



Norwegian University of  
Science and Technology

# Manipulation and Deception with Social Bots: Strategies and Indicators for Minimizing Impact

**Geir Marius Sætenes  
Haugen**

Master in Information Security

Submission date: May 2017

Supervisor: Bernhard Hämmerli, IIK

Norwegian University of Science and Technology  
Department of Information Security and Communication





Norwegian University of  
Science and Technology

# Manipulation and Deception with Social Bots: Strategies and Indicators for Minimizing Impact

Geir Haugen

31-05-2017

Master's Thesis

Master of Science in Information Security

30 ECTS

Department of Information Security and Communication Technology  
Norwegian University of Science and Technology, 2017

Supervisor: Prof. Bernhard M. Hämmerli

## Preface

This thesis is written as a part of the Master of Science in Information Security -program at the Norwegian University of Science and Technology (NTNU), at campus Gjøvik. The work was carried out from January throughout May 2017. Guidance and supervision were performed by Prof. Bernhard Hämmerli. Prof. Hämmerli made me initially interested in the topic of social bots and suggested it as a topic for my Master's Thesis.

Suggested readers are students on a information technology-related Bachelor -or Master's track and/or anyone who have a background in information security/technology. Although the suggested readers are someone with IT-related background, I encourage people with other backgrounds to read the thesis because of the importance of the topic. Since most people are present on Online Social Networks (OSNs), social bots are something that everyone should know about. Being aware of social bots can help protecting users' privacy and raise information-sharing awareness online.

Hokksund, 31-05-2017

Geir Haugen

## Acknowledgment

I would like to thank my supervisor Bernhard Hämmerli. You have been of great support in completing the work in my thesis. Thank you very much for suggesting the theme of social bots.

Many thanks to everyone I have discussed my thesis with. This have been a good exercise for me and have motivated me further in my work. Thank you all for showing interest and for listening to what I were working with.

Also many thanks to my pregnant better half. You have been of great support throughout my work. You have helped me to focus on the work in times where motivation have not been on the top.

I would also like to thank my hamster Viski for remembering me to take breaks in the work. This have been very helpful.

G.H.

## Abstract

Social bots are automated programs that control Online Social Network (OSN) accounts. These social bots can be used to spread spam and malware and to manipulate people online. Since so many people are present on OSNs, it makes a great arena for manipulation. Social bots are used by celebrities and politicians to inflate popularity. A recent study of the 2016 U.S. presidential election showed that social bots were extensively used by both parties. Although most people are aware of social bots on OSNs, they do not know about the underlying dangers of them. Because people add unknown accounts to their friends list, the bots keep spreading spam and keeps influencing public opinion. The use of social bots can threaten the democracy and deceive people online. Therefore it is important that we suppress the power that these social bots inhibit.

Advancements in certain fields such as AI makes it harder to differentiate between human and social bot activity. New social bots coming out renders the previous detection methods useless. This raises the importance of developing new methods for detection. By doing this, we can stay ahead of the social bots and prevent them from spreading malware and causing manipulation.

In this thesis, the different aspects of social bots are discussed. The thesis look at the different types of social bots and elaborates on how they operate and cause harm. Social bot anatomy and their intentions are also discussed. The thesis mainly focus on how to indicate and detect social bots on Twitter. Existing approaches and solutions for detecting social bots are described and discussed.

The main contribution for the work in this thesis involved development of a indicator program for indicating presence of social bots. The indicators incorporated in the program are partly based on new variations of existing methods and partly new ideas of features to analyze. The indicators are based on both behavior and feature analyses. Instead of having a detection program, the program developed in this thesis is meant as a tool for indicating social bot activity. The idea behind the solution as a whole is that if indicators are present, it should be further inspected by a human. Humans can see inconsistencies that are hard to define in algorithms. And by implementing crowdsourcing as a human part, this process can be highly effective. The developed program analyze several different features of a Twitter account. Included is analyses of profile features, tweet timings and analyses of URLs in tweets. By analyzing many different features, the program can indicate presence of many different types of bots.

New contributions are highly needed to fight social bots, the work in this thesis is meant as a contribution to this fight.

## Abbreviations

- AI** - Artificial Intelligence
- API** - Application Programming Interface
- CAPTCHA** - Completely Automated Public Turing test to tell Computers and Humans Apart
- DoS** - Denial of Service
- DPA** - Digital Personal Assistant
- ID** - Identification
- IRC** - Internet Relay Chat
- NLP** - Natural Language Processing
- OSN** - Online Social Network

## Contents

<b>Preface</b> . . . . .	<b>i</b>
<b>Acknowledgment</b> . . . . .	<b>ii</b>
<b>Abbreviations</b> . . . . .	<b>iv</b>
<b>Contents</b> . . . . .	<b>v</b>
<b>List of Figures</b> . . . . .	<b>vii</b>
<b>List of Tables</b> . . . . .	<b>viii</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Problem description . . . . .	2
1.2.1 The subject . . . . .	3
1.3 Research Questions . . . . .	4
1.4 Purpose of the study . . . . .	4
1.5 Methodology . . . . .	4
1.5.1 Information gathering . . . . .	5
1.6 Definitions . . . . .	6
1.7 Literature . . . . .	6
1.8 Assumptions, Limitations and Delimitations . . . . .	7
<b>2 Online Social Networks</b> . . . . .	<b>8</b>
2.1 Different OSN platforms . . . . .	9
2.1.1 Purpose and methods of communication . . . . .	9
2.1.2 OSNs and privacy . . . . .	9
2.2 People’s behavior in relation to social bots . . . . .	10
<b>3 Social bots</b> . . . . .	<b>12</b>
3.1 Classification of bots . . . . .	12
3.1.1 Spam bots . . . . .	13
3.1.2 Chat bots . . . . .	14
3.1.3 Information collecting bots . . . . .	15
3.2 Anatomy of social bots . . . . .	15
3.3 Social bots and AI . . . . .	17
3.3.1 AI gone wrong . . . . .	18
3.3.2 The Turing test . . . . .	18
3.4 The intentions of social bots . . . . .	19
3.4.1 Purchasing fame . . . . .	20
3.5 How are bots successful or unsuccessful? . . . . .	21
3.5.1 What makes social bots harder or easier to detect? . . . . .	21
3.5.2 How does a social bot remain undetected? . . . . .	21



3.6	Beneficial Parties . . . . .	22
3.7	Preventive mechanisms and methods for social bots . . . . .	22
3.7.1	CAPTCHA . . . . .	22
3.7.2	Modeling user’s susceptibility . . . . .	24
<b>4</b>	<b>Social bot detection . . . . .</b>	<b>26</b>
4.1	Detection approaches . . . . .	27
4.1.1	Behavioral analysis . . . . .	27
4.1.2	Content and language analysis . . . . .	29
4.1.3	Crowdsourcing . . . . .	30
4.1.4	Honeypots . . . . .	31
4.2	Publicly available solutions . . . . .	32
4.2.1	BotOrNot . . . . .	32
4.2.2	Twitter Audit . . . . .	34
4.2.3	TweetDeck . . . . .	34
4.3	A neverending arms race? . . . . .	35
4.3.1	Advancements of social bots . . . . .	35
4.3.2	Proposed solution to the arms race . . . . .	35
<b>5</b>	<b>The proposed indication scheme . . . . .</b>	<b>37</b>
5.1	Components . . . . .	37
5.1.1	The Twitter REST API . . . . .	37
5.1.2	Python . . . . .	37
5.1.3	Virustotal . . . . .	38
5.2	Data collection . . . . .	40
5.3	The solution . . . . .	41
5.3.1	Program Flow . . . . .	42
5.3.2	User analyses . . . . .	43
5.3.3	Tweet analyses . . . . .	45
5.3.4	Score calculation . . . . .	51
<b>6</b>	<b>Future work . . . . .</b>	<b>53</b>
<b>7</b>	<b>Conclusion . . . . .</b>	<b>55</b>
	<b>Bibliography . . . . .</b>	<b>56</b>

## List of Figures

1	Anatomy of a social bot. . . . .	16
2	A conversation with cleverbot. . . . .	17
3	An example of a CAPTCHA. . . . .	23
4	Ranking of the top features for determining susceptibility to social bot attacks. . . . .	25
5	Tweet time distribution of automated processes and humans . . .	28
6	Result yielded by BotOrNot for @realDonaldTrump. . . . .	33
7	Result yielded by Twitter Audit for @realDonaldTrump. . . . .	34
8	The flow of the proposed solution. . . . .	43
9	Twitter's default profile picture. . . . .	44
10	Tweet hours of the account boy2bot compared to the average of a set of accounts. . . . .	46
11	Tweet hours of the account justinbieber compared to the average of a set of accounts. . . . .	47
12	Example of different tweet sources. . . . .	49

## List of Tables

1	Examples of types of names. . . . .	44
2	Example score of a fictional analysis. . . . .	51

## 1 Introduction

Since the start of the 21st century, the number of Online Social Networks (OSNs) and blog services have exploded. At the start of the 21st century, more people had acquired a home computer. And Internet started to become a regular installment in homes around the world. OSNs were and are so successful because they can make people closer to one another even if they are in a different city or on the other side of the planet. One of the most successful OSNs, Facebook (founded 2004), started off as a social network for students at some universities in the U.S. But soon facebook expanded its customer segment to the whole world. From 1 million active users in December 2004, facebook has grown to have 1.23 billion daily active users [1]. Another OSN, or more specifically a microblog platform, Twitter, have drawn users since 2006 and now have over 300 million monthly active users (as of June 30, 2016) [2]. The high number of users present on OSNs, makes a great arena for manipulation, spreading of malware and personal data collecting. Social bots poses a great threat to other users on OSNs. Social bots that appear to be human can be used to deceive real people. They can be used to lure people into receiving malware, influence people's opinion and to infiltrate social networks. Social bots have been given more attention in the recent years. This is because of the active use of them in political discussions and elections. Although it cannot be measured that the use of social bots have influenced elections, they are being used. Social bots have been around for a long time, but as technology have developed, and more people are present on OSNs, the use of them is increasing.

"We can only see a short distance ahead, but we can see plenty there that needs to be done."

–**Alan Turing**

The quote above is taken from Alan Turing's work "Computing Machinery And Intelligence" [3]. What Turing means here is that we can only know what will happen in near future, still there is many things that must be done. The quote can be related to the fight against bots as well as other information security aspects. In the context of social bots, the quote means that we know that there is many things that need to be done, and we also need to be ready for the future. Social bots are becoming more advanced by time, implementing AI and machine learning techniques to appear more human. Because the bots are

becoming more advanced, we also need more advanced methods to be able to keep detecting them. If we stick with the detection methods we currently have, the bots will advance and raze the OSNs and remain undetected. It is good that research on social bots is performed. But we need to keep in mind that when new detection methods are released, social bots can be designed to evade these new methods. Social bot developers can, like anyone else, find research about detection methods online. Still, it is important that research is available so that other researchers and detection method developers know how social bots can be identified. Also, by constantly developing new methods for detecting social bots, we can maintain dominance over them.

Although there are numerous OSNs with millions of users, this study focus on the Twitter platform. According to Twitter, their vision is "To give everyone the power to create and share ideas and information instantly, without barriers." [2]. By using Twitter, information can be spread worldwide in a matter of seconds. And the information is shared to a broad audience. Meaning, information shared on Twitter can influence many people. Because of the ability to reach a high number of people, there is also a chance of misuse. As in the case with social bots.

### 1.1 Motivation

My motivation for writing this thesis and performing the study is because of the active use of bots in social media. My interest in social bots is rooted in how they are used to influence both people and events. Bots are used for several purposes, both malicious and benign. The 2016 U.S. presidential election is an example of an event where bots were used. In this election social bots were widely used [4]. Although it is questionable that the bots were used to influence the election, they were used. The fact that computer programs can be used to influence important events is what made me interested in social bots in the first place. Several solutions and methods have been developed for the detection of bots [5][6][7][8][9]. Many of the solutions developed are quite effective for detecting bots and sybil accounts on OSNs. They use different approaches for detecting bots on social networks. But since social bots are becoming more advanced, the methods for detecting them also have to advance. The need for newer detection methods as well as deterring manipulation are my main motivations for developing the algorithm.

### 1.2 Problem description

This section describes the problem using the five W's; what?, who?, when?, where? and why?. The answers to these questions describe the problem as thoroughly as possible.

**What is the problem?** People and events being manipulated by social bots. This happens because people do not know that the manipulation is taking place. The other part of the problem is that social bots are being developed con-

tinuously. This leads to the current bot detection methods being rendered useless.

**Who does the problem affect?** The victims of malicious social bots are potentially every entity and human being connected to a OSN. People with a presence on a OSN can be victims of social bots that appears to be legitimate users. Stock markets can also be victims of social bots. Many stock price predictor tools use OSNs to predict market prices of stocks. A well designed social bot can spread information that gets picked up by these tools to affect the stock prices and trends. Another potential victim of social bots is democracy. Social bots can be used by politicians to influence people's opinion and slander opponents in an election. This is a threat to democracy because it can change the outcome of an election.

**When does the problem occur?** The problem with social bots occur when new social bots are developed and people come in contact with them. This is an arms race, therefore new detection methods need to be developed continuously. If we do not keep developing new detection methods continuously, social bots can manipulate and deceive without being stop.

**Where is the problem occurring?** Social bots are a problem on OSNs and in any setting where human users communicate digitally. Human users on OSNs unintentionally come in contact with social bots. Either because of people being ignorant or that the bots are designed so well it is impossible for anyone to identify them as bots.

**Why is it important that we fix the problem?** The social bots are being developed all the time. New bot designs mitigate the current detection methods being used. The problem of social bots cannot be fixed permanently. But what can be done, is that we keep developing new methods to detect the presence of bots. It is also important that we detect and remove social bots so that their impact can be minimized. Section 4.3 will discuss the problem and possible solutions to this arms race.

### 1.2.1 The subject

Social bots are automated computer programs with a presence on OSNs or other digital platforms where humans communicate. Social bots can be divided into two main groups; malicious and benign. Examples of benign social bots are bots that deliver news or weather forecasts, tools or interfaces to communicate with a remote system or just a bot that answer simple questions. Malicious social bots are the type of social bots that we need to worry about. The malicious social bots appear as legitimate users through user profiles on OSNs. The malicious social bots can be used to influence public opinion and to manipulate other people on a OSN. By influencing and manipulating, they can also change the outcomes of

certain events such as elections and prediction of stock markets. These bad types of bots are also used to spread spam, phishing and various types of malware.

### 1.3 Research Questions

1. *What is the problem of using bots?*
  - a) *Why are social bots unethical or ethical to use?*
  - b) *To what degree can social bots manipulate people or events?*
  - c) *How can different actors benefit by using bots?*
2. How can social bots be detected?
  - a) What is the main problem regarding detection of social bots?
  - b) Are the current detection methods sufficient?
  - c) Which detection methods or algorithms currently exist?
3. How can bot detection be more efficient?
  - a) Can presence of social bots be indicated by analyzing at which hour of a day a tweet is posted?

These research question are answered throughout the thesis. The questions are answered where the related topic is discussed. Research question 1 (in cursive) is not answered in the thesis. As this thesis is written as part of the technology track, this question is more related to the management track of the study course.

### 1.4 Purpose of the study

The main purpose of the study is to develop a program that can give indications of social bots on Twitter. There have been many contributions to find new methods to detect social bots. Because there is an ongoing arms race for social bots and social bot detection methods, new detection methods are needed continuously. By developing a new indication program, the study will contribute to this arms race. Contributions to develop new detection methods for social bots is very important because social bots are used for manipulation, spreading of malware and influencing people's opinion.

### 1.5 Methodology

This thesis focus on indication of bots on Twitter. The choice of OSN fell on Twitter because of the increased use of the platform in politics and business. Recent studies have shown that social bots on Twitter is being used extensively by political parties before elections [4]. Twitter is also a platform where adversaries spread spam and malware. Some of the most ominous behaviors of social bots is that they are able to manipulate people and influence the public opinion.

The thesis provide a program which analyze a Twitter account to see if there are indications of social bot activity. There have been many recent contributions to the fight on malicious social bots. But as the social bots adapt to the newer

detection methods, it is important that new detection methods are developed continuously. The reason for developing such a program is to contribute to the fight on social bots.

The developed program analyze several different features which can reveal different types of social bots. Some of the analyses were inspired by earlier studies, but is performed differently. The other analyses are based entirely on new ideas from analyses of account information and tweets on Twitter. The reason for having a broad set of analyses is because several earlier studies shows better and more accurate detection when analyzing several features. The initial idea was to, in addition to behavior and feature analyses, to include analyses of words and language in tweets. The use of certain words and sentiment in tweets is a promising field when it comes to detection of social bots [5][10]. By including language analysis, the solution would be able to have an even broader set of indicators to indicate social bot activity. Language analyses can also detect anomalies that does not surface through feature or behavior analyses. Unfortunately, including language analyses was out of scope for the work on this thesis. Working on language analyses is a time-consuming process. And the time given for working on the thesis was not adequate for including this in the work. The idea behind the solution is to have a program that can show if a Twitter account give off indicators of social bot activity. If indications are present, a human part should perform further evaluation of the account. The human part can be one dedicated individual, or a group of people enabled through the promising approach of crowdsourcing. The reason for having a human part is because humans can detect small inconsistencies that can be hard to define in algorithms.

### 1.5.1 Information gathering

To gather information about social bots, several academic databases have been used (e.g. link.springer.com, dl.acm.org, sciencedirect.com). When these databases have not been sufficient, Google (google.com) have been used. Some blogs and news articles about social bots have also been used when gathering information about the topic. Many of the blogs and news articles reference scientific work. This makes it easy to find the source of information and to get further details of the different studies.

Because social bots is a fairly new topic, there are not many books on the topic. However, there have been many studies and research contributions in the form of research papers. The research on social bots is very extensive and detailed, making them good subjects to research social bots.

Most of the sources used in the thesis are from studies within the five last years. Some very few studies from 2011 and 2010 have been used. These are some of the first studies on the topic. The study results and concepts have been analyzed and they are still relevant for today's situation about social bots.

In the initial part of the research on social bots, blogs, news articles and youtube



were used to lay the foundation of knowledge on the topic. These sources are very good to learn the basics and information in an overview. By doing this first, the scientific articles and studies were much easier to understand.

## 1.6 Definitions

**Artificial Intelligence** Software that is able to function and learn by itself. Used to mimic human behavior by social bots.

**Crowdsourcing** The process of distributing a task amongst a crowd of people.

**Honeypot** A type of decoy used to expose anomalies. This type of decoy is often presented as a vulnerable entity so that adversaries are more likely to "attack" it. Honeypot accounts can be deployed on OSNs to unveil social bots.

**Online Social Network** A social media platform where users can communicate and create networks.

**Source** When discussing tweets, source means the origin from which a user posts a tweet. Where appropriate, "device" is used to describe the source of tweets.

**Spam** Distribution of unsolicited content.

**Sybil** Another name for fake account. Can both be connected to a bot or be human-controlled.

**Twitter username** Another name for screen name. In this thesis, when describing Twitter accounts, username is often used. Username is used in parallel with screen name.

## 1.7 Literature

There have been many studies on social bots and contributions to detection methods. Ferrara have performed several interesting studies on social media and on social bots [4][11][12]. In one of his more recent studies he studied the use of bots in the 2016 U.S. presidential election. This study is what gave me motivation to contribute to the fight against social bots. Many studies with focus on different social bot detection methods have been performed. Some of studies on behavior-based detection methods [6][7][9][13][14][15] have been great sources of inspiration. These studies have helped substantially for knowing how social bots behave. The use of a human component to identify social bots is central to the proposed solution. There is one study in particular that was crucial for considering this approach [8]. This study evaluated how effective humans can be to detect fake accounts on OSNs. Fake accounts are closely related to social bots, therefore, a human part can be effective to also detect social bots.

## 1.8 Assumptions, Limitations and Delimitations

An assumption this thesis is based on, is related to the arms race between social bots and detection methods. The assumption is that social bot designers research which approaches are used to detect social bots. And then design the new bots so that they will not get detected by the current detection methods. Another assumption is that if new detection or indication methods are developed continuously, we can have the upper hand on the social bots. From a logical view, this assumption should be true; By having several detection approaches at hand, at least one or more approach will always work for detecting social bots.

One of the limitations in this thesis is that the indication program have not been tested on social bot accounts. Social bot accounts are difficult to find. Therefore, to test the program on actual bots, we need to know that an account to be analyzed is of an actual social bot. A good approach to find social bots is the use of honeypot accounts. Honeypot accounts are OSN accounts that can attract social bots. Deploying honeypot accounts and waiting for social bots to contact them is a time-consuming process. A good honeypot-study should be deployed over longer times (from several months to a year). Because of this, the solution could not be tested on actual bots. But several of the indicators used in the program is based on already tested concepts and are proven to be good indicators for social bots. Another limitation is that the indicators that are based on new ideas are not proven to work. Further testing of the solution on actual social bots are needed to prove if these are good indicators.

A delimitation of the work in the thesis is that the developed program is able to solely analyze Twitter accounts and tweets. Twitter were chosen because of the increased use of this platform in politics and because the presence of social bots on Twitter is increasing. Although the main focus is on the Twitter platform, the indicators in the developed program should be applicable to other environments such as Facebook and Instagram. The developed program analyze data downloaded right before analysis. The approach for streaming live tweets were considered. But this approach is time consuming because of the need to wait for tweets to be posted. In addition, performing live streaming of data is not necessary for the types of analyses used in the program. Live streaming of data is more appropriate when performing language analyses of tweet content. This way, social bot tweets can be detected as they appear. Another delimitation is that the developed program does not perform analyses of language in tweets or instant messages. Developing good algorithms for analyses of language is time consuming. This functionality would be implemented if more time were given for the work.

## 2 Online Social Networks

A OSN is a web site where users can create a networks of social connections and share different content such as statuses, news, photos and other digital content [16]. As explained by Steinfield et al. 2008[16], OSNs have 3 essential components:

1. A public or semi-public user profile constructed by the user.
2. A set of connections to other users.
3. The capability to view your own connections and connections made by other users.

Different OSNs have different abilities and different methods to communicate, but these components is the core of all OSNs.

OSNs were initially a social media channel in which people could connect to other people across the world. OSNs gained popularity as the world got more Internet-connected. The first example of digital social media is perhaps Internet Relay Chat (IRC). IRC enabled users across the world to communicate through a chat-interface. Different chat-rooms divided the topic of what the users could talk about. From a simple chat-interface, social media developed into OSNs. Higher Internet bandwidth enabled users to, amongst other things, upload pictures and share data of larger sizes. Throughout the 1990's more and more OSNs emerged. The more successful OSNs appeared in the first years of the 21st century. At the turn of the century, most of the homes in the world were Internet-connected which gave presence to the more successful OSNs. Some OSNs such as MySpace gained huge popularity in the first years of the 21st century, but its popularity recessed because of the introduction of more successful OSNs. Some of the more successful OSNs such as Facebook, Twitter and Instagram are OSNs that became massively popular all around the world. The largest OSN today, Facebook, have approximately 1.23 billion daily active users [1]. Respectively, Instagram have about 500 million activer users [17] and Twitter having approximately 300 million monthly active users [2]. These OSNs are still gaining popularity today and is still growing.

Most of the OSNs have smartphone applications available. This enables the users to communicate on the OSN without the need of a desktop or laptop computer. This definitely have an impact on the amount of daily and monthly active users. This chapter will discuss different OSN platforms and OSN-users behavior in relation to social bots. Section 2.1 discuss different OSN platforms, including how communication is surveyed and how privacy is handled. Section 2.2 discuss the relationship between social bots and other people on a OSN.

## 2.1 Different OSN platforms

There exists many types of OSNs. Each one offering a variety of methods for users to communicate and expand their social or professional network. Some OSNs target users in specific countries whilst others target users in different social situations, for instance people looking for jobs or a life companion. Different OSNs also have different complexities. In this section, complexity of OSNs will be generalized into Facebook (more complex) and Twitter (less complex) except for a few exceptions where noted.

### 2.1.1 Purpose and methods of communication

As mentioned in the beginning of this section, all OSNs have a common core of 3 components. But naturally, not all OSNs are similar. All the different OSNs have their own niche and/or forms of communication. On Instagram, the users communicate by sharing photos. The users can "follow" other users to get updates about the photos a particular user posts. The users can also subscribe to certain "hashtags" to be updated on images posted that is tagged with these "hashtags". Twitter is more of what you could call a microblogging-platform. On Twitter, the users communicate with posts limited to 140 characters. The users can mention someone in a post or contact them directly by adding a "@" followed by another user's username. Just as with Instagram, "hashtags" can also be added in a post to tag it with a certain topic. Facebook on the other hand is OSN that covers a broader set of functionality. On Facebook, the users can interact by several means. Users can publish posts with any type of digital media (photo, video, files etc.). Facebook also enables the users to talk via a chat-interface.

Although the different OSNs have different functionality, some OSNs target users of a certain geographical or social location. For instance in China, Sina Weibo is a popular OSN. Sina Weibo is similar to Twitter in many ways. The users are limited to 140 characters per post, and may add a "@" or a "hashtag" to a post to interact with other users. There are OSNs present in several countries that specifically targets the population of that particular country. Other OSNs such as LinkedIn targets people with a focus on their job network. LinkedIn is in many ways similar to Facebook, but with a focus on connecting people in a professional setting, either to find a new job, promote achievements or to just expand one's job network.

Regardless of complexity or which user is targeted by the OSN, the majority of OSNs provide the users the functionality of posting various content or sharing information.

### 2.1.2 OSNs and privacy

Because OSNs become more popular, they are also attractive to adversaries that want to collect personal information which they can sell or exploit. Being more aware about privacy on OSNs, can help people to not get their information col-

lected and profited on by adversaries. The different OSNs manage privacy differently. On Twitter, all profiles are public by default. On Facebook, the user profiles are not entirely public by default. Instead, profiles are publicly available, but with very limited information.

#### *Twitter*

Although Twitter profiles are public by default, Twitter enables the user to restrict who can view your own tweets. When this functionality is enabled, only the ones who the user have approved will receive future tweets. Follow requests are accepted by default, but a user may enable manual follower approving. This way, following users will only receive tweets if they have been approved by the other user they want to follow. Other less complex OSNs like for instance Instagram and Youtube have approximately the same privacy options as Twitter.

#### *Facebook*

To be able to view the entire profile on Facebook, the viewing user have to become "friends" with the user being viewed. LinkedIn is also very similar to Facebook in how profiles are visible. Facebook which is a more complex OSN than for instance Twitter, have many more options regarding privacy. On Facebook, a user can determine what profile information should be available to all users, friends of friends or just friends. A user can also determine on the go, who a post should be available to. In addition, a user can determine who can send friend requests and managing who can look up the profile.

Even though LinkedIn is very similar to Facebook in many ways, privacy management alternatives are somewhat different. On LinkedIn, there are not that many privacy regulations a user can manage as on Facebook. Some of the features on LinkedIn like sending message to a user is restricted to other people you are connected to.

What can be a problem on OSNs is that people are not necessarily aware about their privacy. Users that do not regulate privacy on their accounts, can have their information stolen by social bots that collect information. As a worst case scenario, a person that do not regulate privacy can have their identity stolen and misused. In 2014, Facebook users were prompted about checking their privacy settings. That a OSN network does this is a good way to make users more aware of what they share on the Internet.

## **2.2 People's behavior in relation to social bots**

How bots succeed in spreading of malware and manipulation is rooted in how people behave on OSNs. Users' awareness of threats such as social bots can vary. And there have been raised awareness for the different threats on OSNs the lat-

est years. But even though people know about the different threats they do not have a deeper understanding of these threats [14]. There is a social etiquette on OSNs that involves following back or adding someone as friend if they follow you or add you as a friend. When these "unknown" friends share links, people often click on them without considering that an "unknown" friend shared it.

A solution to these problems could be to only add people you know as friends on OSNs. Or in other words, only add your friends as friends. By only having friends as friends on a OSN, the chance of being exposed to malicious content is lowered significantly. Many social bots rely on the etiquette of following back or adding someone as friend [11]. By only adding people one know as friends and following back people one knows, the degree for which social bots reach out is lowered significantly and perhaps eliminated.

To further raise awareness about the different problems on OSNs, people must be informed. Information should be presented through many different channels; in news, blogs and on the different OSNs. Information must be shared and displayed through many channels so that people realize the seriousness of the issue. Informing people on the issue can be difficult. But by informing people extensively is very important. Therefore, informing people should be included on the war against social bots.

## 3 Social bots

Social bots or just bots for short are computer programs that controls OSN accounts. Social bots come in many forms. Some social bots are harmless and used like an interface for a tool, and some bots have malicious purposes including distributing misinformation and malware or influencing people. This chapter will cover the different aspects of social bots including how they are constructed, what they can be used for and who the beneficial parties are. With a few exceptions, this chapter will mainly focus on malicious social bots.

Social bots come in many different variations, each with their own purpose. Section 3.1 provides an in-depth classification of the different types of bots. Furthermore, this chapter includes an elaboration on to what degree social bots can influence people and events. The beneficial parties using bots will also be discussed in this chapter. Lastly, the chapter will discuss the different defensive and preventive mechanisms and methods that can be used to deter social bots.

### 3.1 Classification of bots

Social bots are designed to perform specific actions. Some are benign and can be used to receive news, weather information or to interact with other systems. But the concerning type of social bots are the malicious type. Malicious social bots can be used to spread misinformation, influence people or events and spread malware or spam. Social bots can be divided in 3 groups:

1. **Fully-automatic:** After being deployed, this type of bot acts completely on its own.
2. **Semi-automatic:** This type if bot is automatic, but is regularly interacted with by an operator.
3. **Fully-manual:** This group of bots is always being interacted with by a human operator. This group of bots have a human operator meaning there is a human brain in place instead of code that runs automatically.

Since bot means robot, fully-manual bots can not be considered a bot by definition. But the only difference between automatic types and fully-manual bots is that fully-manual bots are human beings instructed by another human what to do. The fully-manual bots are often referred to as sybils. On the contrary, automatic bots are programmed by a human and perform the instructions on a computational level. What all bots do have in common, is that they have an operator. Even if they are fully-automatic or manual, every bot have been designed and deployed by a human being. Semi-automatic bots fall in a category between

fully-automatic and fully-manual. Semi-automatic social bots can be automatic for the most part, but is being regularly commanded or updated by an operator. With a few exceptions of types of spam bots, social bots generally want to blend in. Most social bots are designed to be stealthy. Meaning that their goal is to blend in with the crowd. This is a crucial factor for social bots since their goal is to appear human. A social bot behaves like humans by mimicking the behavior of legitimate users or by simulating user behavior with the use of AI [18]. Both fully-automatic and semi-automatic social bots rely on machine learning (ML). ML can be defined as a sub-category of AI. By using ML, the social bots can learn from data available on the OSN (or from anywhere else on the Internet). ML can be used by social bots to learn language and how people behave on OSNs. An example of a social bot that uses ML is discussed in section 3.3.1. AI in relation to social bots is discussed further in section 3.3. This section cover the different types of social bots on OSNs that can be identified individually. Although some bots are hybrids, the bot types categorized here are types of bots with distinct operational and functional use.

### 3.1.1 Spam bots

Many studies have been performed on detecting spam bots. One of the more interesting studies is the study performed by Stringhini et al. (2010)[14]. The study gives a great categorization of spam bots as well as good descriptions of their behaviors. This section incorporate many of the ideas and findings of their work. Spam bot is a type of bot that delivers spam on the Internet. Spam is involuntary messages that spread advertisements, malware, phishing or other malicious digital media. Spam is usually delivered to a high volume of receivers. Hence, spam bots are always malicious. Spam bots can be fully-automatic, semi-automatic, or fully-manual. One way that spam bots operate on OSNs is that they add a large number of people as friends. Some people will add them as a friend, opening a connection between the two accounts. Then the spam bot sends the spam tweets to the other user. Since the bot and the other user are connected, the spam tweets sent by the bot will be displayed on the other users timeline. Then a portion of the users that receives the spam will interact with the content sent by the bot, either it is advertisement, malware or links to malicious web pages [13]. Spam bots can differ from another. Stringhini et al. further categorize spam bots into 4 sub-categories based on their behavior. The following categories are:

1. **Displayer.** Only displays spam content on its own page. For the spam content to be viewed, another user needs to visit the spam bot's page.
2. **Bragger.** Shares spam content through a feed (e.g. status updates on Facebook or tweets on Twitter). Spam content will only be visible to other users that are connected to the spam bot's account.
3. **Poster.** Sends direct messages to other users. On Facebook for instance, this would be to post a message on someone's wall.



4. **Whisperer.** Sends private messages to other users. For Facebook, this would happen through the Facebook chat functionality, and for Twitter this would be direct messages.

As Stringhini et al. [14] describes, the different approaches are used for different scenarios, and can have different levels of success for distributing spam. "Displayers" are described as the least efficient way to deliver spam. This is because the user have to actually visit the spam bot's profile page. The "braggers" are a bit more effective since the spam content is displayed on several victim's own feeds. One approach that can be used by "braggers" is to hijack trending topics or hashtags for Twitter. By hijacking trending topics, the bot reaches out to a bigger crowd of people. The most effective way of performing spam on OSNs would be to use a poster [14]. With a "poster", the spam content can be viewed by the user behind the account the spam content is posted to as well as other users who visit this account. The use of "posters" will yield a one-to-many relationship for distribution of spam content. The one-to-many relationship is why this is the most effective method of distributing spam on OSNs. The "whisperers" which only sends direct messages yields more of a one-to-one relationship. Although not as effective as "posters", this is a far more stealthy approach to deliver spam. To be more stealthy and to mitigate detection, spam bots can operate in bursts. This means that the bots operate for a shorter period of time and hibernate for a given period of time before they become active again. This way, the probability of detection is lowered. Bot behavior can be categorized into two main categories; greedy and stealthy [14]. The greedy bots always spread spam content. The stealthy bots mostly sends messages which are legitimate or harmless, and once in a while include spam content. The greedy type is perhaps the most known behavior regarding spam. Because the greedy bots always include spam in the messages, they are naturally easy to detect. When spam is distributed with a high frequency, even simple algorithms can detect it. The stealthy bots on the other hand is much more difficult to detect by simple algorithms. This is because the regular behavior of the bot seems benign. When the majority of communication is benign or harmless, such bots become more difficult to detect. Another finding of Stringhini et al. show that legitimate users are more active than spam bots and that most profiles of spam bots distribute below 20 messages in total. This makes it very hard for detection algorithms to detect the spam or the source of the spam itself. These examples of bot behavior shows that spam bots on social networks have adapted in order to avoid detection.

### 3.1.2 Chat bots

Chat bots are bots that are connected to any interface that distribute communication between two parties (e.g. chat interfaces, comments etc.). Depending on the intended use, chat-bots can be simple conversational bots that people can talk to. They can also be tools for which a user can send commands to and get an

intended response in return or support solutions (e.g. Digital Personal Assistants (DPA)). But chat bots are not necessarily benign. Chat bots can also target people and talk to them to influence them or send them malware or spam. By using artificial intelligence, markov chains and various machine learning techniques, chat bots can perform human-like conversations. The use of these methods can make it harder for a human to know that it is talking to a bot. The development of chatbots seem to have been a driving factor for the development of AI. Bots and AI is discussed further in section 3.3.

Some OSNs such as Facebook does not require users to be friends for them to send messages to each other. Although there are settings that can prevent non-friends to not send a user messages, this is not activated by default. Therefore, a huge portion of users on Facebook can be prone to attacks from chat bots. On Twitter, users cannot receive messages from non-friends by default. To be able to receive messages from non-friends on Twitter, the user have to explicitly activate this feature. But if a user follows a social bot, the social bot can message the user. Regardless to any settings the user have activated or deactivated.

### 3.1.3 Information collecting bots

When considering the name "bot", one may think that it always perform communication with a user. But some bots does not communicate with users. Some bots are designed to scrape information from a OSN. Because these bots are scraping for information, they are often referred to as "scrapers". The data collected can for instance be emails, personal data or any other content on a OSN. OSNs like for instance Twitter provide an apprehensive API that allow anyone with a Twitter account to retrieve various information about twitter users and communication. But Facebook for instance does not provide an API as comprehensive as Twitter does. This is because Facebook gives the user more options regarding privacy. Therefore, information scraped from Facebook is far more valuable for adversaries than that of Twitter. But these scrapers have a good counterpart by the name "crawlers". Crawlers are of benign nature and is often used to retrieve system information or to index web pages to a search engine.

## 3.2 Anatomy of social bots

As mentioned in the introduction of this chapter, social bots are computer programs that interact through a OSN account. On OSNs, a social bot acts as a normal (human) user and performs the tasks it is designed to do. Some bots want to remain hidden while other bots want to be as visible as possible. Some bots communicate with users whilst others just perform surveillance. Whatever the purpose is, all social bots have two main components; a face and a brain.

### *The face*

The face is the part of the bot that is visible to other users on the OSN. It can

be considered the user profile with which a bot communicates over the OSN. To blend in with the crowd, it is crucial for a social bot profile to contain profile information just as any other user. The information the bot have on its face can be added manually or be scraped from the web. Information scraped from various sources can be mixed together to create variations when implementing several bots. A bot can create a dedicated OSN profile. This is desirable if the bot needs a customized profile. On the other hand, a social bot can also hijack an existing account. This can be desirable when the goal of a bot is to infiltrate an already existing group or network of people. Some bot accounts can be very easy to distinguish from real ones. Some bot accounts can give of hints that they are created automatically. An example of this can be that the profile image is of an old woman, but the other info on the profile suggests that the account belongs to a 12 year old boy. But if bot accounts are designed carefully, they can appear to be as real as any other human operated account [19].

Figure 1 simply describe the anatomy of social bots. As described in the figure, the face of the social bot acts as a communication port to the OSN. The face is not part of the social bot by definition. It is part of the OSN, but is controlled by the social bot by an API such as the Twitter REST API.

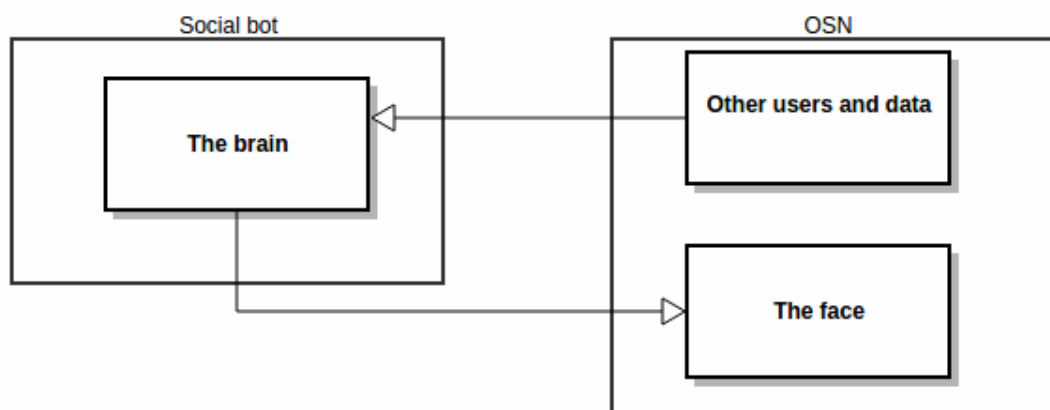


Figure 1: Anatomy of a social bot.

### *The brain*

The other part of any social bot is the brain. The brain is the computational part of the bot, the algorithm itself. Just as a human brain is used with fully-manual bots, the brain of automatic bots are pre-programmed computer programs. The brain is the part of a bot that performs all the actions (i.e. scraping, messaging, analyzing and collecting data etc.). As shown in figure 1, the social bot only consist of the brain. The brain reads data from the OSN and decide what to do. To perform actions (i.e. communicate), the social bot use the face. In many ways

the face acts as a communication port for the social bot on the OSN. The brain determines and decides what to do, and uses the face to perform the actions. For semi-automatic and fully-manual bots, the brain also includes an interface for which the instructor can make the bot perform the desired actions. A simple brain could for instance only be using an API (e.g. Twitter REST API or Facebook's Public Feed API). But more advanced bots might use a web crawler to retrieve information from a OSN.

### 3.3 Social bots and AI

As discussed in section 3.1, most social bots want to blend in. The first stage of blending in with the crowd is to have a profile with credentials that looks genuine. By having a profile that seems legitimate, the bots have greater success of communicating with other people. The next stage is to perform communication without standing out. If a bot shows automatic/robotic tendencies which humans usually do not show, it stands out. This can lead to people deter communication with them, or even report them to the OSN it appears on. This is where AI comes in. With the help of AI, social bots can communicate without revealing their robotic nature. AI can be used in many different technologies. Amongst

Hello.

Hello, how are you?

I am fine, how are you?

I'm fine, thank you.

Who are you?


I'm a human ;).  share!

Figure 2: An example of a conversation with cleverbot. Performed at [cleverbot.com](http://cleverbot.com) [20].

other things, AI is used in video games to make characters behave natural, in self-driving vehicles and in chat bots. Chat bots implement AI so that when a human talks with it, the conversation is performed naturally just as when a human talking with another human. The use of AI can both help people and can be used

by bots to deceive. Another consideration that comes to mind when discussing AI is how effective it is to convince a human being that it is not a machine. The Turing test is a test developed to test a machine's ability to inherit intelligent behavior like that of a human. The Turing test is discussed further in section 3.3.2. Since AI was described by Alan Turing, it have developed to become quite advanced in the recent years. Both technological and for amusement. A range of conversational bots are available online for people to test and talk with. An example of such a conversational bot is cleverbot [20]. Figure 2 shows an example of a conversation performed with cleverbot (characters in black is the writer's, and blue is cleverbot's responses). Cleverbot does give many good answers to questions you ask it, and it asks questions back. But after conversing for a little while, it is easy to determine cleverbot as AI. Although AI is used for technological enhancements and for amusement, what seems to be a goal of AI researchers and developers is to have AI pass the Turing Test. This would be a significant achievement in the development of ai and a merit for humans creating life.

### 3.3.1 AI gone wrong

An interesting history of AI "gone wrong" is the bot Tay. Tay is a fully automatic social bot developed by Microsoft. On Twitter, Tay is known as TayandYou. Tay was developed as an experiment in AI. Tay uses machine learning techniques, and learn from Twitter. What went wrong with Tay is that after being present on Twitter for a little while, it began posting offensive tweets. It started to tweet racial offensive tweets and tweets with other disturbing content. Although Tay did some horrendous things and "went wrong", it can also be considered a success. It did succeed in learning from content on Twitter. In addition, it revealed the awful truth about what people talk about and share on Twitter.

### 3.3.2 The Turing test

As mentioned in the introduction of this chapter, the goal of the Turing Test is to test a machine's ability to inherit intelligent behavior. What we today know as the Turing Test was invented by Alan Turing and was first proposed in his paper *Computing Machinery and Intelligence* [3]. In his work, Turing refer to the test as *The Imitation Game*. Initially, the Imitation Game involves a man and a woman, and an interrogator of either sex. The goal of the game is for the interrogator to determine which of the subjects are a man and which is woman. The interrogator asks questions to the two subjects and tries to determine which sex the subjects are.

"I propose to consider the question, "Can machines think?""

–Alan Turing

In the machine perspective, the Imitation Game involves one subject of either sex, a computer and a judge. In this version of the Imitation game, the task of the judge is to determine which one of the subjects is machine and which is

human. The above quotation is taken from Alan Turing's *Computing Machinery and Intelligence* [3]. Turing wrote this in the beginning of the paper. Further into the paper, the question is changed into "Are there imaginable digital computers which would do well in the imitation game?" [3]. The revised question seems to have been a great motivator for researchers and developers of AI. In the science of AI, the Turing Test have become a method for measuring how successful a machine is in being intelligent. The Turing Test seems to have been a great motivator for the development of AI. Many chatbots have been made in attempt to pass the Turing Test. In 2014, a chatbot named Eugene Goostman allegedly passed the test. But the allegedly passing of the test have received criticism. Many of the chatbots developed use psychology tricks and exploits assumptions and emotions that people (the judges in the test) have. Because it is easier to play on emotion and psychology, most of the bots being developed just become simple conversational agents. Although the Turing test seems to have been and is a great motivator in AI development, the test also have its disadvantages. A great disadvantage is that the Turing test only tests conversational intelligence [21]. This is a disadvantage because intelligence does not only appear in conversations. Intelligence can appear as for instance thought or any other action that a biological creature needs intelligence to perform. Therefore, and because Turing himself discarded the question "Can machines think?", we need to take the Turing test with a pinch of salt.

### 3.4 The intentions of social bots

As mentioned in the beginning of this chapter, social bots can be harmless. They can be news feed entities that grab news from different sources and delivers it to the subscribers' feeds. They can also be amusement tools that delivers a specific type of message or edits an image and return it. They can be used for numerous useful purposes. But what is concerning about social bots is that they can be malicious and cause harm. Even benign bots (e.g. bots that deliver news, weather information and bots that are entirely for amusement purposes) can contribute to spreading unverified information [11]. An example of this can be a news bot that picks up a news article from an unreliable source. Unverified information or intentionally fake news can be grabbed by bots and be reposted. When this is done, false information can rapidly be spread to many people in a very short period of time. This is bad because it manipulate people and can make them believe things that are not true. However, the social bots that are intentionally malicious are the ones that we really need to be aware of. These bots are specifically designed to cause harm, to spread malware, misinformation, spam and to manipulate. The malicious type of social bots can for instance be used to influence public opinion by posting fake reviews on product sites or by spreading fake news. They can also be used to influence elections, stocks and

various events. Something that is a great concern when it comes to social bots being used to manipulate elections, is that it threatens the democracy [11]. In 2016 Bessi et al.[4] studied the use of bots in the 2016 U.S. presidential election. Their findings showed that bots were responsible for about 20% of the conversation related to the election. Although not proven, the extensive use of bots in elections can influence what people thinks and maybe also the outcome of the election itself. If elections are being manipulated, one of the core principles of democracy, which is election, is being threatened. The use of bots in elections is not entirely new. The bots were used in attempt to slander an opponent in the 2010 U.S. midterm elections [11]. But bots could and probably have been used on earlier occasions without them being detected.

Bots can also be used to influence the stock market. Automatic trading algorithms are being used to perform stock trades. As Ferrara et al. [11] mention, some of these algorithms use Twitter signals to predict stock changes. Although this method might be effective to predict the stock market, it might also be a weakness that can be exploited. As Ferrara et al. also explains, this actually happened. An automatic trading algorithm picked up a tweet which resulted in the market value of a stock being multiplied by 200. This one tweet was not posted by a social bot. But the example shows how easy it can be to manipulate crucial parts of the economy such as stocks. Potentially, social bots can exploit the use of automatic trading algorithms to manipulate values of stocks.

#### 3.4.1 Purchasing fame

Another problem that is present and which is enabled by social bots are purchasable reputation. On OSNs, users that get many likes or have many followers is considered more popular. But this popularity and fame can be bought. There is a large industry for selling likes and followers. A quick google search for "buy followers" yields many services that sells likes and followers. An example of a service that sells likes and followers are mysocialfollowing.com. Via this service, 10,000 followers can be bought for as little as 89 USD.

In a study from 2014, Shen et al.[22] explains that likes and followers are purchased for two reasons; to be more famous, and to inflate advertisement. The problem these services cause is that they create artificial fame, which deceive people to think people are more popular or famous than they really are. They also create artificial popularity of products which can manipulate the opinion that people have of a product or brand.

In their study, Shen et al. proposed a method for detecting fake followers on Sina Weibo, a microblogging platform very similar to Twitter. Their proposed detection method focused on features such as the ratio of followee count and follower count, the percentage of bidirectional friends, average post frequency and proportion of nighttime posts. By analyzing these features, the researchers were able to detect fake accounts with an accuracy of over 90%. By analyzing

over 30,000 accounts on Sina Weibo, they found that ordinary users have about 14% fake followers. Furthermore they found that celebrity users have about 42% fake followers. This means that almost half of a celebrity's followers are fake. But this does not mean that all celebrities buy followers. A social etiquette on Twitter is to follow people back if they follow you. This etiquette is exploited by social bots. Therefore, a large portion of the findings by Shen et al. might also include social bots as well as fake accounts.

### **3.5 How are bots successful or unsuccessful?**

There are several factors that can determine how a bot is successful or not. And the factors can vary between OSNs. The degree for how a social bot can be successful often comes down to how it is designed and how it behaves. This section describe how bots can remain detected and what makes them easier or harder to detect.

#### **3.5.1 What makes social bots harder or easier to detect?**

Most social bots are designed to be stealthy. This means that they do not separate themselves from the rest of the users on a OSN. Naturally it is not very easy to detect something that is stealthy. AI have become very advanced in the recent years. And it have become very hard for humans to distinguish from talking to a social bot and an actual human being. The use of AI also makes it harder for detection methods to detect social bots. This is because text generated with AI can be very close to text that humans generate. If the social bot is designed in such a way, it can be harder to distinguish from human users.

Classic forms of spreading spam can be very simple to detect. Classic forms of spam involves spreading the same spam content several times from the same source. But new approaches to spreading spam deviates from the classic form. New approaches for spreading spam with social bots involve not sending the same spam content repeatedly from the same account. Instead, the distribution of spam is spread over multiple accounts. Therefore, analyzing one account does not necessarily show indications of spam.

The OSN profile of the social bot can also determine how easy it can be detected. A simple made profile with obviously fake or generic information stand out from the crowd of legitimate users. Fake or generic information include for instance profile picture of someone else (e.g. a celebrity) or random generated name or username. By having customized profile information, it is much harder to differentiate between a social bot profile and a legitimate human profile.

#### **3.5.2 How does a social bot remain undetected?**

To remain undetected, a social bot needs to behave as close to humans users as possible. This can involve things like not spreading spam in an obvious way, not sharing obviously malicious content and not posting tweets at the same times. Generally, a social bot remain undetected by not behaving in a way that is used



to detect them. To keep blending in with the crowd of legitimate users, a social bot should be designed with the current detection methods in mind. If a social bot is designed without considering how it can be detected, they can be easier to detect. Therefore, for a social bot to be successful, a bot designer needs to research how detection of social bots is performed and what is measured. By for instance knowing that social bots are detected by analyzing at which hours or minutes tweets are posted, the social bot can be designed to post tweets at more random times. By always knowing how social bots can be detected, the social bot designer can patch the social bot. By doing this, the social bot can remain undetected by the current detection methods.

### 3.6 Beneficial Parties

There are many different entities that can benefit from using social bots. Social bots and fake accounts can be used to inflate popularity of advertisement. Companies performing advertisement on OSNs can order (fake) likes to boost the popularity of the advertisement. This can influence people to believe the product being advertised is more popular or better than it actually is.

Another entity that can benefit from using social bots is politicians. As studied by Bessi et al. [4], social bots were extensively used in the 2016 U.S. presidential election. Just as inflating advertisement, politicians use social bots to raise their popularity. Inflating political support can influence other people on OSNs to determine which politician to vote for. Strategies for social bots ordered by politicians include for instance spreading positive messages. When people on OSNs see these messages, they can be influenced to think differently of a politician. The use of social bots in politics can be very dangerous because it can threaten the foundation of our societies which is the democracy.

Other beneficial parties can include terrorist organizations that want to spread their ideologies.

### 3.7 Preventive mechanisms and methods for social bots

This section describe some of the mechanisms and methods for preventing social bots. Section 3.7.1 describe the use of CAPTCHA, and section 3.7.2 describe how to determine a user's susceptibility to social bots.

#### 3.7.1 CAPTCHA

There are not many mechanisms to prevent bots to creating new accounts on OSNs. One mechanism that have become a security standard for online registration forms is CAPTCHA. CAPTCHA stand for **C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part. CAPTCHAs can also be referred to as a "reverse Turing test". Instead of a regular Turing test in which a human determine if the other part is machine or human, CAPTCHAs allow a computer to determine if the other part is machine or human [24]. Most implementations

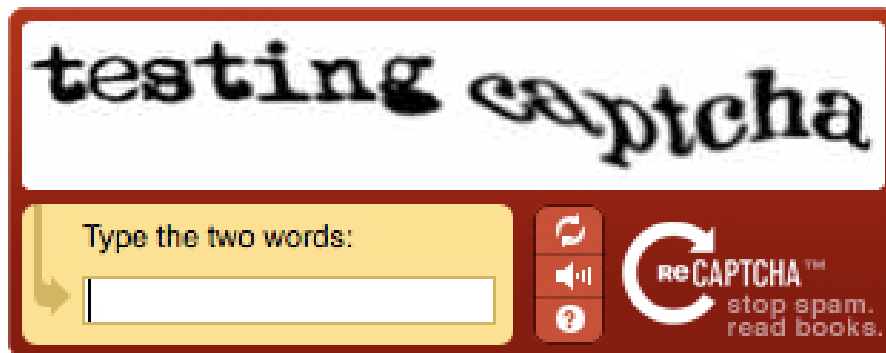


Figure 3: An example of a CAPTCHA. Created at [fakecaptcha.com](http://fakecaptcha.com).

of CAPTCHA displays one or more distorted letters or words to the user. Figure 3 shows an example of a simple captcha implementation. A captcha usually have two parts relevant to the user. It has an image with distorted text in it, and an input field. The user needs to type the characters of the text in the image into the input field. If the characters that the user typed in matches the text in the image, the test is passed. On the contrary, if the characters typed into the field does not match the ones in the image, the test fails.

The CAPTCHAs that have irritated people for over a decade is in place for a good reason. As the name CAPTCHA suggests, it is a test to see if an actor is either human or machine. Having CAPTCHA implemented will to some degree prevent both DoS attacks and bots generating online accounts automatically. But CAPTCHA can easily be beaten by creating advanced image-recognition algorithms. Because of this, different versions of CAPTCHA have been developed. Some CAPTCHAs have images in bad quality in which the user needs to find numbers or words. The bad quality makes it harder for automated processes to read and identify the properties (i.e. letters/numbers) in the image. But the bad quality is still adequate enough for humans to read. Other CAPTCHAS ask the user to select images that have special settings. Then the user have to click on every image that depicts the setting required to be identified.

An interesting and successful attempt to break CAPTCHAs is the solution developed by Bursztein et al. 2011 [24]. In their study they found out that 13 of 15 implementations of CAPTCHA from popular web sites are vulnerable to CAPTCHA-attacks. They achieved this by first eliminating the noise/background used to make the image harder to read for a computer. Then different segmentations are performed to separate the characters in the CAPTCHA-image. Further, each character is recognized using a pre-made training set for character identification. Lastly, depending on background information about the CAPTCHA in focus a spell checking is performed (e.g. if a CAPTCHA implementation uses dictionary words, a dictionary can be used to perform the spell checking)[24].

One may ask one self if we really need to use CAPTCHAS when they can be broken. Yes, some break-ins will happen. However, by using CAPTCHAS, we can stop the majority of automated processes trying to register online accounts. An example of this is when Alta Vista implemented CAPTCHA, the number illegitimate database entries immediately decreased by 95% [25]. This might not be the case for every implementation of CAPTCHAs, but it shows that implementing CAPTCHAs can help.

### 3.7.2 Modeling user's susceptibility

In a study from 2012 [26], Wagner et al. studied OSN users' susceptibility to social bot attacks. This were done by studying network features (e.g. user-follower-relations and retweet behavior), behavioral features (e.g. lexical variety and question coverage) and linguistic features (i.e. structure and content of messages). The conclusion of the study was that users who have a large social network and who actively interact with other users are more susceptible to social bot attacks. As the researchers explain, the reason for this is perhaps because these users are more open and active. The results of the study showed that the susceptible users incorporated one or more of the following features:

- Interacting more actively.
- Using more verbs.
- Talking to a variety of users with mostly conversational purposes.
- Showing a variety of affection words.

As Wagner et al. discuss, users who interact more, are more susceptible to social bot attacks. These users are used to communicate, and seem to be more open, hence they more easily make new connections. Susceptible users also use more verbs. More specifically they use present, past tense and auxiliary verbs and more pronouns. As explained by Wagner et al. the usage of these verbs and pronouns indicate that susceptible users talk about what they are currently doing. Furthermore, susceptible users seems to have higher conversational variety than non-susceptible users and that most of the communication have conversational purposes. They do not only communicate with their closest friends, but a variety of other users. This also indicates more openness for susceptible users. Another feature that might explain why susceptible users are more open, is the use of affection words, positive emotion words, social words, motion words and adverbs. The use of such words mean that the susceptible users often talk about their activities. Figure 4 shows the rankings of the measured features by importance. As the table in the figure shows, the researchers found that the degree of interaction, the use of verbs, conversational variety and coverage and use of present tense verbs is the most important features for determining susceptibility to social bots. As a part of the study, the researchers found out that the best feature for determining susceptibility to bots is the high out-degree and in-degree in the interaction network. Furthermore this means that susceptible users tend to

Feature	Importance
out-degree (interaction network)	100.00
verb	98.01
conversational variety	96.93
conversational coverage	96.65
present	94.66
affect	90.15
personal pronoun	89.71
first person singular	89.27
conversational balance	87.28
motion	87.28
past	86.56
adverb	86.20
pronoun	84.41
negate	84.33
positive emotions	83.25
third person singular	82.38
social	82.02
exclusive	81.86
auxiliary verb	81.70
in-degree (interaction network)	81.66

Figure 4: Ranking of the top features for determining susceptibility to social bot attacks [26].

have many points of both ingoing and outgoing contacts. Such modeling of susceptibility can be used as a defensive mechanism for social bots. When a user's susceptibility have been modeled, the user can be informed or other precautions can be taken. This way, the user can avoid being victims of social bots.

Modeling users' susceptibility to social bot attacks seems to be a very promising defense for social bots. By finding out who is susceptible to social bot attacks, the degree for which social bots can manipulate and spread malicious content can be greatly suppressed.

## 4 Social bot detection

Detection of social bots is an arms race. As new detection methods are developed, the social bots also evolve, making the current detection methods less effective. Since social bots are constantly being developed and evading detection, it is important that new methods for detecting them are also being developed continuously. The difficulties lie in the fact that social bots are designed to appear human and to blend in with the crowd. An exception of this are the spam bots on OSNs. Spam bots are usually easier to detect by the fact that they deliver spam. Distributing a high amount of similar posts or messages can easily be picked up both by humans and computer algorithms.

When social bots create accounts with information that seems legitimate, other users are easily tricked into believing that the bot is another person. This removes or at least suppress the ability for humans to identify bots on OSNs. What can make it even harder for humans as well as computer algorithms to detect bots, is the use of hijacked accounts. Social bots can hijack existing OSN accounts that is already in use by humans. A hijacked account works as a proxy for the bot. Using such a proxy makes it very difficult for a human to identify that a bot is behind the account in focus. Not only is the account information legitimate, but the account can also be confirmed. A confirmed or approved account is an account that is confirmed to be identified with a person or a company. Using hijacked accounts also make it harder for detection algorithms to detect bot activity. This is because a hijacked account have behavioral and feature characteristics of a legitimate account. This can render behavioral and feature based detection methods useless. As an answer to research question 2a ("What is the main problem regarding detection of social bots?"), the main problem regarding detection of social bots is that they are advancing continuously. When new detection methods come out, the old social bots are redesigned to evade detection. This ends up in what seems to be a never ending arms race. By developing new detection methods continuously, we can stay one step ahead of the social bots. Having a bigger arsenal of detection methods will make us ready for new social bots that is deployed. Regarding research question 2b ("Are the current detection methods sufficient?"), the current detection methods are somewhat sufficient. But because social bots are being redesigned continuously, they might not be sufficient for too long. The detection methods that we have today might not be sufficient for the bots that come out the next day. This chapter covers the different approaches for the detection of social bots. In section 4.1 several existing approaches is discussed. In section 4.2, publicly available solutions for bot detection and rating of followers

are discussed. This section also describe a tool which can make it more difficult for detecting social bots. In section 4.3, the arms race problem between detection methods and social bots is discussed. This section also provide a solution to this problem.

#### 4.1 Detection approaches

Detection of social bots can be performed by several approaches. Each of these approaches are better for some scenarios, whilst they are worse for others. Some of the approaches are better to use on specific OSNs. This is because the various OSNs have data available in various degrees. Twitter for instance, makes data about users and posts available through their REST API. Facebook on the other hand does not provide as much information as Twitter does through their API. Naturally this is because Facebook provides more privacy alternatives for a user and that it is a more complex OSN. The complexity and privacy measures of an OSN can have an impact on what detection approach is best to use. In addition, the majority of Twitter profiles are public, whilst on Facebook, most profiles are private. Having publicly available profiles means that more features can be analyzed. Private accounts on the other hand, means that a very limited amount of information regarding a profile. The good thing about publicly available accounts is that more information is available for analyses. Meaning that detection is much easier to perform.

Something that influence the choice of approach is what type of social bot that is to be detected. Different types of social bots communicate through different channels and in different volumes. And some social bots are more stealthy than others. Certain detection approaches can detect different types of bots more successfully than others. A good approach to detect chatbots would for instance be to use language analyses. Behavioral analyses is not necessarily usable with chatbots since they do not give of a behavioral profile.

Numerous contributions to the detection of social bots and bot activity have been made in the recent years. Some studies focus on analyzing the content of tweets, and others focus on feature analysis. What seems to be the most trending detection or indication methods of social bots is behavioral analysis. The reason for this is because behavior can be measured by many different approaches. This section will answer research question 2c ("Which detection methods or algorithms currently exist?"). The section present the different detection methods and approaches that currently exist.

##### 4.1.1 Behavioral analysis

One method to detect social bots is by analyzing behavioral patterns. Behavioral patterns can be patterns of any action performed on a OSN. This can for instance be the time difference for when a user posts a message, or which action a user first performs after logging into the account. Behavioral patterns of bots

and humans often differ from one another. Behavior based detection seems to have been a popular approach the latest years. Many contributions have been made in behavioral analysis and detection of social bots. One of the earlier contributions made by Zhang et al. (2011) [7] focused on analyzing timing patterns of tweets. To test that the behavior belongs to either human or automated pro-

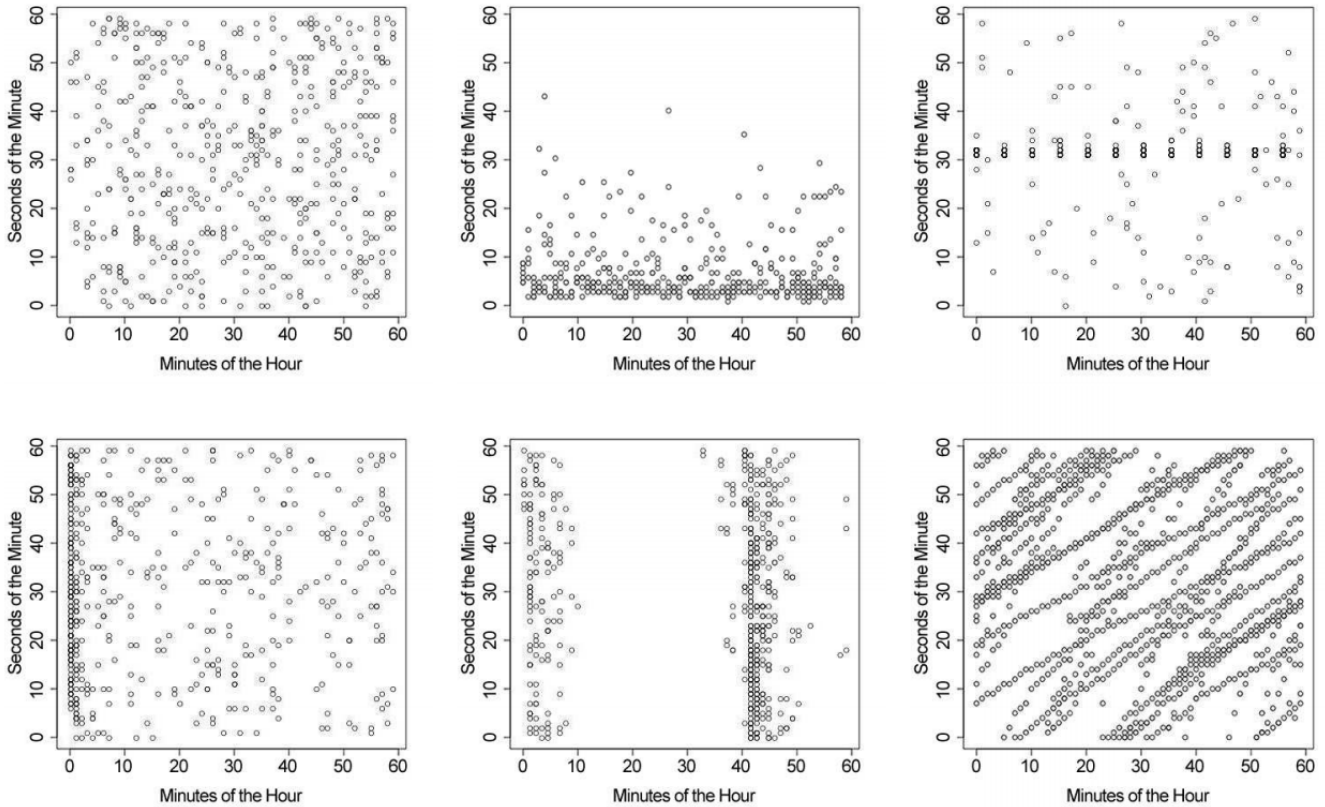


Figure 5: Tweet time distribution of automated processes and humans [7].

cesses (bots), the Pearson’s  $X^2$  test (chi-squared test) was used. The test were used to check if timestamps were consistent with time distributions of human users. The test yields a probability of how uniformly tweets are distributed across second-of-minute and minute-of-hours. As Zhang et al. describes, a low probability indicates tweet time distribution of a human. And a high probability indicate time distribution that is very unlikely for a human to produce, or in other words, likely to be produced by automated processes. As seen in figure 5, tweet time distribution of different accounts vary significantly. The X axis describe the minutes of an hour and the Y axis describe the seconds of the minute. Tweet time distribution from a human is expected to be be more random and with less clus-

terings. The upper left graph in the figure shows a great example of how tweet time distribution of a human user should appear. Time distribution of automated processes is expected to be more clustered (i.e. of less random distribution). The more clustered the distributions are, the more likely it is to be related to automated processes. The problem with this detection method is that it can easily be evaded. This can be done by randomizing the timings of tweets. If a social bot is designed with a high degree of randomness, tweet times can be heavily randomized. This can further make it very hard for this detection method to actually detect uniformity or non-uniformity, and make it harder to distinguish human and bot activity.

In 2012, another study was performed on detecting bots on Twitter with a behavioral approach [13]. In this study tweeting behavior, content of the tweets and account properties were analyzed to classify users as humans, bots or cyborgs. In their work, they refer to cyborgs as users who have both human and automated characteristics. More in detail, their study focused on number of friends, number of tweets, tweet frequency and which device were used and many more. When analyzing relations between friends and followers, the study showed that humans usually have friends and followers of approximately the same number. Bots on the other hand usually have a lower number of followers than number of friends (true for about 60% of bots [13]). This is due to a strategy used by bots. As explained by Chu et al., following back is sort of an etiquette on Twitter. Bots exploit this etiquette by following a large number of users where some of the users will following them back. Having many followers makes a social bot reach out to more people and being more influential. When analyzing number of tweets, the study showed that humans post more tweets than bots. The explanation for this is that social bot accounts either hibernate or get suspended from time to time. But when the bots are active, they tweet more frequently. Studying the devices usage, the results showed that 50.53% of humans tend to access and use Twitter via the twitter.com website. And 42.39% of bots access Twitter through the API.

When testing their detection solution, the results were 98.6% correct detection for humans and 97.6% for bots. Detection of cyborgs were a little less accurate. Both humans and bots were also misclassified as cyborgs. The explanation for this is that the cyborgs are accounts that is used by both humans and bots (i.e. having behavioral properties of both humans and bots).

#### 4.1.2 Content and language analysis

While some social bot detection approaches look at behaviors found in metadata such as timestamps, other approaches analyze the actual text content of tweets. Social bots can be detected by analyzing the mood a tweet have, and which types of words are used. In the development of the social bot detection tool BotOrNot, Davis et al. [12] implemented analysis of linguistic properties through Natural



Language Processing (NLP) and sentiment features in language. By using NLP, a computer program can analyze if a piece of text is of natural (human) language. Hence, by using NLP, a computer program can also determine if some text is of unnatural nature. To be able to analyze text in such a way, the computer program needs to learn what natural language is. This is done by mapping human generated text in such a way that a computer can use it to differentiate natural and unnatural language.

In 2014, Dickerson et al. [5] researched how to differentiate between humans and bots by analyzing linguistics of tweets. Regarding the content of tweets, their solution look at the average number of hashtags, the average number of user mentions, average number of links and the average number of special characters. Regarding semantics, the solution analyze features such as average topic sentiment on each topic by user, sentiment polarity fractions, contradiction rank and positive and negative sentiment strength. In their study, 7.7 million tweets and over 555,000 users were analyzed. One of their findings showed that humans tend to have stronger positive sentiment than bots. This means that humans tend to show positive feelings much stronger than bots do. Their study also showed that humans change their their sentiment more often than bots do. Through the study they also discovered that bots tend to disagree much less than humans do.

In addition to other types of analyses (e.g. behavior analysis and crowdsourcing), content and language analyses seems to be a very promising **additional** method for detecting social bots. Language analyses can be used to successfully differentiate between human and machine generated text. Therefore it is a detection approach that should be considered when creating solutions for detecting social bots. In addition to detecting social bots, content and language analyses have other applications such as modeling users' susceptibility to social bots (see section 3.7.2).

#### 4.1.3 Crowdsourcing

Another social bot detection method that differentiates from the other detection methods is crowdsourcing. Crowdsourcing works as a social Turing test in which people participate in determining if a given OSN account is fake (i.e. a bot). Wang et al. (2012)[8] performed a study in which they examined the feasibility of such a crowdsourcing solution. Their study show that crowdsourcing detection of social bots is highly effective. As Wang et al. states, crowdsourcing seems to be a very effective method for detecting social bots for several reasons:

- Human intelligence can solve problems that computer algorithms can not.
- The workload is spread over many people so that one specific person or group is not overburdened with the task.
- The size of the work group can change dynamically.

- New workers can be recruited on-demand.

What is perhaps the greatest feature of crowdsourcing social bot detection is the use of human intelligence. Humans can detect certain inconsistencies that automated processes cannot. Humans can easily detect inconsistencies in certain areas such as language and context. Spreading the workload of detecting social bots over many people can eliminate a OSN's need to employ people dedicated for the task. If not eliminating the need for dedicated employees, it does spread the workload. Crowdsourcing offers scalability by dynamically changing the size of the work group as well as on-demand recruitment of workers.

As [8] states, non-experts tends to have lower accuracy than experts when it comes to detecting social bots. Non-experts also tend to have a goal of finishing the task quickly. Meaning that their only focus is to get paid. But these accuracy problems can be eliminated by having experts calibrating ground truth filters. Crowdsourced detection of social bots seems to be a very promising approach. It offers scalability, it is cost-effective and yields a low false-positive rate.

#### 4.1.4 Honeypots

All of the other detection approaches discussed in this section are active approaches. Meaning that accounts and their related data is analyzed to look for certain anomalies based on a theory. A more passive approach is to deploy honeypot accounts on OSNs. Honeypot accounts are accounts that are designed to attract social bots. The account activity on these honeypot accounts is inspected to see the properties of the accounts that enables contact with the honeypot accounts. As part of a study in 2010, Stringhini et al. [14] deployed 300 honeypot accounts on several OSNs including Twitter. The honeypot accounts were active for 12 months. When analyzing the data, the researchers found out that a great portion of accounts that contacted the honeypot profiles were actually legitimate human accounts. Therefore, to separate human and bot controlled accounts manual inspection of all the accounts were needed. After inspection, the results showed that on Twitter, out of 397 contacts, 361 were spam bots. Although the results were very different on Facebook and MySpace, this study showed that honeypot detection of bots on Twitter is very successful.

The problem of this approach for detecting social bots is that it is time consuming. Data needs to be collected over longer periods of time. Over longer periods of time, the social bot accounts can already be deleted, or even abandoned. Deploying honeypot profiles for detection of social bots is not as efficient as the other approaches. But the honeypot approach is very good for researching social bots and identifying the features and behaviors that the social bot accounts inhibit.

## 4.2 Publicly available solutions

As social bots are a rising concern, many studies have been made to develop new methods for how to detect them. Some of the methods and approaches developed have been released as publicly available tools. This section describes two publicly available tools that can be used to detect social bots and fake followers. In addition, one tool that can make detection of social bots harder is discussed. Section 4.2.1 describes BotOrNot which has academic literature available for it. This tool is described because it was developed by one of the leading persons on social bot research Emilio Ferrara. Section 4.2.2 describes the tool Twitter Audit. Twitter Audit does not have academic literature available to describe the tool in detail, but it is an interesting tool that gives a lot of interesting information regarding fake followers. Section 4.2.3 discusses the tool TweetDeck and the problem it can cause for social bot detection.

### 4.2.1 BotOrNot

In 2014, Davis et al. [12] developed a service that analyzes if a Twitter account inhabits human or bot characteristics. The service is available through <https://truthy.indiana.edu/botornot/>. To check the bot and human characteristics of a Twitter account, the service uses a Twitter screen name (i.e. username) as input. After pressing "check user", the service gives a score on how likely the given account is to be a bot. The service provides interesting details such as graphs about retweets or mentions and a sentiment score. The service provides interesting information on retweets, mentions, sentiment and analysis of tweet content (e.g. use of verbs or nouns). In addition to being available through the website, BotOrNot is also available through a python api. This allows for flexibility and availability for developers. Figure 6 shows the result yielded by BotOrNot when analyzing one of the Twitter accounts of the U.S. president Donald Trump (@realDonaldTrump). As the figures show, BotOrNot determines a 57% chance of the account is of a bot.

BotOrNot uses the Twitter REST API. Therefore, to be able to use the service, a Twitter authorization is needed. This means that to use the service, a user needs to log into Twitter. As described by Davis et al., the service does not gather information on who submits the requests. But the results of the requests are stored to further improve the service.

The service analyzes six groups of features:

**Network features.** Networks are made by the use of retweets, mentions and hashtag co-occurrence. The statistical features are extracted and used in the algorithm.

**User features.** Metadata related to a Twitter account (e.g. language and geolocation).

**Friends features.** Features of the social contacts of a Twitter account. Including

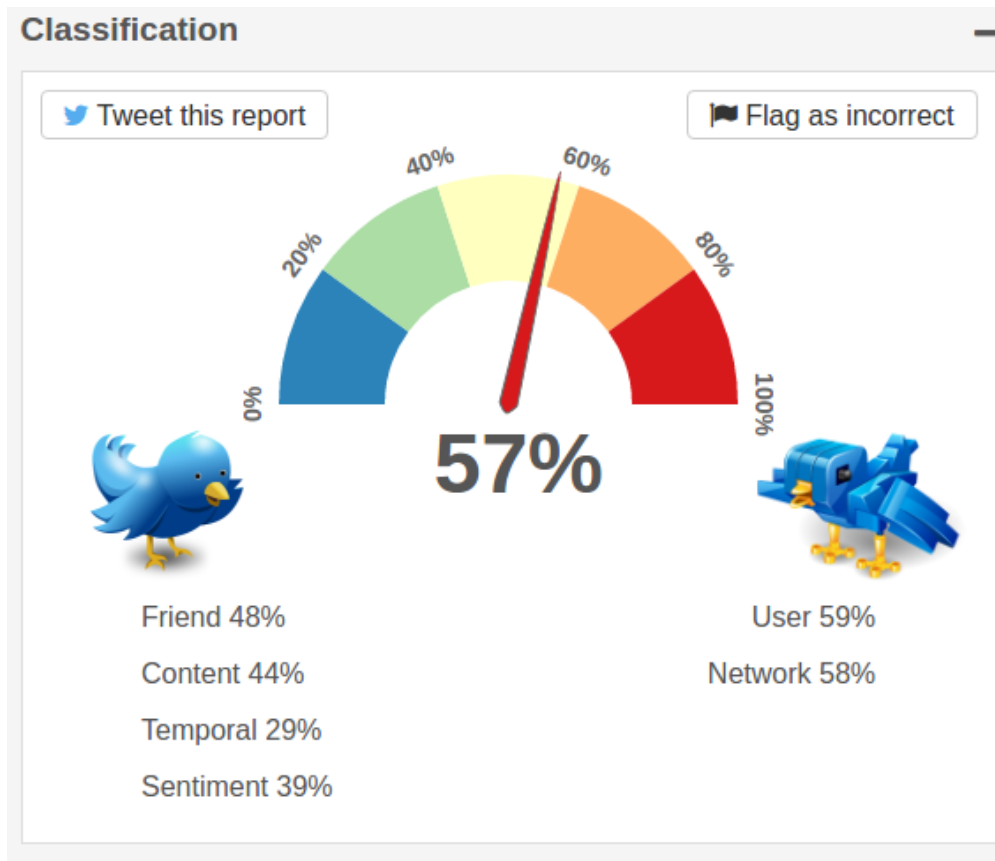


Figure 6: Result yielded by BotOrNot for @realDonaldTrump.

followers, followees and tweets.

**Temporal features.** Timing patterns such as tweet rates and inter-tweet time distribution.

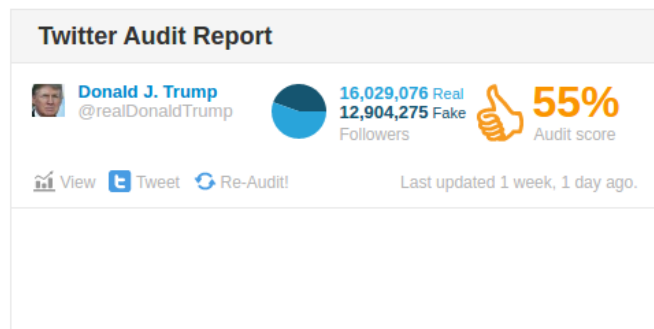
**Content features.** Linguistic properties of tweets analyzed through Natural Language Processing (NLP).

**Sentiment features.** The degree of happiness, emotion scores and arousal-dominance-valence.

Within the different groups, there are over 1,000 different features. To classify presence of bot or human, BotOrNot use all of these features. To train the classifier, Random Forest is used. BotOrNot use a total of 7 classifiers. One for each group of features, and one for the overall score.

### 4.2.2 Twitter Audit

Twitter Audit is a tool that is used for analyzing how many followers of an account is fake. The Twitter Audit tool is available at <https://www.twitteraudit.com/>. There is no scientific article related to Twitter Audit. So information on how analyses are performed is non-existent. But it is a solution that have gotten a lot of press coverage and popularity in the online community. Just as with



How TwitterAudit sees @realDonaldTrump

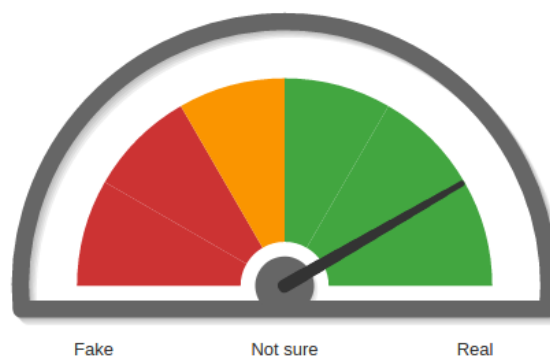


Figure 7: Result yielded by Twitter Audit for @realDonaldTrump.

BotOrNot, a Twitter screen name is used as input. The output shows a score of the account analyzed, a quality score per follower and "Real Points" per follower. The output also shows how many of the account's follower are real and fake. Figure 7 shows the results of analyzing the account @realDonaldTrump. Interesting as it is, the analysis shows that 44% of @realDonaldTrump's followers are fake. Although Twitter Audit is not a social bot detection tool, it is an interesting tool for analyzing a user's followers on Twitter.

### 4.2.3 TweetDeck

Twitter offers an extended tool, TweetDeck. With this tool users can manage multiple timelines and performed management of accounts that extend the stan-

standard Twitter framework. One of the things that users can do with TweetDeck is to schedule tweets. When a user can schedule tweets to be released at certain times, detection methods involving timestamps can give higher false positive rates. TweetDeck is a great tool for Twitter's users. And it can be very useful in many cases. But the problem is that it can make it harder for detection methods that analyze behaviors in tweet timings. By using TweetDeck, detection methods that analyze at which hour or minute a user posts a tweet can give higher numbers of false positives. Naturally, false positives is something one wants to avoid when detecting anomalies. A high number of false positives leads to less effective detection of social bots.

### **4.3 A neverending arms race?**

As described in the introduction section of the thesis, the detection of social bots seems to be a neverending arms race. For every new bot detection method, the bots find workarounds to evade detection. And for every new workaround made for the bots, new detection methods are developed. This is not beneficial in the long haul. Making new detection methods costs money and does require a great amount of work-time. Because of this arms race problem I propose the question "How can we end the war between social bots and detection methods?". This section describes the arms race problem of social bots and detection methods. A proposed solution to this problem is discussed in section [4.3.2](#).

#### **4.3.1 Advancements of social bots**

One of the greatest challenges when it comes to detection of social bots is that new social bots are developed continuously. And these new bots are adjusted to not be detected by the current detection methods. The latest years, social bots have become very sophisticated. And many social bots are now able to behave like humans on OSNs [11]. Because the behavior of social bots is so close to that of humans, it has also become more difficult to separate the behaviors of the two. Many of the techniques developed just a few years ago can be useless to detect the new and more advanced social bots. Because social bots are becoming more advanced continuously, it is important that new approaches for detecting them are being researched.

#### **4.3.2 Proposed solution to the arms race**

One solution to the arms race could be to implement personal identity authentication when registering a OSN account. By implementing a solution like this, every OSN account is related to a person. Therefore, there cannot be any fake accounts. There can still be accounts used by bots. For social bots to create these accounts, a human identity is needed. And it is very unlikely for a human to "donate" his or her identity. This is because activity of the bot can be traced back to the human identity. The suggested solution can be thought of as MinID. MinID is a identification system for Norwegian citizens. With MinID, people in

Norway can sign official documents, perform banking online and many more actions connected to their identity. As a best case scenario a solution like this could eliminate the problem of malicious social bots. If not eliminating the problem of social bots, such a solution would significantly decrease the presence of them on OSNs. It might not eliminate remove social bots because there might be a black market for personal identities. If such an authentication system were to cover the entire human population, certain areas where people are not present on the Internet could be subject to such a black market.

A world-covering identity authentication solution might work well, but it would be comprehensively complex. The world have about 8 billion inhabitants. A system based on today's technology might not be able to handle such a complex register. In addition, this worldwide ID service needs to be regulated. And it could be a problem to determine who should regulate such a system. Each state would want to control the IDs of its inhabitants. And the different states would not necessarily accept IDs from other countries. With such a privacy sensitive system, there is also the problem of spying, hijacking of identities and creation of bogus identities. This can be a problem between states such as the U.S. and Russia. For a system like this to not be prone to such exploits, it must be designed in a way so that it can not be misused.

Although such a solution could end the arms race between detection methods and social bots, it is a utopia. And probably would not work very well in practice.

## 5 The proposed indication scheme

This chapter is dedicated to the proposed solution. Section 5.1 describe the different components and technologies used for developing the solution. Section 5.2 describes how data from Twitter is handled. In section 5.3, the solution is described in detail with focus on the different method of analyses. Section 5.3 also describe how the solution works.

The programmatic part of the solution is available in its wholeness at <https://www.dropbox.com/sh/j0fychymjwm3u98/AACsGN0Ce0Jy-BjfQ1eqPTY0a?dl=0>.

### 5.1 Components

This section describe the different components and technologies used in the solution.

#### 5.1.1 The Twitter REST API

To access content on twitter, the Twitter REST API were used. Through the REST API, Twitter enables interaction with the Twitter platform. Through this API, developers can send and retrieve a variety of information to and from Twitter. One of the greater aspects of this API is that it makes a lot of information available. Twitter provides great documentation for the API. And there are several resources available online that provides examples of how to use it.

To be able to use the API, a twitter account is needed and a Twitter application has to be made on Twitter. In order to perform calls with the Twitter API, a set of authorization keys are needed. These authorization keys are available when a Twitter user have created a Twitter application. Each Twitter application have its own set of keys. When these prerequisites are in place, a library for the API is needed. Twitter have libraries available for most programming languages. But only libraries for Java and Objective-C is built and maintained by Twitter. Other libraries are developed and maintained by other entities, but are accepted by Twitter. In this study, the *twython* library, a python wrapper for the Twitter API were used. This library is very easy to use, and is well maintained. It gives simple, yet complete access to the Twitter REST API. In combination with the Python programming language, *twython* makes interaction with Twitter on a programmatic level very easy.

#### 5.1.2 Python

To be able to interact with Twitter, a programming language is needed. For the programmatic part of this study, the programming language Python was used.



Python is a high-level programming language which is very easy to learn. Python is a great tool for fast prototyping of computer programs. In addition, Python have an extensive community online, making it a great programming language for both beginners and experienced programmers. Because of this, as well as the good reputation of the *twython* library, Python came as a natural choice of programming language.

Python mainly comes in two versions; versions 2 and 3. Because of personal preferences and earlier experience, the choice fell on version 2. More specifically the latest release 2.7.13 in the time of writing.

To be able to interact with python and read data, a *twython* object is needed. With this object, data can be read and written to Twitter. To instantiate a *twython* object, the two sets of keys and secrets are needed. The first set includes the *consumer key* which is used to identify which twitter application is used, and the *consumer secret* which is a password used to authorize the client. The other set which enables API calls to be performed consists of a *access token* for identification and a *access token secret* for authorization. A *twython* object can be instantiated as follows:

```
twitter = Twython(
    app_key='XXXXXXXXXXXXXXXXXXXX',
    app_secret='XXXXXXXXXXXXXXXXXXXX',
    oauth_token='XXXXXXXXXXXXXXXXXXXX',
    oauth_token_secret='XXXXXXXXXXXXXXXXXXXX')
```

(actual keys are replaced by X'es)

In the *twython* library "app\_key" is *consumer key*, "app\_secret" is *consumer secret*, "oauth\_token" is *access token* and "oauth\_token\_secret" is *access token secret*. Now, the *twython* object (in the code example above named *twitter*) can be used to read and write to Twitter.

### 5.1.3 Virustotal

The ability to analyze URLs is made available through the services of Virustotal (virustotal.com). Virustotal is a subsidiary of Google which enable analyses of files and URLs. Files can be uploaded to virustotal.com to analyze if it have malicious contents. Through their web interface, URLs can also be analyzed for malicious content or malware. The services of Virustotal is available through their public API . By using this API, developers can analyze files and URLs. In the study, the Virustotal API were tested for malicious URLs. Testing was performed with a random selection of URLs from a URL blacklist provided at <http://www.urlblacklist.com/?sec=download>. All the URLs from the blacklist that were tested was reported in at least one of the 65 databases that Virustotal check a URL against. To access the API and make calls, developers need to register an account for Virustotal. When an account is created, an API key becomes

available. The API key needs to be included every time a call to the Virustotal API. The following code snippet shows how a URL can be analyzed with the Virustotal API:

```
params = {'apikey': 'XXXXXXXXXXXXXXXXXX', 'resource': url}
response = requests.post('https://www.virustotal.com/vtapi/v2/url/report',
                        params=params)
```

(actual api key is replaced by X'es)

To be able to perform analyses of URLs, a dictionary containing the api key and the url to be analyzed is needed. Both values needs to be strings. Then a simple HTTP POST request is performed at <https://www.virustotal.com/vtapi/v2/url/report>, with the dictionary params as parameters. The service will respond with a response containing the results for the analysis. The response contains details about which out of 65 sources that determine the URL of being malicious. The following example shows the response when querying the URL "02seo.com/iag7a".

```
{u'filescan_id': None,
 u'permalink': u'https://www.virustotal.com/url/83b779a16a7da35af9d57a3ef4c51839b90c64eda613030b58164375d2fa2fc2/analysis/1494488300/',
 u'positives': 8,
 u'resource': u'02seo.com/iag7a',
 u'response_code': 1,
 u'scan_date': u'2017-05-11 07:38:20',
 u'scan_id': u'83b779a16a7da35af9d57a3ef4c51839b90c64eda613030b58164375d2fa2fc2-1494488300',
 u'scans': {
  u'ADMINUSLabs': {u'detected': False, u'result': u'clean site'},
  u'AegisLab WebGuard': {u'detected': True, u'result': u'malicious site'},
  ...
  u'AutoShun': {u'detected': True, u'result': u'malicious site'},
  u'Avira': {u'detected': True, u'result': u'malware site'},
  ...
  u'BitDefender': {u'detected': True, u'result': u'malware site'},
  ...
  u'ESET': {u'detected': True, u'result': u'malware site'},
  ...
  u'Fortinet': {u'detected': True, u'result': u'malware site'},
  ...
  u'Kaspersky': {u'detected': True, u'result': u'malware site'},
  ...
  u'Sophos': {u'detected': True, u'result': u'malicious site'},
```

```

    ...
  },
  u'total': 64,
  u'url': u'http://02seo.com/iag7a',
  u'verbose_msg': u'Scan finished, scan information embedded in this object'}

```

The URL used in this query is taken from the blacklist provided by [urlblacklist.com](http://urlblacklist.com). In the example, every negative (False) finding is removed and replaced with "...", except from the first entry in "scans". As seen in the example, every entry in "scans" contain details about which source defines the URL as clean or malicious. For the URL queried in this example, a total of 8 sources defined the URL as either "malicious site" or "malware site". If the URL have not been scanned before, the response contains information about this. When the request is performed, the developer have the alternative to submit a URL for scanning. This can be done by adding the entry 'scan': 1 to the dictionary params. Scanning a URL does take a great amount of time, therefore the solution does not include submitting URLs for scanning. As is, the solution is only able to retrieve reports on already performed scans.

## 5.2 Data collection

To be able to check for the various indicators, the program collects data as an initial step. This means that the program collects the data just before the evaluation is in process. All the data fetched from Twitter is retrieved as json data. Because there are several functions within the program, tweet data and user data needs to be saved to the disk. This way, every function that need tweet data or user data can read the data as many times as needed from a file. Collecting and saving the data also allows for repeated or reenacted analyses of the data. By saving the data, we also have it available as evidence. Tweet and user data holds a lot of meta-information that can be used for further analyses.

When using python with the twython library, user information can be fetched with the following line of code:

```
user = twitter.show_user(screen_name=username)
```

This code will retrieve all information available through the Twitter REST API of a particular user. In this particular case, user data is retrieved with the use of the *screen name* of an account. In the code, *username* is a string with the *screen name* to retrieve user data from. Very similarly, a user's tweets can be retrieved with the following code:

```
tweets = twitter.get_user_timeline(screen_name=username, count=200)
```

The code example above will retrieve the 200 last tweets of a user with a *screen name* equal to the string *username*. All the tweets retrieved will be constructed to an array with json objects. One json object for each tweet. Because of the Twitter REST API having read limits, the solution is able to retrieve a maximum of approximately 3200 tweets from a user.

After the data is retrieved, it is saved to a file with a filename containing the *screen name* and what type of data the file holds. A file containing user data will be named "screenname\_userinfo", and if it contains tweets it will be named "screenname\_tweets".

The function `savetojson` takes care of writing data to a file with correct filename. But to get the correct filename format, a string with the ending of the filename is needed as a parameter. The functions `displayuserinfojsonfile` and `displaytweetjsonfile` are used to load user data and tweets from files and into a variable. This way, every function that needs to use user or tweet data can retrieve it in a simple manner instead of downloading it several times. These two examples shows how easy information can be fetched from Twitter.

The ability to capture new tweets were considered in the solution. This functionality is provided by Twitter through the Twitter Streaming API. When using the Streaming API, the developer can instantiate a listener to listen for certain hashtags or tweets from a particular user. However, when this API is used, only a small portion of data can be captured. To be able to capture every tweet that matches a certain criteria, a service called *Firehose*. *Firehose* is provided by Twitter through its enterprise API platform *GNIP*. But the *Firehose* service is in the hundred thousand to million dollar price class. Therefore this solution was too expensive to be used in this study. In addition, having a solution that listens to tweets in real-time takes longer time. One of the ideas behind the solution was that it should be fast. By using the Twitter REST API instead of *Firehose*, analyses can be performed in a matter of seconds.

### 5.3 The solution

By looking at the properties of a user's profile and its tweets, we can find traces of behavior for a particular user. Often, human users give off a different behavior than social bots do. The Twitter REST API enables collection of a lot of information. This is good for research purposes and can help researchers identify social bot activity. But the real challenge lies in the difficulty of analyzing the correct properties.

#### *Vision*

The vision of the proposed solution is to give fast and broad analyses of Twitter accounts. Instead of analyzing data collected over time, the solution enables analysis of an account in about one to four seconds (analyses of URLs excluded). Fast detection of social bots is crucial for eliminating the spreading of malware

and spam and other malicious activities these bots can cause. By having an indication program as developed in the work of this thesis, we can minimize the impact that social bots can cause. Many different types of social bots exist, and these often have different behaviors. By analyzing a broad set of properties and statistics, indication of different behaviors can be identified. Hence, different types of bots can be detected.

### *The idea*

Instead of having a social bot detection tool, the idea behind the solution is to have a program that can find indications of social bot activity. When indications of social bot activity is present, the idea is that a human part perform the rest of the analysis. This is because humans can detect small inconsistencies that can be hard for a computer program to find. This can for instance be details such as content and language in tweets, the actual profile image of an account or age of the person in relation to the profile image. To have the best grade of judgement, the human part should be someone with substantial knowledge and training for detecting social bots. A crowdsourcing approach as described in [8] might also be considered instead of having a dedicated group of people. By implementing testing of the "crowd", inaccurate individuals can be eliminated. This way, a crowdsourcing approach can be very effective. The digital part of the solution (the program) is meant as a filter to separate suspicious accounts from legitimate ones. If a suspected account is encountered, this account is then further analyzed by a human part.

This section describe the different approaches used in the proposed solution for indicating bot activity on Twitter. Section 5.3.1 describe the flow of the programmatic part of the solution. Section 5.3.2 describes the indicators related to the profile of a user. Section 5.3.3 describe indicators for analyses of tweets. Calculation of the score is described in section 5.3.4.

#### **5.3.1 Program Flow**

This section describe the flow of the proposed solution. Each step is related to the steps 1 through 4 in figure 8. To use the program of the proposed solution, the user needs to provide a Twitter screen name (step 1). After the program have started, profile information and tweets belonging to the given screen name is downloaded and stored (step 2). Then the program starts to analyze the downloaded data (step 3). In the analysis phase, the user of the program is prompted with various information regarding the findings. If any URLs are found in the tweets, the program prompts the user if analyses of the URLs is wanted or not. If the user answers yes ("y"), the user is prompted again. This time including the approximate time it will take to analyze the set of URLs. If the user answers yes ("y"), analyses of the URLs will proceed. Analyses of URLs are made optional

because it can be a time consuming process. Through testing, most URLs did not take more than 5 seconds to analyze. Therefore, when resolving shortened URLs, the program sets a time limit of 5 seconds. If the process of resolving a URL takes more than 5 seconds, the resolving is aborted and the program proceeds with resolving the next URL.

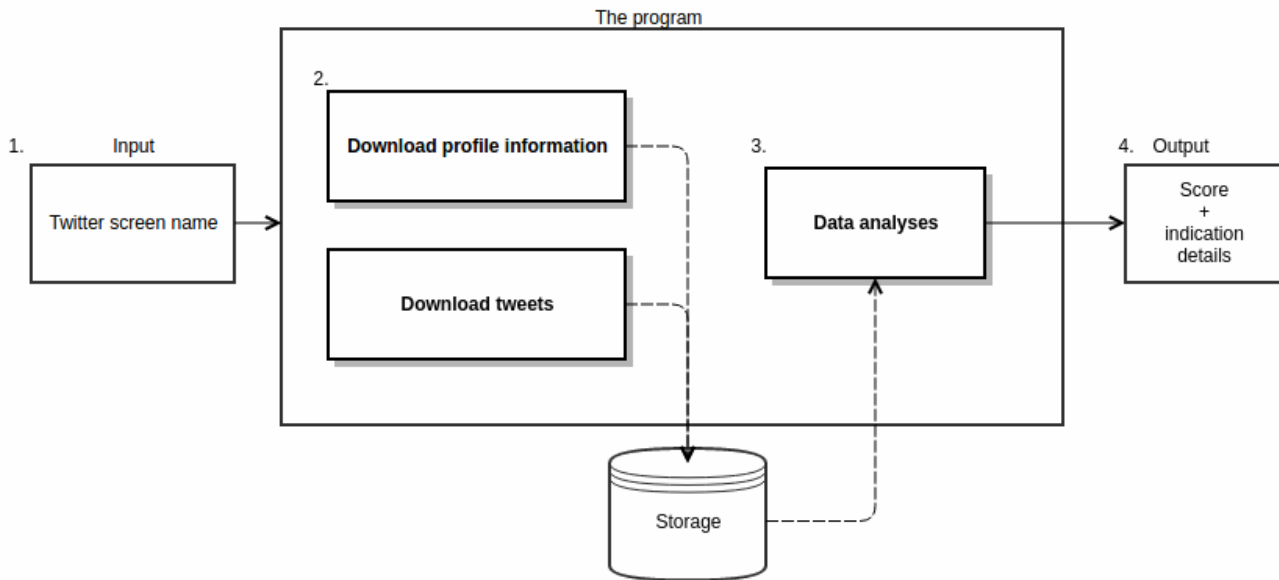


Figure 8: The flow of the proposed solution.

When the program have completed the analyses, the user of the program is prompted with the score based on the findings and which indicators were activated during analysis (step 4). The main part of the work in making the indication program is in step 3 (data analysis). All the different analyses further described in sections 5.3.2 and 5.3.3 are performed in this part of the program.

### 5.3.2 User analyses

This section describe analyses of features related to the user profile. The indicators are described with the ideas for why they should work. The ideas behind some of the indicators were inspired by earlier studies and some are based on entirely new ideas. For each indicator, comments are made if it were inspired by earlier studies. If not commented, the indicator is based on new ideas.

#### *Profile picture*

When social bot accounts are created, the bot designer might not bother to set a profile picture. This might not be necessary depending on the purpose of the

bot. If the bot does not need to look legitimate, a profile picture is not needed. If a profile picture is not set for a Twitter account, the account have the default profile picture (shown in figure 9).

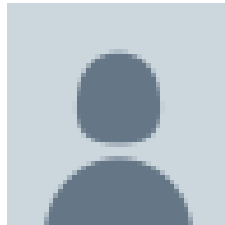


Figure 9: Twitter’s default profile picture.

When user profile information is retrieved, the attribute `default_profile_image` is set to `true` or `false`. If a user have not changed the profile picture and have the default, this attribute is set to `true`. The solution use this attribute to determine if a user account have the default profile picture. Checking an account for default profile picture can be a good indicator for social bots that are simply designed. However more advanced bots will usually set a profile picture to blend in. Therefore, this indicator will only detect social bots that are designed simple. Yet, if an attack is performed with the use of many fast produced Twitter accounts, checking for default profile picture is a good indicator. After all, setting a profile picture is often the first thing a human does when creating a Twitter profile.

#### *Profile name*

The solution also analyze the name associated with the profile. This is a very simple analysis which looks for spaces in a name. This analysis is based on the assumption that all human users usually have at least two names. One or several first names, and one or more last names. Hence, the name string should contain at least one space. The right part of table 1 shows examples of real names (spaces are shown as "\_"). This indication works when bots are automatically generated

Random-generated name	Real name
d9lj3n5dfh2	John_Doe
d9lj_3n5dfh2	Joan_Doe

Table 1: Examples of types of names.

with random names. A random generated name is a string of random letters and digits. An example of a random name is shown to the upper left in table 1. In some cases, when bots are generated in higher numbers, the bot designer might

not bother with giving the social bot profiles actual names. Instead, a much simpler approach would be to generate a random name. If the bot designer want to give the social bot profiles actual names, a list of names need to be generated. This is indeed a very simple task. But it is an extra task to do. And if the social bots to be generated does not need to be stealthy, this extra task does not need to be performed. Inherently, this analysis does not necessarily indicate presence of bots that are more advanced and stealthy. In addition, this indication can easily be evaded. If a bot designer name a bot with a random name, a space can be added (example in lower left in table 1). Then the name would be verified as a real name according to the analysis. Therefore, this indicator is only good for detecting social bots that are designed with simple auto-generated names.

#### *Inactivity*

To be able to detect inactive accounts, the solution performs analysis of a combination of the account age and the number of tweets. This analysis is based on the assumption that if the Twitter account of a human user is over 30 days old, at least one tweet is posted. Therefore if an account contain no tweets and is 30 days older or more, it must be inactive. At the same time, the days since last activity is also analyzed. The last activity threshold is also set to 30 days. The last activity analysis is based on the assumption that a human user posts at least one tweet per 30 days.

A typical feature of some social bots is that they hibernate for longer periods of time [13]. This can be due to either the account being suspended or incubation for future attacks. By analyzing inactivity of an account, the solution can indicate social bots that hibernate. By analyzing inactivity, stealthy social bots can be detected.

#### **5.3.3 Tweet analyses**

This section describe how the indicator program analyze tweets. Each indicator is described with the underlying idea for why it should work. Some of the indicators were inspired by earlier studies, and some are new ideas based on inspection of metadata in tweets. In cases where the indicator is inspired by earlier studies, this is commented. If no such comment is present, the indicator is based on new ideas.

#### *Tweets at hours of a day compared to the average*

By the assumption that humans only tweet in their waking hours, analyzing at which hour of a day tweets are posted can be used as an indication of social bot activity. Analyzing tweet hours was inspired by Zhang et al.[7]. In their study, tweets at the minute of hours were analyzed to look for uniformity of tweet patterns. Instead of analyzing tweets at minutes of hours, the proposed solution analyze at which hours of a day tweets are posted.



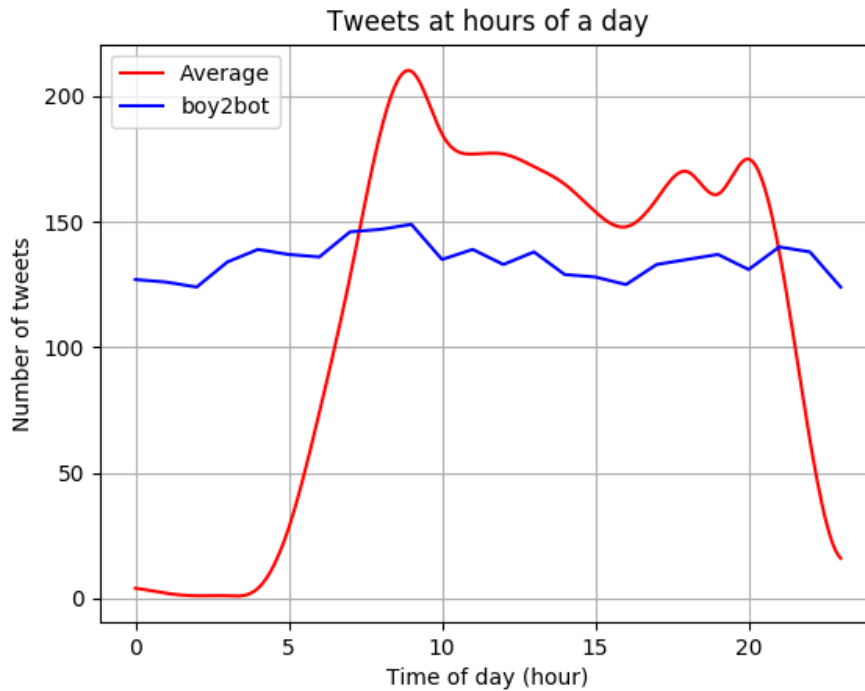


Figure 10: Tweet hours of the account boy2bot compared to the average of a set of accounts.

Instead of looking for uniformity, at which hours of a day a user post tweets, the hours which a user tweet is compared to a pre-calculated average. The pre-calculated average is created by analyzing a set of accounts. The hours of when a tweet is posted is collected from sets of tweets from the set of accounts. When all the hours are collected, the average number of tweets for each hour of a day is calculated by dividing by the number of accounts (set of tweets) analyzed. If the tweet hours for a specific account exceeds a certain threshold for difference between the average, an indication is set. For testing of functionality, this threshold is set to difference of 100 tweets for one specific hour of a day. Figure 10 shows a comparison for the account boy2bot (in blue) and average tweet timings of a set of accounts (in red). In this example the average tweet timings is created from accounts of Norwegian politicians. Norwegian politicians are good representatives for individuals that have typical waking hours for anyone with normal work times. The account boy2bot is a bot that takes random tweets with the word boy in them and change it to bot and retweet the tweet. This bot is used as an example because it works completely automatically. As seen in the graph in figure 10, there are big differences between the average and the bot.

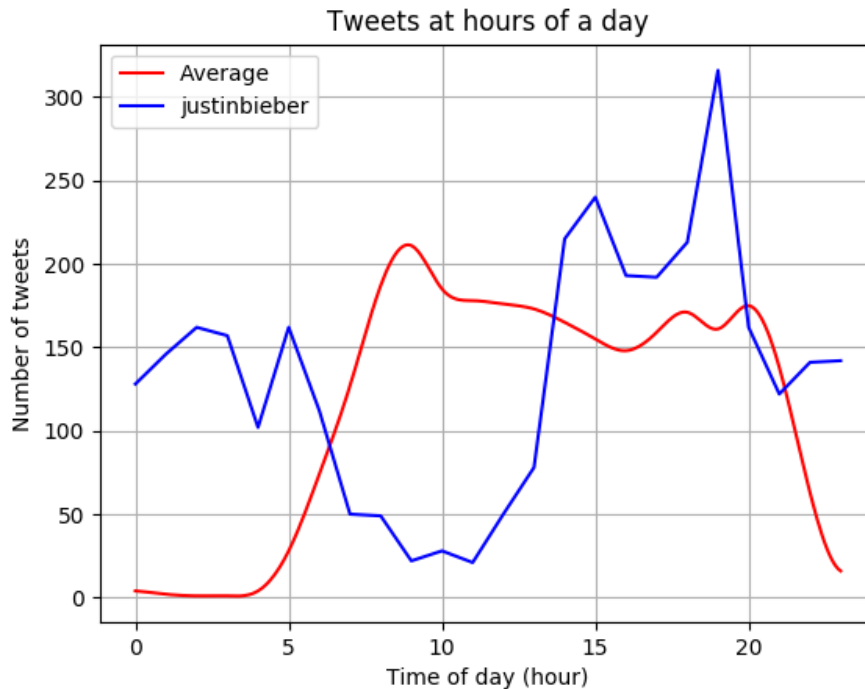


Figure 11: Tweet hours of the account justinbieber compared to the average of a set of accounts.

However, there are a few cases where this analysis will yield a false positive. When testing the solution, analyses of celebrity accounts often gave a completely different result than that of other human accounts. As an example, figure 11 show the tweet hours of the account @justinbieber compared to the same average as in figure 10. The account @justinbieber is a verified account of the musician Justin Bieber. As seen on the chart, the tweet hours of @justinbieber deviates significantly from the average. Therefore, this indicator might give more false positives in cases where the users have different waking hours. Examples are for instance some people that work during the night rather than the day, or that have irregular awake hours.

Research question 3a states: "Can presence of social bots be indicated by analyzing at which hour of a day a tweet is posted?". By testing the solution, the indicator that compares tweet hours to an average set were going to give an answer to this question. Unfortunately, testing became out of scope at the end of work on this thesis. Future testing should involve performing quantitative analyses for both social bots and human accounts. By performing testing of this indicator, the research question can be answered much clearer. However, testing similar approaches earlier have been successful for detecting social bots [7]. Therefore, if the threshold is set adequately, this approach is appropriate for indicating au-

tomated behavior.

#### *Proportion of tweets that have URLs*

It is fairly common to share content on Twitter with the use of URLs. By posting a URL, users can share content provided from different sources. Some users share many URLs, and some users do not share URLs at all. Because URLs can point to other pages, social bots that spread spam also share URLs. Instead of posting malicious content directly as a tweet, a URL is shared to another page that contain the malicious content. Because URLs are commonly used by social bots that deliver spam or malicious content, the solution analyze the proportion of tweets that have URLs. This is done by analyzing a set of tweets and look for the strings "`http://`" or "`https://`". Every find is counted and a percentage is calculated. The percentage is calculated with the amount of tweets that contain URLs, and the total number of tweets analyzed. If the percentage exceeds 60%, an indication is registered.

#### *Malicious URLs*

Spam bots or other malicious social bots on OSNs might share URLs with malicious content. This can be a big problem since other users on the OSN might click on links without thinking where it leads to. And it can often be hard to just look at a URL and tell if it is malicious or not. When malicious URLs are shared on OSNs, the other users are prone to be victims of malware, phishing or other malicious activities. If malicious URLs can be detected, the spreading of malware, spam and other malicious activities can be suppressed or eliminated completely. Most human users would not share a malicious URL intentionally. This can hurt their reputation and can have serious consequences. Therefore, malicious URLs can be an indication of either malicious social bot activity or a compromised account.

To fight the problem of malicious URLs on Twitter, the solution provides analysis of URLs in tweets. By doing this, malicious content can be found and precautions can be taken. The solution also use the presence of malicious URLs as an indication of social bot activity. The ability to analyze URLs is enabled through the Virustotal API (described in section 5.1.3).

To analyze URLs, the solution searches a set of tweets for URLs. URLs are located by looking for the strings "`http://`" and "`https://`" in a tweet. If there exist any URLs in a tweet, the URLs are extracted and checked against Virustotal. On Twitter, URLs are shortened to the format "`https://t.co/restoftheurl`". Instead of checking the "`t.co`" -URLs directly, these shortened URLs are unshortened. By unshortening a URL, the original URL that the shortened URL points to is revealed. This is done by performing a get request on the shortened URL. The functionality of unshortening URLs is only implemented for unshortening "`t.co`"-URLs. In the progress of unshortening URLs, both shortened and unshortened

URLs are printed as output. By doing this, the unshortened URLs are available for further research. When all URLs have been unshortened, each URL is queried against Virustotal. The response from Virustotal is inspected to see if any of the 65 databases report the inspected URL as malicious. If one or more URLs are reported as malicious from one or more of the databases, this is registered as an indicator.

#### *Sources similar to the screen name*

During the research phase of this study, the properties of benign bots were inspected. What many of them had in common was that their main source of tweets were very similar, and often the same as the screen name. This means that the tweets are posted by an API. Normally when humans post tweets, the source is totally different from the screen name. Figure 12 shows examples of different

```
"source": "<a href=\"http://twitter.com\" rel=\"nofollow\">Twitter Web Client</a>"
"source": "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Twitter for Android</a>"
"source": "<a href=\"https://google.com\" rel=\"nofollow\">sorrowjs</a>"
```

Figure 12: Example of different tweet sources.

tweet sources. The first line is from a tweet posted from the Twitter web page. The second line is from a tweet posted by an Android device. The last line is from a tweet posted by the benign bot "SORROWJS". In this case "sorrowjs" is the application registered on Twitter which enables the API access for the SORROWJS bot. All benign bots inspected in the research phase does have a screen name very similar to the source of its tweets. Although some have varying differences, the screen name seemed to be almost identical to the source of the tweets in all of the cases. This indicator is very good for finding benign bots. But in some cases it can also be used to find malicious bots. But this might as well work in cases where a bot is designed with a Twitter application's name being the same as the screen name of the profile of the bot.

#### *Unknown sources*

If a bot is in use of the Twitter REST API to post tweets, a Twitter application is needed. When a social bot use the Twitter REST API, the source of the tweet will be the same as the name of the application (see last line in figure 12). When a human user post tweets, the source is usually referred to as the device the user is posting the tweet from. A device can for instance be iPhone, Android or a web browser (i.e. Twitter Web Client). Examples of expected sources of human-composed tweets are shown in the two first lines in figure 12. Because there are differences in sources between humans and bots, an unknown source could be an indication of bot activity. It is not possible to name a Twitter application as

for instance "Twitter for Android" or "Twitter for iPhone". Therefore, a malicious social bot can not falsify the source from which its tweets originates.

The solution analyze all the sources in a set of tweets and compare it to a pre-defined list of known sources (i.e. sources used by humans). Known sources include the following:

- Twitter for iPhone
- Twitter for Android
- Twitter Web Client
- Twitter for Mac
- Twitter for Windows Phone
- Tweetdeck
- Instagram
- Twitter for iPad

The list above is all the known or verified sources that the solution accepts. If any source that is not in the list is found, this is registered as an indicator. This list is not complete. There do exist other sources which relates to various third-party programs that are intended to be used by human users. A list of known or verified sources is not provided by Twitter. The list above is compiled by inspecting the sources of tweets from several Twitter accounts. Because the list is not complete, this analysis is expected to have a relatively high number of false positives. The number of false positives will naturally be lowered as the list of known sources expands. For such a whitelist of acceptable sources to work well, it needs to be as complete as possible. One way to expand the whitelist is to analyze high number of tweets and inspect every new source that is not in the whitelist already. If a source is identified as a known or acceptable, it is added to the whitelist.

#### *Duplicate tweets*

A typical property of social bots that deliver spam is that they spread the same content several times. Although this method of delivering spam can be quite effective, it is also inherently a weakness. Such activity can be found by analyzing a set of content that is posted. If any content appear several times in a set, this could be an indication that spam is being delivered.

This method of detecting spam is implemented in the proposed solution. The program analyze each tweet in a set of tweets and see if the content of that one tweet is in any of the other tweets. As is, this indicator is set of if the text content of two tweets are exactly the same. This can be problematic in some cases. A simple workaround to evade this indicator would be to for instance change the emoticon in a tweet. If two tweets are equal text-wise, but with different emoticons, this indicator would see the two tweets as different from each other. A better approach would be to analyze portions of the text content in tweets to

calculate a similarity score. This way, the indicator can determine that two tweets with different emoticons are similar even though they are not exactly equal.

*Time difference between tweets*

Very similarly to the inactivity indicator, the solution also analyze the maximum time difference between tweets. The idea behind this indicator was initially to analyze both the maximum and the minimum time difference between tweets. In the time of finish the work on the thesis, the analysis for checking the minimum time difference were not working. To analyze the maximum time difference between tweets, the timestamps are extracted from each tweet. Then, time differences are calculated based on the timestamps. If the maximum time difference exceeds the threshold (28 days), the maximum time difference indicator is activated. As is, this analysis is just a prototype. The idea behind this analysis is that social bots that deliver spam hibernate for a longer period of time [13]. And in their active time, tweets are posted with high frequency. By analyzing both the maximum and the minimum time difference between tweets, the solution would be able to indicate such a social bot.

**5.3.4 Score calculation**

When the different analyses have been performed, the program presents a score. The score is based on all the analyses used. For each analysis performed, a new entry is added to a dictionary (Python dictionary structure). A new entry is added at the beginning of each analysis with a value of 0 (zero). If an indication is found throughout each analysis, this value is changed to 1 (one). Finally, the score is

Analyses num.	Names	Values
1	'defaultprofileimage'	0
2	'nospacesinname'	1
3	'noactivity'	0
4	'thirtydays'	0
5	'duplicatetweets'	1
6	'maliciousurl'	1*
7	'averagedif'	1
8	'applicationcontrolled'	1
9	'urlpercentage'	1*

Table 2: Example score of a fictional analysis.

calculated based on all the entries in the dictionary. All the values are added together, and divided by the number of entries. This gives an average-based calculation of all the values. Represented by the formula:

$$\frac{\text{Values summed}}{\text{Number of values}} = \text{Score}$$

Table 2 shows an example of a score with a selection of indicators. The first column shows the different indication names and the second column shows the values. In this example, the program have analyzed if:

1. the user have the default profile picture.
2. there are spaces in the name.
3. the account contain any activity at all (if the account is over 30 days old).
4. there have been any activity the last 30 days.
5. the account contain duplicate tweets.
6. any tweets have malicious URLs.
7. the tweet timings of the account differs from the average.
8. the account is application controlled.
9. the percentage of URLs in tweets exceeds a certain threshold.

There are two indications (marked with a \*) that is not always used. If no URLs are present in any of the tweets, no analyses are performed on the URLs. To complete the score calculation based on table 2, we first sum all the values. All the values sum up to 6. And there are 9 different indicators. So we end up with the score:

$$\frac{6}{9} \approx 0.67$$

As is, the score is only based on the number of indicators that is found during analyses. The score calculation does not inherit specific weighing of the different indicators. All the indicators are equally weighed. The idea behind the solution is that if any indicator is present, the inspected account should be further inspected by a human part. This means that, if inspection of an account gives any score above 0.0, it should be further inspected.

## 6 Future work

The indication program developed in the work of this thesis is partly based on theory and partly based on earlier studies and research. Some of the indicators are based on already tested concepts, but are still different from these concepts. What is missing in the thesis is actual testing of the indication program. Although partly based on working concepts, the program needs to be tested on both human controlled Twitter accounts and accounts controlled by bots. By testing the solution on both human and bot accounts, the indicators can be adjusted to achieve better and more accurate indications. There is also the problem of testing the solution on actual bots. When testing the solution on a random Twitter account, we cannot know that this account is truly of a human or a bot. By deploying honeypots, social bots can be revealed. When we have identified social bots through the use of honeypots, the solution can be tested on actual bots. Just as the development of detection methods should be continuous, honeypots should also be deployed continuously. Future studies on social bot detection will benefit from this since solutions can be tested on actual bots to see if they work. In a testing scenario, performance of the solution in this thesis should also be measured. The solution should be compared to other solutions to see how well it performs. This is important because some of the indicators might be more effective than existing indicators or detection methods. Knowing which indicators are more effective for revealing social bots is helpful for future studies. The solution in this thesis is meant as a indicator program. Therefore, future testing of the solution should also include the human part of the solution. As a human part, both dedicated people (e.g. people with a background of studying social bots) and a crowdsourcing approach should be tested. Crowdsourcing have been tested before [8], but because social bots are becoming more advanced, further testing is needed on this subject. This is to see if this approach is still viable for detecting social bots.

Improvements on the specific indicators are suggested where they are described (see sections 5.3.3 and 5.3.2). The scoring system is fairly simple because it does not include weighing of the different indicators. By including different weights for the different indicators, the score can be more accurate. With weighing of the indicators, the score can also reflect on how advanced a bot is, or classify which type of bot is being inspected.

Another thing that should be added to the solution in future work is language analysis. Some studies have shown that analyses of language and sentiment is a promising approach for detecting social bots [5][11]. By adding analyses of



the language in tweets, the solution would have an even broader set of analyses to indicate presence of social bots. The studies on language analyses also shows that this approach can detect social bots where behavior analysis do not.

## 7 Conclusion

This thesis have enlightened the topic of social bots and how to indicate their presence. As a practical part of the work in this thesis, a program for indicating social bots on Twitter were developed. The presented solution have both a technological and a biological aspect. The technological aspect is the indicator program. The program analyze an account to look for indications of social bot activity. The different indicators used in the program were inspired by several earlier studies on social bots. As a result of analyzing data available through the Twitter API, some of the indicators are also based on new ideas. The program uses analyses of several different features and behaviors. By implementing several analyses, the program should be theoretically able to detect social bots with different behaviors. Instead of having a detection tool, the program analyze an account to look for indications of social bot activity. If any indications are present, the analyzed account is to be further inspected. This is where the biological aspect comes in. Further analyses can be performed by either a dedicated person or by crowdsourcing. Crowdsourcing is a promising approach of analyses. It is effective and spreads the workload over several people. The biological aspect of the solution is included because humans can detect small inconsistencies that is hard to define in algorithms.

The indicator program works in near real-time. Right before analysis, account information and tweets from the account is downloaded. Right after account and tweet data is retrieved, the analyses are performed. The analyses are performed in about one to four seconds (url-analysis excluded). Having a fast-working indication program can be crucial for fast detection of social bots on Twitter. By having fast indication of social bots in place, the impact of the social bots cause is minimized.

Social bots have become very advanced in the recent years. And they continue to develop. Advancements in fields such as AI, makes detection of social bots more difficult. By continuing to develop methods for detecting them, we can be ready for the next types of social bots. To stay ahead of new social bots that come out, we also need to research new approaches for revealing their presence them.

## Bibliography

- [1] Facebook.com. 2017. Company info | facebook newsroom. Accessed on February 9, 2017. URL: <http://newsroom.fb.com/company-info/>.
- [2] Twitter.com. 2017. Company | about. Accessed on February 9, 2017. URL: <https://about.twitter.com/company>.
- [3] Turing, A. M. 1950. Computing machinery and intelligence. *Mind*, 59, 433–460.
- [4] Bessi, A. & Ferrara, E. 2016. Social bots distort the 2016 u.s. presidential election online discussion. *First Monday*, 21(11). URL: <http://journals.uic.edu/ojs/index.php/fm/article/view/7090>.
- [5] Dickerson, J. P., Kagan, V., & Subrahmanian, V. S. Aug 2014. Using sentiment to detect bots on twitter: Are humans more opinionated than bots? In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, 620–627. doi:10.1109/ASONAM.2014.6921650.
- [6] Dewangan, M. & Kaushal, R. 2016. SocialBot: Behavioral Analysis and Detection. In *Security in Computing and Communications*, Mueller, P., Thampi, S. M., Alam Bhuiyan, M. Z., Ko, R., Doss, R., & Alcaraz Calero, J. M., eds, volume 625 of *Communications in Computer and Information Science*, 450–460. Springer Singapore. URL: [http://dx.doi.org/10.1007/978-981-10-2738-3\\_39](http://dx.doi.org/10.1007/978-981-10-2738-3_39), doi:10.1007/978-981-10-2738-3\_39.
- [7] Zhang, C. & Paxson, V. 2011. Detecting and Analyzing Automated Activity on Twitter. In *Passive and Active Measurement*, Spring, N. & Riley, G., eds, volume 6579 of *Lecture Notes in Computer Science*, 102–111. Springer Berlin Heidelberg. URL: [http://dx.doi.org/10.1007/978-3-642-19260-9\\_11](http://dx.doi.org/10.1007/978-3-642-19260-9_11), doi:10.1007/978-3-642-19260-9\_11.
- [8] Wang, G., Mohanlal, M., Wilson, C., Wang, X., Metzger, M., Zheng, H., & Zhao, B. Y. December 2012. Social Turing Tests: Crowdsourcing Sybil Detection. URL: <http://arxiv.org/abs/1205.3856>, arXiv:1205.3856.
- [9] Wang, G., Konolige, T., Wilson, C., Wang, X., Zheng, H., & Zhao, B. Y. 2013. You are how you click: Clickstream analysis for sybil detection. In *Proceedings of the 22Nd USENIX Conference on Security, SEC'13*, 241–256, Berkeley,

- CA, USA. USENIX Association. URL: <http://dl.acm.org/citation.cfm?id=2534766.2534788>.
- [10] Igawa, R. A., Barbon, S., Paulo, K. C., Kido, G. S., Guido, R. C., Júnior, M. L., & Silva, I. N. March 2016. Account classification in online social networks with LBCA and wavelets. *Information Sciences*, 332, 72–83. URL: <http://dx.doi.org/10.1016/j.ins.2015.10.039>, doi:10.1016/j.ins.2015.10.039.
- [11] Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. June 2015. The Rise of Social Bots. URL: <http://arxiv.org/abs/1407.5225>, arXiv:1407.5225.
- [12] Davis, C. A., Varol, O., Ferrara, E., Flammini, A., & Menczer, F. 2016. Botnot: A system to evaluate social bots. In *Proceedings of the 25th International Conference Companion on World Wide Web, WWW '16 Companion*, 273–274, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee. URL: <https://doi.org/10.1145/2872518.2889302>, doi:10.1145/2872518.2889302.
- [13] Chu, Z., Gianvecchio, S., Wang, H., & Jajodia, S. 2010. Who is Tweeting on Twitter: Human, Bot, or Cyborg? In *Proceedings of the 26th Annual Computer Security Applications Conference, ACSAC '10*, 21–30, New York, NY, USA. ACM. URL: <http://dx.doi.org/10.1145/1920261.1920265>, doi:10.1145/1920261.1920265.
- [14] Stringhini, G., Kruegel, C., & Vigna, G. 2010. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference, ACSAC '10*, 1–9, New York, NY, USA. ACM. URL: <http://doi.acm.org/10.1145/1920261.1920263>, doi:10.1145/1920261.1920263.
- [15] Ruan, X., Wu, Z., Wang, H., & Jajodia, S. Jan 2016. Profiling online social behaviors for compromised account detection. *IEEE Transactions on Information Forensics and Security*, 11(1), 176–187. doi:10.1109/TIFS.2015.2482465.
- [16] Steinfield, C., Ellison, N. B., & Lampe, C. November 2008. Social capital, self-esteem, and use of online social network sites: A longitudinal analysis. *Journal of Applied Developmental Psychology*, 29(6), 434–445. URL: <http://dx.doi.org/10.1016/j.appdev.2008.07.002>, doi:10.1016/j.appdev.2008.07.002.
- [17] Instagram.com. 2017. About us - instagram. Accessed on February 13, 2017. URL: <https://www.instagram.com/about/us/>.

- [18] Boshmaf, Y., Muslukhov, I., Beznosov, K., & Ripeanu, M. 2012. Key challenges in defending against malicious socialbots. In *Proceedings of the 5th USENIX Conference on Large-Scale Exploits and Emergent Threats*, LEET'12, 12–12, Berkeley, CA, USA. USENIX Association. URL: <http://dl.acm.org/citation.cfm?id=2228340.2228358>.
- [19] Xiao, C., Freeman, D. M., & Hwa, T. 2015. Detecting clusters of fake accounts in online social networks. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security*, AISec '15, 91–101, New York, NY, USA. ACM. URL: <http://doi.acm.org/10.1145/2808769.2808779>, doi:10.1145/2808769.2808779.
- [20] Carpenter, R. 2017. Cleverbot.com - a clever bot - speak to an ai with some actual intelligence? Accessed on February 23, 2017. URL: <http://www.cleverbot.com/>.
- [21] Khanna, A., Pandey, B., Vashishta, K., Kalia, K., Pradeepkumar, B., & Das, T. 2015. A study of today's ai through chatbots and rediscovery of machine intelligence. *Int. J. ue Serv. Sci. Technol*, 8, 277–284.
- [22] Shen, Y., Yu, J., Dong, K., & Nan, K. 2014. Automatic Fake Followers Detection in Chinese Micro-blogging System. In *Advances in Knowledge Discovery and Data Mining*, Tseng, V., Ho, T., Zhou, Z.-H., Chen, A., & Kao, H.-Y., eds, volume 8444 of *Lecture Notes in Computer Science*, 596–607. Springer International Publishing. URL: [http://dx.doi.org/10.1007/978-3-319-06605-9\\_49](http://dx.doi.org/10.1007/978-3-319-06605-9_49), doi:10.1007/978-3-319-06605-9\49.
- [23] Freitas, C. A., Benevenuto, F., Ghosh, S., & Veloso, A. May 2014. Reverse Engineering Socialbot Infiltration Strategies in Twitter. URL: <http://arxiv.org/abs/1405.4927>, arXiv:1405.4927.
- [24] Bursztein, E., Martin, M., & Mitchell, J. 2011. Text-based CAPTCHA Strengths and Weaknesses. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS '11, 125–138, New York, NY, USA. ACM. URL: <http://dx.doi.org/10.1145/2046707.2046724>, doi:10.1145/2046707.2046724.
- [25] Sudani, W. A., Gill, A., Li, C., Wang, J., & Liu, F. 2010. Protection Through Multimedia CAPTCHAs. In *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '10, 63–68, New York, NY, USA. ACM. URL: <http://dx.doi.org/10.1145/1971519.1971533>, doi:10.1145/1971519.1971533.

- [26] Wagner, C., Mitter, S., Strohmaier, M., & Körner, C. 2012. When social bots attack: Modeling susceptibility of users in online social networks. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.221.6121>.
- [27] Cao, Q., Sirivianos, M., Yang, X., & Pregueiro, T. 2012. Aiding the Detection of Fake Accounts in Large Scale Social Online Services. In *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, NSDI'12, 15, Berkeley, CA, USA. USENIX Association. URL: <http://portal.acm.org/citation.cfm?id=2228319>.