Types of Bots: Categorization of Accounts

Using Unsupervised Machine Learning

by

Matthew Davis

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved October 2019 by the
Graduate Supervisory Committee:

Huan Liu, Chair
Guoliang Xue
Fred Morstatter

ARIZONA STATE UNIVERSITY

December 2019

ABSTRACT

Social media bot detection has been a signature challenge in recent years in online social networks. Many scholars agree that the bot detection problem has become an "arms race" between malicious actors, who seek to create bots to influence opinion on these networks, and the social media platforms to remove these accounts. Despite this acknowledged issue, bot presence continues to remain on social media networks. So, it has now become necessary to monitor different bots over time to identify changes in their activities or domain. Since monitoring individual accounts is not feasible, because the bots may get suspended or deleted, bots should be observed in smaller groups, based on their characteristics, as types. Yet, most of the existing research on social media bot detection is focused on labeling bot accounts by only distinguishing them from human accounts and may ignore differences between individual bot accounts. The consideration of these bots' types may be the best solution for researchers and social media companies alike as it is in both of their best interests to study these types separately. However, up until this point, bot categorization has only been theorized or done manually. Thus, the goal of this research is to automate this process of grouping bots by their respective types. To accomplish this goal, the author experimentally demonstrates that it is possible to use unsupervised machine learning to categorize bots into types based on the proposed typology by creating an aggregated dataset, subsequent to determining that the accounts within are bots, and utilizing an existing typology for bots. Having the ability to differentiate between types of bots automatically will allow social media experts to analyze bot activity, from a new perspective, on a more granular level. This way, researchers can identify patterns related to a given bot type's behaviors over time and determine if certain detection methods are more viable for that type.

ACKNOWLEDGMENTS

A special thank you to Dr. Huan Liu for all of your constant support and belief in me during both my undergrad and graduate degrees. Tahora Hossein Nazer thank you for your mentorship and guidance throughout all of my research. Thank you Raha Moraffah for teaching me statistics and for being a great friend. Thank you to everyone in the Data Mining and Machine Learning lab: Isaac Jones, Ghazaleh Beigi, Kai Shu, Lu Cheng, Nur Shazwani Kamrudin, Ruocheng Guo, Kaize Ding, Mansooreh Karami, and Jundong Li for all of your support. I would also like to thank Dr. Guoliang Xue and Dr. Fred Morstatter for agreeing to serve on my thesis committee. I also benefited from discussions with David Koelle and Adrian Flowers of Charles River Analytics. This research was funded by the Office of Naval Research (ONR) through research grant N000141812108.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

One of the most challenging aspects about social media bot detection is that bots are evolving and deviating from their previous behaviors to avoid detection by state-of-the-art detection methods [13]. To add to this field of research, it would be interesting to examine the types of bots that are present on these social media networks. Then, an individual type of bot can be singled out and analyzed, and detection methods will become stronger as a whole given this additional information.

What is a type of bot? These types are defined as groups of automated accounts that share some common combination of features, behaviors, structure, and networks. Thus, a type can be as ambiguous or as specific as a researcher wants. Take dormant bots for instance. Basically any bot on a social media network who has not interacted with other users for some time may be considered dormant. On the contrary, bots such as weather bots should only include accounts whose sole purpose is to post about the weather. Other examples of types of bots from previous proposed frameworks include spambots, chatbots, social bots, and sockpuppets [15]. Some other researchers have asserted that bots can be either malicious or benign [22]. For example, chatbots which are helpful bots that can serve as a company representative to interact with potential customers, may be labeled as benign. Most typologies provide an overview of possible types of bots but fail to answer the question of: how many types can exist within a given domain? Furthermore, do different types of bots require different detection methods? Is it possible to detect multiple types of bots in existing labeled datasets without prior knowledge of those bots' presence?

Almost all of the existing research on social media bot detection is focused on removing all bot accounts, regardless of their intent, the unique features of the accounts, or the domain where these bots are present. Moreover, most approaches focus solely on detecting bots within a distinct dataset and then training new models to fit successive datasets. What happens if there are more than one type of bots present in these datasets or if another dataset does not contain these types? If so, are these bots connected? If not, why are only certain types of bots prevalent in certain domains? To start answering these questions, there has been some recent research performed which cites different types of bots and the topic discussions these bots participate in [15, 20, 27]. That research suggests categorizing all bots together and removing them from the social media platform, without the consideration of the type of bot, is not the best solution. This raises the following questions: Is it possible to group bots by their different types instead of generalizing them into a single bot category? If so, is there a way to do this without using manual annotation as has been the case in most previous research [15, 22, 27]? As most other research is focused on detecting a single type of bots in a given domain, how can previous works be expanded to encompass multiple types of bots or bots of the same type but in different domains?

While most existing works focus on only detecting one type of bots within a given dataset, there could be more than one type of bot present in that domain. It would be nice to group these bots by type while doing bot detection. However, as there is almost no research on comparing bot types while doing bot detection to distinguish between automated accounts and humans, this research will focus solely on bot categorizing bot accounts separately using a previous typology on bots. Therefore, the aim of this research is to identify types of bots in a dataset subsequent to determining that the accounts are bots. As such, the social network chosen for this research was Twitter

due to the high volume of existing research on the social media platform and the large availability of data. This research presents a methodology to categorize bots into types automatically using unsupervised machine learning. Having the ability to differentiate between types of bots automatically, could not only make existing bot detection methods more efficient, as it would allow social media companies to only remove certain types of bots which harm human users, it will also ensure helpful bots can continue to exist. After bots have been categorized into types, social media experts will be able to analyze bot activity on a more individual level and identify patterns related to their behaviors. Additionally, the ability to separate bots into types will allow future researchers to track changes to specific types of bots over time and determine if certain detection methods are more viable for any given type.

The rest of this work is organized as follows. Chapter 2 summarizes previously published works on types of bots and unsupervised machine learning for bot detection. Then, Chapter 3 describes the datasets used in this study; how they were collected, what they contain, and the reasons for choosing each dataset. Next, Chapter 4 first presents a methodology for bot type categorization then introduces the unsupervised machine learning algorithms used for the experiments and the related metrics necessary to evaluate the clusters obtained by these algorithms. Following that, Chapter 5 showcases the experimental results and analysis culminating in the discovery of types of bots based on a previous typology. Finally, Chapter 6 reviews the contributions of this work and provides a future direction for other researchers interested in this topic.

Chapter 2

RELATED WORK

This chapter presents the existing work on both types of bots and unsupervised machine learning for bot detection. Section 2.1 will summarize all of the previous works that have attempted to categorize bot accounts or create bot typologies. Then, since limited research has been done on addressing the problem of types of bots using machine learning, unsupervised machine learning works presented that are not specific to determining types of bots are addressed in Section 2.2. This unsupervised learning research is still relevant to consider since the methodology used in this paper is similar to a multi-class case of bot detection. The important takeaways from Section 2.2 are the features used by previous unsupervised machine learning methods and the rational for using this approach for bot detection.

2.1 Types of Bots

Recently, there has been a big increase in the literature on the types of bots that exist on social media. Of these, Gorwa and Guilbeault [15] present one of the most cumulative works. Yet, their work is more of an overview of the research area than new methods to solve these problems. In their work, the authors identify some of the weaknesses of existing bot categorization efforts and present a new typology of bots. This typology lists bots within six distinct categories: 'Web Robots', Chatbots, Spambots, Social Bots, Sockpuppets and Trolls, and Cyborgs and Hybrid Accounts. Additionally, Gorwa and Guilbeault present a framework to categorize bot accounts

into their typology based on three considerations: structure, functionality, and ethics. They emphasize that identifying the structure of the bot can be done by observing how it works and the domain it operates in. Similarly, they examine the functional capabilities of the bot by checking if it appears to engage in conversation with other users. The authors determine the bot's ethics by evaluating the intent or social impact of the bot. While the Gorwa and Guilbeault framework seems very inclusive, the real-world application of their work is limited as the authors did not explicitly mention any data examples of bots in their work.

Another recent work which discusses types of bots is Yang et al. [27]. In their work, the authors admit that bots can be categorized based on their characteristics, domain, and features. There, like the Gorwa and Guilbeault typology, bots can be grouped into types as either simple bots, sophisticated bots, fake followers, or botnets based on these characteristics. Yang et al. also mention that it is possible to distinguish between bots based on the domain that they are present in; for example health, politics, fake news, or terrorism. In addition, the authors include a section on traditional bot detection as they identify six features that help to discern bots from human accounts. These features consisted of user metadata, friend metadata, retweet/mention network structure, content and language, sentiment, and temporal features. Despite including this categorization of bot types in their work, Yang et al. do not explicitly label bots by type and only use dataset labels from previous research to identify bot type. However, since the authors provided a good description of each type of bots, their work is more reproducible than the work by Gorwa and Guilbeault.

Others that ponder this question of how to categorize bots include Stieglitz et al. [22] and Lee et al. [17]. In the work by Stieglitz et al., the authors combined two previous works to form a two-dimensional framework for bot categorization that has

intent (measured as ternary values: malicious, neutral, or benign) as one dimension and imitation of human behavior (measured as binary values: low to none or high) as the other. The main weakness of their categorization is that the authors manually chose where to place each type of bot in their chart rather than using metrics to determine the values. Contrarily, the work by Lee et al., which was primarily a dataset paper, included a section where they used Expectation Maximization to organize bot accounts into categories of duplicate spammers, duplicate @ Spammers, malicious promoters, and friend infiltrators [17]. While this was a promising start, Lee et al. mentioned that there were probably more types of bots within their dataset but did not elaborate on how to interpret these other types [17].

Furthermore, there have been several works which address "evolving bots". While these papers do not explicitly mention about types of bots, it can be conjectured that the evolved bots have some noticeable differences (since bots have become increasingly more sophisticated to keep up with improved detection methods) from their previous counterparts. Thus, some of these works also mention features that help differentiate between older and newer bots. One such paper that examined evolving twitter spammers, Yang et al. [26], identified some trends of newer spambots which avoid detection by: having more followers, posting more tweets, mixing spam messages with normal tweets, and posting heterogeneous tweets. They propose using multiple groups of features to detect these evolved bots such as: graph-based features including betweenness centrality and bidirectional links ratio, neighbor-based features like average neighbors' followers or tweets, and automation-based features like API ratio and API tweet similarity. Another paper focused on evolving bots by Cresci et al. [10] labels bots as social bots, traditional spambots, and fake follower accounts, with social bots being the most evolved type of bot. In that study, the authors

performed experiments to determine if Twitter was capable of removing these evolved bots, if humans could discern between the newer bot and human accounts, and if state-of-the-art machine learning techniques could spot these updated bots. Their results show that the evolved bots are much harder to detect now than in the past.

In closing, there have been many different typologies of bots presented to date. Each typology has a slightly different idea as to what constitutes a bot belonging to any one given type. For example, some typologies categorize based on account features and some group solely based on the intent and behavior of bot accounts. Yet, most of the typologies agree that there are at least three major types of bots namely: simple or spam bots, sophisticated or social bots, and some flavor of fake follower or friend infiltrator bots. The main limitation of the previously proposed frameworks is that all of the typologies presented have done little besides theorizing these types or doing some manual annotation to apply these typologies to real-world data. Therefore, there should be a methodology to automatically categorize bots into types which will promote further research in this area.

2.2 Unsupervised Learning for Bot Detection

Unsupervised machine learning is used for bot detection because it does not rely on having "ground-truth data" – a labeled set of bot accounts provided by direct observation – prior to training. Most bot detection using unsupervised machine learning works by clustering accounts based on their features (i.e. profile information, tweet content, etc.). Clustering can be done using many methods, but most algorithms utilize some sort of distance measure to compare features. Researchers hope that the clusters obtained from these unsupervised machine learning models are differentiated

based on the features and make an assumption that the features are capable of confirming whether an account is a bot or a human. Unsurprisingly, these features, such as the percentage of retweets over total tweets, are indicative of bot activity [14]. While unsupervised bot detection methods are usually used for detecting large groups of bots, there is limited research on identifying individual bots using this learning technique. The biggest issue with using unsupervised learning for bot detection is determining how to validate the model. Existing bot detection algorithms that use unsupervised machine learning cannot evaluate the effectiveness of their methods the same way that traditional supervised methods using ground-truth data can. Therefore, previous works attempt to show their effectiveness by using these traditional supervised learning algorithms to detect bots discovered by their models. Since this evaluation still uses supervision to confirm the results of the unsupervised method, it is still bound by the limitations of supervised learning.

The following subsections present some unsupervised machine learning methods that have been previously used in the field of bot detection. For the sake of comparing bot detection approaches, the approaches are separated into two main methods of unsupervised bot detection: group-based methods which are discussed in Section 2.2.1 and individual-based methods presented in Section 2.2.2. The main distinction between the two methods is that group-based methods work by finding groups of accounts that are likely to have coordinated with one another, as is in the case of botnets, whereas individual methods compare accounts with one another directly. Therefore, the proposed methodology of this work is more similar to an individual-based method since it relies on having a predetermined set of bot accounts and does not account for coordination, but differs in that all of the methods below are focused on strictly separating bots and humans accounts rather than categorizing bots.

2.2.1 Group-based Methods

Group-based unsupervised machine learning bot detection approaches are based on the assumption that bots are most effective when they are implemented as coordinated groups working together to achieve a common goal. This coordination imposes some inherent similarities between the bots in terms of the accounts' profile information, posting behaviors, and activity patterns. Using this assumption, Chavoshi et al. [6] proposed a method to detect bots based on which groups of users have activities that are abnormally aligned. In their work, the authors estimate the probability that two or more users in a group of users tweet or retweet in a window of some w seconds from each other during an hour. They show that for users who have at least 40 tweets in an hour, the probability that two human users share these same activities is nearly zero. Chavoshi et al. [5] then expanded their work to create a lag-sensitive hashing method to find groups of bots based on the similarity of their actions over time. Another group-based unsupervised bot detection approach includes monitoring specific shortened URLs included in the tweet text to determine if multiple users post the same links [7]. Likewise, another work groups users by their interactions, hash-tagging, and URL utilization [1]. Finally, a recent work represents users as DNA-like sequences of tweeting activity patterns and then observes the Longest Common Subsequence length in order to group similar accounts together [9]. While all of the group-based methods show promising results for bot detection tasks, it is unknown how well these methods perform on bots that are not coordinated.

2.2.2 Individual-based Methods

Contrary to group-based unsupervised learning methods, individual-based methods attempt to focus on unknown patterns in the data rather than learning the network structure or discovering explicit coordination. Thus, there are several methods which seek to detect individual bots. These methods attempt to learn the difference between bot and human users on some datasets where researchers already know the labels. For example, one of the first works on this topic, Zhang and Paxson [28], attempted to detect bots using differences between observed human and bot tweeting patterns. Their method involved looking at the minute-of-the-hour and second-of-the-minute differences of tweeting timestamps and grouped accounts based on the distributions of each. More recently, Chino et al. [8] proposed a method called VolTime that is a generative model based on the inter-arrival time and volume of activities. Another recently proposed method, BotWalk [18], works by generating a representation of each social media user by four feature categories and then using four unsupervised anomaly detection algorithms to give each user an overall anomaly score.

Chapter 3

DATA

Table 1. Statistics of the datasets used in this study

Property / Dataset	**Caverlee 2011**	**Morstatter 2016**	**Cresci 2017**
Authors	Lee et al. [17]	Morstatter et al. [19]	Cresci et al. [10]
Tweets	5,613,166	332,475	6,637,615
Retweets	197,850	96,796	836,646
Human Accounts	19,252	2,166	2,167
Bot Accounts	20,601	2,029	9,114
Bots Still Active	14,321	1,952	5,813
Bot Ratio	51.69%	48.37%	81.33%
Labeling Approach	Honeypot	Honeypot	Manual

To analyze the behavior and characteristics of different types of bot accounts, the first step is to obtain a set of bot accounts. Therefore, this research utilizes three existing labeled datasets represented in Table 1 to show that the proposed model is robust with respect to the language, topic, time, and labeling mechanism. The methodology proposed in this paper attempts to categorize types of bots within these datasets by first aggregating them. Then unsupervised learning is used to find commonalities between accounts across each of the datasets, although it may be feasible to find different types of bots within a single dataset as well. While the original datasets contained human accounts, these accounts were not used in this study since the focus is on categorizing types of bots. So, in future work, it should be sufficient to obtain a set of bot accounts without collecting a parallel set of human account, which is typical of most bot detection datasets, and still categorize those bots into types. The following sections describe how each of these raw datasets were collected and what they contain.

3.1 Morstatter 2016 Dataset

The first dataset is a honeypot dataset collected by Morstatter et al. [19]. The dataset contains tweets in both Arabic and English. It was collected using a network of nine honeypot accounts which tweeted Arabic phrases in addition to randomly following and retweeting each other. Any user who followed a honeypot account was considered a bot because the honeypots were designed to exhibit sporadic behaviors that provided no intelligent information to humans. The original data collection occurred between February 3^{rd}, 2011, and February 21^{st}, 2013, but the dataset was re-crawled using the tweet IDs shared by the original authors in August 2018 and again in October 2019. This dataset is referred to as 2016 because that was when the work was published. It is interesting to note that of the original 2,029 bots in the dataset, only 77 of those accounts have been removed from Twitter; with 57 being suspended and the other 20 deleted. Thus, this dataset contains somewhat inconspicuous bots, since these bots have yet to be removed from the network in over five years. However, since only some of the tweets in this dataset are in English, the authors of this work feel that it is necessary to include another dataset such as the one in the next subsection whose text corpus spanned more of the English language.

3.2 Caverlee 2011 Dataset

The second dataset used was the Social Honeypot Dataset [17], or the "Caverlee 2011" dataset in this work. This dataset was chosen because it is one of the most commonly cited ground-truth bot datasets across existing bot detection literature [11, 23]. Moreover, this dataset is significantly larger than other labeled bot datasets, so

it provides an opportunity to see if the models can adapt to this scale. The original authors of the dataset, Lee et al., collected it using 60 social honeypot accounts [17]. Their honeypot accounts lured bot accounts by posting four different tweet variations including tweets with text, web URLs, "@" replies or mentions, and current trending topics on Twitter [17]. Similarly to the Arabic Honeypot Dataset collection, these honeypot accounts were intentionally designed to avoid interactions with real human users so they only mentioned other honeypot accounts in their tweets. The original authors' intuitions were that, "given the behavior of the social honeypot accounts, there is no reason for a user who is not in violation of Twitter's rules to be tempted to message or follow them" [17]. The social honeypot system ran from December 30, 2009, to August 2, 2010, after which it had attracted over 20,000 seemingly automated accounts. All of these 20,000 plus accounts had followed at least one honeypot account and were active for at least two hours on Twitter [17]. Collecting the 200 most recent tweets for each user yielded 2,353,473 tweets from these bots. When recrawled in October 2019, 4,716 of this dataset's bots had been suspended by Twitter.

Although this dataset is very thorough since it contains the largest single corpus of bot accounts to date, it is relatively (over 8 years) old now. Moreover, it could be hypothesized that current-day social bots have since evolved beyond traditional bot behaviors to more closely mimic human account characteristics and avoid existing bot detection methods. Therefore, this study includes a more recent dataset in the next subsection to attempt to combat these bot adaptations.

3.3 Cresci 2017 Dataset

The third dataset in this study combines multiple small datasets introduced by Cresci et al. [10] in their previous work on detecting social spambots on Twitter. The small datasets included in this "Cresci 2017" dataset are labeled as: Fake Followers, Genuine Accounts, Social Spambots #1, Social Spambots #2, Social Spambots #3, and Traditional Spambots #1, in the original work [10]. To give further insight into these datasets, the following is a summary on what type of bots each dataset contains. Social Spambots #1 contains social bots that were discovered during the 2014 Mayoral election in Rome, Italy, which were used to retweet a candidate within minutes of his original posting. Social Spambots #2 are social bots that promoted a hashtag, `#TALNTS`, which advertised a mobile phone application. Social Spambots #3 includes social bots that advertised products on *Amazon.com* by deceitfully spamming URLs which point to the products. Traditional Spambots #1 are 1000 bots that tweet malicious links and were captured using a honeypot. The Fake Followers small dataset is bot accounts purchased in April 2013 on fastfollowerz.com, intertwitter.com, and twittertechnology.com. All of the bot accounts were collected manually by the original authors. Combined there are 9,114 total bots between those datasets that are represented in this "Cresci 2017" dataset.

Table 2. Cresci 2017 Dataset breakdown

Sub-Dataset	**Accounts**	**Bot Description**
Fake Followers	3202	Purchased in April 2013 on sites such as fastfollowerz.com
Social Spambots #1	994	Found during the 2014 Rome Mayoral election
Social Spambots #2	3457	Found promoting #TALNTS, a mobile phone appication
Social Spambots #3	464	Found advertising products available on Amazon.com
Traditional Spambots #1	1000	Captured using a honeypot that tweet malicious links
Total bots	**9114**	

Chapter 4

METHOD

As described in Chapter 3, the first step to categorize bots by type is to obtain a set of bot accounts to analyze. But, since bot detection is not a contribution of this research, this process has been omitted in favor of using existing datasets to simulate this step. To this aim, this research utilizes the previously labeled Twitter bot datasets mentioned in Chapter 3. It should be noted that in the application of this work in the future, any state-of-the-art bot detection methods such as Indiana University's Botometer [11] can be used to obtain a set of bots prior to categorizing the bots into types. Because this method requires that some bots be previously labeled or discovered, it can be considered a weakly-supervised approach. However, the method can be fully unsupervised if unsupervised bot detection, as described in Section 2.2, is used instead of a supervised machine learning algorithm to find bots. Once a set of bot accounts has been obtained, features will need to be selected based on a given bot typology. Section 4.1 of this work describes such a typology used in previous research. Then, Section 4.2 specifies how the raw data from Chapter 3 was preprocessed using content-based feature extraction methods. After the features have been extracted in accordance with a given typology, an unsupervised clustering algorithm is run successively. Each run will modify the number of clusters the algorithm uses until it can be mathematically shown that each clustering contains accounts with different properties. (Unsupervised algorithms are listed in Section 4.3 and the metrics used to select this number of clusters are discussed in Section 4.4). Then, each clusters' properties can be used to identify types of bots according to the original typology.

However, in cases when it cannot be shown that there are different types of bots in the clusters, it may be possible that these clusters yield similar types of bots but show distinct domains that they participate in even within a single dataset. This explanation will still provide a better insight into the role that these bots play and where they exist. Thus, this methodology can be considered "automated" since bot types are assigned algorithmically.

4.1 Bot Typology

This research uses a typology of bots presented by Yang et al. [27] as previously discussed in Section 2.1. However, the methodology presented in this research can be applied on top of any typology, assuming that the chosen typology has a framework which correlates bot features to given bot types. Most likely there may be other types of bots in a given dataset, but the following types outline a baseline for future work on bot categorization.

- *Simple Bots*: As Yang et al. note in their typology, these bot accounts only post content automatically [27]. Thus, simple bots are the most common example of bots on social networks, in large part, due to their obvious bot behaviors. They usually have generic profile information, low numbers of followers and high numbers of friends, and many tweets which contain URL links. These URLs link users to external content that the creators of the bot accounts would like to promote. Sometimes these simple bots will utilize hashtags or mentions to make their URL links more visible. But the main indication of this type of bot is the large quantity of posts and the evident bot-like behavior of their posts.

- *Sophisticated Bots*: This type of bot exploits retweets, mentions, and hashtags in an effort to associate with humans. In the Yang et al. typology of bots, the authors note, "[sophisticated] bots can identify and generate appropriate content around specific topics to gain trust and attention from people interested in those topics" [27]. Sophisticated bots may appear to be human, but often push some unknown agenda through retweets or URLs.
- *Fake Follower Bots*: These bots seek to inflate the popularity of some tweet or user in order to lend credibility to that user. This can be done by increasing the follower count of the target users and by liking the target users' tweets. These bots may attempt to evade bot detection methods by random posting content and often avoid posting many tweets with mentions, URLs, or hashtags.
- *Botnets*: The final category of bots in the typology proposed by Yang et al., botnets are groups of bot accounts that use coordination to interact with each other. Botnets are often used to amplify certain tweets by human authors to make them seem more popular. Since detecting botnets usually requires knowledge about the graph structure of a given network, this type is excluded from this research in favor of categories that can be found without this knowledge.

4.2 Feature Extraction

Following the assumption that bot accounts are usually created to serve specific purposes, their tweet content can be a strong indicator to expose such potentially automated accounts [19]. As such, this work selects certain content-based features that are indicative of specific types of bots presented by Yang et al. [27]. For example, one feature included is the percentage of retweets as compared to all tweets. This metric

would be 0 if none of the tweets were retweets for a given user, a decimal between 0 and 1 depending on the frequency of retweets, and 1 if every tweet in the user's last 200 were retweets. This feature can be used to identify sophisticated bots since it allows a bot to easily copy what others have said in order to gain popularity on the network. Similarly, the percentage of mentions, percentage of URLs, and percentage of hashtags were all calculated. Furthermore, the utilization of mentions, URLs, and hashtags were determined by comparing the number of each to the total number of words tweeted by the user. Mentioning others is another way that sophisticated bots interact with human users on the network, URLs allow simple bots to promote a product or service, and hashtags have an amplifying effect to both bots.

Then, as an additional content-based feature, the author used latent Dirichlet allocation (LDA) [3] to determine if there was any topic significance within the data. LDA, which treats each document as a distribution over topics and where each topic is a combination of the vocabulary in the dataset, was previously proven useful for extracting latent semantics of documents [23]. In this work, each user is considered one document and the content of that document is the user's tweets (note due to Twitter's API rate limit this is a maximum of the 200 latest tweets per user). Thus, the LDA model can show the a topic distribution of each user meaning that it can determine which topics a user is most interested in. The assumption is that, since bots are naturally more interested in certain topics, denoting each account as a distribution over different topics may help to better identify these bots' intent. A separate LDA model was created for each dataset described in the previous section as well as a model that combined the three datasets together. This research includes the LDA topic probability features in order to determine if all the bots in the data operate within the same domain, or set of topics, on the network.

Table 3. List of extracted features

Feature	**Formula**	**Type of Bots**
Topic Probabilities	Generated using LDA model with 200 topics	–
Retweet %	$\frac{\text{\# of tweets that are retweets}}{\text{total \# of tweets}}$	Sophisticated bots
Hashtag %	$\frac{\text{\# of tweets that contain at least 1 hashtag}}{\text{total \# of tweets}}$	Sophisticated bots
Hashtag Utilization	$\frac{\text{total \# of hashtags}}{\text{total \# of words in all tweets}}$	Sophisticated bots
Mention %	$\frac{\text{\# of tweets that contain at least 1 user mention}}{\text{total \# of tweets}}$	Sophisticated bots
Mention Utilization	$\frac{\text{total \# of mentions}}{\text{total \# of words in all tweets}}$	Sophisticated bots
URL %	$\frac{\text{\# of tweets that contain at least 1 URL}}{\text{total \# of tweets}}$	Simple bots
URL Utilization	$\frac{\text{total \# of URLs}}{\text{total \# of words in all tweets}}$	Simple bots

A complete list of the features used in this study is shown in Table 3 along with the corresponding type of bots that this feature may be indicative of.

Table 4. Comparison of unsupervised clustering algorithms

Algorithm	**Parameters**	**Distance Measure**	**Use Case**
k-Means++	Number of clusters	Squared Euclidean	General purpose, Even cluster size, Not many clusters
Gaussian Mixture Models	Number of clusters or distance threshold	Mahalanobis	Good for density estimation
Ward Hierarchical Clustering	Many	Euclidean	Many clusters, possibly connected clusters

Table 4 compares the unsupervised clustering algorithms described in Section 4.3.

4.3 Unsupervised Clustering Algorithms

The following unsupervised cluster algorithms were utilized for this study. All of the algorithms were implemented in Python 3 using the Scikit-Learn package.

- *k-Means++* [2]: The traditional k-Means algorithm is widely used for general purpose clustering tasks. The goal of the algorithm is to choose some k cluster centers such that the centers minimize the sum of the squared distances between each point and its closest center [2]. One of the drawbacks on k-Means is that is converges to a local minimum instead of the global minimum, since the cluster centers are randomly initialized. k-Means++ attempts to mitigate this risk by calculating a more precise starting configuration prior to training. This algorithm will be referred to simply as k-Means moving forward in this study.
- *Gaussian Mixture Models* [24]: A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalizing k-Means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians. This algorithm was chosen because it does not use a Euclidean distance measure as k-Means does.
- *Ward Hierarchical Clustering* [25]: This algorithm is a general agglomerative hierarchical clustering procedure, where the criterion for choosing the pair of clusters to merge at each step is based on the optimal value of an objective function. In this work, the objective function sought to minimize the variance of the clusters being merged. Ward hierarchical clustering may be capable of detecting more structure within the data than the other algorithms.

4.4 Metrics for Unsupervised Cluster Evaluation

In order to solve the problem of determining which types of bots are found within a given dataset, first, one can determine how many different types of bots exist in that dataset. An empirical way of determining how many different types of bots exist within a dataset is by performing unsupervised clustering while modifying the number of clusters until it can be mathematically shown that each clustering contains accounts with different properties. Therefore, this study implores two different metrics that are commonly used to select the optimal number of clusters for a given clustering algorithm. By using the optimal number of clusters, it is empirically shown the clusters are separate and distinct, thus each cluster may contain different types of bots.

The first of these metrics is called Silhouette Analysis [21] and it can be used to study the separation distance between clusters that are provided by some unsupervised learning algorithm. This measure has a range of $[-1, 1]$. Silhouette coefficients (as these values are referred to as) near $+1$ indicate that the sample is far away from the neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters and negative values indicate that those samples might have been assigned to the wrong cluster. To get the coefficients for each $x \in C_i$, the *Silhouette Score* [21] can be computed as:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{4.1}$$

$$a(i) = \frac{1}{|C_i| - 1} \sum_{y \in C_i, x \neq y} d(x, y) \tag{4.1a}$$

$$b(i) = \min_{j \neq i} \frac{1}{|C_j|} \sum_{z \in C_j} d(x, z) \tag{4.1b}$$

where x, y, and z are given instances, C_i is a cluster assignment, and $|C_i|$ is the magnitude of a cluster or the number of instances assigned to that cluster.

Once each silhouette coefficient has been calculated, a silhouette plot can be generated. This plot displays a measure of how close each point in one cluster is to points in the neighboring clusters and thus provides a way to assess parameters like number of clusters visually. The optimal silhouette plot should be roughly equivalent in size and thickness for each cluster, demonstrating that the clusters are well separated.

The second commonly used metric to determine the optimal number of clusters for unsupervised clustering algorithms is the *Caliński-Harabasz Index* [4] which is also known as the Variance Ratio Criterion. The metric is defined as Equation 4.2 below:

$$\frac{SS_B}{SS_W} \times \frac{N-k}{k-1} \tag{4.2}$$

$$SS_B = \sum_{i}^{k} \sum_{x} \|x - \mu_i\|^2 - \sum_{i}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{4.2a}$$

$$SS_W = \sum_{i}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{4.2b}$$

where k is the number of clusters, N is the total number of instances (data points), x is a given instance, C_i is a cluster assignment, μ_i is a cluster centroid, SS_W is the overall within-cluster variance, and SS_B is the overall between-cluster variance.

For clarity, SS_B measures the variance of all the cluster centroids from the dataset's grand centroid. Hence, a large SS_B value means that all of the centroids are well separated from each other. Conversely, SS_W measures the density of the cluster, or how close each of the points are to the centroid. Intuitively, SS_W will keep decreasing as the number of clusters goes up since each cluster becomes smaller and tighter. So Caliński and Harabasz [4] reasoned this SS_W value would quickly decrease until the optimal number of clusters and then, after that number, the decrease would become less drastic. Therefore, for the Calinski-Harabasz Index, the ratio of $\frac{SS_B}{SS_W}$ will be largest at the optimal clustering size.

Chapter 5

EXPERIMENTS AND FINDINGS

5.1 Experiment 1: Validating the Data

The first experiment performed in this study was an attempt to determine if aggregating the datasets would produce meaningful results when trying to demonstrate that there are different types of bots that are not unique to one datset. So, it was necessary to validate that clustering algorithms could be utilized across the entire data and that these algorithms would not produce trivial results. For example, a trivial clustering might separate the data into distinct clusters each containing a different original dataset. To do this validation, the three datasets (recall the Cresci 2017 dataset was actually comprised of five smaller datasets, for a total of seven datasets overall) in Chapter 3 were combined and the features were extracted as described in Section 4.2. Then, the three clustering algorithms (k-Means, GMM, and Ward Hierarchical clustering from Section 4.3) were used to see if it was possible to separate each individual dataset from the combined data. Since it was already known that were seven datasets in the combined data, the analysis was done with only $k = 7$ clusters. The assumption was that if the datasets were completely separable, then there would be no reason to attempt to cluster any other number of clusters since seven should produce the optimal clusters. Figure 1 shows an example cluster decompositions for this experiment using k-Means. The figure displays the number of bots within a given cluster as well as which proportions of those bots belong to each dataset. In this example, almost 60% of the bots assigned to the Cluster 1 are

from the Caverlee dataset, 33.7% are from Fake Followers, 5.4% of Cluster 1 bots are from the Morstatter dataset, and the rest are from Traditional Spambots #1. While Clusters 1-4 do not seem to be completely represented by one specific dataset, Clusters 5, 6, and 7 almost exclusively contain the Social Spambots #1, Social Spambots #3, and Social Spambots #2, respectively. This raised some suspicions that perhaps the Social Spambots (#1, #2, and #3) datasets contained features much different than the other datasets. In that case, comparing these datasets to the others would be ineffective due to the inherent bias of the data. For example, if a dataset contained only combinations of numbers and no words, then clustering using content-based features would always separate that data from other datasets that contain natural language tweets. In order to confirm this suspicion on the bias in the Social Spambots datasets, the LDA topic probabilities of each dataset were individually analyzed. The result was that a majority of bot accounts in each dataset shared the exact same topic with the largest probability. For instance, in Social Spambots #1 almost every bot had the same most contributing LDA topic. Figure 3 shows this analysis. Thus, these Social Spambots datasets were indeed biased. To validate this claim, an additional experiment was performed to do a traditional bot detection task of distinguishing bots from humans. One of the results is shown in Figure 2 for Social Spambots #1. This figure shows that almost all of the bots were correctly classified using solely the tweet content features and unsupervised clustering (the f_1 score was 99.56% for k-Means, 99.56% for GMM, and 99.14% for Ward Hierarchical clustering). Therefore, these datasets (Social Spambots #1, #2, and #3) were removed from the rest of the experiments due to the existing data bias. It is assumed that the rest of the dataset should avoid the limitation of data bias because the content of the bots in those datasets are not obviously separable and appear in more than one cluster.

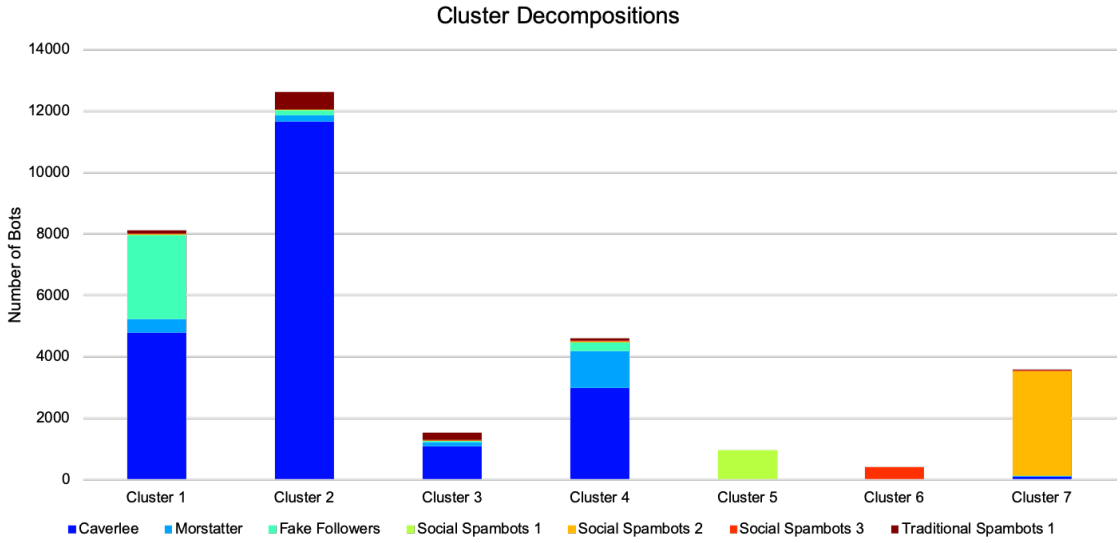

Figure 1. Cluster decompositions using k-Means with $k = 7$. Social Spambots #1, Social Spambots #2, and Social Spambots #3 datasets are nearly separable.

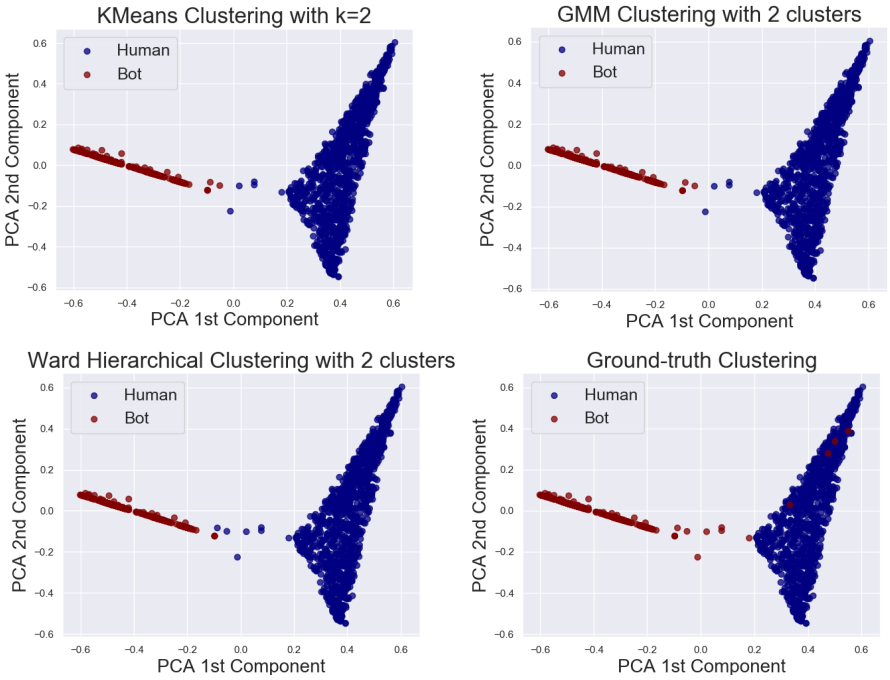

Figure 2. Empirical demonstration of dataset bias in the Social Spambots #1 dataset using clustering to do traditional bot detection. Principle Component Analysis (PCA) [16] was performed prior to graphing in 2-dimensions but the unsupervised algorithms were trained and tested without the use of dimensionality reduction.

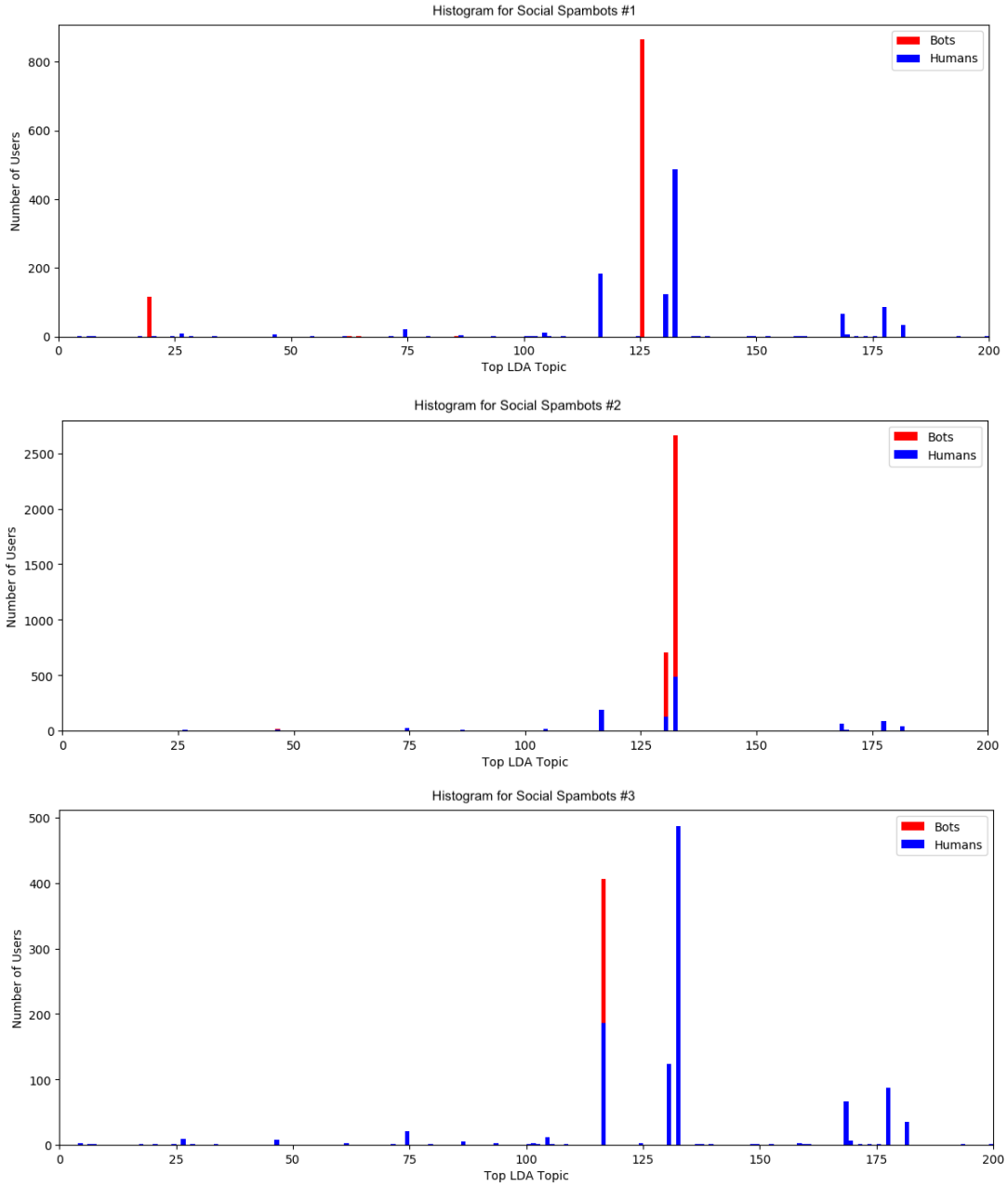

Figure 3. Analysis of the largest (top) LDA topic probabilities for the Social Spambots (#1, #2, and #3) datasets, respectively from top to bottom. Note the humans accounts are the same for each dataset since there was only one set of these users in the Cresci dataset. The results show that all of the bots in each dataset roughly share the top LDA topics meaning their content is extremely similar to each of the other bots in that datset. Due to this content bias in the datasets, these datasets were removed for future experiments.

5.2 Experiment 2: Choosing the Optimal Number of Clusters

Since the collection of Social Spambots datasets were determined to be biased in the previous experiment, they were excluded from this experiment. So, this experiment was completed on the combined data of the Caverlee 2011, Morstatter 2016, Fake Followers, and Traditional Spambots #1 datasets. This experiment attempted to use the metrics from Section 4.4, the Silhouette Score and the Caliński-Harabaz Index, to determine if there was an optimal number of clusters for each of the three clustering algorithms. Figure 4 shows eight silhouette plots to provide a complete picture of a Silhouette Analysis for k-Means. The clusters are plotted individually and the silhouette coefficients for that cluster are shown as the area under each curve. The red dashed-line represents the average silhouette coefficient value across all clusters. As mentioned in Section 4.4, the optimal cluster size should be represented by the silhouette plot where each cluster is nearly equivalent in size and all coefficient values should be positive. However, from the results, it is hard to determine which number of clusters is optimal using this metric. The plots for $k = 3$ and $k = 4$ have roughly the same average silhouette coefficient as the plot for $k = 2$. But in $k = 2$, and from $k = 5$ to $k = 9$, the largest cluster contains some individual negative silhouette coefficients. Figure 5 shows the Caliński-Harabaz Index value for each cluster size k. While the results of the Caliński-Harabaz Index varied by algorithm, there was a general trend where the index value decreased as the number of clusters increased. Overall, k-Means produced the best results for this metric regardless of the cluster size. Combining the results of the Silhouette Analysis and Caliński-Harabaz Index, the optimal number of clusters for the k-Means clustering algorithm is 3 or 4. As such, the following sections examine both of these number of clusters.

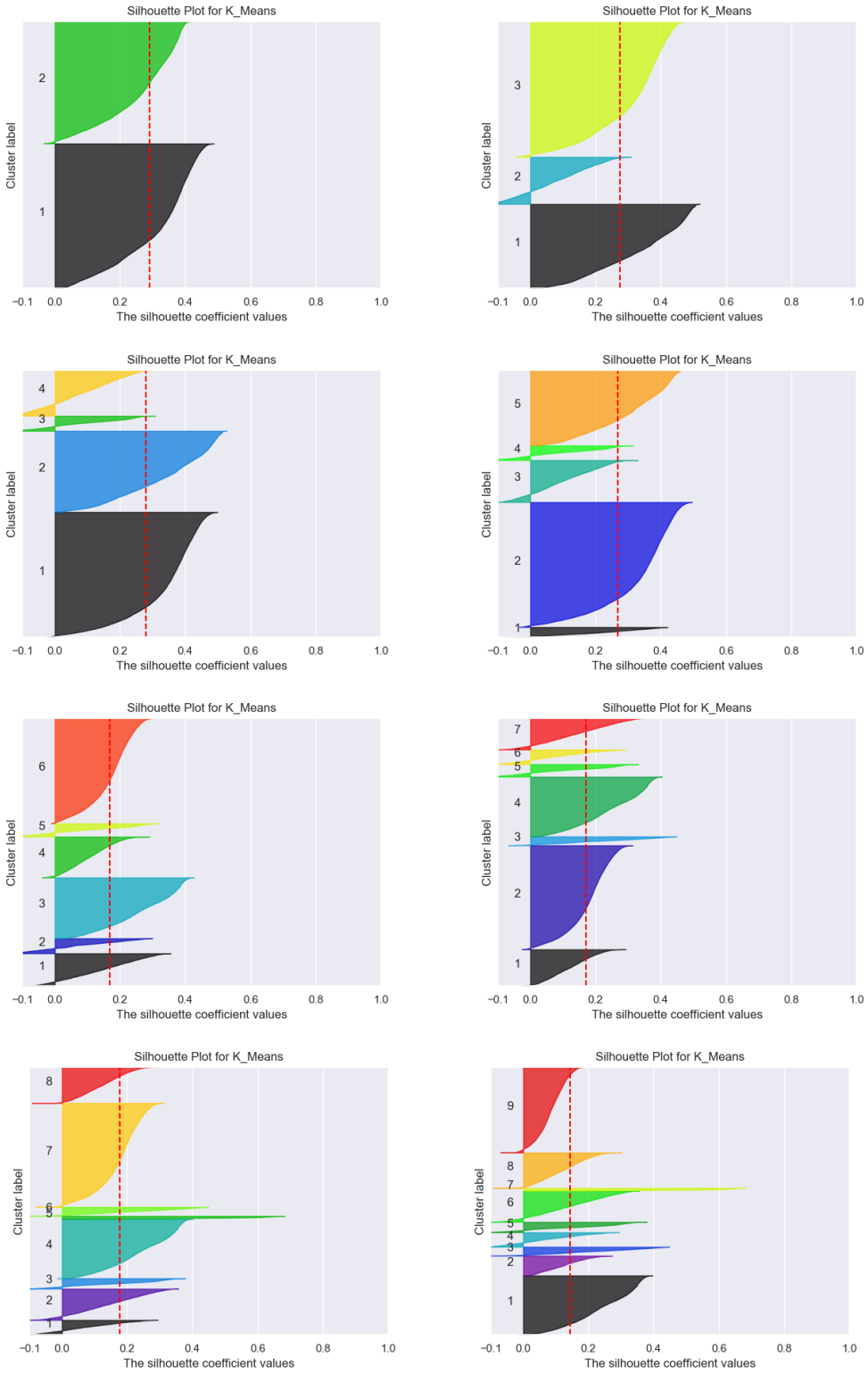

Figure 4. Silhouette Analysis for k-Means Clusters

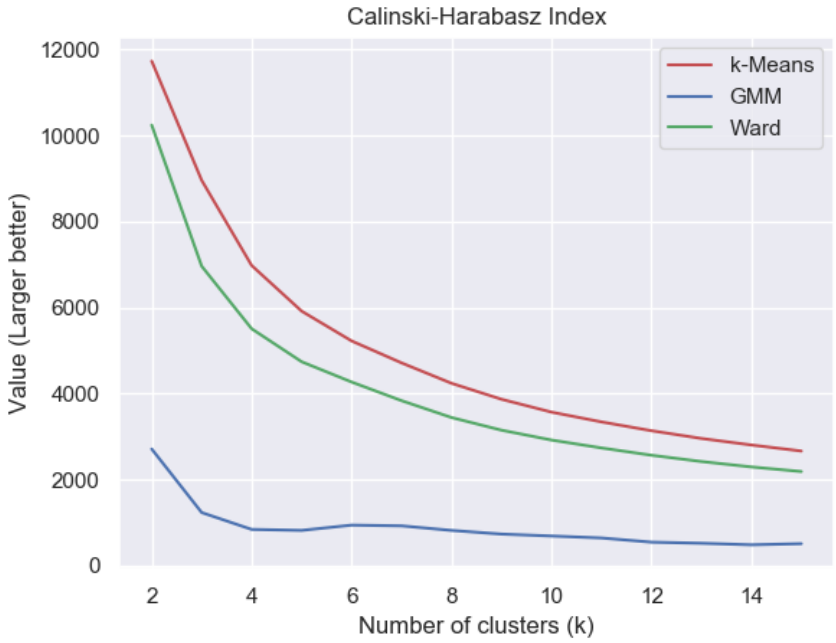

Figure 5. Calinski-Harabasz Index for Each Clustering Algorithm. The results, combined with the Silhouette Analysis, show that k-Means with $k = 3$ and $k = 4$ produces the best (most well-separated and dense) clusters.

Figure 6. Word Clouds for k-Means with $k = 4$. This is introduced in Section 5.3.

5.3 Experiment 3: Determining Domain Significance in Clusters

The next experiment was conducted to determine if the domain that the bots were present in had any effect on the final clusters obtained using the unsupervised learning algorithms. First, the LDA topic probabilities were examined for each cluster using the same method as Experiment 1 (comparing the "top" topics with the largest probability). It was challenging to determine if the top topics for each cluster correlated back any one of the original datasets, especially since the Caverlee dataset contained the largest number of bots in each cluster. But, it was clear that the aggregated topic probability distributions were somewhat similar for each cluster. Next, Word Clouds were generated for each cluster in an attempt to identify topics that were prevalent in only some of the clusters. This was done for all of the clustering algorithms at their respective optimal number of clusters. This work used word frequency, i.e. Word Clouds, in an attempt to find similarity between the clusters. Other NLP techniques exist to find differences between or highlight the uniqueness of each cluster, but that was not the focus of this experiment. An example of the Word Cloud output is shown in Figure 6 for k-Means with $k = 4$ clusters. By examining these Word Clouds, it is obvious that all four of the clusters found using k-Means algorithm have similar topics. The top words shared between the bots in each cluster were "free" and "twitter" for all four clusters. Additionally, words such as: "money", "news", "blog", "online", and "marketing" can all be found in each of the clusters. Since the top words are very similar between each of the clusters, it can be assumed that the topics are similar. Thus, the domain where the bots are participating in is similar if not the same for all clusters, which means the bots are clustered based on other features.

5.4 Experiment 4: Clustering Bots by Type

In this experiment, unsupervised machine learning was used to find different types of bots on the combined data previously used in Experiments 2 and 3. First, these algorithms were run with the optimal number of clusters determined in Experiment 2. Then, Word Cloud and LDA topic probability distribution analyses were used in Experiment 3 to check that the domain was consistent across all of the clusters. To ultimately determine the types of bots, the actual features of the accounts that had been grouped in each cluster were analyzed. The results for this comparison are shown in Figures 7 and 10 for $k = 3$ and $k = 4$, respectively. In Figure 7, Cluster 1 has less than 10% of all major features except for URLs. These bots may only interact indirectly with other users in the network, and do not attempt to spam or directly contact others using mentions or hashtags. So, it may be assumed that these are fake follower bots. Cluster 2 has high interactions with almost 70% of their tweets mentioning other users. These bots could fall into the category of sophisticated bots. Sophisticated bots often disguise themselves as human users in an attempt to convince others to promote their content. Cluster 3 contains bots that post URLs in 90% of their tweets but do not combine these with any other text features. These bots can be considered simple bots since they are usually the easiest for humans to identify. Figure 10 shows the clusters for k-Means with $k = 4$. Here, Cluster 1 has similar features as Cluster 3 in Figure 7. The high URL rate is a defining characteristic of simple bots. Likewise, Cluster 2 in Figure 10 is similar to Cluster 1 in Figure 7. Since this group does not interact much with others, these accounts could be indicative of fake follower bots. Cluster 3 contains a type of bots which was not seen in the $k = 3$ clusters. This group uses many URLs (over 75% of their tweets) but also uses hashtags to promote

these URLs. Since there is no category between simple and sophisticated bots, this group can still be labelled as simple bots, but future typologies should be expanded to include this phenomenon. Finally, Cluster 4 is the sophisticated bot group which often mentions other users and may retweet others in an attempt to gain viewership.

To further understand the differences of each cluster, or bot type, two additional analyses were conducted for both the $k = 3$ and $k = 4$ case. First, the current state of the bot accounts in each cluster were checked using the Twitter API in October 2019. The state of each account could be one of three labels: 1) *active* meaning that the accounts could still participate on the network, 2) *suspended* meaning that the accounts had violated Twitter's terms and conditions and was temporarily blocked from network participation, or 3) *deleted* meaning that the original authors or Twitter had removed the account from the network. This current state analysis is shown in Figures 8 and 11. Interestingly, over 40% of the bots in the clusters previously associated to fake follower bots were suspended by Twitter for both $k = 3$ and $k = 4$. Subsequently, the process conducted in Experiment 1 was repeated to determine the decomposition of each cluster. This is shown in Figures 9 and 12. Since this step relies on having ground-truth information about the dataset, it does not need to be performed to group bots by type. The only necessary part of the methodology is comparing the cluster features to determine types. However, this analysis ensured that each of the clusters contained bots from several of the original datasets and helped to validate the type labels. While the Caverlee dataset made up the largest portion of each cluster due to the sheer size of the dataset, the majority of the Morstatter and Fake Followers datasets were found in the clusters corresponding to sophisticated bots and fake follower bots, respectively. These two analyses help to prove that common types of bots can be identified across the datasets.

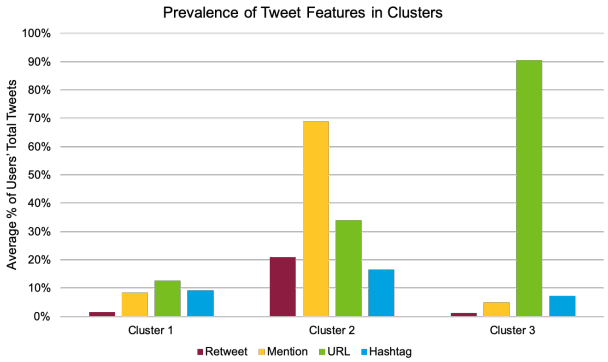

Figure 7. Features by cluster for k-Means with $k = 3$. Cluster 1 is labeled as fake follower bots, Cluster 2 sophisticated bots, and Cluster 3 simple bots.

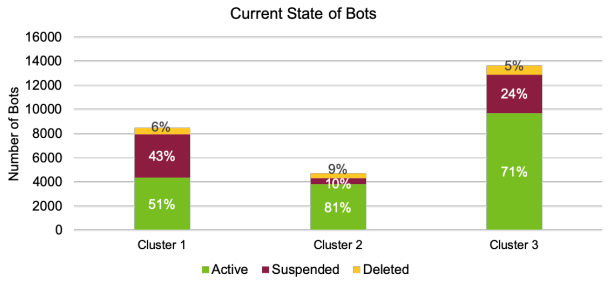

Figure 8. Current state of bots as of October 2019. Cluster 1, fake follower bots, had both the largest number and percentage of bots suspended compared to the other clusters. On the other hand, Cluster 2, sophisticated bots, had the least.

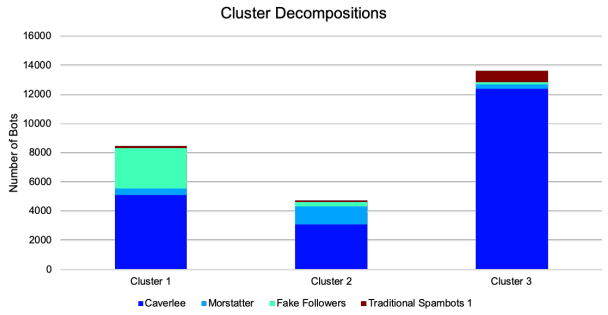

Figure 9. Decomposition of clusters by dataset. Cluster 1, which was previously labeled fake followers bots, contains 86.41% of bots in the Fake Followers dataset. 61.03% of bots in the Morstatter dataset fall in Cluster 2 (sophisticated bots).

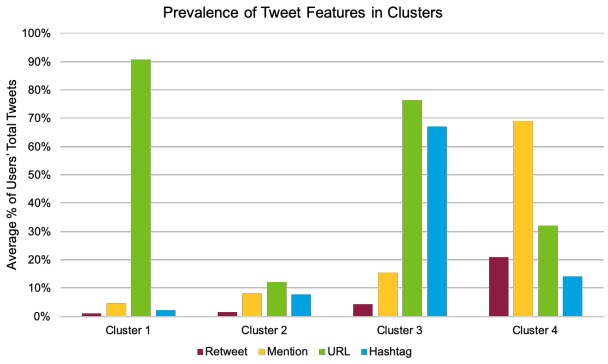

Figure 10. Features by cluster for k-Means with $k = 4$. The clusters are labeled as: simple bots, fake follower bots, simple bots, and sophisticated bots from left to right.

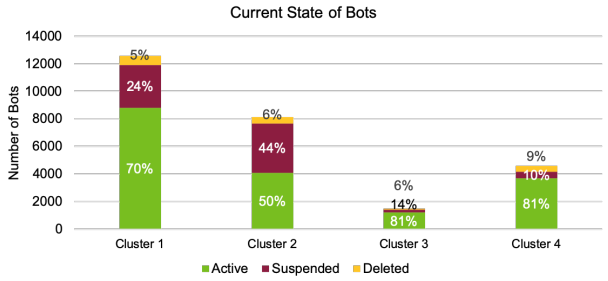

Figure 11. Current state of bots as of October 2019. Cluster 2, fake follower bots, had both the largest number of bots suspended compared to the other clusters. Cluster 1 had the 2nd largest but Cluster 3 had the least, though both were simple bots.

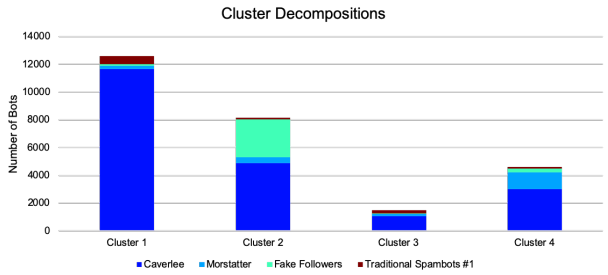

Figure 12. Decomposition of clusters by dataset. 85.43% of bots from the Fake Followers dataset are grouped in Cluster 2. 59.40% of the bots from the Morstatter dataset are in Cluster 4. The number of bots in each cluster is similar to the $k = 3$ case previously shown in Figure 9.

5.5 Experiment 5: Observing Bot Types Over Time

This experiment was performed to determine if there were different distributions of bot types in the datasets since these datasets span multiple years of the Twitter data. In other words, have bot types changed over time? To this aim, Experiment 5 aggregated the "created_at" tweet feature to group tweets by month and then plotted the number of tweets by month for each dataset. This was further broken down by cluster as previously obtained by the k-Means with $k = 4$ during the Experiment 4. Figure 13 shows this distribution of each cluster in terms of the number of tweets per month by bots in that clusters. The number of tweets is only compared within individual datasets since the dataset collection techniques differed. For instance, the Morstatter dataset is much different in terms of the number of tweets by month since the authors used the honeypot collection method to collected user_ids for the bot accounts [19]. Once all bot user_ids had been collected, in 2015, the authors used the Twitter Searching API to crawl the bots' timelines and get their latest tweets. By contrast, the other datasets were tracking the tweets of bots real-time for several months using the Twitter Streaming API. This collection difference explains the difference in the number of tweets by month in Figure 13. The datasets are shown in order of time from oldest, Caverlee and Traditional Spambots #1, to the latest. It is clear, by examining the figure, that the oldest datasets have a larger percentage of tweets from bots in Cluster 1, or simple bots, than the others. The Fake Followers dataset is dominated by a majority of tweets from the bots in Cluster 2, or fake follower bots. Then, the Morstatter dataset is mostly comprised of tweets from bots in Cluster 4, or sophisticated bots. This shows that the most prevalent bot type has changed over time and all types of bots can be found in each dataset.

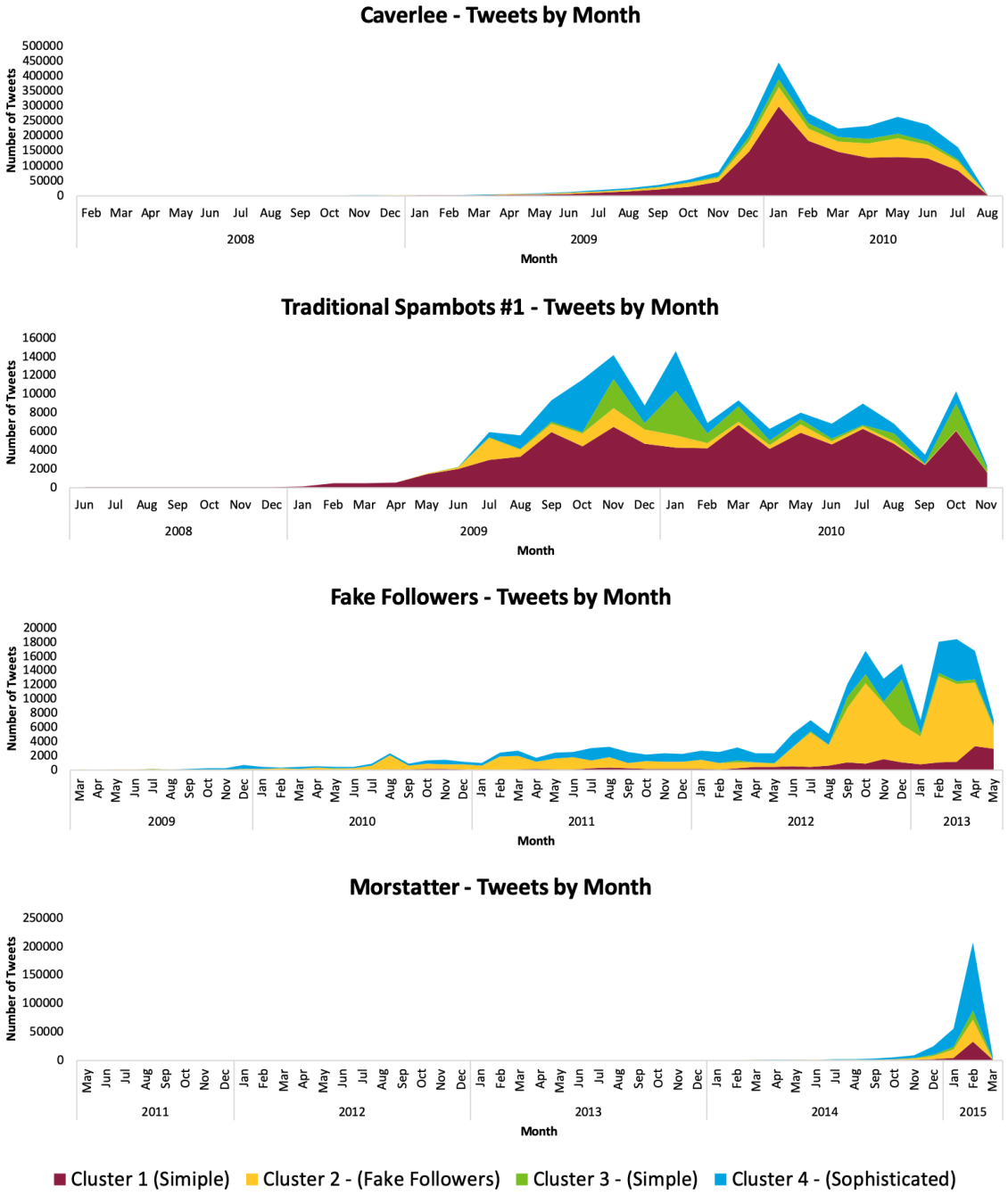

Figure 13. Temporal analysis of bot types by dataset. The area under the curve shows the distribution of the number of tweets by cluster. It is clear Cluster 1 (simple bots) is most prevalent in the first two (older) datasets. Eventually there is a transition in the distribution of tweets which represents more bots from Cluster 4 (sophisticated bots) are now present in the more recent dataset.

5.6 Experiment 6: Applying the Method to Individual Datasets

For the last experiment, the author wanted to cluster bots by type as in Experiment 4, but strictly cluster within one individual dataset rather than using the aggregated dataset presented in Chapter 3. This way, future researchers can be assured that the methodology presented in this work can be applicable to their data, even if they do not combine multiple datasets. So, the methodology from Chapter 4 was applied to the datasets individually and all prior knowledge about the datasets was ignored. k-Means, GMM, and Ward Hierarchial Clustering were all performed on each dataset and the Caliński-Harabasz Index and Silhouette Analysis were used to determine the optimal number of clusters. Using this method, the results for the Caverlee 2011 and Morstatter 2016 datasets were very similar to the results found in Experiment 4. The optimal number of clusters was three for both datasets and the analysis of the clusters' features yielded three unique types of bots that matched the types (simple, fake follower, and sophisticated) found in the typology by Yang et al. However, the third dataset, Cresci 2017, had much different results. The Silhouette Analysis and Caliński-Harabasz Index metrics found k-Means with $k = 8$ to be the optimal clusters.

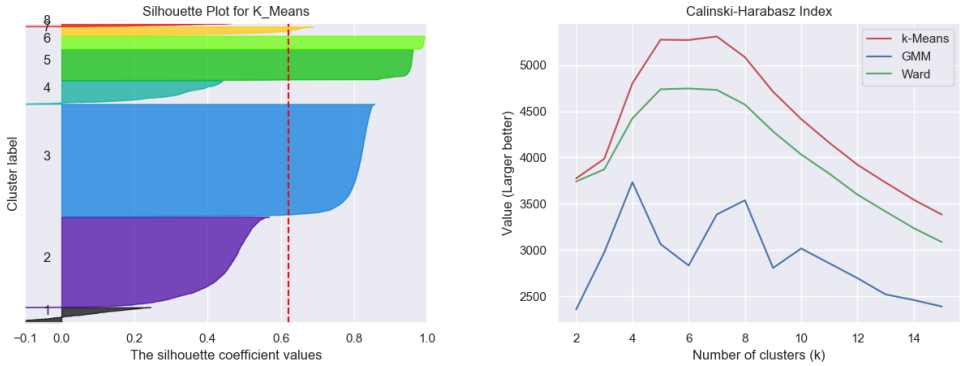

Figure 14. Silhouette Plot and Caliński-Harabasz Index for k-Means on Cresci Dataset. The optimal number of clusters was found to be $k = 8$ for this dataset.

The unique results for this Cresci dataset might be attributed to the fact that the authors combined several smaller datasets manually. Experiment #1 already showed that there was some sort of data bias in the Social Spambots #1, #2, and #3 datasets. If the dataset were from different domains or contained very different bot accounts as far as the bots' content, then it would make sense that they might be well separated into small clusters of distinguishable accounts. Figure 15 shows the prevalence of tweet features by cluster for this Cresci 2017 dataset. Using the typology proposed by Yang et al., Clusters 1 and 8 could be labeled as sophisticated bots, Clusters 4, 6, and 7 as simple bots, and Clusters 2 and 5 as fake follower bots. However, Cluster 3 does not fit within the existing typology as the bots in that cluster mention other users about 30% of the time but do not ever retweet them or post any URLs in their content. Future work should expand the typology to include this phenomenon.

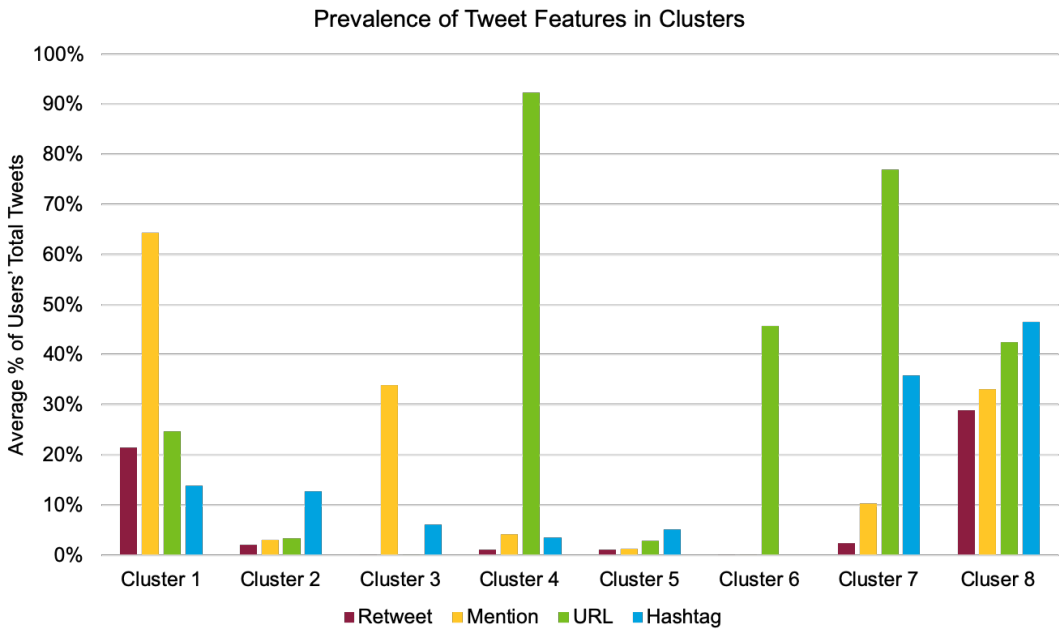

Figure 15. Features by cluster for k-Means with $k = 8$. Cluster 3 is much different from all of those in the previous experiments and, thus, could not be labeled within the three types presented by Yang et al.

Chapter 6

CONCLUSION AND FUTURE WORK

The contributions of this research are as follows: this work summarized the existing research of types of bots in social media networks and surveyed the current unsupervised bot detection models. Then, the author created an aggregated dataset and performed experiments to ensure that combined the dataset did not contain any bias and that all bots in the dataset operated within a single domain. Finally, the researcher experimentally demonstrated that it is possible to group bots by their respective types by testing several unsupervised machine learning and showing that k-Means could separate these bots into types. Analysis performed on the clusters obtained from these experiments proves that multiple types of bots can be found within a single domain and that the most prevalent type of bot on the social network changes over time.

As a continuation of this work, more features from the bot accounts should be extracted to solidify the existing types of bots and create more unique types as subcategories of the previous framework. For example, profile features, such as the ratio of friends to followers, can be used to further differentiate bots. Other possible features include the graph structure of the network in the form of connections between users either in their tweet interactions or their friend/follow network, account metadata like creation date or geo-location, and intent of users which could be found using natural language processing sentiment techniques. Additionally, the methodology presented in this work should be expanded to include other unsupervised algorithms like DBSCAN [12] which are capable of automatically calculating the optimal number

of clusters as opposed to using Silhouette Analysis or the Caliński-Harabasz Index to determine this number of clusters. This, in conjunction with adding more features, should make the final clusters more unique and, thus, make it easier to distinguish between different types of bots in the final analysis.

Another future direction of this research will be to choose one type of bot identified in this research and study the areas on the social media network where this type is found. Additionally, tracking this type of bot over time could indicate if or how that bot is evolving to evade detection within the network. Moreover, a given type of bots' topic probability distribution (found using LDA on the accounts' tweet corpus) can be compared to previous distributions to see which topic areas fluctuate the most. While comparing LDA topic probability distributions is somewhat trivial as it is obvious there will be certain changes to the topics as new online trends emerge over time, it might be possible to discover that bots within a certain domain either change their tactics or migrate to another domain entirely.

REFERENCES

[1] Faraz Ahmed and Muhammad Abulaish. "A generic statistical approach for spam detection in Online Social Networks". In: *Computer Communications* 36.10 (2013), pp. 1120–1129. DOI: 10.1016/j.comcom.2013.04.004.

[2] David Arthur and Sergei Vassilvitskii. "k-means++: The advantages of careful seeding". In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics. 2007, pp. 1027–1035.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.01 (2003), pp. 993–1022.

[4] Tadeusz Caliński and Harabasz JA. "A Dendrite Method for Cluster Analysis". In: *Communications in Statistics - Theory and Methods* 3 (Jan. 1974), pp. 1–27. DOI: 10.1080/03610927408827101.

[5] Nikan Chavoshi, Hossein Hamooni, and Abdullah Mueen. "DeBot: Twitter Bot Detection via Warped Correlation". In: *2016 IEEE 16th International Conference on Data Mining (ICDM)*. Dec. 2016, pp. 817–822. DOI: 10.1109/ICDM.2016.0096.

[6] Nikan Chavoshi, Hossein Hamooni, and Abdullah Mueen. "Identifying Correlated Bots in Twitter". In: *Social Informatics*. Ed. by Emma Spiro and Yong-Yeol Ahn. Cham: Springer International Publishing, 2016, pp. 14–21.

[7] Zhouhan Chen and Devika Subramanian. "An Unsupervised Approach to Detect Spam Campaigns that Use Botnets on Twitter". In: *CoRR* (Apr. 2018).

[8] Daniel Y. T. Chino, Alceu Ferraz Costa, Agma J. M. Traina, and Christos Faloutsos. "VolTime: Unsupervised Anomaly Detection on Users' Online Activity Volume". In: *SIAM International Conference on Data Mining SDM*. Philadelphia, PA, USA: SIAM, 2017. DOI: 10.1137/1.9781611974973.13.

[9] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. "DNA-Inspired Online Behavioral Modeling and Its Application to Spambot Detection". In: *IEEE Intelligent Systems* 31.5 (Sept. 2016), pp. 58–64. DOI: 10.1109/MIS.2016.29.

[10] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. "The Paradigm-Shift of Social Spambots: Evidence, Theories, and Tools for the Arms Race". In: *Proceedings of the 26th International*

Conference on World Wide Web Companion. (2017), pp. 963–972. DOI: 10.1145/3041021.3055135.

[11] Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. "BotOrNot: A System to Evaluate Social Bots". In: *The Web Conference*. 2016, pp. 273–274.

[12] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *Kdd*. Vol. 96. 34. 1996, pp. 226–231.

[13] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. "The Rise of Social Bots". In: *Communications of the ACM* 59.7 (June 2016), pp. 96–104. DOI: 10.1145/2818717.

[14] Zafar Gilani, Ekaterina Kochmar, and Jon Crowcroft. "Classification of Twitter Accounts into Automated Agents and Human Users". In: *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. ASONAM '17. Sydney, Australia: ACM, 2017, pp. 489–496. DOI: 10.1145/3110025.3110091.

[15] Robert Gorwa and Douglas Guilbeault. "Unpacking the Social Media Bot: A Typology to Guide Research and Policy". In: *Policy and Internet* (2018), pp. 1–30. DOI: 10.1002/poi3.184.

[16] Harold Hotelling. "Analysis of a complex of statistical variables into principal components." In: *Journal of educational psychology* 24.6 (1933), p. 417.

[17] Kyumin Lee, Brian David Eoff, and James Caverlee. "Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter." In: *Proceedings of the 5th International Conference on Web and Social Media (ICWSM)*. AAAI. The AAAI Press, 2011, pp. 185–192.

[18] Amanda Minnich, Nikan Chavoshi, Danai Koutra, and Abdullah Mueen. "BotWalk: Efficient Adaptive Exploration of Twitter Bot Networks". In: *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. ASONAM '17. Sydney, Australia: ACM, 2017, pp. 467–474. DOI: 10.1145/3110025.3110163.

[19] Fred Morstatter, Liang Wu, Tahora H. Nazer, Kathleen M Carley, and Huan Liu. "A New Approach to Bot Detection: Striking the Balance between Precision and Recall". In: *ASONAM*. IEEE. 2016, pp. 533–540.

[20] Richard J. Oentaryo, Arinto Murdopo, Philips K. Prasetyo, and Ee-Peng Lim. "On Profiling Bots in Social Media". In: *International Conference on Social Informatics* (2016), pp. 92–109. DOI: 10.1007/978-3-319-47880-7_6.

[21] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20 (Nov. 1987), pp. 53–65. DOI: 10.1016/0377-0427(87)90125-7.

[22] Stefan Stieglitz, Florian Brachten, Björn Ross, and Anna-Katharina Jung. "Do Social Bots Dream of Electric Sheep? A Categorization of Social Media Bot Accounts". In: *CoRR* (2017), pp. 1–11. DOI: 10.1007/s00253-010-2538-y.

[23] VS Subrahmanian et al. "The DARPA Twitter bot challenge". In: *arXiv preprint arXiv:1601.05140* (2016).

[24] Santosh Vempala and Grant Wang. "A spectral algorithm for learning mixtures of distributions". In: *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.* IEEE. 2002, pp. 113–122.

[25] Joe H Ward Jr. "Hierarchical grouping to optimize an objective function". In: *Journal of the American statistical association* 58.301 (1963), pp. 236–244.

[26] Chao Yang, Robert Harkreader, and Guofei Gu. "Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers". In: *IEEE Transactions on Information Forensics and Security* 8.8 (2013), pp. 1280–1293. DOI: 10.1109/TIFS.2013.2267732.

[27] Kai-Cheng Yang, Onur Varol, Clayton A. Davis, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. "Arming the public with artificial intelligence to counter social bots". In: *Human Behavior and Emerging Technologies* (2019), e115. DOI: 10.1002/hbe2.115.

[28] Chao Michael Zhang and Vern Paxson. "Detecting and Analyzing Automated Activity on Twitter". In: *Passive and Active Measurement*. Ed. by Neil Spring and George F. Riley. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 102–111. DOI: 10.1007/978-3-642-19260-9_11.