, e.g. tweeting from a suspicious source. Therefore we use reconstruction error to measure how likely a user is compromised.

For each view v, the reconstruction error for a user u is computed as follows:

$$f_v(u) = \|\mathbf{x}_v^{(u)} - \Psi_v(\Phi_v(\mathbf{x}_v^{(u)}))\|^2, \tag{5.1}$$

where $\Phi_v(\cdot)$ denote the encoding function and $\Psi_v(\cdot)$ denote the decoding function. For example, using the principal component analysis approach [107], $\Phi(\mathbf{x}^{(u)}) = \Sigma^T \mathbf{x}^{(u)} \equiv \hat{\mathbf{x}}^{(u)}$, where Σ is a matrix of eigenvectors associated with the covariance matrix computed from the data matrix X and $\Psi(\hat{\mathbf{x}}^{(u)}) = \Sigma \hat{\mathbf{x}}^{(u)}$.

One way to extend the formulation to multi-view learning is to apply a weighted combination of the reconstruction errors from multiple views:

$$f(u) = \sum_v \alpha_v f_v(u) = \sum_v \alpha_v \|\mathbf{x}_v^{(u)} - \Psi_v(\Phi_v(\mathbf{x}_v^{(u)}))\|^2, \tag{5.2}$$

where α_v is the hyperparameter associated with the error in the v-th view. The limitation of the preceding formulation is that the reconstruction error for each modality, $f_v(u)$ may vary depending on the scale, dimensionality, and number of latent factors associated with each view of the data. This makes it difficult to determine an optimal set of hyperparameters $\{\alpha_v\}$ that can effectively detect the compromised accounts. Therefore, the key design issues in CADET are as follows:

1. How to design an appropriate reconstruction error function, $f(u)$, that is robust to the varying scale, dimensionality, and number of topics for different data modalities?

2. How to design flexible encoder ($\Phi(\cdot)$) and decoder ($\Psi(\cdot)$) functions for compromised account detection?

In the subsections below, we first describe the proposed approach used in CADET to address the first question. This will be followed by discussion on the nonlinear encoding method used to address the second question.

5.4.1 Multi-view Reconstruction Error

CADET learns a global low-rank encoding of the multi-view data to detect compromised accounts using a 2-level framework, as shown in Figure 5.5. The first layer provides an initial encoding of each view, $\Phi : \mathbb{R}^{d_v} \rightarrow \mathbb{R}^{m_v}$. This provides the flexibility needed to deal with the varying dimensionality, scale, and number of latent features associated with each modality of the data. For example, some of the views, e.g. places, have a large number of features that are correlated, e.g. some users may tweet from multiple nearby cities like San Francisco and San Jose, California. Similarly, a user could be tweeting mostly at 8am, but occasionally

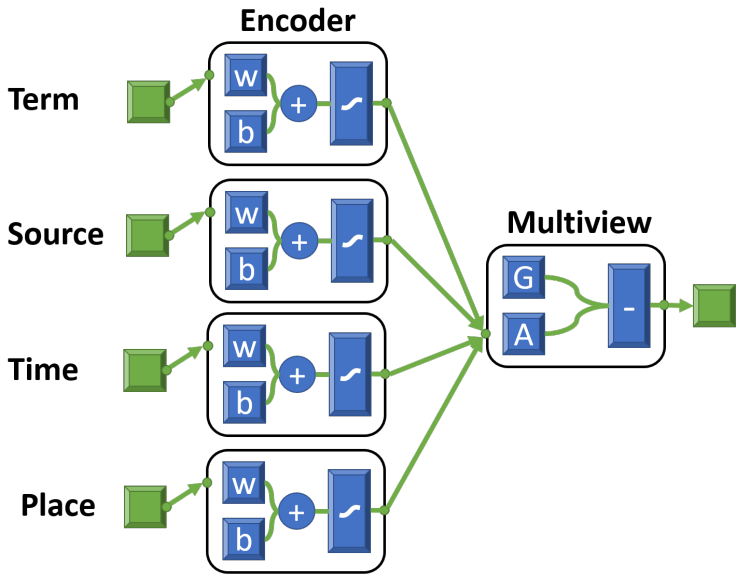


Figure 5.5: CADET Framework. CADET is a two-layer, multi-view learning framework. In the first level, each view is encoded independently to learn a lower-dimensional representation for each data modality. The second layer maps the encodings from multiple views to a shared latent space.

at 7am and 9am. The initial encoding reduces the dimensionality by mapping correlated features in a given view v to the same latent feature, $\Phi_v(X_v)$.

The second layer maps the set of latent features from each view, $\{\Phi_v\}$ to a shared latent subspace, G. There are two advantages for projecting the latent features to a common subspace. First, it alleviates the need to specify the optimal set of hyperparameter values α_v when combining the reconstruction error of different views, unlike the multi-view learning approach shown in Equation (5.2). This is because without any labeled examples available, there is no rationale for preferring one set of hyperparameter values over another. Second, the shared subspace provides a common basis for aggregating the reconstruction errors as the dimensionality of the matrices used are the same for all the views. Specifically, the multi-view reconstruction error provided by the second layer encoding can be written as

follows:

$$\min_{\{G,A_v\}} \sum_{v=1}^{k} \|G - \Phi_v(X_v)A_v\|_F^2 + \Omega(G), \tag{5.3}$$

where A_v is the transformation matrix for the v-th view, G is the latent features in the shared space, and $\Omega(G)$ is a regularization term. The A_v matrices allow each single-view encoding $\Phi_v(X_v)$ to have a different number of features.

The formulation in the second layer of CADET is quite similar to the generalized canonical correlation analysis (CCA) [18] approach for multi-view learning, except the shared latent space is learned by projecting the encoded features from the first layer. Furthermore, the formulation given in Equation (5.3) can accommodate different types of regularization, e.g., sparsity constraints, unlike the generalized CCA approach, which restricts G to be an orthogonal matrix. Nevertheless, the orthogonality constraint has the advantage of producing a closed form solution for learning G and A_i. Specifically, the objective function for CADET with orthogonality constraint becomes:

$$\min_{G,A} \sum_{v=1}^{K} ||G - \Phi_v(X_v)A_v||^2 + Tr[\Gamma(G^\top G - I)], \tag{5.4}$$

where Γ is assumed to be a diagonal matrix. Solving for G and A_v, we obtain:

$$G(I + \Gamma) = \frac{1}{K} \sum_{v=1}^{K} \Phi_v(X_v)A_v \tag{5.5}$$

$$A_v = (\Phi_v(X_v)^\top \Phi_v(X_v))^{-1} \Phi_v(X_v)^\top G \tag{5.6}$$

By substituting Equation (5.6) into Equation (5.5) yields the following equation:

$$G\Lambda = MG, \tag{5.7}$$

where $\Lambda = I + \Gamma$ and

$$M = \frac{1}{K} \sum_{i=1}^{K} \Phi_v(X_v)(\Phi_v(X_v)^\top \Phi_v(X_v))^{-1} \Phi_v(X_v)^\top$$

This reduces the problem of learning the shared multi-view representation G to an eigenvalue decomposition problem on M. To ensure G is low rank, only the eigenvectors from the top τ eigenvalues are used. τ is selected at the point where the eigenvalue curve begins to level out, i.e. the elbow in the curve. Once the shared latent subspace G is obtained, we can replace it into Equation (5.6) to learn the matrix A_v for each view. Finally, the reconstruction error for each user, u, is computed as follows:

$$f(u) = \sum_{v} \|g_u - \Phi_v(X_v^{(u)})\|^2,$$

where g_u corresponds to the u-th row of matrix G.

5.4.2 Single View Encodings

The first layer of CADET uses a nonlinear autoencoder, specifically a multi-layer perceptron with one hidden layer trained to learn a lower-dimensional encoding of the input features, as shown in Figure 5.6. Unlike NMF and PCA, it performs a nonlinear transformation of the original features by employing an activation function such as the sigmoid function. The nonlinear autoencoder is trained to minimize the reconstruction error of its input data. We used Matlab's implementation of autoencoder, where the weights of the network are trained using the scaled conjugate gradient algorithm, with L2 and sparsity regularizers. Each autoencoder is trained for 800 epochs. The activation function used is logistic sigmoid.

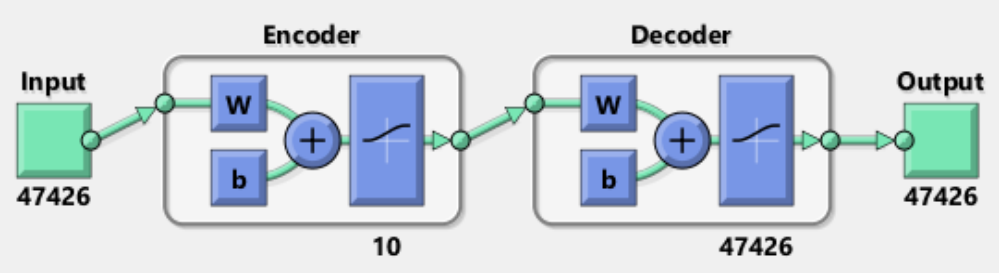


Figure 5.6: A nonlinear autoencoder that learns a nonlinear, lower-dimensional embedding of an input data matrix.

5.5 Experimental Evaluation

This section presents the datasets used in our experiments along with the experimental setup and results obtained to demonstrate the effectiveness of CADET.

5.5.1 Data Collection

This experiment used the same dataset described in the previous chapter. The entire dataset contained 5889 users, with 484 of them identified as compromised. However to ensure we had a sufficient number of previous tweets from each user to identify their normal behavior, users with fewer than 150 tweets were removed. Our final dataset contains 462 compromised accounts and 5065 normal (i.e. not compromised) accounts.

5.5.2 Experimental Setup

We compared the performance of CADET against the following baseline algorithms:

- **Distance-based**: In this approach, we compute the cosine similarity between all pairs of users for each view. For each user, we select their average distance (i.e., 1 - average similarity) to other users to be the outlier score for the view. The users are then ranked based on their average outlier score across all four views.

- **PCA** [107]: This approach applies principal component analysis (PCA) to the data matrix and uses its reconstruction error to determine whether an account is compromised. We consider two ways to extend PCA to our multi-modal Twitter data. The first approach, termed **PCA-1** [107], simply concatenates the multi-modal features into a single feature vector before applying PCA. The second approach, termed **PCA-2**, applies PCA on each view separately and then combines their reconstruction errors, assuming the error from each view is weighted equally.

- **Multi-view NMF** [4]: In this approach, we extended the approach in [4] from 2 to 4 views and added an orthogonality constraint on each of the topic-feature matrices. The reconstruction errors from each view are then combined. Specifically, the objective function for the multi-view NMF is:

$$\begin{aligned} \min_{V,H,K,L,M \geq 0} & ||X_1 - VH||_F^2 + ||X_2 - VK||_F^2 \\ & + ||X_3 - VM||_F^2 + ||X_4 - VL||_F^2 \\ s.t. & HH^\top = I, KK^\top = I, MM^\top = I, LL^\top = I \end{aligned} \tag{5.8}$$

- **COMPA** [28]: This supervised approach creates a user's behavior profile from different types of tweet features. Next COMPA trains a classifier on the anomaly scores of the withheld tweets to detect anomalous tweets, i.e. those that do not belong to the profile. A user's anomaly score is given by the maximum of their tweet anomaly scores.

For fair comparison, we set the number of latent factors associated with each view to be 4 for all of the approaches (PCA, NMF, and CADET). Since PCA-1 is the concatenation of four views, the number of components used is the sum of the number of components found for each view. For CADET, the number of latent features in the multiview layer was

selected as the inflection point in the eigenvalue curve of the G matrix. For evaluation purposes, we rank all the users by their reconstruction error and measure their performance in terms of their precision at top-K% and their percentage improved from random of the area under the precision-recall curve (AU-PR). Specifically, precision at top-K (P@K) measures the proportion of the top K% of users who are compromised. A large P@K signifies that the highest ranked users are true compromised accounts. We choose P@K as it is a commonly used measure for security related applications, in which the top-K most suspicious accounts are presented to domain experts or account holders for verification purposes. For our experiments, P@K is evaluated at top 0.1%, 0.2%, 0.5%, 1%, 3%, 5%, 8% and 10% users.

5.5.3 Experimental Results

In our experiments, we first investigate the benefit of using a two-level framework in CADET to detect compromised accounts. We then compare the performance of CADET against the baseline approaches.

5.5.3.1 Single View Encoding

To determine the number of latent features to be used, we applied PCA to each of the four data views—place, source, hour, and terms—and selected the number of latent features based on their proportion of variance explained, which correspond to the elbow of the eigenvalue curve, as shown in Figure 5.7. We found that the best performance for PCA on each view was at 4 latent features. For a fair comparison, we also set the number of latent factors to 4 for NMF and nonlinear autoencoders.

To demonstrate the value of using multi-modal data, we compared the performance of CADET against the results of using reconstruction error for each view. From Figure 5.8,

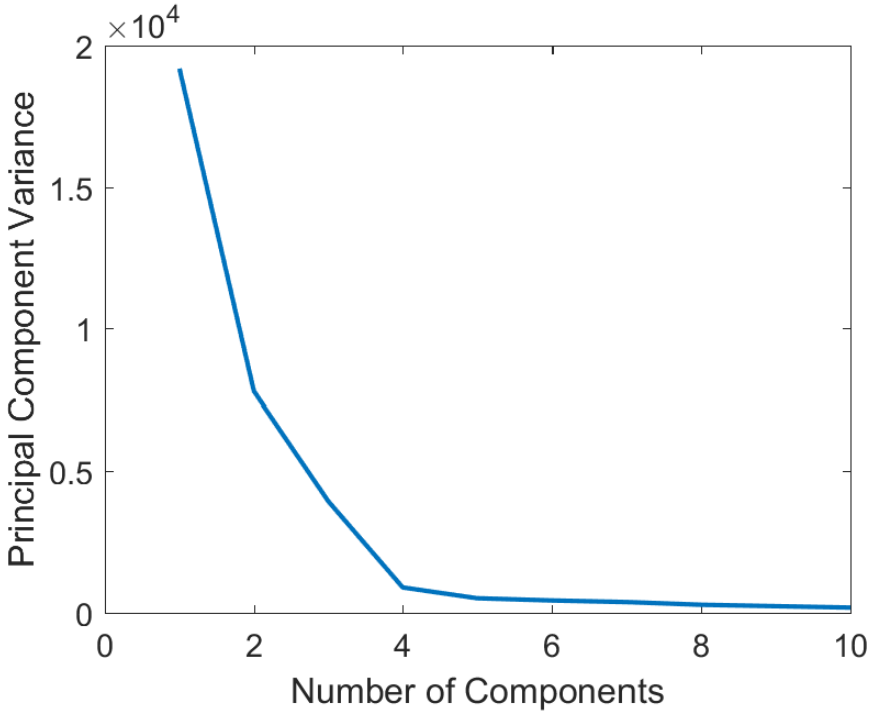


Figure 5.7: Variance reduced by the addition of each principal component of the user-source matrix. The inflection point occurs at 4 principal components.

it is evident that the time users post and their location are good detectors of compromised accounts. To determine whether a simple linear combination of the views is sufficient, each reconstruction error was normalized by the number of features of its respective view, and the vectors were averaged across views, denoted as View Avg in Figure 5.8. The results show that simple linear averaging of features from multiple views may not necessarily improve performance over single view encoding. However, by projecting the multiple views into a shared subspace, CADET is able to detect more compromised accounts among its highest ranked users compared to any of the single-view autoencoders.

5.5.3.2 Comparison against Baseline Algorithms

This section compares the performance of CADET against several baseline algorithms. Our results demonstrate that CADET is better at detecting compromised accounts compared to

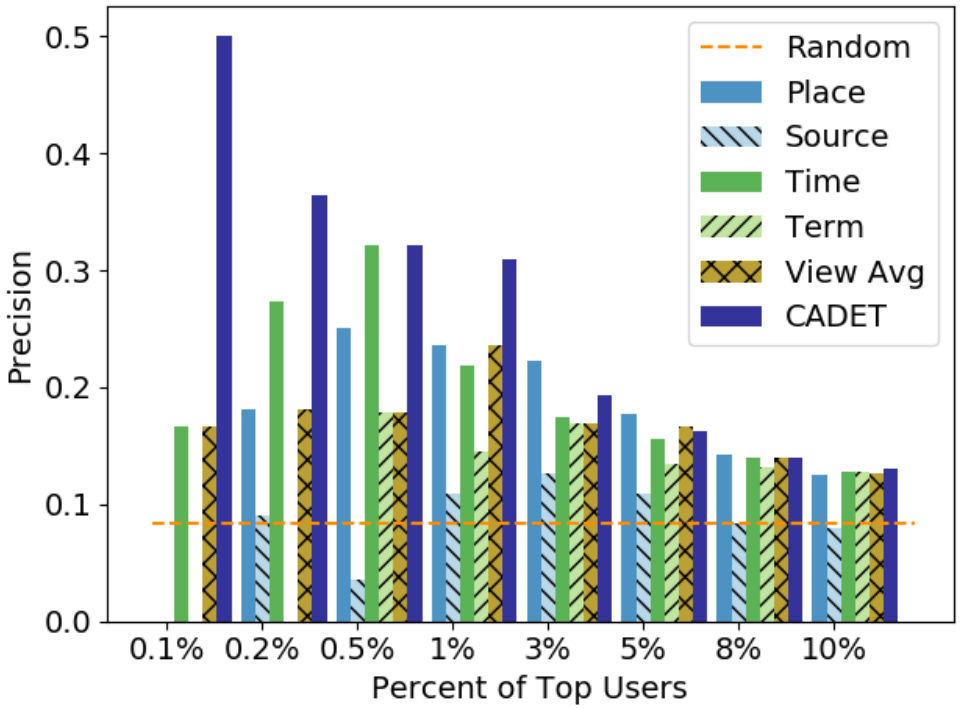


Figure 5.8: Performance comparison of single-view autoencoders, combining view scores via averaging, and CADET. CADET achieves the highest precision for the top 0.1%, 0.2%, 0.5% and 1% of users. Place and Time views alone achieved good precision. Using all four views alone, via averaging view scores, was insufficient. However by projecting each view into a shared space, CADET achieved higher precision.

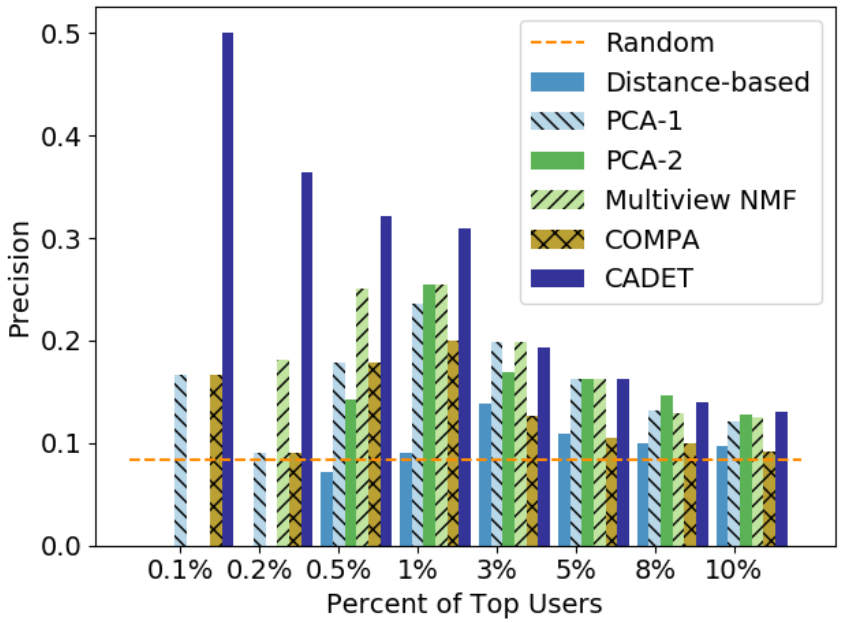


Figure 5.9: Performance comparison of multi-view algorithms, Distance-based, PCA-1, PCA-2, Multi-view NMF, COMPA, and CADET. CADET achieves the highest precision for the top 0.1%, 0.2%, 0.5% and 1% of users. For more users, all five algorithms had similar precision.

the other algorithms. From evaluating the performance using on top k% precision, CADET, shown in Figure 5.9, is able to identify more compromised users among its highest ranked users compared to other methods. For example, among the top 0.2% of users, which consisted of 11 users in total, CADET correctly identified 4 compromised users whereas PCA-1 and COMPA identified only 1 compromised user and Multi-view NMF found 2 compromised users. CADET consistently achieved the highest precision within the top 1% of users. However, as the number of users increased, all of the algorithms except Distance-based saw decreased performance due to an increase in the number of false positives. Furthermore, the precision achieved by CADET is consistently better than random guessing (0.0835) unlike some of the other baseline algorithms. Surprisingly COMPA, which is supervised, had significantly lower precision than our approach. COMPA was designed to detect groups of accounts with the same type of anomalous tweets at the same time [28]. As a result, the

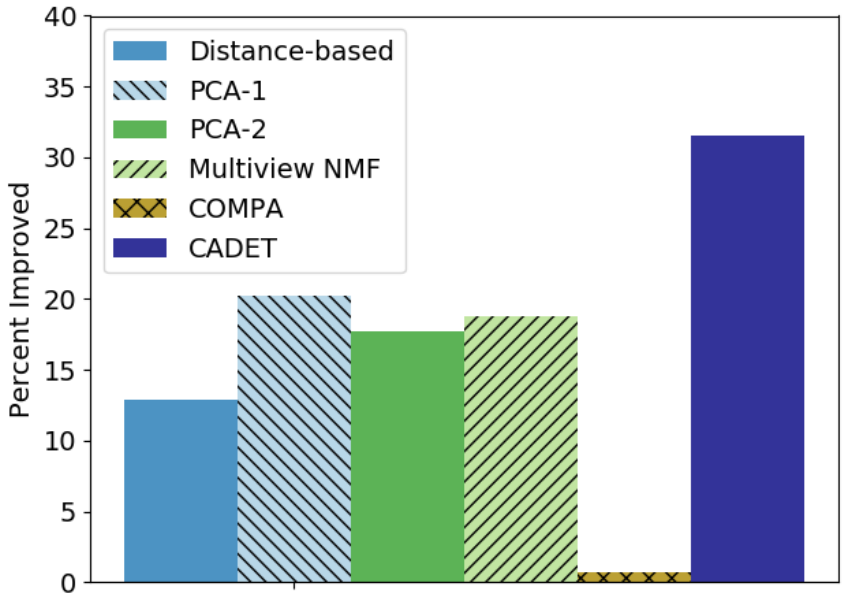


Figure 5.10: Comparison of percent improved from random guessing in terms of Area under Precision-Recall curve (AU-PR) for CADET against the other baseline methods.

types of anomalous tweets it detects tend to be spam tweets. However in this dataset, only about 15% of the compromised tweets were spam [103].

Additionally these algorithms were evaluated based on their precision-recall curves. The percent improved of the area under the precision-recall curve (AU-PR), shown in Figure 5.10, demonstrates that overall, CADET has the largest AU-PR improvement followed closely by PCA-1. Furthermore, by analyzing the precision-recall curve shown in Figure 5.11, it is evident that CADET's higher AU-PR is a result of achieving significantly higher precision at low recall rates compared to PCA-1 and other baseline algorithms. The heavy tail of the precision-recall curves are unavoidable due to the unsupervised nature of their learning algorithms. Even supervised learning, i.e. COMPA, also suffered from the heavy tail in the precision-recall curve. As the number of users increases, all 6 algorithms are susceptible to more false positives. However, their overall performance is significantly better than random guessing.

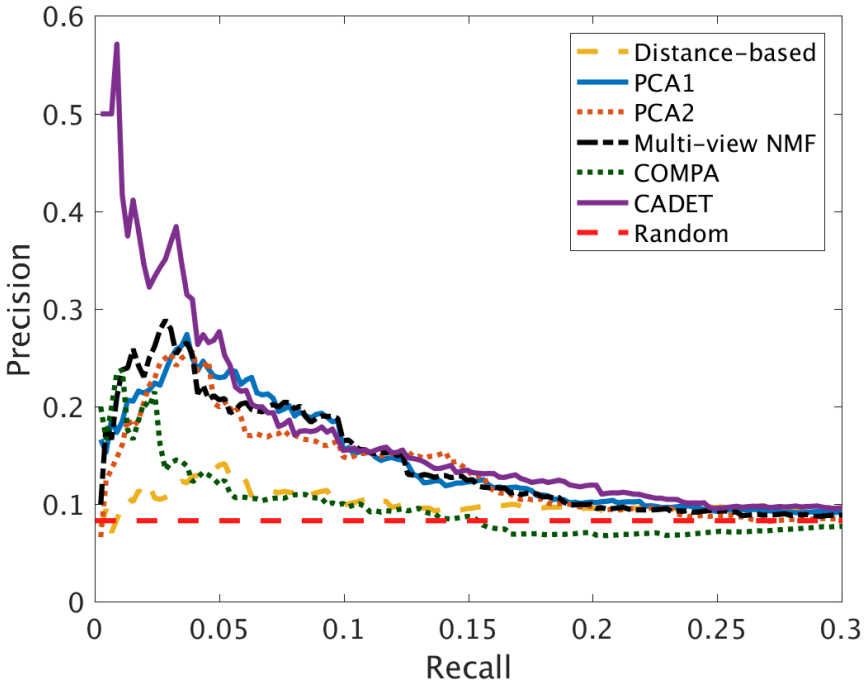


Figure 5.11: Performance comparison based on the Precision-Recall curves of multi-view algorithms, PCA-1, PCA-2, Multi-view NMF, COMPA, and CADET. CADET achieves higher precision than the other methods for recall under 5%.

5.6 Conclusion

This chapter presented CADET, an unsupervised multi-view framework for compromised account detection. CADET combined nonlinear encodings from multiple views to learn a user-encoding, using a variant of generalized canonical correlation analysis. CADET was able to detect compromised accounts with higher precision and higher area under the precision-recall curve than several existing approaches for compromised account detection. In the next chapter, I present a framework for detecting compromised tweets, which may identify when an account was first compromised. I explore a more complex deep learning approach to improve compromised account detection performance.

Chapter 6

Compromised Tweet Detection using Deep Learning

Compromised accounts are often used to spread misleading information, including spam and false information. Victims of compromised accounts may lose their friends [96], or worse still, if a reputable account is compromised, such as a news agency's Twitter account, the hackers' posts can lead to public panic and volatile fluctuations of the stock market [72]. Thus, accurate and early detection of compromised accounts is essential to protect social media users from such malicious threats and mitigate their potential damages. Early detection means identifying compromised accounts at the post level. This chapter proposes an algorithm to detect compromised posts, i.e. the hackers' posts.

Compromised post detection is a challenging problem for several reasons. First, a compromised post may resemble a normal post for another user. For example, a hacker may lie about the user's sexuality to spread a rumor, but such posts are not unusual for some users who discuss their sexuality openly on social media. Detecting compromised posts is therefore more difficult compared to other tasks such as spam detection as spam posts tend to have features that are be more easily distinguishable from the regular posts of normal users. Although there has been previous research to address this challenge by learning the behavioral profile of each user [28, 68, 97], they require a significantly large sample size to

build a reliable user representation. If the number of tweets used to generate the profile is too low, it will flag too many false alarms. If the number of tweets needed is too high, then it cannot be effectively used for early detection. In addition, it is computationally expensive to generate a profile for every user.

Second, the raw features of the social media data (e.g., the text messages and other meta-features of user tweets) are often sparse and noisy. For example, tweet messages tend to be short, i.e., 140 characters long, and contain typos and other non-standard lexical variations, making it difficult to effectively use them for training a robust compromised account detection model. While there has been previous research focusing on deriving reliable feature representation for detecting compromised accounts, they are mostly limited to linear (e.g., using principal component analysis [107]) and unsupervised [102] learning methods. As a result, the derived features may not be optimal for compromised post detection purposes.

To address these challenges, I propose CAUTE (Compromised Account User Tweet Encoder), a deep learning framework that simultaneously learns the nonlinear embeddings of users and their posts, and detects whether a post is compromised. CAUTE considers both lexical and meta features of a tweet to determine whether it was posted by the genuine account holder or a hacker. It accomplishes this by learning a pair of encoders to transform the raw features into more informative features while reducing their sparsity and noise. The first encoder, *tweet2user*, learns a latent representation to help transform the tweet features into user features while the second encoder, *user2tweet*, learns a representation that helps predict the content of a tweet from the user's features and tweet's meta-features. The hypothesis is that if a user is indeed the author of a post, then the errors associated with the tweet2user and user2tweet encoders for a given (user, tweet) input pair are expected to be low. Otherwise, the errors are likely to be high if the tweet was composed by another user

(hacker). Instead of applying some arbitrary threshold, the residual errors of the encoders are fed into a fully-connected neural network layer, *res2class*, to predict whether the post is compromised for that given user. In principle, since the user features can be derived from the user's profile, CAUTE is applicable even in a cold start scenario, when the user has not posted any tweets, unlike other existing methods. However, unless the user profile includes meaningful information about the topics of interest to the user, it may not be sufficient to train a robust model. To enhance CAUTE's performance, the user features can also be derived from a small set of their initial tweets. Unlike other approaches that require a large training set to learn a reliable profile of the users, our empirical results suggest that CAUTE can effectively identify compromised posts after observing only as few as 10 of their initial tweets.

The main contributions of this chapter are as follows:

- Propose CAUTE, a deep learning framework that can detect compromised posts by modeling the user and post features simultaneously.
- Show that the nonlinear embeddings derived by the tweet2user and user2tweet encoders are informative predictors of compromised posts.
- Demonstrate that CAUTE outperforms state-of-the-art baseline algorithms in terms of their accuracy as well as their ability for early detection without generating a large number of false alarms.

The remainder of this chapter is as follows. The compromised post detection problem is formally defined in Section 6.1. Section 6.2 describes the CAUTE framework. Experimental results are presented in Section 6.3, followed by our conclusions in Section 6.4.

Figure 6.1: A typical attack scenario of compromised account on Twitter. The original user will tweet before and after their account has been compromised, denoted as normal posts. When hackers take control of the account, they will publish tweets, i.e. compromised posts. When the user realizes the account was compromised, they will alert their followers of the compromise in an announcement post.

6.1 Problem Statement

Let $\mathcal{U} = \{u_1, u_2, \cdots, u_N\}$ be the set of social media users and $\mathcal{T} = \{T^{(1)}, T^{(2)}, \cdots, T^{(N)}$ be the set of all postings, where each $T^{(i)} = \{t_1^{(i)}, t_2^{(i)}, \cdots, t_{m_i}^{(i)}\}$ is the set of posts associated with user u_i's account. A user's account is compromised when an unauthorized person, i.e. hacker, gains access to the user's account and perform some actions, e.g. posting, from that account without the user's consent. We denote $y(t_j^{(i)})$ as the genuine author of the j-th post from user u_i's account.

Definition 6.1.1. Compromised Post: A post, $t^{(i)}$, from user u_i's account is said to be compromised if it was written by another user, i.e., $y(t^{(i)}) \neq u_i$.

Definition 6.1.2. Compromised Account: The social media account of user u_i, is said to be compromised if it is associated with at least one compromised post, i.e., $\exists t^{(i)} : y(t^{(i)}) \neq u_i$.

As proof of concept, this research focuses on compromised Twitter accounts, where each post corresponds to a user's tweet. The proposed framework can be easily generalized to other social media platforms or to other types of hackers' actions, e.g. liking or browsing.

Figure 6.1 demonstrates a typical attack scenario of compromised accounts on Twitter. The genuine user initially publishes a series of tweets. When the account is compromised, the hacker will publish one or more tweets, shown in dark orange in Figure 6.1. When the original user discovers the account was compromised, he or she will perform actions to prevent the hacker from further posting, e.g., by changing password, and post an *announcement post*, shown in black, to inform their followers that the hackers' tweets were not written by the original user.

In this chapter, we cast the compromised post detection problem as a binary classification task. Specifically, we classify each user-tweet pair, $(u_i, t^{(i)})$, either as positive class if $t^{(i)}$ is a compromised post or negative class if it is a genuine post. To predict the user-tweet pair, let $\mathbf{x}_t^{(i)} \in \mathbb{R}^d$ be the set of features associated with the post $t^{(i)}$ and $\mathbf{z}^{(i)} \in \mathbb{R}^p$ be the set of features associated with the user u_i. We further categorize the tweet features $\mathbf{x}_t^{(i)}$ into two types: *content* or *meta* tweet features. Content-related tweet features include the text message itself, hashtags, mentions, and URLs whereas tweet meta features include the location, source (application), and language used by the user to post the tweet. The tweet meta features are expected to have lower variability since a user will likely tweet from a limited number of applications (e.g., from a Web browser or via their smartphone app) and locations, usually in the same language. In contrast, content features such as the terms used, hashtags, and mentions may vary from one tweet to another. Such distinction between content and tweet meta features will be utilized by our proposed CAUTE framework. Finally, the user features $\mathbf{z}^{(i)}$ can be derived from the user profile or based on the user's initial posting behavior. The latter can be obtained by extracting features from the first k% tweets associated with the user. This is similar to the approach used in COMPA [28, 29].

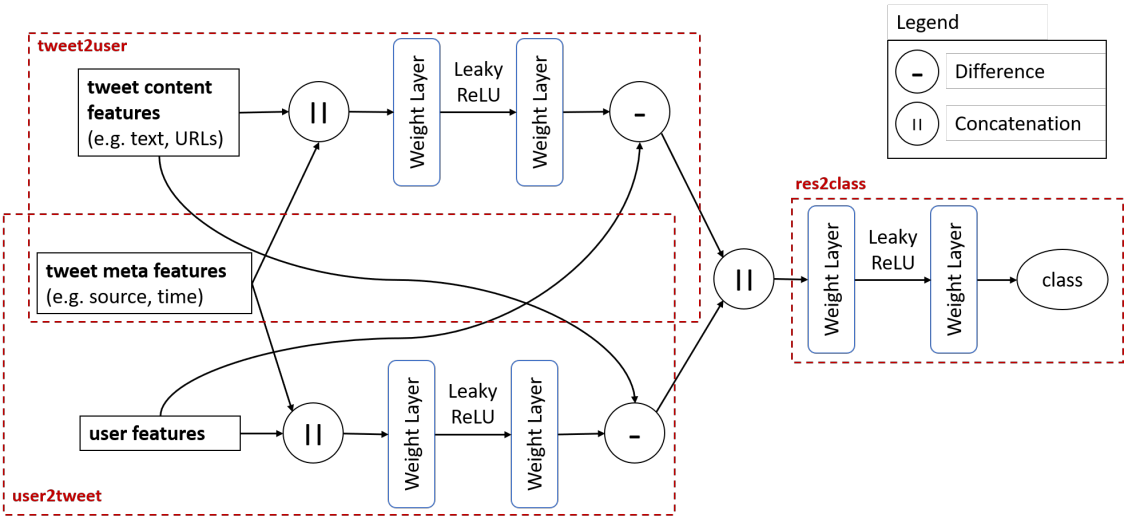


Figure 6.2: CAUTE Framework.

6.2 Proposed Framework

CAUTE is a 3-component neural network architecture that simultaneously learns the feature embedding of a given user-tweet pair (u, t) and uses the residual error of the embedding to determine the likelihood that user u is not the author of tweet t. A high likelihood would suggest that the tweet has a high probability of being a compromised post.

As noted in the Introduction section, the raw user and tweet features, which are typically represented using one-hot encoding, are often too sparse to be effectively used for detecting compromised posts and accounts. To address this challenge, the first two components of our architecture, *tweet2user* and *user2tweet*, are feature encoders designed to learn nonlinear embeddings of the raw features, which are then used to approximate the user and tweet features. The third component, *res2class*, uses the residual errors from the tweet2user and user2tweet encoders to predict whether the tweet was written by the user. The rationale behind our proposed framework is as follows. If user u is the author of tweet t, then the tweet features $\mathbf{x}_t$ can be used to predict the user features $\mathbf{z}$, and vice-versa. If the user is not the author of the tweet, then the residual errors of the predictions are likely to be large.

A high-level schematic illustration of the CAUTE framework is shown in Figure 6.2.

6.2.1 tweet2user Encoder

The goal of the tweet2user encoder is to learn a nonlinear feature embedding of the tweet that can be used to predict the user features. One potential challenge to learning the embedding is that tweets with similar content can be posted by more than one user. To address this challenge, instead of learning the embedding from the tweet content features alone (e.g., the terms in the tweet), CAUTE also considers the tweet meta features to help identify the user who authored the tweet. The latent features derived by the tweet2user encoder thus represent not only the topics pertaining to the tweet content but also some information about the user who posted the tweet.

The tweet2user encoder is a feed-forward neural network, shown in the top left portion in Figure 6.2. Specifically, given a user-tweet pair, $(u_i, t^{(i)})$, with the corresponding user feature $\mathbf{z}^{(i)} \in \mathbb{R}^p$ and tweet feature $\mathbf{x}_t^{(i)} \in \mathbb{R}^d$, the network is trained to learn a pair of mapping functions $g_1 : \mathbb{R}^d \to \mathbb{R}^k$ and $g_2 : \mathbb{R}^k \to \mathbb{R}^p$ such that the following residual error:

$$\|g_2(g_1(\mathbf{x}_t^{(i)})) - \mathbf{z}^{(i)}\|^2 \tag{6.1}$$

is minimized for all user-tweet pairs in the training data. The mapping function g_1 is implemented using a Leaky ReLU applied to the output of the first weight layer shown in Figure 6.2 while the mapping function g_2 is implemented using a linear activation function applied to the output of the second weight layer. The output of the Leaky ReLU unit, $g_1(\mathbf{x}_t^{(i)})$, thus provides a nonlinear embedding of the tweet features.

The input of the tweet2user encoder is a concatenation of the tweet content features

($\mathbf{x}_{t,c}^{(i)}$) and tweet meta features ($\mathbf{x}_{t,u}^{(i)}$), i.e., $\mathbf{x}_t^{(i)} = [\mathbf{x}_{t,c}^{(i)}, \mathbf{x}_{t,u}^{(i)}]$. The output of tweet2user is trained to approximate the user features $\mathbf{z}^{(i)}$. How the user is represented, i.e. its raw features, can affect the user encoding accuracy of tweet2user. Ideally, the features should be unique for each user, i.e., no two users should have the same feature representation. A simple representation of users is a 1-hot encoded feature vector of length N, corresponding to the total number of users. However as social networks are continuously growing, this fixed-length feature vector would prevent CAUTE from detecting compromised posts of the users who are not in the training set.

6.2.2 user2tweet Encoder

As multiple users can post similar type of tweets, it is possible that the tweet2user encoder would predict the features of one user when the tweet is posted by a different user. To boost our confidence that a user is indeed the author of a tweet, a user2tweet encoder is simultaneously trained to recognize the type of tweets posted by the user. Since a user can post multiple tweets with different content, it is insufficient to use the user features alone as input to the user2tweet encoder as the encoder will always produce the same output given the same input features. In this case, the embedded features are likely to be the average feature representation of all the tweets by that same user. To overcome this limitation, the user2tweet encoder in CAUTE leverages the tweets' meta features to identify their content features. Specifically, the meta tweet features are concatenated with the user features before being provided as input of the user2tweet encoder, as shown in Figure 6.2. Formally, given a user-tweet pair, $(u_i, t^{(i)})$, with the corresponding user feature $\mathbf{z}^{(i)} \in \mathbb{R}^p$, tweet meta feature $\mathbf{x}_{t,u}^{(i)} \in \mathbb{R}^{d_1}$ and tweet content feature $\mathbf{x}_{t,c}^{(i)} \in \mathbb{R}^{d_2}$, the encoder is trained to learn a pair of mapping functions $h_1 : \mathbb{R}^{p+d_1} \rightarrow \mathbb{R}^k$ and $h_2 : \mathbb{R}^k \rightarrow \mathbb{R}^{d_2}$ such that the following residual

error:

$$\|h_2(h_1(\mathbf{z}^{(i)}, \mathbf{x}_{t,u}^{(i)})) - \mathbf{x}_{t,c}^{(i)}\|^2 \tag{6.2}$$

is minimized for all user-tweet pairs. By optimizing the error function, the approximation will be close to the actual tweets' content features if the tweet is authored by the selected user. However if the tweets are authored by a different user, then their residual errors are likely to be large.

6.2.3 res2class Classifier

The tweet2user and user2tweet encoders provide approximations of the user and tweet features respectively. The third component of CAUTE, res2class, takes the residual errors of the two encoders as input to predict whether a tweet was authored by the corresponding user. The output of res2class is a positive class if the tweet was written by the user, and negative class if the tweet was written by a different user.

Given a user-tweet pair, $(u_i, t^{(i)})$, let $\Delta_{t2u}(u_i, t^{(i)}) = g_2(g_1(\mathbf{x}_t^{(i)})) - \mathbf{z}^{(i)} \in \mathbb{R}^p$ be the element-wise residual error of tweet2user and $\Delta_{u2t}(u_i, t^{(i)}) = h_2(h_1(\mathbf{z}^{(i)}, \mathbf{x}_{t,u}^{(i)})) - \mathbf{x}_{t,c}^{(i)} \in \mathbb{R}^{d2}$ be the corresponding element-wise residual error of user2tweet. The residuals from the tweet2user and user2tweet encoders are concatenated and provided to a feed forward neural network with Leaky ReLu activation function. A cross entropy loss function was used to train the res2class classifier, enabling the output of res2class to represent the likelihood of the class.

6.3 Experimental Evaluation

This section describes the experiments conducted to evaluate the performance of CAUTE and other state-of-the-art algorithms for compromised account detection on a real-world Twitter dataset. We first describe dataset collected along with the features used to describe the users and their posts. We also discuss our approach for training the network before presenting the experimental results and discussion.

6.3.1 Data

We employed the Twitter streaming API to download the tweets posted between between April 27, 2015 and May 6, 2015. We extracted the IDs of users who posted the tweets and collected the 200 most recent tweets[1] by each user. As noted in [103], identifying compromised posts is a challenging annotation task. Following the approach used in Trang et al. [97], we artificially inserted compromised posts into the data by swapping posts from one user and assigning them to another user. With this approach, we created a dataset that contains tweets from 5524 users, where each user has posted, on average, about 173 'genuine' tweets. In addition, we artificially injected 957,392 'compromised' tweets by assigning each tweet to a random user. The entire dataset thus contains 1,917,342 (user, tweet) pairs with 50.1% pairs are genuine posts and the remaining 49.9% pairs are compromised.

Learning the encoding from a tweet's representation to its respective user's representation and vice versa is heavily reliant on the features used. Instead of manually identifying the discriminative features to be used for compromised account detection, CAUTE is designed to automatically learn the informative features using the tweet2user and user2tweet encoders.

[1]Although we requested for 200 tweets, the number of actual tweets returned may vary since the user may delete some of their tweets before they were collected

For each tweet, we consider the following raw tweet features:

- Content features:
  - Hashtags: hashtags that appear in the body of the tweet.
  - Mentions: which users were mentioned in the tweet.
  - URLs: domain name of any URLs in the tweet.
  - Text: terms that appear in the tweet.
- Meta features:
  - Time of Day: when was the tweet published.
  - Language: the language in which the tweet was written.
  - Source: the application used to post the tweet.

We used one-hot-encoding to represent each feature except for text. To reduce the sparsity of text features, we applied Singular Value Decomposition (SVD) with the number of components set to 30. To select this component set, we varied the number of components from 10 to 100 and measured the average pairwise distance between users and selected the number of components that produced a large pairwise distance.

As previously noted in the Introduction section, in principle, CAUTE is applicable even in a cold-start scenario unlike other existing methods, e.g., by using the user profile information to create the user features. By applying CAUTE on the user and tweet features, we can detect whether the tweet is likely to be posted by the user. However, in practice, the detection performance under the cold-start scenario tends to be poor unless the user features are indicative of the type of tweets posted (e.g., if the user profile features include the topics

of interest to the user). In our experiments, we reserve a subset of the initial posts made by each user to define the user features. Specifically, we applied SVD on the initial tweets and used their derived components as user features. The subset of tweets will be excluded from the dataset used to train the compromised detection model.

To ensure fairness in experimental evaluation, we reserved the same subset of the tweets made by each user to learn the behavioral profile of the users for all the baseline algorithms considered in this study. We explored two percentage values, 5% and 10% of the 200 tweets for each user, as our initial subset. Ideally, only a small percentage of tweets would be available to learn a user representation, even though having more tweets used to generate a profile would lead to better predictions of compromised posts.

6.3.2 Training a Neural Network

Neural network training is challenging due to the numerous hyperparameters that need to be tuned, including batch size and number of epochs. If the batch size is too large, the neural network over-generalizes the patterns learned, whereas a batch size that is too small leads to longer training time. We tested batch sizes of powers of 2 up to 2048 tweets and found that user-specific embeddings emerge when the batch size is either 16 or 32 tweets, without suffering a noticeable increase in training time. For these experiments, the tweet2user and user2tweet encoders were trained using batch size of 16 tweets. The res2class encoder focused on learning more general patterns, so batch size was set to 128 tweets.

In order for a neural network to be generalizable to unseen users, a sufficiently large sample of users is needed for training. However training on more users leads to a longer training time. To overcome this challenge, users were split equally into three sets for training, validation, and testing. Using all tweets from the users in the training set could still lead

to long training time. However training on a random sample of their tweets could be just as effective as training on all of a users' tweets. After removing the tweets used to generate the user features, we selected 40% of the users' tweets to train the network, because it had similar performance to using 100% of the tweets. The model was evaluated on the withheld tweets from the training set users to determine when to stop training the neural network. If the loss increased by 0.1% for these tweets, training was terminated.

The datasets and their purposes are denoted as follows:

- Train: Approximately 1/3 of users and 40% of their tweets were used to train the neural networks. These tweets were also used to perform SVD in order to obtain the user features or their behavioral profile. The remaining 40% of tweets from these users determined how many epochs to train each neural network

- Validation: Approximately 1/3 of users that do not appear in the training set. This dataset was used for hyperparameter tuning, e.g. number of nodes in the neural network.

- Test: Remaining 1/3 of users. The (user, tweet) pairs for these users are used for evaluating the performance of the various algorithms.

For the validation and test set, the initial subset of tweets used to generate the SVD-based user feature vector were excluded. For training and validation sets, instances of compromised posts were created by matching each tweet with a random user that was not their respective user. That random user, assigned to be the hacker, was another user in the training or validation set respectively. For the test set, instances of compromised posts were created by matching all the tweets of a user in the test set to the same random user in the test set. Thus, for each user u_i in the test set, the tweets associated with u_i include all of their

original tweets ('genuine' posts) as well as all the tweets from another user in the test set u_j ('compromised' posts) where $i \neq j$.

6.3.3 Experimental Results

In this subsection, we present the results of our experiments. In particular, we design the experiments to answer the following questions:

- How good is the feature embeddings learned by CAUTE?
- How well does CAUTE perform compared to existing methods for compromised post detection?
- What value does each component of CAUTE contribute to its overall performance?

6.3.3.1 Evaluating the Utility of CAUTE's Latent Features

The first research question is to evaluate the efficacy of using the features learned by CAUTE to detect compromised posts. Specifically, we use the feature embeddings learned from tweet2user and user2tweet encoders to train a classifier for predicting compromised posts. To do this, we first create a dataset containing a set of (user, tweet) pairs, where the user features correspond to the latent features of the user2tweet encoder while the tweet features are the latent features of the tweet2user encoder. Each (user, tweet) pair is assigned a class label of +1 if the tweet is a compromised post or -1 if the tweet is posted by the user. We evaluated the performance of CAUTE's latent features using both logistic regression and random forest as our classifiers.

We compare the performance of CAUTE's latent features against those generated by the following baselines:

Table 6.1: Comparison of feature representation methods. Logistic regression and random forest classifiers are applied to detect compromised posts using the user and tweet features obtained from various methods. Results shown are evaluated in terms of AUC score for compromised posts.

Method	Logistic Regression	Random Forest
Raw Features	0.5036	0.5000
Doc2vec	0.5017	0.5353
SVD	0.5012	0.5022
Tweet2User & User2Tweet Features	**0.5964**	**0.5708**

- Raw features: In this approach, the user features are the same as the original user features provided as input to CAUTE (i.e., SVD applied to the text features of a subset of initial tweets posted by each user). For the tweet features, we applied one-hot encoding to the tweet content and meta features listed in Section 5.5A. Due to memory limitations, we also applied SVD on the text features with number of components set to 30.

- Doc2vec: [51] This is a popular representation learning approach for documents. To create the user features, we first concatenated the text of a user's tweets into a single document. We repeated this process for each user to obtain 5524 documents. We then trained a doc2vec model on these documents to obtain the feature embedding for each user. To obtain the tweet features, we applied the doc2vec model to the text of each tweet.

- SVD: For user features, we use the same features as CAUTE. For tweet features, all the tweet content and meta features (e.g. source, text, language, etc.) were concatenated before applying SVD.

Table 6.1 summarizes the results of our experiment. We use Area under the ROC curve (AUC) for the positive class (compromised post) as our evaluation metric. For logistic

regression, the results suggest that none of the baseline methods (raw features, doc2vec, and SVD) were able to produce classifiers that perform significantly better than random guessing (AUC score around 0.5). For random forest, doc2vec is the only baseline method that performs slightly better than random guessing, with an AUC score around 0.53.

These results showed the challenges of finding effective features for compromised post detection. Using the raw features alone is insufficient to produce classifiers that perform better than random guessing. The data contains too much variations that standard methods such as SVD and doc2vec were unable to create useful features that can accurately identify compromised posts. Transforming the features using a nonlinear encoder may lead to higher performance. Specifically training a random forest using doc2vec embeddings achieved slightly higher AUC than using the raw features or SVD. One rationale for this is that doc2vec tries to capture meaning within its embeddings. However, since the doc2vec features are learned in an unsupervised way, there are no guarantees that they would be effective for compromised post detection.

CAUTE's latent features derived from the tweet2user and user2tweet encoders were significantly better predictors for compromised post detection than the other baseline algorithms. The results hold for both logistic regression and random forest classifiers. Given that the tweet2user and user2tweet encoders are designed to recognize tweets coming from the genuine users, their embeddings of tweets and users should be good predictors of whether a post is compromised.

6.3.3.2 Performance Comparison

Next, we evaluate CAUTE's overall performance as a whole and show that CAUTE can detect compromised posts better than existing methods. We consider both the overall per-

Table 6.2: Comparison of compromised post detection between CAUTE and the baseline algorithms; COMPA and PCA. Algorithms are evaluated in terms of their AUC score. The number of posts used to generate the user features varied from 5% of their posts to 10%. CAUTE consistently outperforms the baselines by at least 2%. All of the algorithms improve performance when more tweets are used to generate the user features.

Algorithm	5% Tweets	10% Tweets
COMPA	0.6415	0.6779
PCA	0.5064	0.5088
CAUTE	**0.6707**	**0.7014**

formance, weighing false positive and false negatives equally, as well as early detection of compromised accounts. We compare CAUTE against the following state-of-the-art algorithms;

- COMPA: We applied the Trang et al. [97] adaptation of the COMPA algorithm [29, 28], which builds a profile for each user based on their tweets' features. Previously unseen tweets are compared to the profile to determine their likelihood of being anomalous, i.e. assigned an anomaly score. The tweet's anomaly score is the weighted average of the anomaly score of each of its features.

- Principal Component Analysis (PCA): This is an unsupervised anomaly detection approach developed in [107] to extract principal components of the data and then reconstruct the original features from the principal components. The likelihood of a post is compromised is measured by its reconstruction error. In this experiment, we concatenate the users' terms (from the tweets used to create user features) with the tweet features of their tweets. Next we learn the principal components of this matrix for users in the training set. From the learned principal components, we transform user-tweet pairs and calculate their reconstruction error.

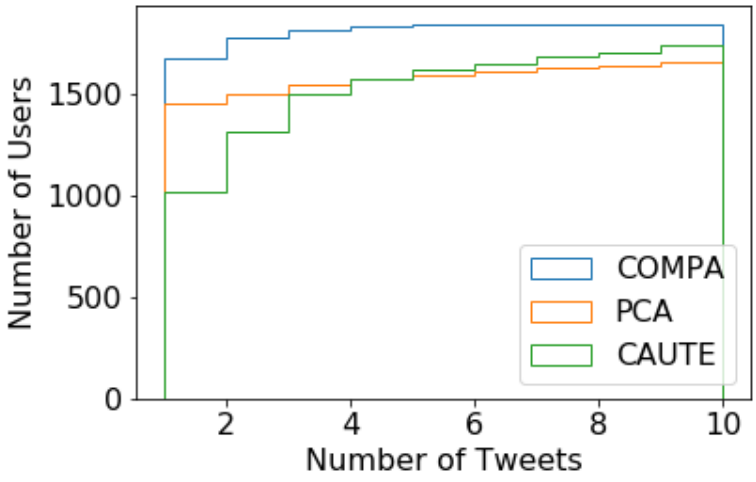
CAUTE is consistently better at detecting compromised posts than the baseline algo-

rithms, as shown in Table 6.2. PCA has the lowest AUC, comparable to random guessing, which is not surprising given the difficulty of the detection task and the unsupervised nature of the algorithm. COMPA and CAUTE were significantly better than PCA. Additionally, both CAUTE and COMPA were better at detecting whether a post belonged to the genuine user when more posts were used to generate the user representation.

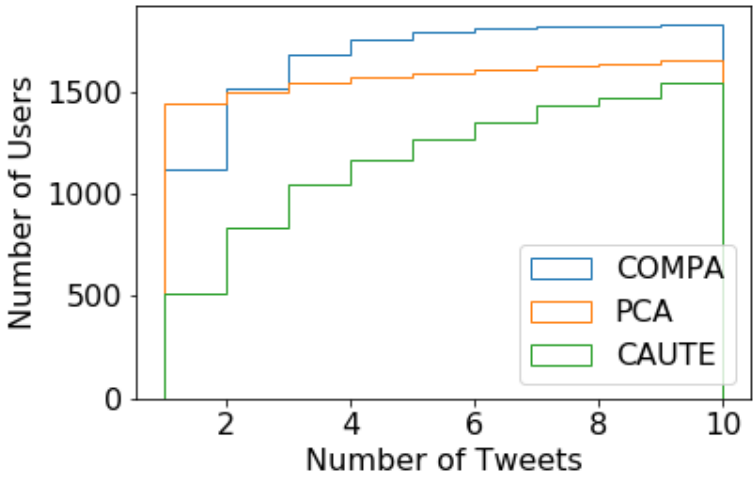
In addition to detecting whether a tweet was published by its respective author, it is crucial to detecting the compromised accounts early. In this experiment, we provide each user with a set of compromised posts and test how many such posts are observed before one of them is flagged as compromised. Ideally, the first post should be detected as compromised, to mitigate any further damages posed by the hacker.

On average, COMPA detects compromised posts earlier than CAUTE. However, given its lower AUC, it also classifies more posts as false positives, i.e. genuine posts classified as compromised post. We observe that COMPA detects the first message as compromised for the majority of the 1841 users in the test set, shown in Figure 6.3a. CAUTE is more conservative when flagging a post as compromised. Within the first 10 tweets, CAUTE can detect most compromised accounts.

To verify that these algorithms were not predicting genuine posts as compromised, we also provided each user with their own posts, and tested how many posts before each algorithm would identify one of them as compromised. Both PCA and COMPA mis-classify genuine users as compromised early, shown in Figure 6.3b. PCA mislabels 1447 of the 1841 users as compromised from their first tweet. COMPA incorrectly predicts only 1125 users as compromised from their first tweet. However, by their 10th tweet, COMPA incorrectly predicts 1832 users as compromised. CAUTE has significantly fewer mis-classifications on the first tweet than the other algorithms. It only incorrectly classifies 520 genuine users



(a) Compromised accounts detected



(b) Genuine accounts predicted as compromised

Figure 6.3: Measurement of how many posts from each user are observed before one of them is flagged as compromised. In (a), all of the tweets provided to the user were compromised. In (b), all of the tweets were their own tweets. COMPA detects most compromised accounts from the first tweet, but also predicts genuine tweets as compromised at a higher rate. CAUTE detects most compromised accounts within the first 10 tweets, and has a significantly lower false positive rate from the the genuine users' posts.

as compromised based on their first tweet. By the users' 10th tweet, CAUTE incorrectly identified only 1572 genuine users as compromised.

Thus, while COMPA and PCA can detect a compromised account earlier than CAUTE, it is at the cost of incorrectly classifying most users' genuine posts as compromised. CAUTE detects compromised accounts slightly later, e.g. from the 3rd compromised tweet. However, CAUTE also detects the genuine users' posts correctly more often than COMPA or PCA, thus raising fewer false alarms. With fewer false alarms, the user is more likely to take actions to mitigate the attack, e.g. changing their password, when a compromise is detected.

6.3.3.3 Evaluating the Importance of Each Component

Next we analyze the importance of each component of CAUTE. This analysis considers how the removal of one or two of the components of CAUTE affects performance, shown in Table 6.3. The tweet2user encoder is better at detecting tweets from a hacker than the user2tweet encoder. Despite appending some tweet information to the user in the user2tweet encoder, it still had lower performance. The inclusion of the res2class encoder yields significantly higher AUC for both encoders. This is expected because the res2class is trained on the specific classification task, identifying whether a tweet was written by their respective user, whereas the other components are focused on minimizing the approximation error. The addition of the user2tweet embedder into CAUTE yields slightly higher performance than using the tweet2user and res2class encoders.

To train the res2class component of CAUTE, the residuals could either be frozen or updated during back propagation. If the residuals are frozen, res2class only learns the neuron weights between the residual and the class. In CAUTE, the residuals are updated as back propagation would update the weights of the user2tweet and tweet2user encoders

Table 6.3: AUC of the tweet2user and user2tweet encoders in comparison to CAUTE. For tweet2user and user2tweet embedders, tweet-user pairs were scored based on the sum of absolute residuals. Tweet2user encoder was better than the user2tweet encoder at identifying whether a tweet matched its respective user. Both encoders provided some information to CAUTE, which achieved higher AUC than either of the individual encoders.

Component	AUC
tweet2user	0.6212
user2tweet	0.5581
tweet2user + res2class	0.6419
user2tweet + res2class	0.5772
res2class (freeze residuals)	0.6691
CAUTE	0.6707

as well as the res2class encoder. The pretrained tweet2user and user2tweet encoders are passed to CAUTE. Frozen residuals is significantly faster at converging, as there are fewer weights that need updating. However allowing CAUTE to back propagation through the entire network yields slightly higher performance, as evident in Table 6.3.

6.4 Conclusion

This chapter proposed CAUTE, a compromised account user-tweet encoder which can identify tweets that do not belong to the designated user, i.e. are compromised. CAUTE is able to identify tweets matched with the incorrect user, from a subset of each user's tweets. Additionally, CAUTE can identify whether tweets belong to users it had not previously observed. In future work, we plan to extend our framework to different types of compromised accounts. Whereas CAUTE focused encoding users and tweets, future work would encode either users with other users to identify suspicious following activity or users with their social media activity patterns to identify hackers seeking information about the compromised user.

Chapter 7

Conclusion

In this dissertation, I explored two approaches malicious actors use to share disinformation on social media; hashtag hijacking and compromised accounts. In hashtag hijacking, a group of users promote a trending hashtag in an organized campaign to change the meaning of the hashtag. This can lead to disinformation about emerging events, such as natural disasters, to affect those involved, e.g. victims of an earthquake. Detection of hashtag hijacking is especially challenging due to hashtags have emergent meaning, i.e. the topic of a hashtag is not defined until people start using it. Therefore this dissertation proposes HASHTECT, an unsupervised learning algorithm to detect hashtag hijacking. HASHTECT learns the topics of a hashtag based on the terms that co-occur with that hashtag and the users who tweet it. It identifies hijacked hashtags by detecting whether there was a significant change in topic and that topic was significantly different from the original intent of the hashtag. HASHTECT can detect hijacked hashtags better than previously proposed algorithms.

Compromised accounts occur when malicious actors take control of the account to impersonate the genuine user. They are used to damage a user's reputation and disseminate misleading information that cause public panic and financial loss. Compromised account detection is a challenging problem, due to the sparsity and noise of social media features, the difficulty of obtaining a ground truth set of posts that were published by hackers, and that some compromised posts resemble normal posts for other users. This dissertation addresses

these challenge by proposing two algorithms, CADET and CAUTE. CADET circumvents identifying the hackers posts by using unsupervised learning to detect compromised accounts. CADET learns a multi-view encoding of each user and identifies the likelihood a user is compromised from their reconstruction error from this encoding. The encoding learns the user's behavior and reduces the noise and the sparsity of features.

CAUTE addresses these challenges by focusing on compromised post detection, i.e. identify if the post was written by a different user than the genuine user of the account. It simultaneously learns a tweet embedding that can approximate the user's features and a user embedding that can approximate the tweet's features. These embeddings are predictive of whether the (tweet, user) pair are a match. CAUTE addresses the compromised account detection challenges by (1) learning an embedding that reduces noise and sparsity and (2) assuming any tweet from another user could be a hacker tweet.

Compromised account detection is also challenging due to the diversity of users. Each user has a different pattern of behavior on social media, however those behaviors have significant overlap, e.g. common software used to publish posts. CADET approaches this challenge by detecting anomalies from multiple views. CAUTE selects the compromised posts to be normal posts from other users. The user and tweet embeddings learned can separate tweets from different users.

There are three key areas of future work for detecting hijacked hashtags and compromised accounts; online learning, cluster-centric learning, and the cold start problem. All of the algorithms proposed in this dissertation were learned via batch learning, i.e. all of the tweets by the user or with the hashtag were provided to the learning algorithms at the same time. In practice, though, social media posts are continually published. A user's topics of interest may change over time, e.g. concept drift, so the user embeddings should change to

reflect these changes. Otherwise these models will produce more false positives, i.e. normal behavior flagged as compromised, as the model becomes out of date. Concept drift can also affect hashtags, so online learning is needed to update the normal topics of the hashtag over time.

Social media platforms are large and continually growing. In this dissertation, the datasets used were relatively small. Thus the algorithms proposed may not scale well, e.g. from thousands of hashtags to hundreds of millions. To address this challenge, we can take advantage of the pairwise similarities between hashtags and between users to cluster the hashtags or users. For hijacked hashtags, the topics would be learned for each cluster instead of each hashtag. For compromised accounts, the embeddings would be cluster-specific instead of user-specific. For example, the hashtags #`Sales` and #`Retail` are closely related. If we clustered all hashtags and learned the topics of each hashtag cluster, I hypothesize that the cluster's topics will be similar to the topics of each hashtag. To detect whether a hashtag is hijacked, we could test whether the recent tweets containing that hashtag belong to one of the cluster's topics. A cluster-centric approach would reduce the model complexity, allowing for more hashtags to be monitored.

As mentioned in Chapter 6, another challenge is the cold start problem, i.e., how to detect compromised posts for previously unobserved users. In this dissertation, I hypothesized that if the user profile contained information about the user's topics of interests, these features could be used as the user features. For some users, their username (e.g. Twitter handle) may provide some information about their interests. They may include the name of a fictional character, a movie, or a game that they enjoy. For many users, their username is based on their actual name, which has no connection to their interests. Thus, additional features need to be augmented to generate the user features.

There is a growing concern among the populous regarding user privacy. As a result, some social media companies have modeled their platforms to remove the content of users' messages after they have been received. For example, Snapchat deletes the snaps (pictures) and chats (messages) from their servers once they have been opened by the designated recipient [42]. Thus, detecting compromised accounts is more challenging with less information. Future work would explore the use of non-content features, such as the user profile and log information, to identify compromised accounts.

Additionally countries, especially the European Union, are passing laws which require machine learning user models to be more transparent. For example, a social media platform would need to make available to the user information on how its models came to a prediction. Deep learning models are harder to interpret, due to their nonlinear embeddings. Recent research focuses on increasing model interpretability, especially of complex algorithms. However, this transparency also increases the information available to adversaries. Hackers looking to evade detection could learn from these algorithms for model transparency.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Mohammad-Ali Abbasi and Huan Liu. Measuring user credibility in social media. In Ariel M. Greenberg, William G. Kennedy, and Nathan D. Bos, editors, *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 441–448, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[2] Robin Abcarian. Boston bombing: Reddit learns how 'witch hunts can start'. *LA Times*, 4 2013.

[3] Abigail Abrams. Pizzagate gunman: i regret how i handled comet ping pong shooting. *Time*, December 2016.

[4] Zeynep Akata, Christian Thurau, and Christian Bauckhage. Non-negative Matrix Factorization in Multimodality Data for Segmentation and Label Prediction. In *16th Computer Vision Winter Workshop*, 2011.

[5] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–36, May 2017.

[6] Tripwire Guest Authors. What happens to hacked social media accounts. *Tripwire*, 2015. https://www.tripwire.com/state-of-security/security-awareness/what-happens-to-hacked-social-media-accounts/.

[7] Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. How Noisy Social Media Text, How Diffrnt Social Media Sources? In *Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP 2013)*, Nagoya, Japan, 2013.

[8] Sylvio Barbon, Jr, Rodrigo Augusto Igawa, and Bruno Bogaz Zarpelão. Authorship verification applied to detection of compromised accounts on online social networks. *Multimedia Tools and Applications*, 76(3), February 2017.

[9] Fabrcio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virglio Almeida. Detecting spammers on twitter. In *In Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*, 2010.

[10] Alessandro Bessi and Emilio Ferrara. Social bots distort the 2016 u.s. presidential election online discussion. *First Monday*, 21(11), 2016.

[11] Mathieu Blondel, Yotaro Kubo, and Ueda Naonori. Online passive-aggressive algorithms for non-negative matrix factorization and completion. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 96–104, 2014.

[12] Jan Bohacik, Antonin Fuchs, and Miroslav Benedikovic. Detecting compromised accounts on the pokec online social network. In *2017 International Conference on Information and Digital Technologies (IDT)*, 2017.

[13] Marcel Broersma and Todd Graham. Social media as beat: Tweets as a news source during the 2010 british and dutch elections. *Journalism Practice*, 6(3):403–419, 2012.

[14] Nick Bryans. How not to deal with an attention seeking troll twitter style, 2012. http://www.shoutingatco.ws/2012/08/06/how-not-to-deal-with-an-attention-seeking-troll-twitter-style/.

[15] Massimo Calabresi. Inside russias social media war on america. *Time*, May 2017.

[16] Anita Campbell. Small business trends: What is hashtag hijacking?, 2013. http://smallbiztrends.com/2013/08/what-is-hashtag-hijacking-2.html.

[17] Qiang Cao, Xiaowei Yang, Jieqi Yu, and Christopher Palow. Uncovering large groups of active malicious accounts in online social networks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 477–488, New York, NY, USA, 2014. ACM.

[18] J. Douglas Carroll. Generalization of canonical correlation analysis to three or more sets of variables. In *Proceedings of the American Psychological Association*, pages 227–228, 1968.

[19] Twitter Help Center. My account has been compromised, 2014. https://support.twitter.com/articles/31796-my-account-has-been-compromised.

[20] Xinran Chen, Sei-Ching Joanna Sin, Yin-Leng Theng, and Chei Sian Lee. Why do social media users share misinformation? In *Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '15, 2015.

[21] Yimin Chen, Niall J. Conroy, and Victoria L. Rubin. Misleading online content: Recognizing clickbait as "false news". In *Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection*, WMDD '15, pages 15–19, 2015.

[22] Zi Chu, Indra Widjaja, and Haining Wang. Detecting social spam campaigns on twitter. In *Proceedings of the 10th International Conference on Applied Cryptography and Network Security*, ACNS'12, pages 455–472, Berlin, Heidelberg, 2012. Springer-Verlag.

[23] Joe Coscarelli. The #mynypd hashtag is not going so well for the police, 2014. http://nymag.com/daily/intelligencer/2014/04/mynypd-hashtag-promptly-hijacked.html.

[24] Anqi Cui, Min Zhang, Yiqun Liu, Shaoping Ma, and Kuo Zhang. Discover breaking events with popular hashtags in twitter. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1794–1798, New York, NY, USA, 2012. ACM.

[25] Anthony Cuthbertson. Hackers hijack isis twitter accounts with gay porn after orlando attack. *Newsweek*, 2016.

[26] Nicholas A. Diakopoulos and David A. Shamma. Characterizing debate performance via aggregated twitter sentiment. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1195–1198, 2010.

[27] Nicholas Difonzo and Prashant Bordia. *Rumor Psychology: Social And Organizational Approaches*. American Psychological Association, 1 edition, 9 2006.

[28] Manuel Egele, Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. COMPA: Detecting Compromised Accounts on Social Networks. In *ISOC Network and Distributed System Security Symposium (NDSS)*, 2013.

[29] Manuel Egele, Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Towards detecting compromised accounts on social networks. *IEEE Transactions on Dependable and Secure Computing*, 14(4), 2017.

[30] Dina ElBoghdady. Market quavers after fake ap tweet says obama was hurt in white house explosions. *The Washington Post*, April 2013.

[31] Austin Fracchia. Top 7 hashtag hijacking fails, 2014. http://www.business2community.com/social-media/top-7-hashtag-hijacking-fails-01026291.

[32] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y. Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, pages 35–47. ACM, 2010.

[33] Pamela Geller. Isis fighters, supporters hijack #baltimoreriots twitter hashtag, discuss race issues, urge attacks on policemen, 2015. http://pamelageller.com/2015/04/isis-fighters-supporters-hijack-baltimoreriots-twitter-hashtag-discuss-race-issues-urge-attacks-on-policemen.html/.

[34] Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. Using topic models for twitter hashtag recommendation. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13 Companion, pages 593–596, 2013.

[35] Chris Grier, Kurt Thomas, Vern Paxson, and Michael Zhang. @spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security*, CCS '10, pages 27–37. ACM, 2010.

[36] Asmelash Teka Hadgu, Kiran Garimella, and Ingmar Weber. Political hashtag hijacking in the u.s. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 55–56, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[37] Kohei Hayashi, Takanori Maehara, Masashi Toyoda, and Ken-ichi Kawarabayashi. Real-time top-r topic detection on twitter with topic hijack filtering. In *Proc. of the 21th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2015.

[38] Alfred Hermida. Twittering the news: The emergence of ambient journalism. *Journalism Practice*, 4(3):297–308, July 2010.

[39] Harold Hotelling. The generalization of student's ratio. *The Annals of Mathematical Statistics*, 2(3):360–378, 08 1931.

[40] Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. Social spammer detection with sentiment information. In *Proceedings of the 2014 IEEE International Conference on Data Mining*, ICDM '14, pages 180–189, 2014.

[41] Rodrigo Augusto Igawa, Alex Marino Goncalves de Almeida, Bruno Bogaz Zarpelao, and Sylvio Barbon, Jr. Recognition of compromised accounts on twitter. In *Proceedings of the Annual Conference on Brazilian Symposium on Information Systems: Information Systems: A Computer Socio-Technical Perspective - Volume 1*, SBSI 2015, pages 2:9–2:14, Porto Alegre, Brazil, Brazil, 2015. Brazilian Computer Society.

[42] Snap Inc. Privacy policy, 2018. https://www.snap.com/en-US/privacy/privacy-policy.

[43] Twitter Inc. About verified accounts. https://help.twitter.com/en/managing-your-account/about-twitter-verified-accounts.

[44] Infowars.com. Pro-toothpaste hashtag hijacked by anti-fluoride activists, 2014. http://www.infowars.com/pro-toothpaste-hashtag-hijacked-by-anti-fluoride-activists/.

[45] Gary G. Koch J. Richard Landis. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

[46] Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, and Shiqiang Yang. Catchsync: Catching synchronized behavior in large directed graphs. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 941–950, 2014.

[47] Krishna B. Kansara and Narendra M. Shekokar. A framework for cyberbullying detection in social network. *International Journal of Current Engineering and Technology*, 2015.

[48] Howard Koplowitz. #keepcalmvotedem hashtag trends on twitter, gets hijacked by republicans, 2014. http://www.ibtimes.com/keepcalmvotedem-hashtag-trends-twitter-gets-hijacked-republicans-1716095.

[49] K. P. Krishna Kumar and G. Geethakumari. Detecting misinformation in online social networks using cognitive psychology. *Human-centric Computing and Information Sciences*, 4(1), Sep 2014.

[50] Ludmila I. Kuncheva. Change Detection in Streaming Multivariate Data Using Likelihood Detectors. *Knowledge and Data Engineering, IEEE Transactions on*, 25(5):1175–1180, May 2013.

[51] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, 2014.

[52] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press, 2001.

[53] Kyumin Lee, James Caverlee, and Steve Webb. Uncovering social spammers: Social honeypots + machine learning. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 435–442. ACM, 2010.

[54] Kar Wai Lim and Wray Buntine. Twitter opinion topic model: Extracting product opinions from tweets by leveraging hashtags and sentiment lexicon. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 1319–1328, New York, NY, USA, 2014. ACM.

[55] Bruce R. Lindsay. Social Media and Disasters: Current Uses,Future Options, and Policy Considerations. Technical Report R41987, Congressional Research Service, September 2011.

[56] Teun Lucassen and Jan Maarten Schraagen. Trust in wikipedia: How users trust information from an unknown source. In *Proceedings of the 4th Workshop on Information Credibility*, WICOW '10, pages 19–26, 2010.

[57] Zongyang Ma, Aixin Sun, and Gao Cong. Will this #hashtag be popular tomorrow? In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 1173–1174, 2012.

[58] David Mack. New york gay bar claims it was hacked after angry tweets to bill de blasio and al sharpton. *BuzzFeed News*, 2014.

[59] Alexis Madrigal. #bostonbombing: The anatomy of a misinformation disaster. *The Atlantic*, 4 2013.

[60] Walid Magdy, Kareem Darwish, and Ingmar Weber. #failedrevolutions: Using twitter to study the antecedents of ISIS support. *CoRR*, 2015.

[61] Jennifer Marshall. Nearly two-thirds of u.s. adults with social media accounts say they have been hacked. Technical report, University of Phoenix, 2016.

[62] Ashish Mehrotra, Mallidi Sarreddy, and Sanjay Singh. Detection of fake twitter followers using graph centrality measures. In *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, pages 499–504, Dec 2016.

[63] Scott Menard. *Applied Logistic Regression Analysis*, volume 106. Sage Publishing, 2 edition, 1997.

[64] Scott Menard. Standards for standardized logistic regression coefficients. *Social Forces*, 89(4):1409–1428, 2011.

[65] Merriam-Webster. Disinformation, 2015. http://www.merriam-webster.com/dictionary/disinformation.

[66] Chris Messina. Groups for twitter; or a proposal for twitter tag channels, Aug 2007. https://factoryjoe.com/2007/08/25/groups-for-twitter-or-a-proposal-for-twitter-tag-channels/.

[67] Benjamin Murauer, Eva Zangerle, and Guenther Specht. A peer-based approach on analyzing hacked twitter accounts. In *Hawaii International Conference on System Sciences*, 01 2017.

[68] Meike Nauta. Detecting Hacked Twitter Accounts by Examining Behavioural Change using Twtter Metadata. In *Proceedings of the 25th Twente Student Conference on IT*, 2016.

[69] Nielsen. Global trust in advertising, 2015. http://www.nielsen.com/us/en/insights/reports/2015/global-trust-in-advertising-2015.html.

[70] Markham Nolan, Assaf Uni, and Gilad Shiloach. Hostage steven sotloff is the center of new isis propaganda campaign. *Vocative*, 2014.

[71] Kenneth Olmstead and Aaron Smith. Americans and cybersecurity. Technical report, Pew Research CCenter, 2017.

[72] Shira Ovide. False ap twitter message sparks stock-market selloff. *Wall Street Journal*, 4 2013.

[73] Symeon Papadopoulos, Kalina Bontcheva, Eva Jaho, Mihai Lupu, and Carlos Castillo. Overview of the special issue on trust and veracity of information in social media. *ACM Transactions on Information Systems*, 34(3), April 2016.

[74] Liza Potts, Joyce Seitzinger, Dave Jones, and Angela Harrison. Tweeting disaster: Hashtag constructions and collisions. In *Proceedings of the 29th ACM International Conference on Design of Communication*, SIGDOC '11, pages 235–240, New York, NY, USA, 2011. ACM.

[75] The Associated Press. U.s. indicts 3 it ties to syrian electronic army for hacking. *AP in the News*, 2016.

[76] Jacob Ratkiewicz, Michael Conover, Mark Meiss, Bruno Goncalves, Alessandro Flammini, and Filippo Menczer. Detecting and tracking political abuse in social media. In *International AAAI Conference on Web and Social Media*, 2011.

[77] Jacob Ratkiewicz, Michael Conover, Mark Meiss, Bruno Gonçalves, Snehal Patil, Alessandro Flammini, and Filippo Menczer. Truthy: Mapping the spread of astroturf in microblog streams. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 249–252, 2011.

[78] Xin Ruan, Zhenyu Wu, Haining Wang, and Sushil Jajodia. Profiling Online Social Behaviors for Compromised Account Detection. *IEEE Transactions on Information Forensics and Security*, 11(1):176–187, 2016.

[79] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 851–860, 2010.

[80] Erich Schubert, Michael Weiler, and Hans-Peter Kriegel. Signitrend: Scalable detection of emerging topics in textual streams by hashed significance thresholds. In *Proc. of the 20th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2014.

[81] Neil Shah, Alex Beutel, Brian Gallagher, and Christos Faloutsos. Spotting suspicious link behavior with fbox: An adversarial perspective. In *2014 IEEE International Conference on Data Mining*, pages 959–964, Dec 2014.

[82] Richard Shay, Iulia Ion, Robert W. Reeder, and Sunny Consolvo. "my religious aunt asked why i was trying to sell her viagra": Experiences with account hijacking. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 2657–2666, 2014.

[83] Elisa Shearer and Jeffrey Gottfried. News use across social media platforms 2017, 2017. http://www.journalism.org/2017/09/07/news-use-across-social-media-platforms-2017/.

[84] Yi Shen, Jianjun Yu, Kejun Dong, and Kai Nan. Automatic fake followers detection in chinese micro-blogging system. In *Advances in Knowledge Discovery and Data Mining*, pages 596–607. Springer International Publishing, 2014.

[85] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *SIGKDD Explorations Newsletter*, 19(1):22–36, September 2017.

[86] Ge Song, Yunming Ye, Xiaolin Du, Xiaohui Huang, and Shifu Bie. Short text classification: A survey. *Journal of Multimedia*, 9(5):635–643, 2014.

[87] Vasumathi Sridharan, Vaibhav Shankar, and Minaxi Gupta. Twitter games: how successful spammers pick targets. In Robert H'obbes' Zakon, editor, *Annual Computer Security Applications Conference (ACSAC)*, pages 389–398. ACM, 2012.

[88] Kate Starbird, Jim Maddock, Mania Orand, Peg Achterman, and Robert M. Mason. Rumors, false flags, and digital vigilantes: Misinformation on twitter after the 2013 boston marathon bombing. In *iConference 2014 Proceedings*. iSchools, 2014.

[89] Kate Starbird and Leysia Palen. (how) will the revolution be retweeted?: Information diffusion and the 2011 egyptian uprising. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, pages 7–16, New York, NY, USA, 2012. ACM.

[90] Gianluca Stringhini, Manuel Egele, Christopher Kruegel, and Giovanni Vigna. Poultry Markets: On the Underground Economy of Twitter Followers. In *Proceedings of the 2012 ACM Workshop on Workshop on Online Social Networks*, WOSN '12, pages 1–6, 2012.

[91] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 1–9, New York, NY, USA, 2010. ACM.

[92] Gianluca Stringhini, Gang Wang, Manuel Egele, Christopher Kruegel, Giovanni Vigna, Haitao Zheng, and Ben Y. Zhao. Follow the green: Growth and dynamics in twitter follower markets. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 163–176, 2013.

[93] My T. Thai, Weili Wu, and Hui Xiong. *Big Data in Complex and Social Networks*. Chapman & Hall/CRC Big Data Series. CRC Press, 2016.

[94] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment in short strength detection informal text. *J. Am. Soc. Inf. Sci. Technol.*, 61(12):2544–2558, December 2010.

[95] Kurt Thomas, Chris Grier, Dawn Song, and Vern Paxson. Suspended accounts in retrospect: An analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, 2011.

[96] Kurt Thomas, Frank Li, Chris Grier, and Vern Paxson. Consequences of connectivity: Characterizing account hijacking on twitter. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 489–500, 2014.

[97] David Trang, Fredrik Johansson, and Magnus Rosell. Evaluating algorithms for detection of compromised social media user accounts. In *2015 Second European Network Intelligence Conference*, pages 75–82, Sept 2015.

[98] Alexander Trowbridge. ISIS swiping hashtags as part of propaganda efforts. *CBS News*, 2014.

[99] Oren Tsur and Ari Rappoport. What's in a hashtag?: Content based prediction of the spread of ideas in microblogging communities. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 643–652, 2012.

[100] Wali Ahmed Usmani, Diogo Marques, Ivan Beschastnikh, Konstantin Beznosov, Tiago Guerreiro, and Luís Carriço. Characterizing social insider attacks on facebook. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 3810–3820, 2017.

[101] Courtland VanDam and Pang-Ning Tan. Detecting hashtag hijacking from twitter. In *Proceedings of the 8th ACM Conference on Web Science*, WebSci '16, pages 370–371, 2016.

[102] Courtland VanDam, Pang-Ning Tan, Jiliang Tang, and Hamid Karimi. Cadet: A multi-view learning framework for compromised account detection. In *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2018*, 2018.

[103] Courtland VanDam, Jiliang Tang, and Pang-Ning Tan. Understanding compromised accounts on twitter. In *2017 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, 2017.

[104] Yannick Veilleux-Lepage. Retweeting the caliphate: The role of soft-sympathizers in the islamic state's social media strategy. In *6th International Symposium on Terrorism and Transnational Crime*, Antalya, Turkey, December 4-7 2014.

[105] Anthony J. Viera and Joanne M. Garrett. Understanding interobserver agreement: The kappa statistic. *Family Medicine*, 37(5):360–363, 5 2005.

[106] Sarah Vieweg, Amanda L. Hughes, Kate Starbird, and Leysia Palen. Microblogging during two natural hazards events: What twitter may contribute to situational awareness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1079–1088. ACM, 2010.

[107] Bimal Viswanath, M. Ahmad Bashir, Mark Crovella, Saikat Guha, Krishna P. Gummadi, Balachander Krishnamurthy, and Alan Mislove. Towards detecting anomalous user behavior in online social networks. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 223–238. USENIX Association, 2014.

[108] Beidou Wang, Can Wang, Jiajun Bu, Chun Chen, Wei Vivian Zhang, Deng Cai, and Xiaofei He. Whom to mention: Expand the diffusion of tweets by @ recommendation on micro-blogging systems. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1331–1340, 2013.

[109] Ingmar Weber. "political polarization of web search queries and hashtags" by ingmar weber, with martin vesely as coordinator. *SIGWEB Newsletter*, (Summer):4:1–4:10, July 2013.

[110] Shan-Hung Wu, Man-Ju Chou, Chun-Hsiung Tseng, Yuh-Jye Lee, and Kuan-Ta Chen. Detecting in situ identity fraud on social network services: A case study with facebook. *IEEE Systems Journal*, 11(4):2432–2443, Dec 2017.

[111] Eva Zangerle and Günther Specht. "sorry, i was hacked": A classification of compromised twitter accounts. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, SAC '14, pages 587–593, New York, NY, USA, 2014. ACM.

[112] Zhedi Zhang, Futai Zou, Li Pan, Bei Pei, and Jianhua Li. Detection of zombie followers in sina weibo. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pages 2476–2480, Oct 2016.

[113] Arkaitz Zubiaga, Damiano Spina, Víctor Fresno, and Raquel Martínez. Classifying trending topics: A typology of conversation triggers on twitter. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, 2011.