Discouraging abusive behavior in privacy-preserving decentralized online social networks

Álvaro García Recuero

ANNÉE 2017

UNIVERSITÉ DE RENNES 1

UNIVERSITE BRETAGNE LOIRE

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Bretagne Loire

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Informatique

École doctorale Matisse

présentée par

Álvaro García Recuero

Préparée à l'unité de recherche INRIA Rennes – Bretagne Atlantique
Institut National de Recherche en Informatique et en Automatique
Université de Rennes 1, ISTIC

Discouraging Abusive Behavior in Privacy-Preserving Decentralized Online Social Networks

Thèse soutenue à Rennes
le 19e Mai 2017

devant le jury composé de :

Carmela TRONCOSO
Chargé de Recherche, IMDEA Software/ *Rapporteuse*

Bogdan CARBUNAR
Professeur à l'Université de Florida International / *Rapporteur*

Pierre ALAIN-FOUQUE
Professeur à l'Université de Rennes 1/ *Examinateur*

Christian GROTHOFF
Advanced Researcher, INRIA Rennes/ *directeur de thèse*

Caminante, no hay camino, se hace camino al andar – Antonio Machado

Acknowledgments

I thank Carmela TRONCOSO, Chargé de Recherche, IMDEA Software, and Bogdan CARBUNAR, Professeur à l'Université de Florida International, for having accepted to read and evaluate my thesis.

I am must also thank Pierre-Alain FOUQUE, Professeur à l'Université de Rennes 1, for having accepted to be in my thesis jury.

I am grateful to my supervisor Dr. Christian Grothoff, for his patience and support overseeing this thesis work until completion. I thank him for advising and directing me to the best of his knowledge until successful completion of this thesis. Special thanks to Cristina Onete and Jeff Burdges for directing us towards Privacy Set Intersection (PSI) protocols when in need of some literature review. I am also grateful to my colleagues at the TAMIS and DECENTRALISE teams at Inria during the past years, namely Florian Dold and Laurent Morin for giving feedback prior to the defense of this thesis, and Aurélien Palisse as well as Ronan Lashermes for their help in providing their "French touch" into the translated summary of the manuscript.

I am also grateful to Carlos Castillo for helping me to shape the last research steps of this thesis work with some remote mentoring in his spare time, which has taken me and the work into the right direction for publication. In this same context, the feedback of all anonymous reviewers has been a very useful insight as well.

In terms of funding, I thank INRIA for the generous support provided during the PhD studies and I acknowledge EIT Digital for their additional funding for traveling and conferences as part of their doctoral school programme.

Last but not least, thanks to my friends and family for supporting me at all times during this journey with their kindness and love.

Abstract

Today popular centralized Online Social Networks (OSN) such as Facebook or Twitter process massive amounts of information associated with user content in their platforms. Their approach creates several threats for privacy of users. Firstly, user data can leak to authorized (e.g., advertising) or unauthorized third parties (external entities). Secondly, centralized OSN are prone to censorship. On the other hand, future decentralized OSN (DOSN) remove the central authority for data management and so effectively such threats. However, they require careful routing to be resilient to failures, access control and data storage at the peer level.

This thesis investigates privacy-preserving protocols that aid in detecting and discouraging abusive behavior in future DOSN. In such settings less metadata is available to participants for analysis. However, to detect abuse we may need to make use of metadata that represents neighborhood knowledge, namely a social graph or network size/structure. Thus we need to provide privacy-preserving protocols that protect such metadata and are compatible with decentralized settings.

We first analyze abusive behavior in Twitter, an existing centralized, subscription based OSN platform. The data model of Twitter is a publish-subscribe messaging infrastructure that allows participants publishing and subscribing to various types of notifications (e.g., news, sports). At the individual level, we conjecture that attackers may be more likely to abuse potential victims if they can address them via the OSN interface with tools as mentions (e.g., tagging) in Twitter. To verify abuse at the individual level, we start collecting messages directed to potential victims using a data mining framework we build and that programmatically crawls Twitter APIs. We managed to retrieve in the order or hundreds of thousands of tweets and metadata of millions of social relationships from Twitter. Then, we extract a number of features, namely individual measurable properties related to the dataset. To obtain abuse ground truth, we develop a light-weight web platform that provides effective and customizable crowdsourcing of label annotation for a sample of our dataset. The sample contains messages directed towards potential victims of abuse and we ask humans to label the nature of messages following a set of abuse guidelines that provide a non-binary classification choice (undecided, abusive, acceptable).

Initially we consider the problem of abuse classification in a centralized OSN (Twitter). Next, we abstract from the Twitter model and consider DOSN where locally, users only have access to a partial view of the metadata available in the network to perform abuse detection. In turn, we analyze the impact of enforcing privacy into features involving neighborhood knowledge, which require collection and computation of

social graph metadata that is not available to the user locally. In order to use these features in DOSN, we design a signed Private Set Intersection (PSI) protocol that protects participant metadata collected and computed by the PSI. In addition, we analyze the resistance of our protocol against adversaries trying to tamper with the value of the PSI features. Finally, we perform data minimization by approximating PSI features and testing supervised learning algorithms for abuse detection. Our results show that approximation of the neighborhood fingerprint that PSI features use for abuse detection is still useful and compatible with DOSN.

Contents

List of Figures

1 Introduction

Contents

1.1 Motivation

Online social networks (OSN) have become one the main vehicles for online communications in the 21st century, with millions of users that participate in these platforms [102]. According to Facebook, over a billion users logged on to the social media site each month in 2013, that is up 23 percent from 2012. Due to this popularity, modern OSN are a very appealing target for abuse and mass-surveillance over citizens, which is a real threat to modern liberal societies [73]. To the best of our knowledge, modern OSN currently face two systemic issues to their well-being: abuse and mass-surveillance. The thesis research presented in this manuscript covers the former only, even though mass-surveillance is arguably just a more advanced kind of abuse.

Abusive behavior is today a prominent issue for OSN platforms such as Twitter[1], who often try but struggle to mitigate the problem[2]. Abusive behavior includes but it is not limited to spreading propaganda aimed at increasing the online presence of a terrorist organisation [105] and their recruitment [60], hoaxes that deceive authorities and citizens about a fake chemical crisis [42] or more generally astroturfing activities [111] that influence public opinion or even worse, enable online abuse of selectively chosen targets [98].

In terms of abusive behavior, cyberbullying gives a definition [119, 91] that refers to harassing another person via any form of digital communication as of today (e.g., cell phones, Internet). Malicious participants in the OSN send messages designed to harm the self-esteem or image of the potential victim, which some types of "social phenomena" in OSN (e.g., social contagion) can exacerbate [86]. These harmful type of communication has been known to be a significant risk factor for suicide [96] and previous studies report it almost doubling in number of attempts [77]. The risk factor for contagion of emotions from participants throughout a OSN is more significant for the well-being of weaker parties involved [87].

Defining an Internet "troll" or cyber-troll is, according to [116] someone who is a member of an online community and posts abusive comments at worst or divisive information at best to create controversy. In this context, in [14] authors note that the most prone users to interact with "trolls" are often those who consume conspiracy news despite their false and satirical nature; namely users who are misinformed or choose not to consume social media from traditional or contrasted sources. This may be difficult to assess in OSN platforms as Twitter, which are often used as unofficial news outlets. Groups of malicious users can easily take advantage of this to create a wrong view on reality [106]. Synchronizing their activities in the OSN, they are able to deny, disrupt, degrade and deceive others [42]. To these ends, such practices can have a non-negligible impact in the manipulation of political elections or even fluctuation of stock markets [20].

To prevent some of the above abuse related problems, OSN platform providers self-impose limits on users' behavior in the network[3], but these mechanisms are very

[1] `https://www.eff.org/deeplinks/2015/02/twitter-harassment-what-can-do`
[2] `https://www.eff.org/deeplinks/2016/01/twitters-policy-reboot-good-bad-and-ugly`
[3] `https://twitter.com/rules`

subjective and even misused to enforce draconian penalties on users of the OSN[4]. It is thus no surprise that these same governments generate numerous information disclosure requests to organizations as Twitter[5]. Therefore, privacy and abuse are highly related to each other. Facebook for example, provides a number of privacy policies to users. Although, being Facebook a centralized OSN, those are insufficient to protect user's privacy from abuse if we adhere to the definition given in Article 12 of the Universal Declaration of Human Rights by the Assembly of United Nations, 1948. The article states the following: "No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honor and reputation. Everyone has the right to the protection of the law against such interference or attacks.". Recently, Cambridge Analytica profiled millions of citizens for targeting them in elections campaigns. They use Facebook data[6].

Nevertheless, privacy has always been an overlooked topic; ever since it was first discussed in the early 1980's [123]. More recently Google[7] was fined for a breach in data privacy laws in France; reportedly due to the insufficient or lack of information provided to users so that they know how their private browsing data is collected and used. Spain[8] has also fined the Internet giant, due to mishandling of user personal data. Similar companies such as Twitter, use this user-oriented meta-data for direct advertising thus providing third-parties valuable information use to target prospective new customers.

1.2 Problem

1.2.1 Discouraging abusive behavior

Centralized OSN platforms typically employ expert staff who search for abuse and privacy related issues in their platforms. Due to the subjective nature of abuse, using human feedback alone to classify abuse is very costly as searching and filtering is often too slow at large scales. Therefore, OSN providers rely on intelligent systems built to support the automation of this task. At Facebook, the Facebook immune system (FIS) [115] automatically detects and acts upon suspicious information based on logged user behavior into the social networking platform. Twitter presumably also has its own abuse detection algorithm, but public technical details are mainly related to the infrastructure and software stack they use [94].

In general, in a centralized OSN the provider is in charge of protecting user content, metadata (e.g., size or structure) and communication patterns of the network. The last two are of extreme importance, as they provide information about social relations of a user into the network graph to potential adversaries. Even though, malicious

[4] `http://www.bbc.com/news/technology-16810312`

[5] `https://transparency.twitter.com/information-requests/2015/jul-dec`

[6] `https://t.co/vWSEdOJ0sz`

[7] `http://www.reuters.com/article/2014/01/08/us-france-google-fine-idUSBREA0719U20140108`

[8] `http://www.thelocal.es/20131219/spain-fines-google-900000-for-privacy-breaches`

users can still befriend their victims using fake accounts to bypass controls of the centralized setting. Users in these platforms can also restrict communications coming from malicious participants in the network by choosing to accept communications only from a closed network of friends in the platform. In deployments as Twitter, public messaging is however the default modus operandi.

Secondly, these OSN were built with great emphasis into owning subscriber's data in order to perform aggressive marketing and advertising[9]. Apart from having a business model that is arguably ethical, these platforms were not conceived with privacy in mind. Therefore, they violate any reasonable privacy expectations of citizens and have become critical keystone in the military-industrial espionage complex [114].

An alternative to these centralized designs are DOSN, which remove the central authority to distributed the task of storage, access control and routing to peers in the network. This approach protects users against censorship and network failures, but becomes way harder to protect the privacy of user metadata and communication patterns even if using encryption.

1.2.2 Privacy in the age of Big Data

Nowadays, with a growing concern regarding maintaining users privacy during data collection and processing, some governments are taking the issue of privacy into serious consideration. The European Union regulator has drawn a new law that will come into effect soon and which will consider the protection of personal, sensitive metadata. In particular the Data Protection Act states that "Personal data should be adequate, relevant and not excessive in relation to the purpose or purposes for which they are processed" [121]. While the definition of adequate, relevant and not excessive remains a more abstract issue, such laws are pushing towards a future of serious data minimization in Big Data. Therefore, companies and users' interests will be inevitable related to each other and such laws will play a fundamental role in application of policies that ensure fair use of citizen data.

In the context of the social web, removing the central authority existing in centralized architectures can reduce censorship and leakage of user content, which allows participants to attain higher privacy levels. However, privacy-preservation has been highlighted as one of the main challenges in moving from centralized to decentralized OSN architectures that remove the need for such a centralized authority [33]. This is due to the difficulty to protect all metadata and communication patterns in the network. It is known, that using just encryption by itself is not enough, as we have passive adversaries that eavesdrop and utilize traffic analysis tools to perform user re-identification through the data mining and analysis of statistical properties of a user/network fingerprint.

In the context of Twitter, in [53] authors assessed how to hide away user sensitive data from a central server, but did not analyze if that leads to an increase or decrease of abuse in the platform (perhaps due to not considering the vision of future decentralized

[9] `http://www.theguardian.com/technology/2015/mar/18/twitter-puts-trillions-tweets-for-sale-data-miners`

OSN designs). Naturally, the level of impact over abuse that increasing user's privacy bring into these systems is not yet clearly assessed or quantified.

Others have investigated fully decentralized protocols that maintain the anonymity of dynamic communities of users without explicit, declared friendship relationships [13]. However, in such scenarios it is important to note if higher levels of privacy or anonymity result into more abusive behavior or unwanted communications among users. Removing the accountability of user's actions in online social networking applications has been proven to increase abusive communications. The platform "Secret" ceased its activity due to issues related to managing cyberbullying [10]. Other semi-decentralized platforms such as Diaspora are known for having been abused by terrorists or malicious groups in the past [11], thus raising concerns on the ethical implications of providing such services to the public without any limitation.

1.3 Contributions

The main research goal of the thesis is to investigate the pressing issue of abuse and find ways to privately detect it in future DOSN that have as paradigm decentralization of the network infrastructure.

To take advantage of all metadata available in a DOSN platform and use it for abuse detection, we investigate how to privately compute features based on social graph metadata that represent neighborhood knowledge about the network size or structure. Considering that in future OSN deployments some metadata may not be available or must be kept private from the prying eyes of other participants, our privacy-preserving protocol is practical and we show evidence of reasonably good abuse classification while respecting privacy requirements. Finally, we perform a data minimization that shows our method is useful and even more efficient to privately compute neighborhood knowledge in DOSN requiring an amortized, low computational cost. Thus, peers can choose to detect abuse using a set of local features available in their proximity or use neighborhood knowledge when required to improve results of abuse detection.

1.3.1 Detecting abuse with machine learning

We evaluate detection of abuse in a centralized OSN deployment, Twitter. To avoid both the censorship issue as well as the manual labor required to investigate incidents, we propose a design that allows potential victims to deploy abusive behavior classifiers locally. The classifiers can then take appropriate action, such as giving messages that are likely to be abusive a lower relevance in a user's ranked timeline. This local approach also has the advantage that it is compatible with decentralized OSN that lack a central service provider. Local computation implies that the classification has to be performed without neighborhood knowledge. We will show that local knowledge together with some privacy-preserving neighborhood knowledge is sufficient for providing a reasonably

[10] `http://www.cnet.com/how-to/anonymous-social-networking-apps-that-are-not-secret/`
[11] `http://www.bbc.com/news/world-middle-east-28843350`

classifications of abuse, which we expect to increase with access to a larger ground truth too. Given that our dataset is limited to Twitter, it is still difficult to generalize the conclusions to other OSN.

1.3.2 Privacy-preserving abuse detection

Abuse today represents a serious concern to users and administrators of centralized OSN as Twitter. However, privacy-preserving abuse detection is beyond what industry considers when looking at the problem. In this thesis we try to realize if providing more privacy to potential perpetrators ("trolls") and potential victims of abuse, supervised machine learning methods can obtain reasonable utility of classifications. That is, the more privacy is offered, how challenging is to provide meaningful results for abuse detection? The rationale for that is that for instance, enhancing privacy may restrict search capabilities in a social graph of a centralized OSN, while in decentralized architectures this is already a constraint. Thus we implicitly study how to make abuse detection work in decentralized settings too.

1.4 Roadmap

The thesis starts surveying the state-of-the-art in abuse detection in centralized OSN. Secondly, collecting data for a use case of abuse detection in a centralized OSN, namely Twitter. Thirdly, it develops a number of features to train supervised learning algorithms in a binary classification problem and evaluate results according to our ground truth.

The hypothesis is to find utility in social-graph based metadata features for detecting abusive behavior while respecting the privacy of users in future decentralized versions of modern OSN as Twitter. Assessing this trade-off will lead us to quantifying how much metadata needs to be disclosed to reach reasonable levels of abuse detection while respecting privacy of participants. The goal is to perform a binary classification of abusive behavior in these online platforms without having to leak sensitive or private metadata to machine learning classifiers. In the long term, we envision these insights useful for developing decentralized OSN (DOSN) that carry over the existing culture of timeline construction of Twitter but provide an underlying network architecture that is less prone to be abused. Naturally, this will also have the advantage that it will preserve users metadata privacy. We address the above challenges in the following manner,

- In chapter 2 we survey the state-of-the-art in abuse detection systems for centralized OSN and expose the effects of online abusive behavior in OSN platforms and microblogging services. We review literature that proposes using different techniques for abuse detection in such systems, notably some of which are deployed in practice by OSN providers themselves (e.g., Facebook Inmune System, CopyCatch, SynchoTrap). An important aspect to these systems is that OSN providers as Facebook have full access to metadata about the network graph of participants, which in turn may be used for effective abuse detection. Unlike in

centralized OSN, we have found no clear evidence in the literature about addressing the issue of abuse detection in DOSN.

- In chapter 3, "Abuse model in Twitter", we present the abuse guidelines and rules that aid us in governing our research methodology. We use a set of guidelines following sociological and psychological studies. Using a modern OSN as Twitter, we then model abuse into messaging and social graph. We then proceed to collect such graph metadata, for which we develop a programatic framework to crawl metadata in the proximity of potential victims. Having such dataset, we build TrollSlayer, a web-based interface to run a Human Annotation Task (HAT) that consider our guidelines for abuse. We store human inputs into a ground truth database that also provides characterization of data patterns for abusive content.

- In chapter 4, "Abuse detection in modern Online Social Networks", we present a data model and mining framework we develop to use the Twitter OSN as baseline. First we experiment with a set of supervised machine learning classifiers for abuse detection that leverage features involving local and neighborhood knowledge. We show we are able to detect abuse reasonably well and discuss the shortcomings inherent to human annotation in our ground truth. We make the necessary adjustments into our classifiers to resolve the class imbalance in the number of samples among the minority class (abusive) and majority (acceptable) in our ground-truth. All supervised machine learning techniques used in this chapter rely on the scikit-learn framework [12].

- In chapter 5, "Abuse detection in decentralized Online Social Networks", we consider results of abuse classifications obtained from Chapter 4 and analyze each of the input features for classification in terms of privacy and resistance adversaries in the system. This approach enables us to asses the feasibility of our abuse detection methodology in future decentralized settings while making it compatible with the culture of timeline construction in existing centralized OSN as Twitter or Facebook. We evaluate abuse detection results using a subset of features that remain safe for use with our classification framework in decentralized settings. Surprisingly, results reveal that local knowledge of unforgeable account attributes, together with a small subset of similarity features based in neighborhood metadata can be useful for abuse detection in decentralized, privacy-preserving settings. Chapter 6 will introduce the technical details of our privacy-preserving protocol.

- In chapter 6, we present our privacy-preserving protocol for abuse detection in DOSN. Our protocol differs from a previous PSI protocol (PSI-CA) in that ours does not need to rely on a Certificate Authority (CA). This is because our approach relies on BLS (Boneh-Lynn-Shacham) signatures for public key management, which is compatible with a DOSN model. In terms of resistance to adversaries, we use standard cut-and-choose notation to protect sensitive metadata

[12] `http://scikit-learn.org/`

which provides neighborhood knowledge about the neighborhood among two parties, sender and receiver.

- In chapter 7 we present a data minimization approach to obtain a fingerprint of the features involving neighborhood knowledge, namely neighborhood information. To offload computation of such fingerprint or representation to peers in the network we employ MinHashes[13]. This allows offline computation at the peer level and also reduces the size of the messages exchanged among peers using our privacy-preserving protocol. Besides, this method is compatible with our privacy-preserving protocol, and compatible with DOSN privacy requirements when using neighborhood knowledge for abuse detection. Useful applications include privacy-preserving collaborative filtering in DOSN that rely on neighborhood knowledge and the use of some sort of ranking or learning algorithm.

- Results show that by using only a limited number of features for classification of abuse, it is possible to obtain meaningful detection rates. Our method improves user's privacy at a fraction of the cost of misclassifications. In addition, approximating some of the features involving social graph metadata can further minimize the amount of sensitive information collected for abuse detection algorithms.

Publications The above roadmap includes answers to important questions related to designing an abuse detection system while respecting privacy of a user's social graph in OSN. We develop a Private Set Intersection protocol in chapter 6 that only yields cardinality of neighborhood knowledge features involving, namely those involving social graph metadata. We acknowledge that the cost of data collection and processing through the Twitter API would be too high for a user to apply neighborhood knowledge using our abuse detection framework in chapter 5. However, in DOSN this can be somehow mitigated with the use of gossip routing algorithms that disseminate information in a best-effort basis in the network. This approach is compatible with out protocol in chapter 6 and can also benefit from the optimizations we present in chapter 7 for further efficiency.

The following contributions have been or are being published as main author at international venues as result of this PhD thesis:

- Efficient Privacy-preserving Adversarial Learning in Decentralized Online Social Networks, preprint [Online].

- Privacy-Preserving Abuse Detection in Future Decentralised Online Social Networks, in the 11th ESORICS International Workshop on Data Privacy Management International Workshop, DPM 2016 (`http://dpm2016.di.unimi.it`), Heraklion, Greece, acceptance of 9 full papers and 5 short papers out of 24 submissions.

- Discouraging Abusive Behavior in Privacy-Preserving Online Social Networking Applications, In 25th International World Wide Web Conference, April 2016,

[13] `https://en.wikipedia.org/wiki/MinHash`

Montréal, Canada. WWW Companion Volume (acceptance to the PhD Symposium was highly competitive, 7 out of 16 submissions were accepted).

2
Related work

Contents

2.1 Abuse detection in centralized OSN

Today, centralized Online Social Networks (OSN) have become a prominent and important tool that democratizes communication in the WWW. The influence of OSN reaches over to millions of users, who sign into these platforms to connect and communicate with their friends and family but also consume social media content in the form of news [90]. Secondly, thanks to their free-speech nature, OSN platforms provide users with the ability to debate about any topic of interest, including discussions about controversial matters as politics, climate change, human rights, etc.

The business model of large centralized OSN providers is to monetize user generated content through third-party advertising, marketing campaigns and similar activities [2]. However, this approach provides opportunities to abuse participants and centralized OSN infrastructures at a marginal cost. Perhaps, the root of the problem falls into the ethics of the business model underpinning these centralized OSN architecture.

Given that centralized OSN architectures are prone to abusive behavior in exchange of offering a free infrastructure to participants, such large OSN providers (e.g., Facebook, Google) need to resort to artificial intelligence (AI) algorithms that automate detection and removal of abuse in their platforms. Tuenti, a Spanish OSN platform, employs humans together with AI to analyze abuse related incidents in their platform [38]. This is because manual search and filtering alone is very costly and slow at large scales or simply because human intervention can not fully address the problem of abuse. In such cases, the OSN provider heavily invests in research and development of AI.

In 2010 Trend Micro had already reported a surge in the number of cyber-attacks targeting OSN [3]. Since then, this threat has been only growing in numbers and today has become a major security concern to firms and business operating such platforms. To tackle this threat, research works in Social Network Analysis and Mining (SNAM) investigate the use of techniques such as belief theory, trust propagation, statistical methods, machine learning algorithms and even linguistics to aid in the detection of abusive behavior, ranging from spam and harassment to sophisticated political campaigns.

Large OSN platform providers such as Facebook, Instagram, MySpace, AskFM or Twitter are often target of many of these attacks, and therefore a lot of the attention in recent research is devoted to these centralized platforms. In this context, some part of the research literature deals with malicious attackers that attempt to tamper quantitative metrics of the platform, namely number of likes in Facebook, followers, favorites or retweets in Twitter, etc. The goal of these attacks is to influence users of these OSN platforms. For detecting this type of abuse in platforms as Twitter, recent works propose to use traditional features and algorithms that are proven useful for spam detection in the WWW [46]. The goal is detecting spambots and social spammers that engage into the OSN platforms for fun and more importantly profit.

A second part of the literature is focused in the issue of abuse as a human-centered problem. In this context, we also find examples of spam detection, cyber-terrorism, and even vandalism detection though text mining approaches. Our research aims to

investigate the gap between these two parts of the literature and experiment with a new set of features that measure abusive behavior and protect the social graph from abuse in adversarial learning environments. Even if we employ classical machine learning methods that are not novel, their application does validate and extends previous research findings that show the difficulty for OSN providers in tackling abuse in adversarial environments [115] where it is not useful to rely on character-level features such as [104] does. Otherwise, attackers could easily obfuscate those to subvert detection and attack the social graph of the OSN.

2.1.1 Graph theory for abuse detection

To deal with abuse without considering the content of the communication, graph-based techniques are used to detect cyber-bullying [65], dishonest behavior [107], as well as fake accounts in OSN [38]. Although, this approach suffers from the fact that real-world social graphs do not always conform to the key assumptions made about the system. Thus, it is not easy to prevent attackers from infiltrating an OSN and deceiving others into befriending their fake accounts. These attackers are known as Sybils [57], which create an impression of being strongly connected to another portion of the network graph made up of legitimate user accounts 2.1. If they do succeed in doing so, graph-based defenses are rendered useless. However, yet in the context OSN, graph-based sybil defenses can benefit from machine learning techniques by extracting OSN user metadata into their feature set in order to predict potential victims of abuse [26].

In this same vein of work, [107] presents two graph-based ranking algorithms for the detection of generic dishonest behavior in online networks. Earlier, they show how to apply trust propagation for detection of trolls in an online social network with ranked nodes [108], namely the Slashdot [89] site. Approaching the problem of detecting abuse as a ranking problem for messages rather than users does also present censorship.

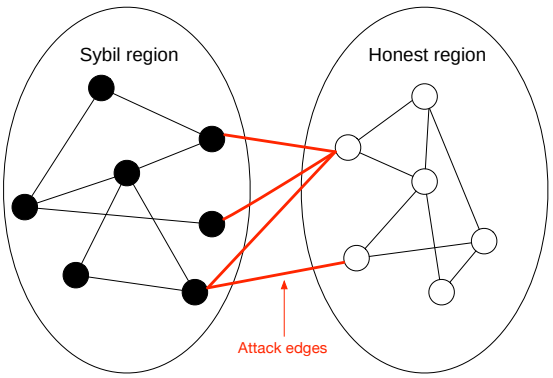


Figure 2.1: Social network layout, from[101]

Social graph-based sybil defenses Modern OSN are vulnerable attacks of such Sybils, where multiple accounts are used to target a set of victims or potential victims.

Sybil accounts create edges to their attack victims to perform any type of malicious activity targeted towards abusing participants and/or the OSN infrastructure. This includes but it is not limited to tampering with quantitative metrics in the OSN (e.g., number of likes in Twitter, Facebook, etc).

To the best of our knowledge, Facebook has been known to deploy abuse detection systems in their platform that have evolved to match the needs to detect these adaptive attackers. In CopyCatch and SynchoTrap [15, 39], analysis of activity patterns over time provides their system a good detection rate of malicious accounts that perform attacks in a loosely synchronized manner. These findings reveal the synchronized nature of abusive behavior in OSN, ranging from spamming to fake followers. In this context, human intervention would not easily detect abuse due to the amount and nature of the abusive behavior (e.g, deceive) of attackers in the OSN platform.

A large number of techniques have been proposed in order to detect identify Sybil regions in a OSN graph by assessing the mixing time of their social graph [127, 128, 49]. Mixing time of a social graph measures how fast a random walk reaches the stationary distribution (proportional to node degrees) or in other words, how well-enmeshed is a graph. While in many previous works authors claim that honest regions of the graph tend to have higher mixing that non-honest ones, according to [101], this intuition has been used without care when designing many sybil defense algorithms based in such mixing property. In the authors view, results such as [93] are merely circumstantial and do not reflect the property of the social graph correctly, either because they explicitly eliminated low-degree nodes (expected shorten mixing time) from the computation or because they use low values of the variation distance for the random walk. In this scenario, the problem is that presumably well-connected nodes would be traversed first (especially in a BFS approach as theirs) so that probabilistic average-mixing guarantees are satisfied while ignoring the slow-mixing portion of the graph; whereas perhaps this may not be the case when considering a worst-case scenario of the mixing.

In that vein, previous OSN literature focused in spam detection claimed that forming hundreds of relationships in a short period of time is considered as a key indicator of abuse [12]. This would be in line with the idea in which OSN that are based in acquaintances actually exhibit faster mixing than those that are not. Therefore being more difficult to make a design choice that remains useful for all types of graphs and OSN.

Such approach to detecting abuse in a OSN is thus closely related to hypothesis' drawn from graph theory and network analysis that assist in the characterization of a social graph used as input. In that context, it would be important and necessary to obtain the correct full portions of the OSN graph in order to perform unbiased graph traversal that keeps graph properties intact.

2.1.2 Supervised learning

In the context of supervised machine learning, many works have focused their attention in the problem of abuse. Researchers of a Spanish university in Bilbao resolved a case of cyber-bullying in a primary school employing supervised learning techniques. In

their work the assume a user behind a "troll" account must have a real profile in the same or another social network, namely being linked to the malicious one as to track interactions with the potential victims distantly [65]. In [126] authors also rely on supervised learning techniques to investigate abuse detection as a means of stopping online harmful communications in OSN. Thus the idea of using supervised machine learning techniques to tackle this problem is not new, but identifying a "non-troll" account that is linked to a malicious one, presumably created by the attacker to mask the interactions with potential victims and so avoiding any suspicion, can be more difficult to achieve.

There has also been extensive research in the area of spam detection in OSNS with supervised machine learning. The goal is to deter spammers in platforms as Twitter, who try to spread promoted campaigns, phishing or even malware through URL shortening. In [92], the authors investigate the effectiveness of honeypots when used together with machine learning to detect spam data in OSN as Twitter and MySpace. Gao et. al [66], performed a characterization of spam campaigns in Facebook, which study reveals that they are launched using compromised accounts to orchestrate such campaigns with messages that contain links malicious URLs, phishing sites, etc. This study concluded that 97 % of malicious accounts investigated were compromised, not fake ones. On top of that, the malicious accounts exhibited diurnal patterns for posting messages in the network when their target users are asleep so presumably increasing the chance for a malicious message being opened by a naive user when first logging into the OSN that morning.

2.1.3 Natural Language Processing

Related to text data mining and analysis, Natural Language Processing (NLP) is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages. While possible to use prior defined rules, nowadays most of these algorithms learn these rules from a large corpora of real-world examples (e.g., linguistics).

In [55], authors perform a binary classification of posts in the YouTube network by following common sense knowledge. They concluded that having a social network graph is only of use in social networking platforms if that yields a better model of the problem. They propose multi-class classifiers for text classification on individual sensitive topics. Natural Language Processing techniques are also used as a means to new computing paradigms that aim to detect how user emotions which lead to certain behaviors, namely Sentic-Computing [36].

In [104] the authors advance the state-of-the-art in abuse detection algorithms collecting comments of the Yahoo Finance and News portals during 14 months. Their dataset, which is presented as a major contribution, contains comments from a forum of Questions and Answers. For building a ground truth of abuse, authors also include in their dataset a set of comments flagged as "abusive" by the internal Yahoo raters or external ones using the platform (end-users). In addition, they have collected a second dataset by crowd-sourcing abuse annotations in Amazon Mechanical Turk. Results

show, the quality of human raters worsens significantly in the second dataset. This is presumably due to the fact that even the same humans may not classify in the same way a content when displayed to them several times. Depending of the context, and the community in which the conversation takes place (e.g., friends, family, colleagues, etc), this may be also different. Therefore authors state that for using supervised learning algorithms, choosing humans carefully for annotating abuse is an important factor that influences the rate of false negatives/positives. In fact, this comes as no surprise, as the usual limitation for training supervised machine learning models is always the minority class, that is the one willing to predict here, abuse.

2.1.4 Ad-hoc approaches

While the problem grows in importance, OSN providers but also users developed new solutions that aim to mitigate the problem. In the Twitter environment, a third-party browser-plugin [1] exists to allow a user to filter abuse by blocking followers directly from the OSN web interface. Both, blocking and suspension, have the disadvantage that they can be viewed as censorship if done at the platform level. Twitter has also enabled its users to report abuse and use a block button that effectively enable a potential victim to ignore abusive comments from abusive senders.

Many similar applications are emerging in order to empower individuals to report and/or block cyber-trolls to build a shared, global blacklist of abusers. For instance, GG auto blocker [2] and the site Block Together [3] both crowd source blacklists of accounts. Some common metrics used to identify abusive accounts are the creation date, number of followers and so forth. This approach come with the advantage of getting users to build a ground-truth of what a troll for free, but has the disadvantage of getting false positives and suffering of abuse itself. Even Twitter itself has introduced the concept of block lists in the OSN [4] as a means to thwart abuse in their platform. This includes functionality for users of the OSN so that they can import and export lists of blocked users. However, there are not reported numbers on the accuracy of these approaches, perhaps due to their subjective and ad-hoc nature.

2.1.5 Other approaches

Other approaches make use of statistical data mining techniques which inform domain services of such malicious activities [120], or focus in studying the personality of Internet trolls [34].

In [48], authors point out that trolling can also be detected though means of user and conversational data mining across not just one but several OSN. This approach relies on the fact that the user will react to abuse by posting about it into the same or different OSN. That implies having to collect data from more sources (usually publicly

[1] `https://chrome.google.com/webstore/detail/twitter-block-chain/dkkfampndkdnjffkleokegfnibnnjfah?hl=en`

[2] `http://blog.randi.io/good-game-auto-blocker/`

[3] `https://blocktogether.org`

[4] `http://tinyurl.com/oymn6pp`

available), therein failing to reach some of the goals in our second chapter of the thesis, minimization of data collected. Besides, a-posteriori analysis of the issue does not prevent trolls from attacking back to the user in question on the second OSN.

Belief function theory is applied to the problem of troll detection in [56]. Given that irrelevant data is also possible, this work goes beyond by stating that irrelevance of a message is a necessary but not sufficient condition for identifying someone as a troll.

Anti-fraud algorithms are an example of how to detect whether users or tweets in a data set are outliers, namely abusive. That is, different from most of the other data. There are existing techniques for machine learning, such as supervised learning, which take that approach. Supervised learning introduces the idea of having a dataset available where records are known and accordingly marked as positive or negative in advance.

2.2 Privacy-preservation in OSN

Recent incidents affecting the privacy of user data and its safety in Online Social Networking applications does again highlight the dangers of entrusting the management of personal data to these platforms. In 2016, Troy Hunt uploaded millions of compromised accounts from the LinkedIn professional Online Social Network to a web site that allows potential victims to check whether the password of their account is still exposed. Considering many users have a same login (email and password) for several online services: email, OSN, Flickr, etc, according to Wired it would be plausible to invest few dollars in buying the LinkedIn dataset in an online black market to access many more services than just the professional OSN[5].

Modern OSN platforms often fail to address the security threats that they face. Even with recent efforts that aim to enforce extra controls on users at the time of logging, namely two-way authentication mechanisms, participants can not be certain that their data will be safe and not compromised at all. Such mechanism is often not enforced by OSN, as not every user is tech-savvy or technical enough to configure these mechanisms on their account/profile. This suggests that this solution is as shallow as deep are the security and privacy problems on these platforms, many leading to the result of thousands of stolen credentials. Designing a systems as a remote service that acts as a single, centralized point of control to manage user data is a tremendous security risk for participants, as well as providers. Users often store a lifetime of media posts and digital communications on such platforms, so the privacy of their data is tighten to the ability of the platform provider to ensure only legitimate access to their personal information. This conflicts with the marketing and advertising interests platform providers have on such user generated content.

Privacy-preserving models based in generic access control policies as those from OSN as Facebook [61], propose to take advantage of such mechanisms to enforce control of privacy when accessing historical communications as well as a social graph topology of contacts. In a similar study in that vein, a middleware-based solution called

[5] `http://www.wired.co.uk/article/linkedin-data-breach-find-out-included`

OpenSocial [100] proposes to enable individuals and organizations to define custom access policies that inter-operate with several OSN and thus provides users with a single pane-of-glass to access and control their data.

However, to solve the problem of privacy in OSN is necessary to rethink all the foundations of these services: architecture, access control, data storage, network protocols, etc. While preserving the privacy of users is important, ensuring the culture of existing OSN carry over can be a determinant factor in the adoption of future decentralized solutions [62]. On the other hand, cryptographic protocols coupled together with decentralized architectures can provide a reasonable alternative to modern OSN architecture as in the case the case of Twister[6], a fully decentralized peer-to-peer microblogging site that looks very similar to Twitter but uses a substantially different architecture and design principles behind the scenes (end-to-end encryption). Diaspora is another system that implements a semi-decentralized or federated OSN where users can create pods where participants join the network.

2.2.1 Decentralization and privacy

While decentralisation may seem to make privacy inherently more robust against an strong adversary, it is also harder to implement correctly if we need to hide all object metadata and information flow, even with encryption [72]. Perhaps for that reason, a number of works have either focused in making computation among distrustful parties privacy-preserving, obfuscating metadata as a means to obtain better level sof privacy, or use cryptographic primitives that hide metadata and secure non-mediated communications.

In the landscape of decentralized, peer-to-peer, secure private networks, projects such as GNUnet focus on the security aspects of networked systems through decentralization. Such system provides no centralized or trusted service. Therefore, offers a GNU Name System as decentralized trust management infrastructure. Systems such as Secushare[7] for instance, rely on GNUnet for peer-to-peer routing and encryption.

Another peer-to-peer (P2P) distributed design is PeerSon. Here, a fully decentralized architecture supports the removal of a central authority server to offer better privacy to users [33].

PeerSon, as well as Secushare, have not yet considered the impact of increasing users' privacy during online social interactions in their OSN deployments. Many advocate for the study of social network theory to aid in the development of mature framework that can aid OSN developers to quantify the trade-off of privacy and safety during online interactions [129].

2.2.2 Graph obfuscation and anonymity

In other cases, to adhere to the goal of preserving users privacy, research literature as [130] and [75] propose to modify edges of a graph. This is done by obfuscating the

[6] `http://twister.net.co/`
[7] `http://secushare.org`

graph via perturbation of its intrinsic properties [43], namely adding noise into the graph with random addition/deletion/switching of edges [19] [21], or grouping of sets of vertexes into a super-vertex [37].

Likewise, in [53] DeCristofaro proposed to hide away user sensitive data from a central server by employing cryptographic techniques such as homomorphic encryption, but did not explore if the anonymity such system provides has any impact over the online interactions among users in the OSN platform.

In AnonyLikes [6] authors present a cryptographic protocol that allows users to provide quantitative feedback in a platform such as Likes Facebook anonymously, namely without disclosing their identity in/to the OSN. All that without reducing or modifying the functionality of the OSN, including but not limited to adhering to existing authentication mechanisms and publicly displaying such Likes while honoring requirements of no duplicate feedback actions on a post. Behind the scenes, AnonyLikes employs homomorphic encryption, so that operations can be performed directly on cipher-text (sum of likes here). To decrypt the final value of the computation, a number of parties involved in the protocol (trustees) must collaborate as the server alone is not able to so by protocol design. Note their protocol allows to obtain only the number of Likes of a given post, not who are the authors of those. This is possible thanks to the properties of a variant of "El Gamal" cryptosystem that satisfies additive operations [45].

2.2.3 Secure multiparty computation

The general solution to the problem of cooperative computations over private inputs is the use of secure multiparty computation [125]. With secure multiparty computation, a function $f(i_k)$ can be computed over a set of private inputs i_k without disclosing anything but the result $f(i_k)$ (and whatever can be derived from $f(i_k)$ and the input of the respective party). The result will be correct assuming an honest-but-curious adversary model, which is applicable in cooperative negotiation systems.

In general, implementing a secure multiparty computation is possible if complex for any finite circuit. The Fairplay compiler [10] can be used to automate the task of constructing a secure multiparty protocol and implementation. Given a function f, Fairplay can generate the code the different parties would need to execute. Fairplay is a good solution for secure multiparty computations for simple functions f. For complex algorithms, it can be difficult to construct the required circuit and the complexity of the resulting protocol and computations might be too high. For example, executing a single 32-bit addition using the Fairplay system can take about one second.

An alternative to automatically generating secure multiparty computation protocols is to design a specific, efficient protocol for a particular problem; this has been done in many areas where the privacy of the participants' inputs is of importance, such as distributed data mining [44], collaborative intrusion detection [35], collaborative forecasting and benchmarking [7], and secure large-scale auctions [18].

Private Set Intersection and Private Set Union A privacy-preserving approach to the problem of sharing information amont two distrustful parties can be approached

as a secure multiparty problem. Such techniques are applicable technique in many real life situations, where ones does not want to disclose information about each participant yet needs to compute a result using collective inputs from participants. Practical situations where this type of protocol can be used is for instance paper submissions systems of conferences, where program chairs may want to check whether a list of paper submissions contains any duplicate submission to any other conference. Naturally this would required some collaboration from the program chairs or organizers among conferences. However, if none of them is willing to reveal authors' identify of received submissions, then a solution is to use a Private Set Intersection (PSI) protocol. PSI is a solution that is informative enough to identify just that a duplicate paper exists without having prior knowledge of its authors. These similarity metrics often measure the Jaccard index [8] among two sets, which can range from 0 to 1, namely no similarity at all or equally similar.

In this context, previous work in PSI [17] has proposed approaches that measure similarity among two sets in a privacy-preserving way. This is done providing two protocols for the privacy-preserving evaluation of sample set similarity. Their work is aimed at computing a privacy-preserving Jaccard index among sets. That statistic is used in comparisons of similarity among sets, which measures such similarity as the size of the intersection divided by the size of the union of the sets [80].

[8] `https://en.wikipedia.org/wiki/Jaccard_index`

3
Abuse in Twitter

Contents

Modeling and tackling the problem of abuse in a centralized OSN such as Twitter requires a clear definition and guidelines for abusive behavior, a judicious methodology to collect potential cases of abuse, as well as human involvement in order to verify them. To this end, first we adapt a set of guidelines that nicely cover and map existing types of online abuse in Twitter to "Behavioral sciences". Such behavioral techniques are used in practice by organizations that abuse in various ways, namely deceiving, disrupting, denying or degrading others in cyber space. Their goals include but are not limited to influencing foreign policy of countries, public opinion or even elections. Secondly, for data collection we develop a victim-centric approach that obtains a Twitter dataset containing metadata of messages exchanged in the network as well as their respective social relationships. Lastly, we obtain abuse ground truth using a crowdsourcing methodology that follows the JTRIG's definition of abuse in 3.2.3. Workers are presented with visual examples of abuse in order to proceed according to our guidelines in the rating of abusive behavior. Using the same set of messages to annotate, we perform the same annotation task into two different platforms. We employ a cost-free crowdsourcing platform we build, namely Trollslayer; unlike Crowdflower, a widely used commercial platform in the research community of SNAM.

We calculate overall and inter-rater descriptive statistics about agreement, showing evidence of higher disagreement in case of abuse, regardless of the crowdsourcing platform being used. We summarize our results and conjecture about the reasons behind such disagreement.

3.1 The problem with abuse

Abusive behavior is today prevalent in centralized OSNs as Twitter and it is rooted into motivations often more complex than a simple form of contemporary spam. It also has deeper implications to our society in the form of human interactions. Centralized OSNs suffer from two systemic external security issues, namely censorship and mass-surveillance, where governments or intelligence agencies respectively block or collect user generated content according to interests of military complex.

In project PRISM[1], the National Security Agency (NSA) in the United States of America performs bulk collection of vast amounts user generated content (e.g., chat, email, voice) irrespectively of citizen consent. This is possible thanks to the Patriot Act and its section 215 for which Jim Sensenbrenner – author of the Patriot Act – even admitted that "no fair reading of the text would allow for this program". Reportedly, for 34 years courts have used this section to approve over 35,000 applications, while only 12 were rejected[2]. This includes applications to access major databases of user content acquired and hosted at WWW giants as Google, Facebook, etc. In terms of censorship, enforcing strict abuse guidelines in an OSN platform[3] can be as abusive as breaking those rules. We keep seing cases of authoritarian governments that impose draconian penalties on citizens that freely express themselves in OSN [4]. This is possible due to the centralized nature of popular OSN, which indeed do need to enforce anti-abuse guidelines for safeguarding users's privacy and safety. However, in practice these guidelines do limit freedom of speech for participants and yet do not discourage abusive behavior completely from the OSN.

3.1.1 Incidents in Twitter

In the Twitter environment, users who have been victim of abusive behavior face a practical dilemma in order to defend themselves: i) closing their account (e.g., Robin Williams' daughter, Zelda Williams closed her account after being repeatedly abused on Twitter[4]), ii) reaching to OSN administrators to demand a prompt response that suspends accounts of abusive users (Leslie Jones[5]).

It is worth to note abusing in centralized OSN deployments is far too easy compared to other systems. So much that the pressing nature of the issue has urged service providers as Twitter to update their site usage policy several times lately [97]. The Twitter guidelines as of today, aim to discourage users from abusing in the platform, but victims as the feminist account "@justkelly_ok" are an example of someone who kept being abused, namely with rape threat messages for a consistent amount of time, due to the lack of automated tools that ban abuse from the platform[6]. These threats are also

[1] `http://en.wikipedia.org/wiki/PRISM_(surveillance_program)`

[2] `https://goo.gl/NJlICD`

[3] `https://blog.twitter.com/2015/update-on-user-safety-features`

[4] `https://www.washingtonpost.com/news/the-intersect/wp/2014/08/13/robin-williamss-daughter-zelda-driven-off-twitter-by-vicious-trolls/`

[5] `https://goo.gl/7fxyGw`

[6] `http://www.bbc.co.uk/newsbeat/article/37244089/twitter-removes-rape-tweet-and-bans-`

possible because of the nature of Twitter, which unlike Facebook, allows participants to publicly address anyone in the OSN with a personal identifier or "handle". The concept of public one-to-one messaging does not exist in Facebook but in Twitter one can mention the "handle" of any other user, thus effectively mentioning them while reaching to a large public audience.

3.1.2 Abuse Countermeasures in Twitter

The frequency and quantity of incidents of abuse in Twitter has raised concerns in many ways about the effectiveness of anti-abuse policies and guidelines. Since such guidelines ought not to be enough, at the individual level, many have decided to create their own, personalised filters or blocking lists. Twitter itself often suspends misbehaving users who do not follow their guidelines and policies, hence indirectly promoting a way of censorship that risks freedom of speech in the WWW. Therefore, the problem with abuse in Twitter shows a tension between freedom of speech and the limits of what should be considered as abusive behavior. Such tension highlights that current OSN platforms were not designed to discourage abuse.

In Twitter, until now, abusers could abuse their victims and then block them to avoid being suspended by Twitter. We will see that like in our final conclusion of the DPM paper [69] for improving upon centralized OSN timeline construction, Twitter is now proposing users a new "quality filter"[7] intended to hide or rank tweets from a person's mentions when the sender of these messages is not followed by the recipient, which also resembles features we used for quantitative measurement in the study of a complex network as Twitter[8]. In the meantime, OSN administrators still need to continuously verify user reports and suspend abusive accounts accordingly, even by being exposed to such abuse content themselves. Due to this difficulty in keeping abuse out of the platform, abusive comments should be automatically detected and dismissed and only in case of doubt reviewed by a human that is able to make a final judgement on whether to accept or refuse the content into the platform.

Nevertheless, abusive users often come back after some time by creating a fresh pseudonym or different identity in the platform. Note Twitter has announced that will tackle these adaptive adversaries by tracking them in the network in order to stop them from registering into the OSN service again.

Privacy concerns in Twitter To defend against privacy abuses as impersonation or leaking of personal information in the OSN network, Twitter like centralized settings typically employ simple Access Control Lists (ACL's) to allow users configure how much information a user is willing to disclose to the public. However, that alone does not prevent or discourages abuse; examples are Twitter or Facebook. Besides, with these ACL privacy settings defaulting to a less restrictive configuration in OSN

user-after-woman-goes-public-with-their-response

[7] http://www.wired.co.uk/article/twitter-tools-harassment

[8] https://en.wikipedia.org/wiki/Reciprocity_(network_science)

as Twitter [84], attackers can even infiltrate the platform as benign users and start misbehaving only when their victim or target is at reach.

Twitter is also enforcing stricter controls at the time of account registration in the platform, effectively tracking phone numbers of not only attackers but also victims to the best of our knowledge[9]. Such database is made up of network participant's phone numbers, potentially useful for tracking their real identity, in a presumably a desperate measure to enforce usage guidelines. However, there are privacy risks associated with holding such metadata about individuals, as having to comply with law enforcement and handling it over to authoritarian governments (not unlikely to happen looking at Twitter's data disclosure requests[10]).

Considering the above problems, a better solution may be designing and deploying decentralized, privacy-preserving online social networking applications that discourage abusive behavior by design. The technical challenges here are due to the decentralized nature of the solution, which require use of epidemic algorithms for message dissemination and balanced design choices for keeping privacy-utility of user identities in the network.

3.2 Defining abuse

In order to discuss abuse, we need a set of guidelines that govern our data methodology in the least intrusive way for users and the OSN provider itself. We restrict ourselves to the use case of Twitter, for which we find in the literature three definitions of abuse, mapping those of Twitter to "Behavioral Sciences" employed at the JTRIG group of GCHQ.

Comparing JTRIG guidelines to Twitter[11], becomes evident that following the former guidelines we are able to nicely map each of the Twitter guidelines to an existing definition for abusing according to the JTRIG.

3.2.1 Twitter guidelines

Twitter Terms & Conditions (TTC)[12] are in constant change nowadays, so they include but are not limited to the following points:

- Violent threats (direct or indirect): users may not make threats of violence or promote violence, including threatening or promoting terrorism. Note the similarity with Deny in JTRIG.
- Harassment: users may not incite or engage in the targeted abuse or harassment of others. Specifically, if the reported account is sending harassing messages to an account from multiple accounts. Note this is also listed in the above guidelines of

[9] `http://www.theverge.com/2015/2/26/8116645/twitter-improves-abuse-reporting-tools-phone-numbers`

[10] `https://transparency.twitter.com/en/information-requests.html`

[11] `https://twitter.com/rules`

[12] `https://dev.twitter.com/overview/terms/agreement-and-policy`

JTRIG under Disrupt, and later in our methodology we model the abuse problem similarly.

- Hateful conduct: serial accounts are forbidden for disruptive or abusive behavior. Again note the similarity with Disrupt.
- Private information: You may not publish or post other people's private and confidential information. Note this resembles the Degrade listing in JTRIG.
- Impersonation: it is not allowed to impersonate others through the Twitter service in a manner that is intended to or does mislead, confuse, or Deceive others.

Twitter also includes a characterizations of abuse in terms of type of spam as follows:

- Username squatting: creating accounts for third-parties or purposes different than personal use (selling). According to previous literature, this has investigated and there is a whole underground black market of fake Twitter account. selling [118].
- Malware and Phishing: users may not publish or link to malicious content intended to damage or disrupt another person's browser or computer or to compromise a person's privacy.
- Spam: this includes high churn of follows and subsequent unfollows to other users. An attacker may be willing to become more noticeable and gain more clicks, follows or attention in the network. Entering many links in tweets and not personal content is also considered spam here. Likewise of hashtags to reference unrelated or popularize trends/topics that fall in that same category. Lastly, unsolicited or duplicate "@"replies/mentions. The list if long, and there are many others. Including some of our assumptions on what is abuse.

We are not concerned in the differentiation of spam types in our abuse guidelines. It is worth to mention filters and third-party personalized plugins already address that issue at the browser level by blocking malicious accounts in order to offer a better user experience. In this thesis we do not target detection of abusive accounts but rather only messages so that developers can build systems less prone to censorship. However, we acknowledge resolving this type of abuse also remains of critical importance yet difficult to centralized platforms as Twitter.

3.2.2 Trolldor guidelines

Trolldor[13], is a platform that allows users to search for statistics of a particular user in Twitter; also allows users to report someone as a "troll". Key reasons Trolldor lists for users to report a Twitter profile as a "troll" include:

- Provocations: users who just provoke others for fun.

[13] `http://trolldor.com`

- Creep: users who fill other users timeline on a daily basis with messages worshiping their idols, friends, relatives and colleagues.
- Retweeter/favoriter: users who never create their own content and just retweet and favorite other people tweets.
- Insult/threat: users who insult or threaten other users.
- False identity: profiles that seek to usurp identity of others.

Looking at the top list of trolls in the ranking provided by Trolldor, we observe their approach lacks of real insight into abusive tweets. This is because users are annotated based on account metrics and using presumable victim's reports. This is dangerous, as creates a new form of abuse, on the reported incidents themselves. Such naive approach in turn renders non-abusive users as "trolls" in the ground truth. This entails some ethical concerns, but also shortcomings, as there seems to be no way to counteract the behavior of users who presumably use Trolldor to undermine the image of someone they dislike or do not agree. Besides, simply reporting someone as a "troll", makes this publicly available in the platform thus ensuring discredit of the victim.

3.2.3 Our definition of abuse

In order to consider a tweet as abuse or not, we abstract from the initial set of existing guidelines of a centralized microblogging site as Twitter and investigate the notion of abuse as defined in behavioral sciences of top secret agencies and groups, namely the Joint Threat Research and Intelligence Group (JTRIG) [98] of GCHQ. Incidentally, recent political events have been reported to rely on behavioral sciences which together Facebook users metadata collection can throw useful insights about influencing voters in political elections. A company called "Cambridge Analytica" managed to successfully profile and address voters of the election campaign in US and of the Brexit referendum in Britain. They reportedly collected the biggest and most valuable psychometrics database of Facebook users to date. In fact, chief Alexander Nix has claimed to have a massive database of 4-5,000 data points on every adult in America. These news are today front page of journals and political debate, as some claim the invasive psychological profiling of OSN users[14] leads to manipulation of election results in representative democracies[15]. Since then, Cambridge Analytica public position is to de-emphasize any use of psychological profiling tools.

We employ a set of guidelines given by the professional trolls at JTRIG, as we find it both simple yet very useful to map existing guidelines of abuse on Twitter to "Behavioral Sciences". Using their four (4 D's) to characterize abuse, we also cover any other existing definitions of abuse in OSN:

[14] `https://theintercept.com/2017/03/30/facebook-failed-to-protect-30-million-users-from-having-their-data-harvested-by-trump-campaign-affiliate/`

[15] `https://en.wikipedia.org/wiki/Representative_democracy`

- Deny: encouraging self-harm to others users, promoting violence (direct or indirect), terrorism or similar activities.
- Disrupt: distracting provocations, denial-of-service, flooding with messages, promote abuse.
- Degrade: disclosing personal and private data of others without their approval as to harm their public image/reputation.
- Deceive: supplanting a known user identity (impersonation) for influencing other users behavior and activities, including assuming false identities (but not pseudonyms).

3.3 OSN model

To discover communication patterns among participants in Twitter, we collect account, message (e.g., tweet), messaging graph (e.g., proportions of messages in time, direction, etc) and social graph metadata (e.g., followers, followees, reciprocal relationships) in Twitter following our method. The assumption here is that such information would be useful if multiple perpetrators are in the same social circle when coordinating their attacks (e.g., Figure 3.1), namely sock puppet accounts or Sybils [57] that leave structural evidence of artificial creation in the network. Figure 3.1 resembles a bipartite graph construction, where users that abuse are tightly connected among themselves and reach out to their potential victims with messages (which form edges) into a disconnected portion of the network graph.

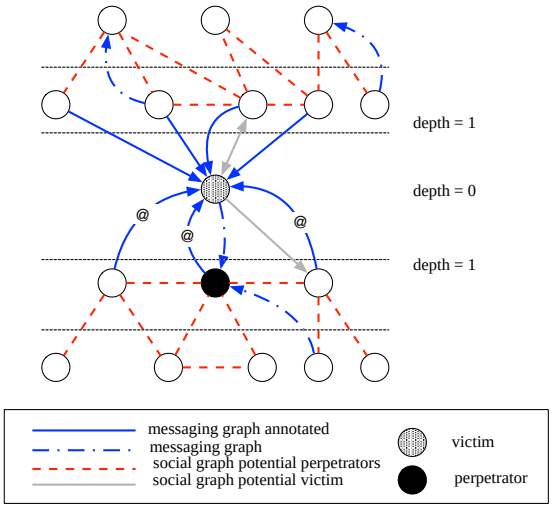


Figure 3.1: Social and messaging graphs between a potential victim and a potential perpetrator

For our Twitter use case, let us consider two directed graphs whose set of vertices are Twitter users. Let $\mathcal{G}_f$=($\mathcal{V}_f, \mathcal{E}_f$) be a directed graph of social relationships with a set

of vertices $\mathcal{V}_f$ that are follower users (those who follow or subscribe to another user's posts), and a set of directed edges $\mathcal{E}_f$ pointing from follower to followee users (those who receive such a follow or subscription request).

Secondly, let $\mathcal{G}_m$=($\mathcal{V}_m, \mathcal{E}_m$) be a directed messaging graph with a set of users as vertices $\mathcal{V}_m$, and a set of directed edges $\mathcal{E}_m$. These edges are created from tweets in two cases: First, they point from users authoring a tweet to users mentioned in the tweet (`@user`). Second, if a tweet is a reply, an edge is created so that it points from the responding user to the author of the original tweet. Thus, $\mathcal{E}_m$ models the tweets that are shown in a user's notifications and are thus a vector for abusive behavior. Users in set $\mathcal{V}_m$ may or may not be in the set of follower users $\mathcal{V}_f$.

For each user $u \in (\mathcal{V}_f \cap \mathcal{V}_m)$, we note the direction of its graph relationships. If it is a follower belonging to the edge set $\mathcal{E}_f$ of the social graph $\mathcal{G}_f$, then

$$\mathcal{E}_f \coloneqq \{(u, v) \mid u \text{ follows } v\}.$$

And if a tweet within the set of tweets $\mathcal{E}_m$ in the messaging graph $\mathcal{G}_m$, then

$$\mathcal{E}_m \coloneqq \{(u, v) \mid u \text{ mentions } v \vee u \text{ has reply to } v\}.$$

For each tweet $\chi_i \in \mathcal{E}_m$ in the messaging graph $\mathcal{G}_m$, our classifier will define a feature (f) based binary oracle function O_f, to predict whether the tweet in question (χ_i) is abusive or not. That is, whether it belongs to a set of acceptable tweets A, or the set of abusive tweets B. The classifier is not allowed to output "undecided", hence A$\cap$B $= \emptyset \wedge$ A$\cup$B $\in \mathcal{E}_m$.

3.3.1 Graph traversal

In order to realize the above OSN model, we develop a crawler which forms a database of tweets collecting data through Twitter public REST APIs[16]. To bootstrap our crawler, we start with a given seed set of users. Next we start a breath-first-search (BFS) traversal on the Twitter graph that it is able to select up to a given maximum depth we set as static threshold, namely maxdepth. In Algorithm 3.1 we summarize such operational mode. Note that we started with the hypothesis that a mastermind may be in control or coordinating potential perpetrators, however this is only useful if we were trying to detect a volume of users inflicting abuse to their victims. In reality, we end up only detecting abusive messages, but the methodology still holds for future efforts in the initial hypothesis.

Note we collect nodes in the neighborhood which are located beyond a threshold if, and only if, those are identified as sufficiently representative to the set of nodes collected at the maximum depth boundary. To identify them, we may look at the in-degree and out-degree properties of each node in such boundary, as our crawler supports both options and it is fully configurable. The idea is that a node beyond the threshold of maxdepth is crawled if there is a minimum number of edges or relationships among itself with the nodes up in the boundary that such threshold defines. For example,

[16] `https://dev.twitter.com/rest/public`

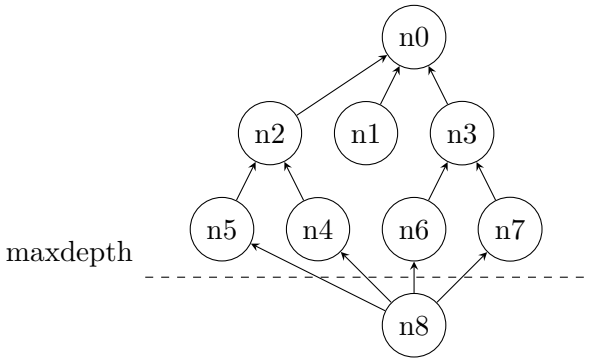


Figure 3.2: Example of crawling in the breadth-first traversal

in Figure 3.2 see node n8 which has an out-degree of four (non dashed-lines). Let's imagine that we had set the out-degree minimum threshold to three, then the condition is met. Likewise, in the configuration of the crawler, a minimum in-degree option can be set independently for also checking the number of edges from the nodes at maxdepth to the node beyond such boundary. This makes the crawl more interesting if we are looking for nodes that are tightly connected and that possibly relate to each other in a sense of community structure.

Boundaries of the data crawl We configure several parameters of our crawler to limit the depth of the graph traversal, and thus getting tweet, account and social graph metadata of each node found along the BFS path. At each depth, we repeat the same process of data collection but we do not go any further than depth 2 in the traversal; except when a user at depth 3 is highly popular among boundary (depth 2) nodes. This is done with an additional parameter passed to our crawler, in-degree and/or out-degree for boundary nodes.

3.3.2 Twitter dataset

In our data collection, the set of users that interact with potential victims in our seed set are potential perpetrators. To avoid capturing entirely unrelated data, initially we do not target data collection of users in the seed set that are verified accounts or accounts with high number of followers, as this would result in an excessively large part of the social graph being collected, which is also unlikely to be related to abusive activity. While we have no data to support the theory that popular accounts do not show abusive behavior under their real name (e.g., "@Nero" was a popular account suspended by Twitter), we do not see how excluding such atypical accounts would introduce an undue bias.

To ensure that a relevant amount of abuse is present in our dataset, we collect a sample of tweets by manually identifying a few user profiles on Twitter that are either i) likely victims of abuse based on the fact that their opinions in the OSN platform are

$\mathcal{G}$	Directed graph $\mathcal{G}$=($\mathcal{V}$, $\mathcal{E}$)
$\mathcal{E}$	Set of edges in graph $\mathcal{G}$
$\mathcal{V}$	Set of vertices in graph $\mathcal{G}$
$\mathcal{V}'$	Set of vertices in graph $\mathcal{G}$ at *maxdepth*

Algorithm 3.1: Bounded Breadth First Search (bBFS)

input : *seed*
input : $maxfollowers$ is a threshold per node.
input : *maxdepth* is a threshold for BFS termination.
input : *mindegree*: threshold to include nodes beyond *maxdepth* in crawl.
function: $\mathcal{F}(x) \coloneqq \{y \mid (y,x) \in \mathcal{E}\}$ gets followers of node $x \in \mathcal{G}$.
function: $Add(x,y) \coloneqq \{(x,y) \mid y \in \mathcal{F}(x)\}$ maps follower y to parent item x.
output : $\mathcal{G}$=($\mathcal{V}$, $\mathcal{E}$)
output : $\mathcal{B} \coloneqq \{\forall y \in \ \mathcal{F}(x),\ y \mapsto x \mid x \in \ \mathcal{V}\}$ set of followers of node $x \in \mathcal{V}$
output : $\mathcal{B}' \coloneqq \{\forall y \in \ \mathcal{F}(x),\ y \mapsto x \mid x \in \ \mathcal{V}'\}$ set of followers of node $x \in \mathcal{V}'$

```
1  begin
2      distance map
3      parent map
4      fifoq ←seed parent ←(seed: ∅) distance ←(seed, 0)
5      while q not empty do
6          u = fifoq.pop() level ←distance(u) followers ← F(u)
7          if level ≤ maxdepth and |F(u)| ≤ maxfollowers then
8              foreach v ∈ followers set do
9                  if level + 1 ≤ maxdepth and |F(u)| ≤ maxfollowers then
10                     fifoq.push(v)
                       //Save parent and distance to root
11                     B ← Add(u, v) distance[v].reassign(level + 1)
12                 else
13                     if |F(v)| ≤ maxfollowers then
14                         B' ←Add(u, v)
           //Decide if follower beyond maxdepth is included
15         if mindegree > 0 then
16             foreach u ∈ B'(u, v) do
17                 if |v| ≥ mindegree then
                       //Include user in crawl
```

	Overall	**Depth 1**	**Depth 2**	**Depth 3**
$\mathcal{E}_m \in \mathcal{G}_m$ directed to seed set	1648	1648	–	–
$\mathcal{E}_m \in \mathcal{G}_m$	773,162	734,896	36,487	1636
# with mentions	374,907	359,302	14,920	567
# with mentions & retweets	1878	1765	113	0
# with mentions & replies	1183	1026	292	284
# $\mathcal{E}_s \in \mathcal{G}_s$	27,017,119	25,042,892	1,636,161	0

Table 3.1: Basic statistics of the data crawled

often a source of debate (e.g., human rights' activists, members of minorities such as feminist, LGBT and politically active communities, etc) ii) randomly selected accounts. Together, these accounts form a "seed set" of potential victims we use to bootstrap our data collection methodology. We ensured that all of these seed accounts were largely active in English so that we could hope to comprehend the interaction. Finally, we made sure that the accounts did not receive an excessive number of messages, as "celebrities" may be easy to manually identify as likely victims, but would likely not be representative of the whole population, and would have also caused excessive manual annotation work. We also collect a messaging graph which consists of all public messages directed towards accounts in the seed set. While abuse may also occur in private messages, our access to Twitter did not allow us to observe such messages. Thus, the messaging graph contains public tweets. The authors of messages in the messaging graph that mention accounts in the seed set are potential first-degree perpetrators in the same sense that the seeds are potential victims.

Table 3.1 shows some basic statistics about the dataset collected, such as the number of tweets directed towards the users in the seed set. In total, we account for 1648 tweets mentioning some of the seeds.

3.3.3 Other datasets

Publicly available OSN datasets that focus on abuse are quite limited. The first and most closely related to our problem definition is the Impermium dataset, which is available from a public Kaggle competition [1]. This dataset provides a number of labeled tweets that can aid in the textual analysis of abuse for detecting insults. While this dataset provides interesting data for application of NLP based techniques to the problem fo abuse detection, it does not seem to fit well with the use of social graph metadata for detecting such abuse. Secondly, as the tweet identifiers from the Imperium dataset are anonymized, it is not possible to reproduce a collection of the text in this dataset.

Table 3.2 lists a number of seminal works for analysis of large scale analysis of the Twitter social graph.

Dataset	Size ($\mathcal{V}$)	Volume ($\mathcal{E}$)	Overall average degree	Maximum degree	Triangle count	Clustering coefficient	Diameter
Twitter (ASU, IBM) ICWSM '10 [50]	465,017 vertices	834,797 edges	3.5904 edges/vertex	678 edges	38,389	0.0613%	8 edges
Twitter (MPI) ICWSM '10 [40]	52,579,682 vertices	1,963,263,821 edges	74.678 edges/vertex	3,691,240 edges	55,428,217,664	0.0937%	18 edges
Twitter (KAIST) WWW '10 [90]	41,652,230 vertices	1,468,365,182 edges	70.506 edges/vertex	3,081,112 edges	34,824,916,864	0.0846%	23 edges
Twitter (Inria Sophia) COSN '12 [64]	537,500,000 vertices	23,950,000.000 edges	n/a	n/a	n/a	n/a	n/a
Twitter (Inria Rennes) WWW '16 [67]	971,586 vertices	230,848,163 edges	n/a	n/a	n/a	n/a	n/a

Table 3.2: Metrics of Twitter datasets with social-graph metadata, ordered increasingly by year of crawl

3.3.4 Database implementation

We store data in PostgreSQL, presumably the world's most advanced open source database technology. To automatically define tables at crawler's runtime, we use a library[17], an object relational mapper (ORM) that maps objects to back-end tables into PostgreSQL. This adds a compatibility layer to our system if we wanted to port it to another relational back-end and reduces literal SQL statements in our code. The data model is defined in the database.py file in Appendix A.

3.4 Abuse characterization

To proceed to characterize and detect abuse, we first need to collect a ground truth representing abuse, so that we can use it for statistical modeling of the problem and refer to it as a human baseline. However, identifying abuse is a hard cognitive problem even for humans, thus obtaining such ground truth is difficult and also costly process due to scarcity of high-quality human workers available for such task. To identify abuse we build Trollslayer [18], a web-based platform that provides a set guidelines to aid humans in the annotation of tweets that may represent abuse. This web based crowd-sourcing platform allows us to enlist various researchers and colleagues to assist us with the annotation task, thus initially avoiding the problem of untrusted workers and the high cost of a cloud based third-party solution. Next we validate the results obtained with our solution and compare them with an existing commercial crowdsourcing platform, Crowdflower. In both cases workers were rewarded appropriately for the annotations, removing from our pool workers those that did not fulfill the requirements to enter our Human Annotation Task (HAT), similar to Amazon's Human Intelligence Task [19].

We aggregate annotations of Trollslayer and Crowdflower ground truth into one, referred to as Both from now on in this thesis. Overall we use 163 trusted raters to code tweets manually. In total they review 14,193 annotations, out of which 9809 are acceptable and 2469 abusive. The remaining 1912 are undecided and 3 are text duplicates (namely retweets) we exclude. We are not aware of any similar, public annotated dataset that can be used as ground truth of abusive behavior in Twitter. We will make the identifiers of the messages (tweet id's) public, but not the rest of the information so that other researchers can replay the crawl for verification of results

[17] `http://www.sqlalchemy.org`

[18] `http://trollslayer.decentralise.rennes.inria.fr`

[19] `https://en.wikipedia.org/wiki/Amazon_Mechanical_Turk`

without jeopardizing privacy of users or breaking Twitter terms and conditions[20].

3.4.1 Voting scheme

To establish our ground truth and handle disagreement among reviewers, we implement a voting scheme that determines which choice most reviewers agreed with. The voting scheme calculates an overall agreement among reviewers of a tweet, the result of this calculation is then the human ground truth, which we will refer from now on as HB. Our voting scheme calculates the overall score of a tweet by adding $+1$ to the score if a tweet was marked as acceptable by a reviewer, -1 for abusive, and 0 when the reviewer was undecided (or failed to annotate). For the final agreement, we consider a tweet to be abusive if the resulting overall score is strictly less than -1 and to be acceptable if it is strictly above $+1$. We exclude the tweet from learning and evaluation if the score is in the range of $[-1, 1]$. The rationale behind this decision is to ensure that the number of agreeing reviews for a tweet is large enough to allow us to consider the result as a good approximation of ground truth. In particular, using the proposed method ensures that we do not classify tweets as abusive or acceptable solely based on the opinion of a single reviewer. In Tables 3.3 and 3.5 we see this approach does not remove a significant amount of tweets from our dataset, but prevents a biased final decision on the tweet. For the % agreement value, we computed the percentage of agreement among reviewers vote on a tweet with each other, namely whether was in agreement with the rest about the tweet. Considering the resulting voting using Both crowdsourcing platforms together, ground truth shows reviewer's agreement of 95.73% on common reviewed tweets, namely c_overall. For c_abusive there is 90% agreement and for c_acceptable 0.96%. As expected, agreement on abusive tweets is significantly lower (independently of platform used for ground truth) but very high on acceptable tweets, which is our Human baseline (HB).

Figure 3.3 summarizes these results in the context of the Human Annotation Task (HAT) in Trollslayer, Crowdflower , as well as with Both as ground truth. Surprisingly, and in despite of the cognitive difficulty of the task proposed to humans, agreement is fairly reasonable and similar in both platforms. Despite providing a supportive context to reviewers, Trollslayer shows lower agreement on abusive tweets. We suspect this is due to counting with a limited number of reviewers only, thus having disagreement in many cases which in turn renders an item not useful for training in later chapters. Also note the percentage of tweets annotated as abusive in Crowdflower is roughly 6 times larger than in Trollslayer.

Secondly, the result of the annotation with Crowdflower shows ratings on acceptable and abusive do not follow a symmetric, bell-shaped distribution [21]. in terms of agreement percentage among reviewers' ratings, see the "mean" and "median" values of the boxplot in Figure 3.3. Outliers are clearly skewing the data distribution in Crowdflower to lower percentages of agreement, which is pulling the mean lower than then median. On the contrary, in Trollslayer, the mean and median are closely related. That

[20]

[21] `http://onlinestatbook.com/`

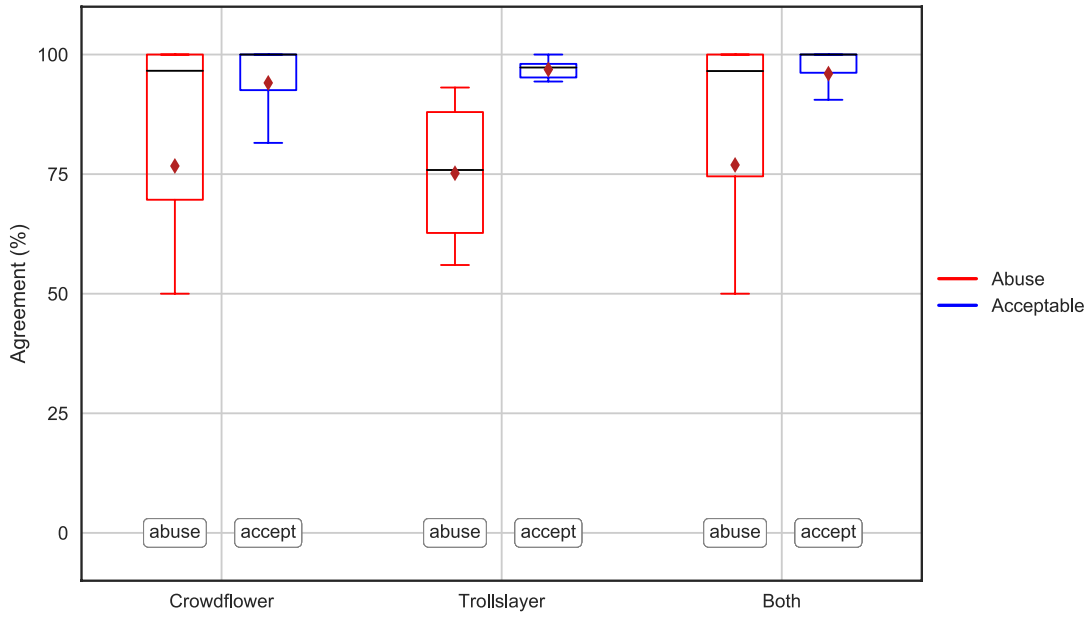


Figure 3.3: Agreement by Crowdsourcing platform

is, the "mean" is nearly equal to the "median", thus minimizing the sum of absolute deviations. Naturally, due to the lack of such outliers in the case of the Trollslayer, the "median" is not dragged down so much and even stays above the mean.

3.4.2 Trollslayer ground truth

Trollslayer can obscure the usernames of authors of a tweet involved in the conversation to minimize exposing private information, and to elicit annotations that are specific to the tweet and not the author. In addition, it display previous and subsequent tweets ordered by timestamp around the message being evaluated. The goal is to help workers understand the context of a particular tweet or discussion.

Reviewer input is a non-binary choice, as workers are able to annotate tweets from $\mathcal{E}_m$ and label them as either acceptable, abusive or undecided. The latter option being important as even with relatively clear guidelines, workers are often unsure if a particular tweet is abusive, especially given a limited context. To further compensate for this uncertainty, each tweet is annotated multiple times by independent workers.

The Trollslayer interface is depicted in Figure 3.4, which displays to raters only tweets in ($\mathcal{E}_m$), namely directed to the set of potential victims as explained following our summarised version of JTRIG's guidelines, Section 3.2.3. We do not annotate other tweets based on the assumption that victims are unlikely to explicitly subscribe to feeds of their tormentors.

In addition to the tweet itself, Trollslayer displays some supporting context, such

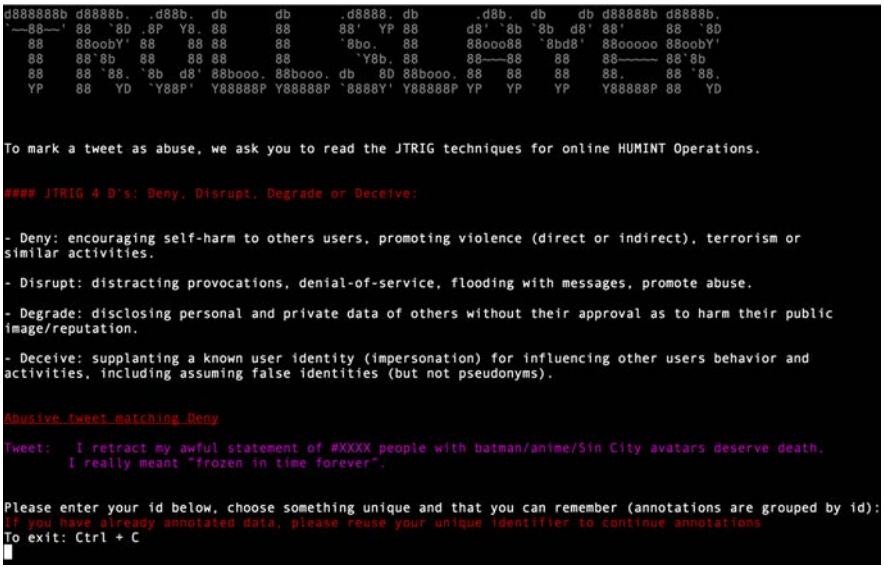


Figure 3.4: TrollSlayer interface

as previous and subsequent tweets in the tweet's author's timeline. To help workers understand the context of a particular tweet Trollslayer obscures the account names involved in the conversation to minimize exposing private information, as to elicit annotations that are specific to the tweet and not the author.

Analysis of annotations in Trollslayer In our Trollslayer platform, we manage to collect 5102 annotations where any of the 47 potential victims from our seed set is mentioned. For evaluation we include scores from the 7 workers, who contributed at least with more than 500 annotations each. In the aggregate, we have over 388 tweets annotated as abusive but only a subset of workers agreed on labeling it as abusive after voting. This suggests that such abuse classifier should be deployed locally, so that indeed potential victims can perform the detection without relying on a centralized authority. On the other hand, it would be expensive to train such classifier at first but a set of pre-labeled messages can be included with the tool once the user installs it. Table 3.3 summarises the annotations of abusive, acceptable provided by our internal workers in our own crowdsourcing platform, Trollslayer. On average, workers reported 3.75% of the reviews as abusive, and 92% as non-abusive. They were undecided for the remaining 18% approximately. Individual workers making independent decisions about the nature of a tweet (abusive, non-abusive) often disagree with the rest.

Incidents during annotation in Trollslayer In Trollslayer, we noticed and that the task of manually coding tweets as abusive, acceptable or undecided is of such subjective nature that can potentially render low levels of agreement among individual workers, due to having insufficient and different number of tweets annotated. In addition, and even though we selected a number of trusted workers to minimise such

reviewer	total # reviewed	% abusive	% acceptable	% agreement	c-abusive	c-acceptable	c-overall
1	694	2.73	93.08	95.82	0.63	0.97	0.96
2	348	4.02	91.09	95.11	0.56	0.98	0.95
3	559	3.22	94.81	98.03	0.62	1.00	0.98
4	602	4.48	91.19	95.68	0.93	0.96	0.96
5	663	3.31	93.66	97.00	0.76	0.98	0.97
6	528	4.73	89.58	94.31	0.89	0.95	0.94
7	686	3.79	90.23	94.02	0.87	0.94	0.94
μ	582.85	3.75	91.95	95.71	0.75	0.97	0.96
σ	121.46	0.71	1.926	1.423	0.15	0.02	0.14

Table 3.3: Human baseline statistics with interval voting *in* Trollslayer. *The c-values are discussed in 3.4.1. Note* undecided *votes are not counted.*

shortcoming, disagreement among workers can still arise, mostly because they are asked to make a decision for tweets which are simply perceived as borderline. The availability of the undecided option in the Trollslayer platform does not fully address this, because workers faced with a difficult choice can then avoid being decisive. We have seen a small number of workers (at least 1) who consistently annotates a few of such borderline tweets as undecided, indicating that the task can become so tedious to the point of discouraging human workers from providing decisive annotations. Below is an example of a borderline case where we observe perfect disagreement among workers in our Trollslayer platform (Table 3.4).

We find several cases where there is perfect disagreement among reviewers (same number of votes for than against). The problem with messages such as the one in Table 3.4 is that the user posting the message is repeatedly tagging or mentioning other users through the special character "@" that Twitter has provided to direct public messages to other participants. We investigated the public profile of the author of this tweet. While looking at its profile description it all seems legitimate: "Food Service 4 Rochester Schools"; taking a look at the linked website to the expanded url in the tweet takes us to a donation site hosted in Ontario, Canada. Browsing on the website we find a reference to a more than doubtful image of the founder of the organization, [22]. Therefore next we decide to check the source of the tweet and find that its value is the URL "https://unfollowers.com", which in turn redirects to URL "https://statusbrew.com/". This is a commercial site that engages online audiences or in other words, a social media campaign management tool. After a quick inspection at the products offered by the social media campaign management site, indeed we note that the site offers to its customers the option of automatically "scheduling content" for online publishing. Such Twitter accounts are known as content polluters or bots (not a human) that also perform a lot of following/unfollowing. The fact that the account links to a presumably fraudulent donations site and that we crawled this same profile back in 2016-01-10 23:02:59, where the account only had 16690 followers as compared to the current 36531, indicates non-human behavior. We plan to investigate such patterns in the future by comparing several of our snapshots of the Twitter social graph.

[22] `https://pureheartsinternational.com/pages/founder`

tweet_ids	677457607502655489	677584862044430336
tweet_id	677497741904334848	674349370792280064
677622407562141696	677530462332538881	
677542880962048000	677617537002151937	

Table 3.4: Tweets with perfect disagreement using Trollslayer *ground truth*

3.4.3 Crowdflower data

To verify the results obtained so far, we decide to experiment with a commercial cloud-based platform which provides workers that can annotate our tweets for a fee. We choose Crowdflower [23], which provides on-demand access to human workers all around the world and several tools for task optimization. The platform offers the possibility of completing large-scale, fine-grained cognitive intelligence tasks where often humans can be more accurate in the annotation than machines [122]. Crowdflower also provides a number of test questions to filter our bad raters and select contributors with the highest quality, namely including only those who have an accuracy equal or higher than 70% percent throughout the 10 test questions (a mix of abusive and acceptable tweets) we ask them prior to entering our HAT. In addition we also take advantage of the option to select only workers from a subset of English speaking countries (Australia, Canada, India, Sweden, United Kingdom, United States) and make sure that at least 3 workers have annotated each tweet. In Crowdflower we account for 156 trusted raters. These raters provide 9088 annotations in Crowdflower, out of which 6287 are acceptable, 2081 abusive; while the remaining 720 are undecided. There are 0 text duplicates (or retweets). In contrast, similar works only account for a limited number of labels for true positives in the ground truth, which is 40-65 for harassment in [126], 31 for abuse in [67] and 7 in case of the NSA [58].

The taks is again annotating a dataset of tweets looking for abusive cases, which is an abstract problem requiring humans to study the contents and context, namely a Human Intelligence Task (HIT) according to Amazon Mechanical Turk [24]. Similarly, Crowdflower provides access to human intelligence tasks that require human knowledge. The main downsides are, the monetary cost associated with obtaining high-quality raters that complete the task in due time following the guidelines we provide and that by default it does not obscure metadata about the author of a tweet as we do in Trollslayer. On the contrary, it assists us in the process of finding workers willing to complete the task and provides timely results that are straight forward to analyse with our framework.

Analysis of annotations in Crowdflower Table 3.5 shows basic statistics on agreement among the 153 raters of our Crowdflower experiment. Unlike in Trollslayer (only 7 raters), we show only aggregated reviewer's statistics. The results show that in average,

[23] `https://www.crowdflower.com/`

[24] `https://www.mturk.com/mturk/help?helpPage=overview`

each rater has a much smaller amount of tweets annotated, meaning having more human workers able to annotate renders a larger number of annotations but individually, each human provides less annotations.

	total # reviewed	% abusive	% acceptable	% agreement	c-abusive	c-acceptable	c-overall
μ	51.35	19.09	74.61	93.70	0.89	0.95	0.94
σ	95.10	19.67	20.79	9.13	0.20	0.09	0.09

Table 3.5: Human baseline statistics with interval voting *in* Crowdflower. *The c-values are discussed in 3.4.1. Note* undecided *votes are not counted.*

Incidents during annotation in Crowdflower In Crowdflower, Table 3.6, we find also cases of disagreement. However, there is a larger number than in Trollslayer. This may explain some of the later results in our automated classification of abusive behavior in Twitter, whereas some crowd-workers simply annotate tweets based on the assumption that a badword is indicative of the definition of abuse given by JTRIG (we consider these samples as adversarial in later states of the thesis).

3.4.4 Ground truth of Both platforms

Finally, we aggregate annotations from both platforms to create a unique ground truth. We also display a few examples of perfect disagreement we find in this ground truth after aggregating votes using our scheme presented earlier on in this chapter.

We calculate intersection amont tweets with perfect disagreement among annotations in Trollslayer and Crowdflower, noting the resulting intersection is zero. This underpins the following analysis where we find raters in Crowdflower to annotate much less on average.

tweet_ids	671006453398962177	670982735331262464
671082628259389441	677242705647411200	677570057472315393
677296141126488064	670930784480460802	671058912423100416
677095274129461249	674387711634112512	670981847111720960
670984874824568832	674469438436020224	671058162317832194
676165807026610177	669981909238190080	671013645719961600
670422679959154689	671065033993035776	677623069578539009
670975934514745344	668304445286031361	670962157182394368
677343184180015105	674390853968723968	
671011520050946048	670935553840312321	

Table 3.6: Tweet id's with perfect disagreement using Crowdflower *ground truth*

3.5 Summary of results

We have presented our data collection methodology, built a"Human Annotation Task" and analyzed the resulting agreement among human raters or reviewers following our JTRIG's definition of abuse in all experiments, either using Trollslayer or Crowdflower.

3.5.1 Dataset

Size and timeline We collected nearly 1M tweets belonging to timelines of users directing at least a tweet with a mention to a set of potential victims we define in a OSN model as per Section 3.3. Our data collection started in December 2015, thus having tweets and social relationships published/formed since then. Note such tweets are not counted as edges in Table 3.2 simply because such arcs belong to a messaging multigraph as discussed later in the thesis.

Finally, the size of our database is about 30GB in PostgreSQL. This is comparable to past works in SNAM shown in Table 3.2, also including large portions of the Twitter social graph.

Ethical considerations While our complete database or dataset includes social graph relationships among nodes in the network, publishing such metadata is forbidden due to TTC, which also apply onto data collected for research purposes. Given this limitation, publishing of tweet related metadata must be anonymized or limited to the tweet identifiers[25]. The tweet identifiers are therefore subject to be made available. This allows researchers to comply with Twitter terms and conditions accordingly. For social graph metadata, the tweet identifier can also be used for replaying collection of a portion of the Twitter social graph.

To advance the research on abuse detection in the Twitter environment, we release an anonymized version of our dataset and annotations[26].

3.5.2 Agreement

In terms of agreement, results are not that different among Trollslayer and Crowdflower, but the latter shows lower agreement. This proves the fact that even humans have difficulty in deciding what is abuse, so in next chapter we evaluate if machine learning algorithms can do better.

To statistically asess this agreement/disagreement among workers, we calculate an inter-assessor agreement descriptive statistic. According to [124], there are number of descriptive statistics such as Light's kappa and Hubert's kappa, which are multi-rater versions of Cohen's kappa. Fleiss' kappa is a multi-rater extension of Scott's pi, whereas Randolph's kappa generalizes Bennett S to multiple raters. Similarly to Cohen's kappa or Fleiss' Kappa, Randolph's kappa descriptive statistic is used to measure

[25] `https://twittercommunity.com/t/twitter-and-open-data-in-academia/51934/5`
[26] `github.com/algarecu/trollslayer`

the nominal inter-rater agreement between two or more workers in collaborative science experiments. We choose Randolph's kappa over the others by following Brennan and Predige suggestion from 1981 of using free-marginal kappa when raters are not forced to assign a certain number of cases to each category (e.g., abusive, acceptable) and using fixed-marginal kappa when they are [30]. Our scenario considers assigning a different number of annotations to each class or category, which satisfies Randolph's kappa requirement.

Given this setting, values of kappa can range from -1.0 to 1.0, with -1.0 meaning complete disagreement, 0.0 meaning agreement equal to chance, and 1.0 indicating perfect agreement above random respectively. According to Randolph, usually a kappa of 0.70 or above indicates good inter-rater agreement [27].

For the dataset annotated, we show Randolph's kappa in Table 3.7. The overall agreement is nearly 0.70 in Trollslayer with a limited number of workers and annotations, which improves significantly when aggregating annotations from Crowdflower into Both. Note that in contrast to the agreement scores from Table 3.3 and Table 3.5, we calculate kappa on agreement among all three possibilities, abusive, acceptable or undecided, thus representing a more strict metric of agreement. For our Crowdflower ground truth, Randolph's kappa is depicted in Table 3.7, which also shows below 0.70 value of kappa.

Platform	Overall agreement (%)	free-marginal
Trollslayer	0.56	0.34
Crowdflower	0.68	0.53
Both	0.73	0.59

Table 3.7: Randolph's multi-rater kappa for all platforms using n=6 and k=3

Countermeasures We find one case of a reviewer using our Trollslayer platform that avoids being decisive by repeatedly selecting the undecided option during tweet annotation. Using our voting scheme presented earlier in this chapter, we exclude malicious raters that ought not to be decisive in the ratings and avoid any impact on consensus of abusive or acceptable tweets. For Crowdflower, we wonder if the provided controls in the platform are sufficient to discourage malicious workers from voting. However this is less of a concern as in the commercial platform as we count with a larger number of raters.

For Trollslayer, we plan to introduce a simple countermeasure that performs on-the-fly statistics of a user's annotations internally in the platform, not just providing controls at the application level. Crowdflower provides these controls during task annotation entry in order to decide if a rater can be trusted or not. This includes a timer that enforces raters to read content and pass several test questions prior to entering the annotation process. This will also help prevent malicious users that taint our ground truth.

[27] `http://justusrandolph.net/kappa/`

In the future we believe approaching the problem in an smarter manner, namely using "active learning" techniques during annotation of tweets can enhance our crowdsourcing approach so that it obtains better and larger ground truth. Besides, we can also make our tool more attractive to raters by introducing a gamification approach where more content of interest is displayed to them as they progress in the task. This avoids raters being discouraged from the tedious task of annotating abuse.

4

Abuse detection in modern Online Social Networks

Contents

In this chapter we extract a set of features from our dataset, previously presented, and then build and evaluate a number of supervised machine learning algorithms with the support of an open source machine learning framework, namely scikit-learn for Python. The features extracted include information such as account date of creation, bidirectional links in the network, network metrics (node in/out degree), etc. which together with a set of novel intersection features introduced here for the very first time, give result to our main findings on abuse detection in Twitter.

In Twitter, users can follow other users, effectively subscribing to their updates or comments. This architecture follows the usual pub-sub that traditionally distributed applications have implemented. In Twitter, these subscriptions create the so called Twitter social graph. In contrast to OSN as Facebook, Twitter allows asymmetric relationships in a directed graph, which provides diverse network dynamics through subscription/subscriber based pattern that may be or not reciprocal in Twitter social graph. This is a notable characteristic of Twitter social graph, which several studies have investigated in order to describe the social dynamics of such graphs [64], which is based on the idea of modeling the web as a graph structure [88] [31].

Machine learning is a subfield of computer science that evolved from the study of techniques such as pattern recognition in artificial intelligence. The advantage of machine learning is that allows scientists to use algorithms that learn without being explicitly programmed. In particular, machine learning is used today for building models that make data-driven decisions in many different areas of computer science, from recommendation systems [1] to image recognition [5].

Supervised machine learning has been used to detect abuse in OSN. There is large body of literature that tries to apply machine learning techniques to such problem. While we follow the same path, our work is characterised by the set and subset of features we input to our classifiers. We show them in Table 4.1. They included but are not limited to metadata about the tweets, users, social and messaging graph, as well as some other aspects we briefly utilise during our analysis.

4.1 Background

To deal with abuse without considering the content of the communication, graph-based techniques have been shown to be useful for detecting and combating dishonest behavior [107] and cyber-bullying [65], as well as to detect fake accounts in OSN [38], but they suffer from the fact that real-world social graphs do not always conform to the key assumptions made about the system. Thus, it is not easy to prevent attackers from infiltrating the OSN by deceiving others into befriending their fake (Sybil) accounts. Consequently, these Sybils create the impression of being strongly connected to a cluster of legitimate user accounts, which in turn makes graph-based defenses not useful. On the other hand and yet in the context of OSN, graph-based sybil defenses can benefit from machine learning techniques by extracting OSN user metadata into their feature set in order to predict potential victims of abuse [26].

Our work is partially inspired by the idea of leveraging victim prediction for robust abuse detection in an OSN. According to [24], abusive users can only befriend a fraction of real accounts. In the case of Twitter, their abuse detection algorithm is presumably considering metrics such as the number of mentions to other users (`@user`), tweets per day of the user, number of follow request sent by the user per day and so forth, as to flag unusual or suspicious behavior [117]. At Facebook, the Facebook Immune System (FIS) automates abuse detection [115]. FIS uses information from user activity logs to

[1] `http://netflixprize.com/`

automatically detect and act upon suspicious behaviors in the OSN. Such automated or semi-automated methods are not perfect. In FIS, [25] found that only about 20% of the deceitful profiles they deployed were actually detected, which shows that such methods result in a significant number of false negatives. In this context, we see that the main limitation of state-of-the on abuse detection is performing classifications with numbers as 40-65 for harassment in [126], or even 7 in the case of the NSA [73]).

4.2 Feature engineering

We extract a set of features related to account, message, messaging-graph and social-graph based metadata. In addition, we add the some information theory set of features, not yet proven to be effective in our experiments. The set of features is presented in Table 4.1.

4.2.1 Account metadata

In the case of user accounts, numerical features include the number of followers, followees, ratio of followees to followers (and viceversa), favorited tweets, ratio of replies over tweets, number of replies from a potential victim, tweets per day, mentions per day and rate of mentions over tweets. The age of the user account is also included in this category as numerical feature.

4.2.2 Message metadata

For message properties, we extract numerical features such as the number of mentions, hashtags in a tweet directed to the seed set, as well as the the count of "bad words" in the message itself (using a pre-processed list of badwords from [81]). Regarding categorical, we evaluate when a message is a retweet, reply, or contains sensitive content according to Twitter.

4.2.3 Messaging-graph metadata

In terms of messaging-graph metadata, we calculate the number of directed edges (messages) in the graph from a potential abuser to the potential victim, how many users reply to a mention in the tweet (edge directed in reverse direction), delay of the reply from a user in the potential victim seed set (if any), and finally mutual hashtags and mutual mentions among sender and received. A categorical feature here is whether a potential victim replies of not to the message.

A tweet is invasive if the potential victim has received a message from a subscriber not listed in her subscriptions, is a categorical feature in the feature set of social-graph metadata.

4.2.4 Social-graph metadata

For social graph-related data, we collect the subscriptions and subscribers of sender and receiver of the message. We then calculate several metrics, listed in Table 4.1, most of the numerical.

We can not annotate tweets based on social-graph metadata, thus our approach is to show to volunteers reviewing tweets only the tweet text, which is generally totally independent of the meaning of each feature, except for "swear words". However, after analyzing the patterns among annotated messages, we find a disjointness among mutual connections in the social graph for abusive compared non-abusive tweets. We see abusive users have less mutual edges directed towards/from their potential victims (e.g., mutual subscribers, mutual subscriptions).

4.2.5 Abuse distribution

To characterize abuse distribution of each feature, we employ a CCDF (Complementary Cumulative Distributed Function) in log-log scale for convenience, as we can pack a large range of values in a relatively short space. How do we get to a CCDF? In simple terms, a CCDF represents the inverse of the CDF (Cumulative Distributed Function), which in turn comes from the summation or integration operation on its probability density function (PDF). The CCDF represents the probability P that a feature having value x (x axis), or more, does not exceed X (y axis).

In Figures 4.1 to 4.5 we show data distribution of both, acceptable and abusive cases in each of the features extracted from our dataset, see Table 4.1. We employ the CCDF curve to highlight how often a feature is below a particular threshold and thus characterize abuse in Twitter dataset accordingly to annotations for the dataset using data Chapter 3.

In the case of message metadata we clearly see that acceptable messages do not contain "badwords" . Because "badwords" appear often for abusive messages, we observe that its CCDF in 4.2c shows a range from 1-7 with higher to lower probability, meaning that more than 7 in a tweet is highly unlikely. This makes a lot of sense, because a tweet is restricted to 140 characters in length. In contrast, the CCDF about number of hashtags shows that acceptable messages often contain more hashtags, which is an indication that abusive messages do not bother using this functionality much. Note these two CCDFs are not in log-log scale as the values are easyly packed into the x axis.

We also observe diverse patterns in message graph metadata features with their respective CCDFs. In the case of "replies" it is clear that abusive messages occur in a setting with a limited number of replies and more frequently than in acceptable messages.

We observe that social and similarity based features do not exhibit a very distinguished pattern among abusive and acceptable messages, so it is difficult to know at this point how they impact the abuse classification. However, in the case of Figure 4.4d, a feature that can be easily manipulated by a malicious adversary, the overall subscrip-

tion ratio over time for abusive messages is clearly kept under those acceptable. This makes sense if a malicious adversary is trying to remain undetected, however they need to control their subscription ratio carefully not to exceed that of legitimate users.

4.3 Classifiers

Table 4.1 summarizes the list of features we extract and develop from the dataset collected. In classification problems, the specific combination of features is critical for classifiers performance so we evaluate their relative importance. In the following experiments, we considered all the aforementioned features, including the count for "swear words" [81].

We experiment with tree based supervised models [16] from scikit-learn[2]. Such models work well in classification problems where data is limited or unbalanced. In particular, we compare decision trees (DT) [29], random forest (RF) [28], extra trees (ET) [70] and the AdaBoost (AB) and gradient boosting (GB) classifier [27]. We use default classification options for 'criterion' (gini) and 'splitter' (best) as well as the same values of 'max_depth', 'class_weight' and 'random_state' in all classifiers for reproducibility, and to avoid biasing in favor of any specific classifier that got more tuning. While we tried other supervised learning algorithms such as logistic regression, k-means clustering and NB-trees, the aforementioned methods performed best, and thus we limit our presentation to those.

4.3.1 Decision trees

In 1983, Breiman [29] first described CART algorithm. It could be used for any of, regression or classification of a binary target. Decision trees are a robust method for machine learning, as they are invariant to the inclusion of irrelevant features and can be easily visualized. CART is a recursive algorithm, which finds the best data splitting at each iteration to maximize the chances of predicting the correct target values. We use the CART implementation of decision trees in Python, which is a close variation of the well-known C4.5, but with no rule sets according to its authors.

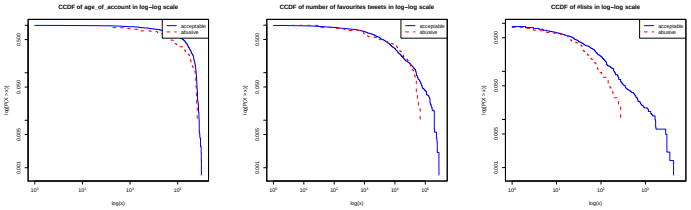
4.3.2 Random forest

A random forest is an ensemble of decision trees [28]. Random forest combines tree predictors in each tree, depending on the values of a random vector sampled independently with the same distribution for all trees in the forest. It uses a number of decision trees, so in theory, random forest should improve on the performance of decision trees.

4.3.3 Extremely randomized trees (extra trees)

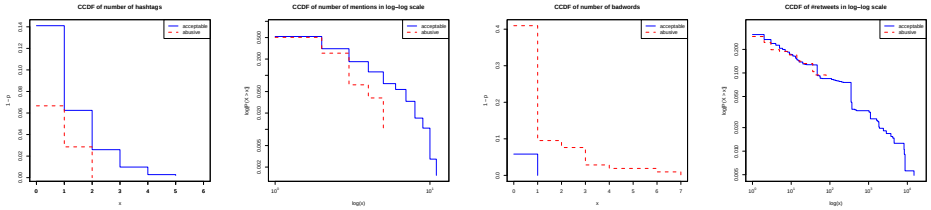
In ExtraTrees, the classifier [70] selects a random subset of candidate features, but unlike random forest, thresholds are drawn at random for each candidate feature and

[2] `http://scikit-learn.org/stable/supervised_learning.html`



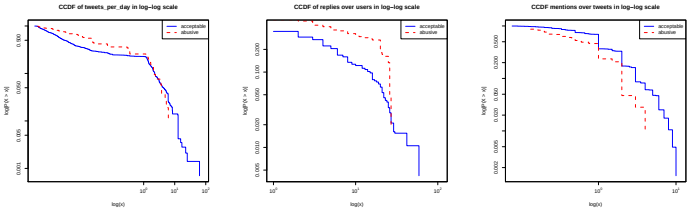
(a) *(b)* *(c)*

Figure 4.1: Account based features



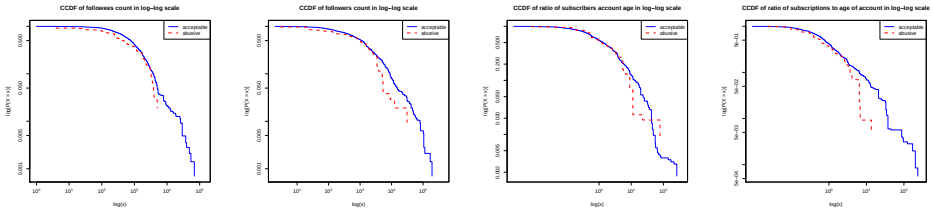
(a) *(b)* *(c)* *(d)*

Figure 4.2: Message based features



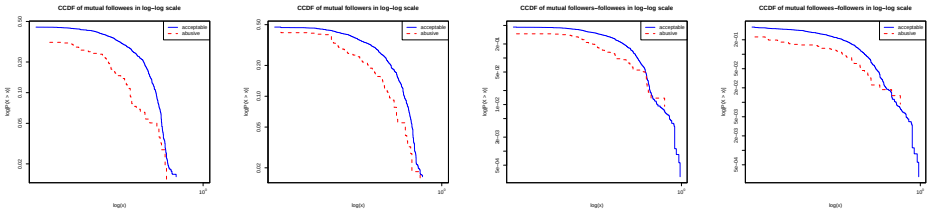
(a) *(b)* *(c)*

Figure 4.3: Messaging-graph based features



(a) *(b)* *(c)* *(d)*

Figure 4.4: Social based features



(a) *(b)* *(c)* *(d)*

Figure 4.5: Similarity based features

the best of these randomly-generated thresholds is chosen as the splitting method. This allows to reduce the variance of the model a bit more, at the expense of a slightly greater increase in bias.

4.3.4 Gradient boosting

This algorithm, also introduced by Breiman [27], is a non-parametric method that can be used for regression or classification. The benefit of gradient boosting is that it works well with features measured at different scales and is able to detect non-linear feature interactions.

4.3.5 AdaBoost

The core principle of AdaBoost is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction [3]. It was introduced in 1995 by Freund and Schapire [63]. The idea is to concentrate in samples that were misclassified in previous iterations so that the error can be reduced.

4.3.6 Support Vector Machines

Support Vector Machines (SVM) are a type of supervised machine learning model that incorporates the idea of mapping in space the different points to classes or categories, so that there is a clear separation among them. SVM comes in two flavors, linear or non-linear. SVM can perform non-linear classification (the one we use for this thesis) by implicitly mapping such points into high-dimensional feature spaces. The scikit-learn implementation is based on libsvm [4] which the full paper in [41] covers for reference.

Support Vector Machines are used for classification, outliers detection or regression. Its main advantages are i) its ability to perform well in high dimensional spaces ii) being effective when the number of dimensions may be higher than the number of samples

[3] `http://scikit-learn.org/stable/modules/ensemble.html#adaboost`
[4] `https://github.com/cjlin1/libsvm`

and iii) only needs a subset of points for training (less computational overhead). Its drawbacks are related to cases with imbalance in the number of features and samples, which may render the classifier useless; or the fact that they do not directly provide probability estimates (need cross-validation).

We evaluate the utility of SVM for abuse detection. In the first instance, we explore a linear SVM classifier. However, it performs very poorly. Therefore, we also try the regular default, non-linear classifier for SVM in scikit-learn, namely Radial Basis Function (RBF). According to the documentation, training an SVM with the Radial Basis Function (RBF) kernel requires two parameters: C and gamma. The first parameter, C, trades off misclassification of training examples for simplicity of the decision boundary. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly. Gamma defines how much influence a single training example has. The larger gamma is, the closer other examples must be to be affected. We set gamma to "auto" for parameter self tuning and C to 1.0 to give the model the freedom to select more samples as decision vectors.

4.3.7 Ensemble voting

We aggregate the results of the previous classifiers into one, namely an ensemble of classifiers. This way we can verify if the combination of several of them can yield better results than when ran individually.

4.4 Evaluation

For evaluation, the dataset is partitioned into two disjoint subsets for training and evaluation of the classification algorithms. The partitions are induced by the seeds as explained in our initial data collection 3.3.1. As a result, all collected information relating to the same seed remains in the same partition. We used 5-fold cross validation for all experiments.

An upper bound for our performance expectations is the human baseline (HB) from Tables 4.2 to 4.4. While the classification algorithms have additional data available to them, it is unrealistic for them to perform better than the individual reviewers who provided the ground truth. A dummy classifier BL is also included in the list of classification algorithms as baseline, which classifies each tweet according to the most predominant class (acceptable in our case) and misclassifies all abusive tweets.

4.4.1 Precision-recall and ROC

Binary classification approaches depends on a given threshold, which is a cutoff value in the prediction probability after which a classifier labels an item. For our classification task, in order to capture this trade-off between the true positive rate (TPR) and the false positive rate (FPR) in a single curve, the receiver operating characteristics (ROC) analysis [74] offers the possibility of visualizing the trade-off resulting from different threshold values. Given the low prevalence of abuse in our data, we also display the

precision-recall curve for each classifier, a more appropriate way to visualize such trade-off when using imbalanced number of samples in the predicted classes, namely of abuse and acceptable. Precision (a.k.a. specificity) and Recall (a.k.a. sensitivity) are briefly introduced next for informational purposes during the later evaluation:

- Precision, $P = \frac{tp}{tp+fp}$, measures the positive predictive power of the classifier, or in other words, which fraction of true positive elements is classified as such.
- Recall, $R = \frac{tp}{tp+fn}$, measures the fraction of relevant elements, which are actually returned by the classifier, either right (True positives) or wrong (False negatives).
- F-score, a.k.a. $F1 = 2 \cdot \frac{P \cdot R}{P+R}$, combines precision and recall, and is the harmonic mean of precision and recall. Usually its value is among the precision and the recall values.

In our benchmark, we denote the class of interest (abusive) as the Negative and the other acceptable as the Positive. It is important to notice this difference, as usually the class of interest is always Positive in the confusion matrix. However, we make this choice to emphasize the nature of the tweet also in the confusion matrix.

In simple terms, high precision means that an algorithm returned substantially more relevant results than irrelevant, while high recall means that an algorithm returned most of the relevant results.

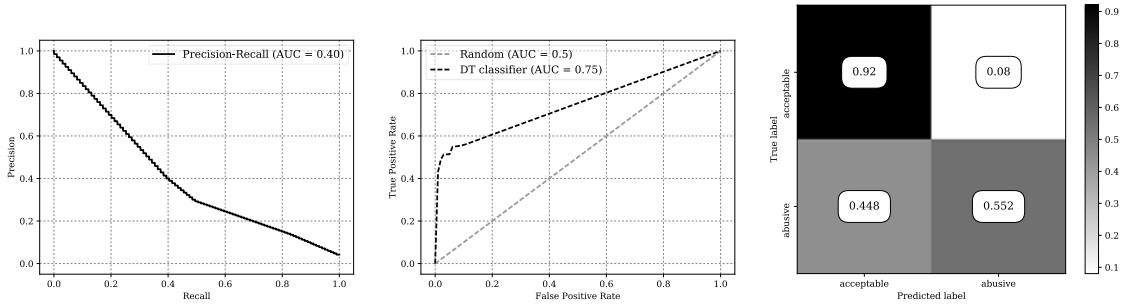
4.4.2 Evaluation on Trollslayer ground truth

We show the raw numbers of precision, recall and the F-score in Table 4.2. Surprisingly, some learning algorithms –especially AdaBoost and SVM– approach the quality of the human reviewers baseline (HB).

In figs. 4.6 to 4.12 we display the precision-recall (P-R) and ROC curves of each classifier alongside with their respective confusion matrix, which allows a visual comparison of the classification results using Trollslayer ground truth.

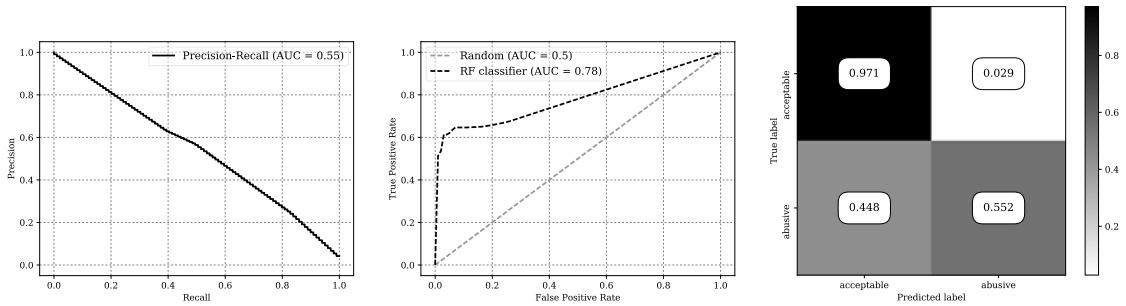
The AdaBoost (AB) classifier is the least confused between abusive and non-abusive tweets. However, it mislabels a slightly higher percentage of abusive tweets compared to other classifiers, such as extra trees (ET). In contrast, AB is nearly perfect on correctly labeling acceptable tweets 4.10c. It performs better than extra trees (ET) for acceptable tweets, which has roughly double amount of confusion for those 4.8c and obviously worse p-r curves 4.8a. The Voting classifier provides a moderate number of false-positives in both, acceptable and abusive, thus having the largest AUC for the ROC (%98.5). Finally, the SVM classifier with Radial Basis Function (RBF) kernel has the best Recall for abusive tweets but mislabels a high amount of acceptable ones also, thus not being able to provide a balanced number of false positives and negatives in the task of abuse detection.

Feature relative importance using Trollslayer ground truth We analyze the relative importance (RI) of each of the features in the classifiers presented in Table 4.1.



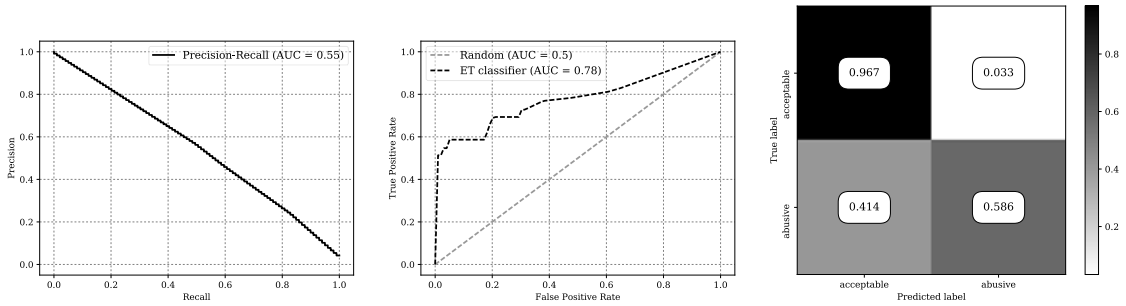
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.6: Evaluation for decision trees using Trollslayer ground truth



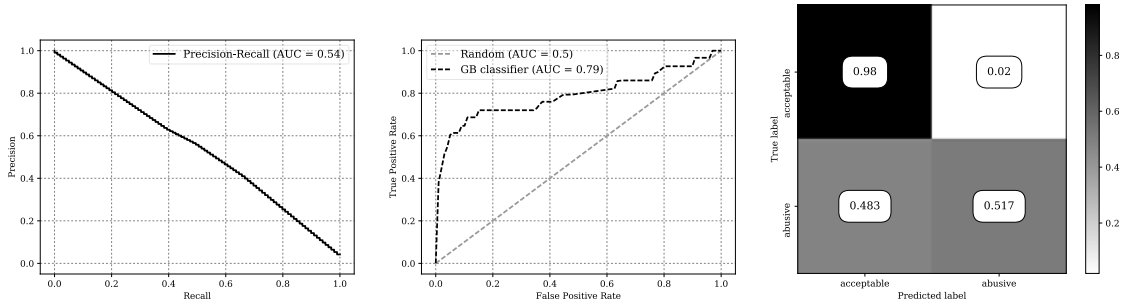
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.7: Evaluation for random forest using Trollslayer ground truth



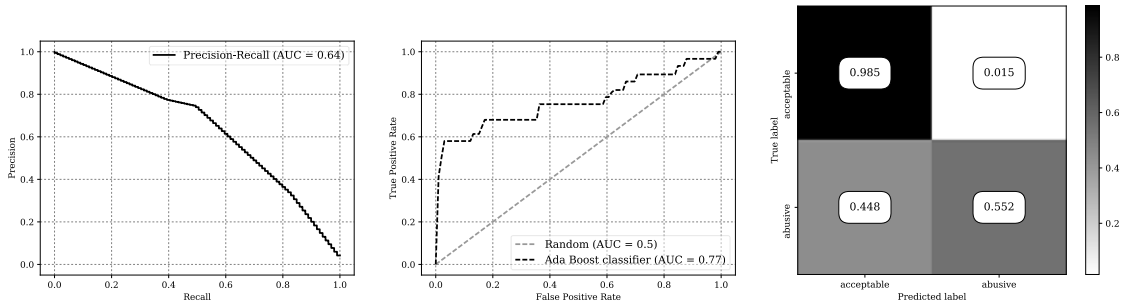
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.8: Evaluation for extra trees using Trollslayer ground truth



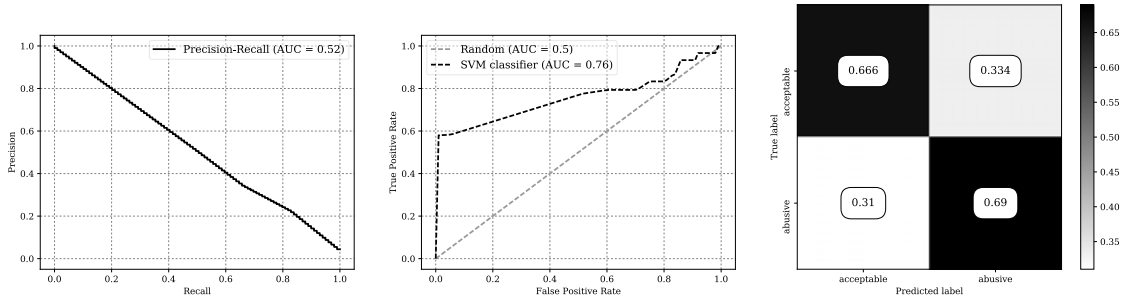
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.9: Evaluation for gradient boosting using Trollslayer ground truth



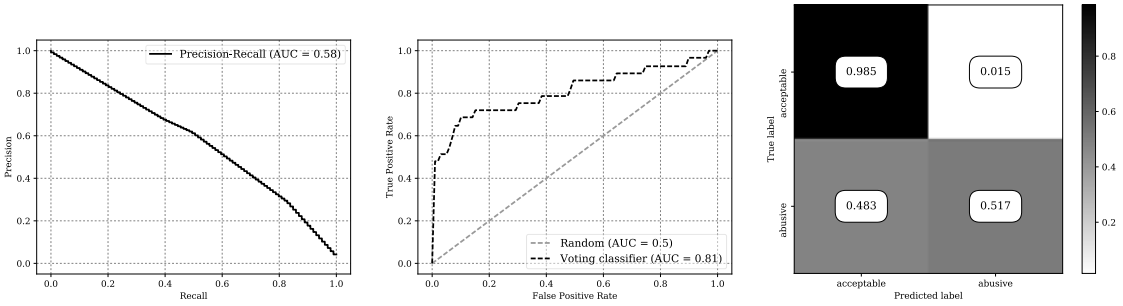
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.10: Evaluation for gradient AdaBoost using Trollslayer ground truth



(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.11: Evaluation for SVM using Trollslayer ground truth



(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.12: Evaluation for Ensemble Voting using Trollslayer ground truth

The Voting ensemble does not provide such information at this time in scikit-learn so we are unable to analyze it.

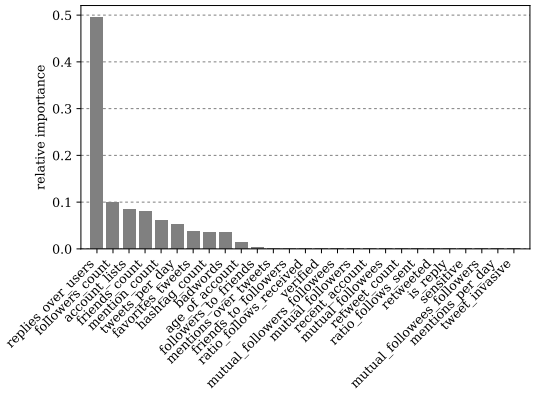
Feature relative importance is not uniform across all classifiers. "Replies over users", "friends count" and "account lists " are often among the top 5 highest features in all classifiers. The exception is ET, which ranks "badwords" and "account age" higher instead. AdaBoost ranks "mention count" as first feature, confirming our hypothesis regarding the fact that abusive message must be able to address a potential victim in such manner on Twitter. AdaBoost also ranks"age of account" higher than ET. Presumably, such combination together with "badwords" and other messaging, social and account based features help the classifier to reach a reasonable performance in the detection of abuse.

The rest of top 5 features in GB are "age of account", "badwords", "followers count", "mention count"'. This indicates that both, social and messaging graph related features from Table 4.1 have a larger impact in the performance of the GB classifier than "badwords" and thus are required to produce a better detection of abuse. Also note that in all classifiers, the feature "tweet invasive" ranks as the most useless features. We suspect this may be related to the fact that such feature is made up of two directional features that define a bi-directional relationship in the social graph among potential victims and perpetrators.

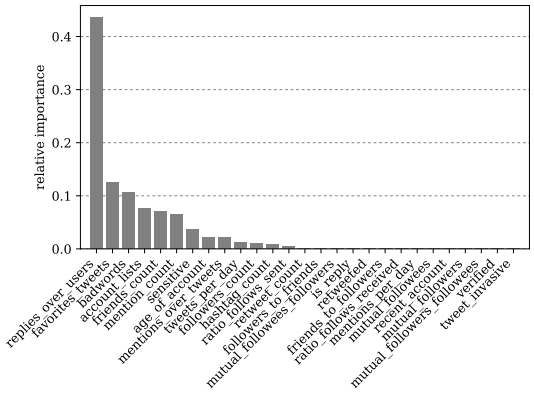
Note we pull out "badwords" from the list of features in the DPM publication [69] in order to highlight the fact that this is feature which can be easily obfuscated by an adaptive adversary. This does not change overall values of abuse detection as we will see, but proves the fact that using lists of "badwords" is not be a reliable manner of detecting abusive content (in particular comments that "degrade") in a modern OSN. Also because as we will see, crowd-workers in Crowdflower often mark a tweet as abusive only based in the existence of such a "badword" in the message. This assumption taints the abuse annotation and does not proof reliable in the classification results obtained in the next chapter, abuse detection, even if as we show the inter-rater agreement is higher in Crowdflower than in Trollslayer ground truth annotation (0.53 versus 0.34).

In general all classifiers, as seen in Figure 4.13, make use of features from Table 4.1

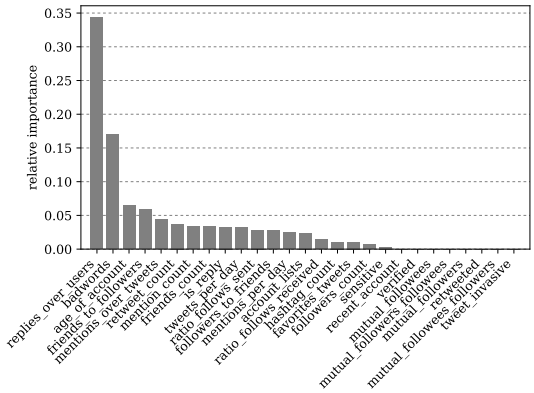
related to account metadata, message metadata, messaging metadata, social graph metadata and to a lesser degree similarity graph metadata . We suspect this is due to some other features containing a subset of information that makes up for more complicated ones (e.g., # followers is a subset of mutual followers, namely Jaccard index among follower sets).



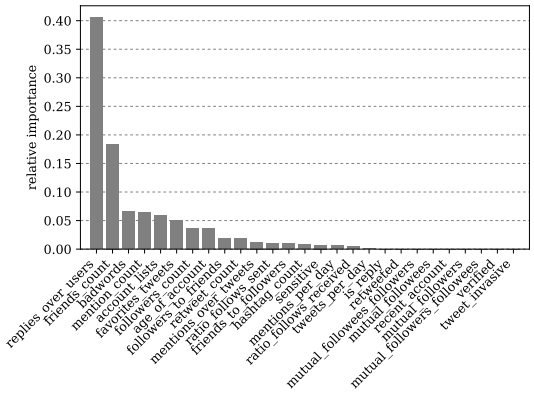
(a) RI of DT



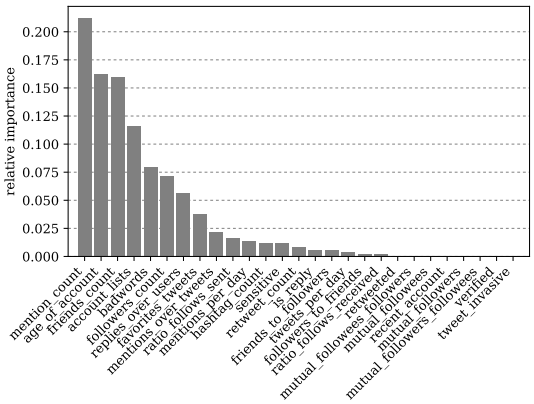
(b) RI of RF



(c) RI of ET



(d) RI of GB



(e) RI of AB

Figure 4.13: RI of classifiers with Trollslayer ground truth

4.4.3 Evaluation on Crowdflower ground truth

We now evaluate our classifiers with ground truth from Crowdflower only. Results in Table 4.3 show that the learning algorithms –especially AdaBoost – perform significantly worse using this dataset. Extra trees (ET) obtains the most reasonable result. Detection of abuse is poorer with Crowdflower than Trollslayer ground truth. This means that the ground truth is significantly different and the classifiers do not generalize so well in such context when tested on unseen data.

In figs. 4.14 to 4.20 we display the precision-recall (P-R) and ROC curves of each classifier side by side along with their confusion matrix allow a visual comparison of their classification performance using Crowdflower ground truth.

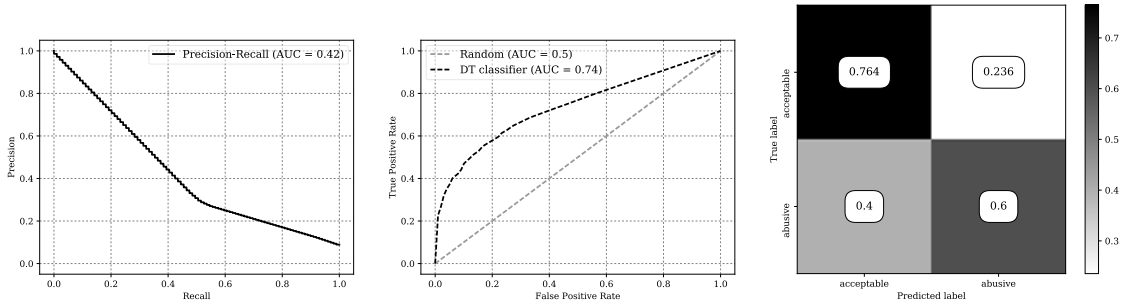
We show the raw numbers of precision, recall and the F-score in Table 4.3. As mentioned, the learning algorithms do not perform so well with this ground truth, especially Gradient boosting, whose performance degrades significantly, due to mislabeling many more samples on both acceptable and abusive cases

The ET classifier is the least confused among abusive and acceptable cases, see Table 4.3. In terms of Recall, DT offers the best result and AdaBoost (AB) outperforms the in precision to the rest.

Feature relative importance using Crowdflower We analyze the relative importance (RI) of each of the features in the classifiers presented in Table 4.1 when using the Crowdflower ground truth. For the voting classifier we do not provide such information at this time because it is not yet available in the scikit-learn library. The same applies to the SVM classifier.

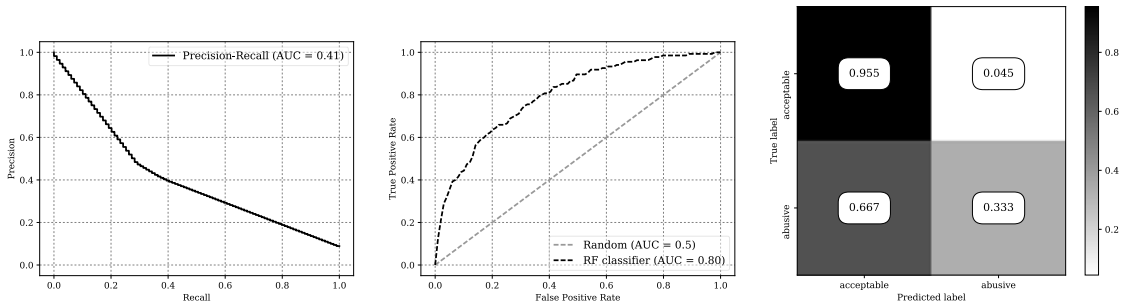
Considering the best performing classifiers from Figures 4.14 to 4.20, the relative importance of DT and ET classifiers is as follows: "badwords" has a large contribution to the detection of abuse in both classifiers. In the first, DT, we observe "followers count" ranks as highest, and in ET "badwords" is first followed by "mention_count" and "age_of_account".

Gradient Boosting GB splits the highest ranking features among quite a few more than the former two classifiers. However, its result is not optimal. This means the feature combination employed is leading to more confusion rather than a better result. In particular, this combination highlighted in Figure 4.21d seems to optimize for best results on detecting acceptable tweets.



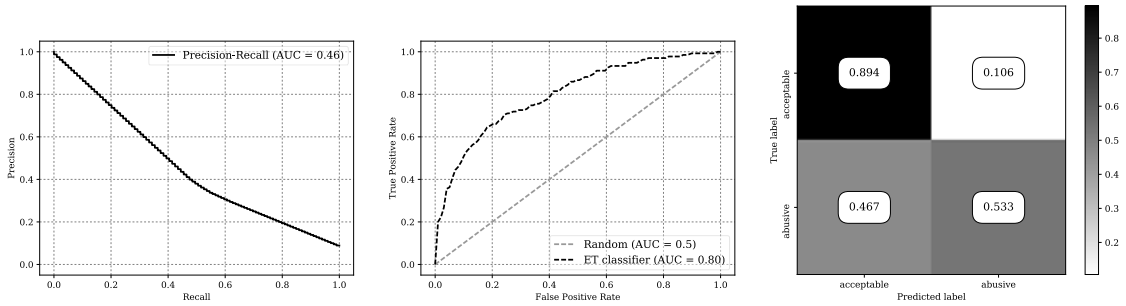
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.14: Evaluation for decision trees with Crowdflower ground truth



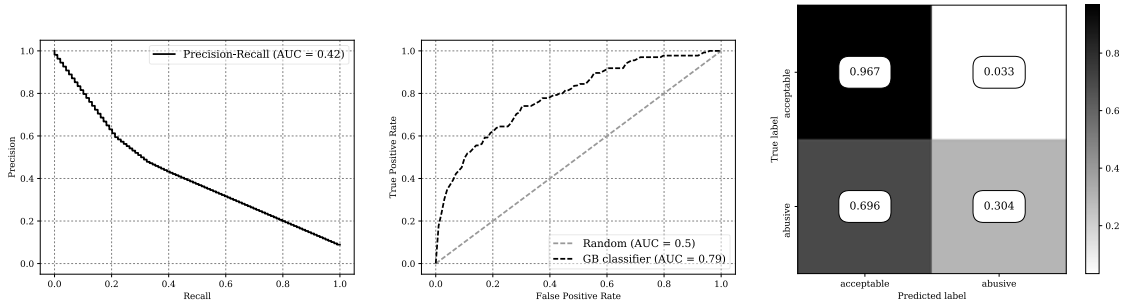
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.15: Evaluation for random forest with Crowdflower ground truth



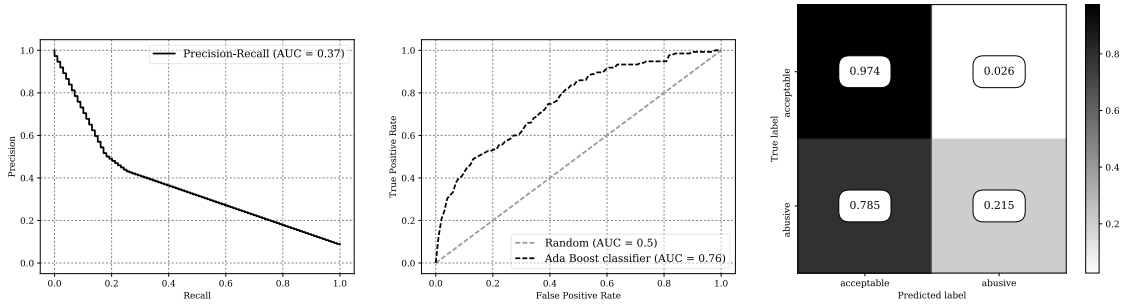
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.16: Evaluation for extra trees with Crowdflower ground truth



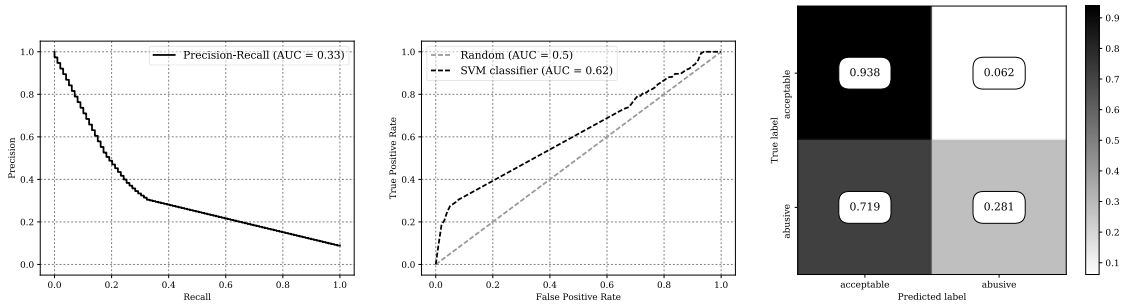
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.17: Evaluation for gradient boosting with Crowdflower ground truth



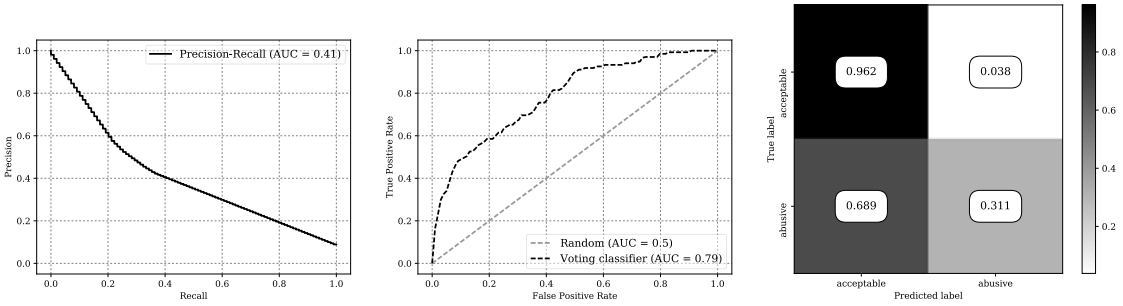
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.18: Evaluation for gradient AdaBoost with Crowdflower ground truth



(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.19: Evaluation for SVM with Crowdflower ground truth



(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.20: Evaluation for Ensemble Voting with Crowdflower ground truth

4.4.4 Evaluation on Both ground truth

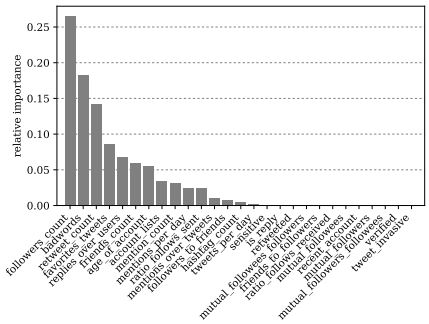
We also analyzed the performance of the classifiers on the aggregate ground truth obtained from both platforms, Trollslayer and Crowdflower.

The results here show highly ineffective classification of abusive tweets with any of the classifiers. Unfortunately, the best performing algorithm for abusive (SVM) provides high recall (0.93) but very low precision for the acceptable class. This imbalance results in a high number of items classified abusive when they are actually acceptable, see Figure 4.27c. On the other hand, extra trees DT, or even ET provide the most balanced classification among the two classes and all the classifiers evaluated. However, it is still difficult for any of these classifiers using our ground truth to approach the human baseline of reviewers for precision and recall of 0.77 as seen in Table 4.4.

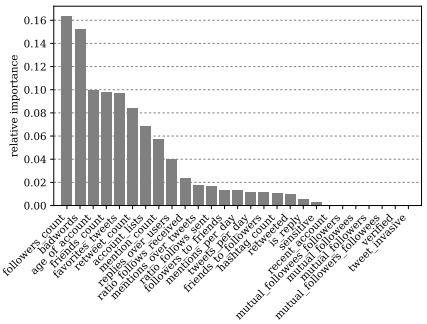
Feature relative importance using Both ground truth We analyze the relative importance (RI) of each of the features in the classifiers presented in Table 4.1. The Voting ensemble does not provide such information at this time in scikit-learn so we are unable to analyze it.

Feature relative importance is not uniform across all classifiers, but “badwords” ranks highest in both, DT and ET, indicators of good classifications as discussed earlier. Second is “age_of_account”, with almost as much relative importance in classifications under decision trees. As usual, the number of “mentions”, “friends”, “replies” and “retweets” complete the most important top 5.

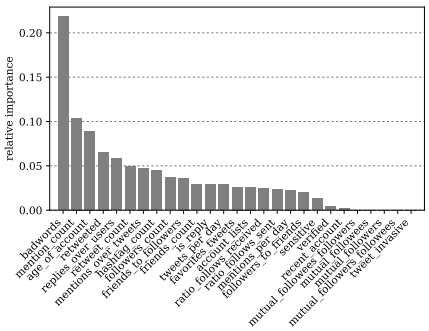
The use of “badwords” and “age_of_account” seem critical then to correctly classify a reasonable amounts of tweets close to the parameters provided by the human baseline in Table 4.4.



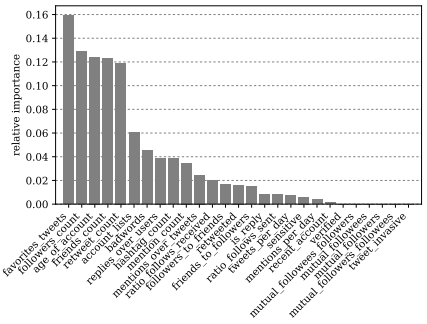
(a) RI of DT using Crowdflower ground truth



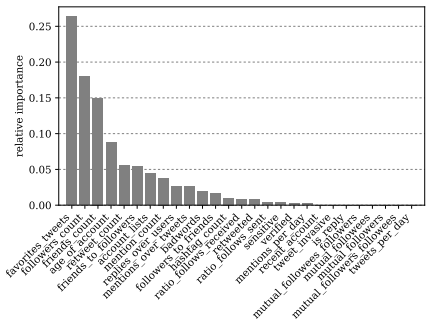
(b) RI of RF using Crowdflower ground truth



(c) RI of ET using Crowdflower ground truth

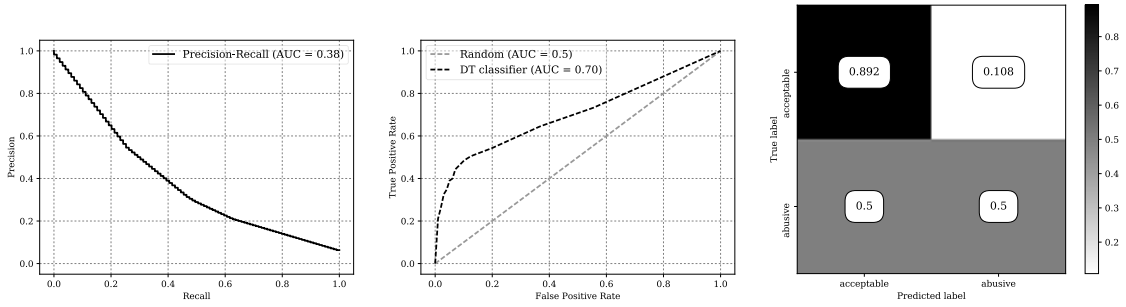


(d) RI of GB using Crowdflower ground truth



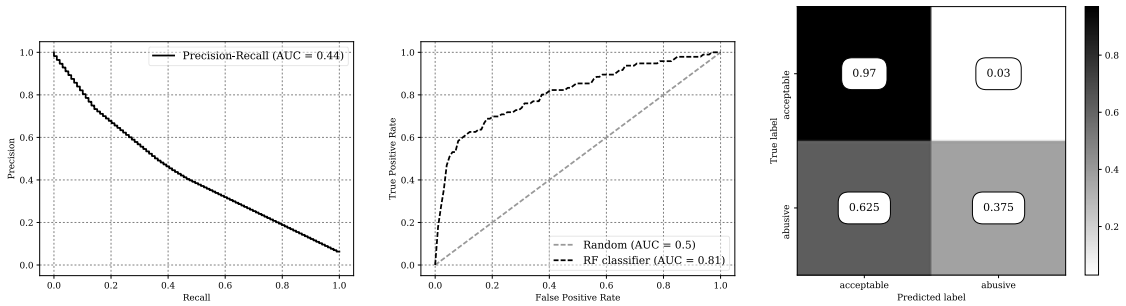
(e) RI of AdaBoost using Crowdflower ground truth

Figure 4.21: RI of classifiers with Crowdflower ground truth



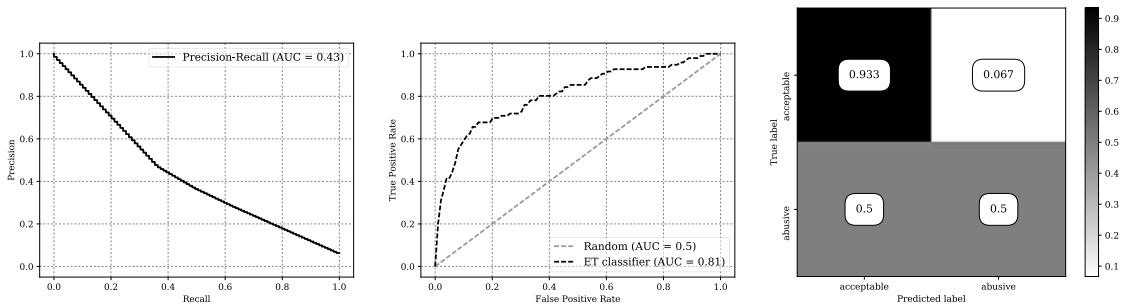
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.22: Evaluation for decision trees with Both ground truth



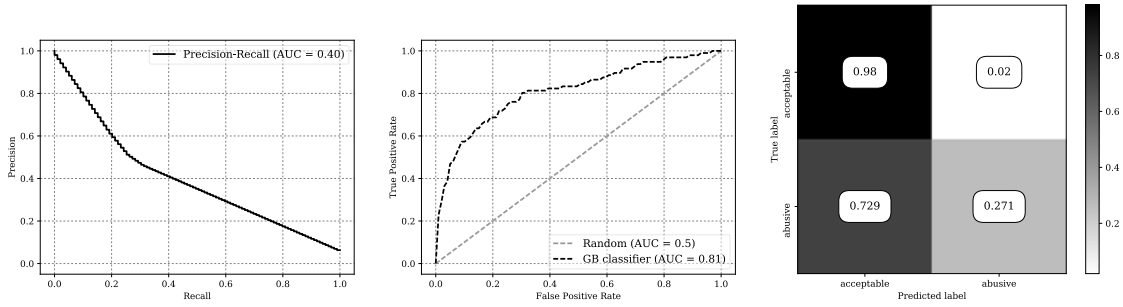
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.23: Evaluation for random forest with Both ground truth



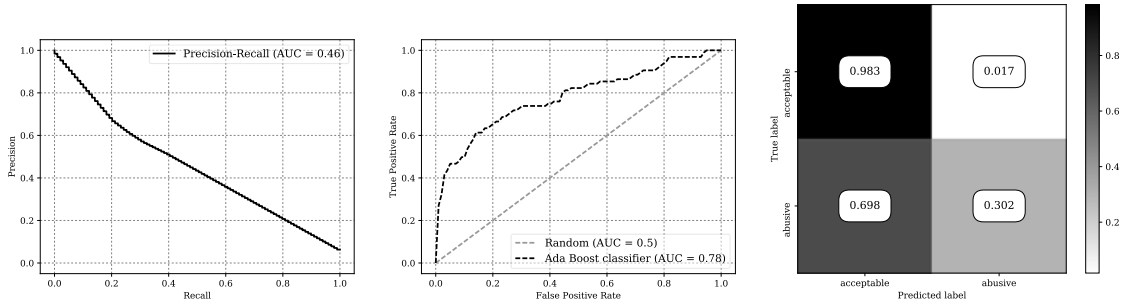
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.24: Evaluation for extra trees with Both ground truth



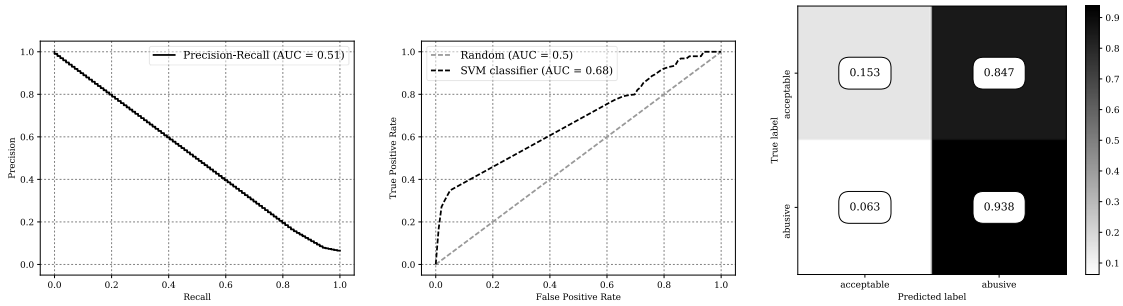
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.25: Evaluation for gradient boosting with Both ground truth



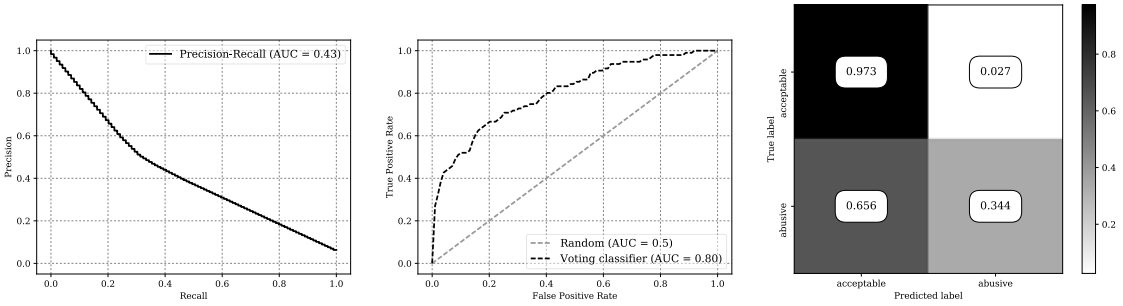
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.26: Evaluation for gradient AdaBoost with Both ground truth



(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.27: Evaluation for SVM with Both ground truth



(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 4.28: Evaluation for Ensemble Voting with Both ground truth

4.5 Summary of results

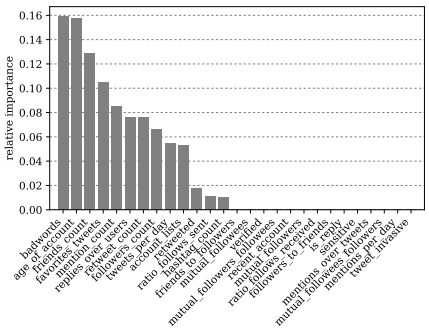
It is insightful to compare the results with each ground truth. We observe that Trollslayer provides the best approximation to reasonable abuse detection results. We suspect this is due to agreement, as seen in chapter 3, which means more reviewers and more annotations does not really help to train supervised machine learning classifiers if the consensus or quality of the ratings decreases.

AdaBoost performs best with Trollslayer ground truth as we seen in Table 4.2. We believe this is due to the ability of this type of classifier to adjust the weights of incorrectly classified instances so that subsequent iterations of the classification focus in more complicated or extreme samples. It is important to note that we use AdaBoost with a default algorithm in scikit-learn, namely real SAMME.R bossting algorithm [131].

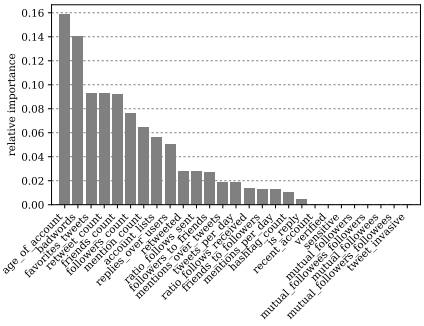
Another insight is that performance of classifiers using Crowdflower data decreases significantly due to classifiers mislabeling more cases that are considered acceptable in the respective ground truth. The sum of the precision and recall is not weighted by class predominance, therefore we obtain a more pessimistic result that what in theory one could do so with our methodology, yet we are able to detect some abuse fairly well.

Finally, comparing each classifier's confusion matrix (CM) with the results from the human reviewers in Table 3.3, reviewer number 1 only agreed in 63% of the cases with his peers on abuse, which is nearly indistinguishable from ET classifier, which had 59% agreement 4.8c with the reviewers' human baseline. Note that on average, in Trollslayer reviewers agreed on 75% and 97% of the cases for abusive and acceptable respectively. Similarly, Crowdflower provides a consensus of %89 and %95, but with more outliers below the median value.

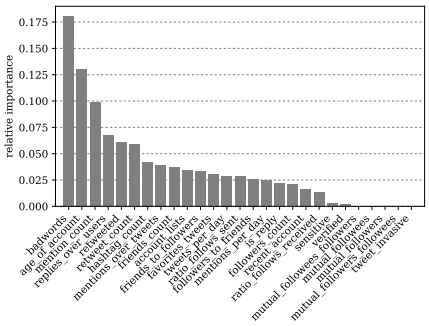
Finally, using all features proves to be most beneficial with SVM classifier 4.11, which provides high Recall (true abusive ratio) at the cost of mislabeling a significantly higher percentage of acceptable cases, thus leading to poorer overall Precision than most classifiers (except decision trees), 65%.



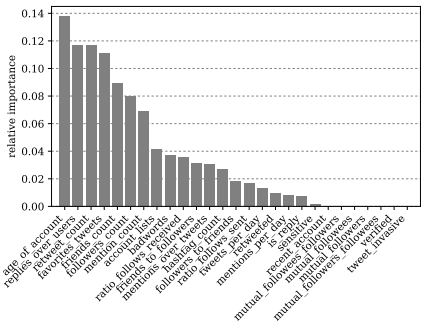
(a) RI of DT with Both ground truth



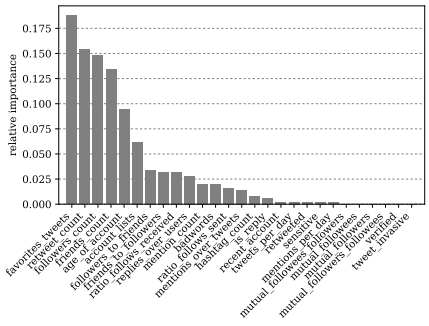
(b) RI of RF with Both ground truth



(c) RI of ET with Both ground truth



(d) RI of GB with Both ground truth



(e) RI of AdaBoost with Both ground truth

Figure 4.29: RI of classifiers with Both ground truth

Feature	Description	Type
account metadata		
age of account	days since user account creation.	Numerical
verified account	whether sender account is verified by Twitter or not	Categorical
# lists	how many lists the sender has created	Numerical
# favorited messages	number of tweets favorited by sender account	Numerical
recent account	check if user account has equal or less than 30 days	Categorical
message metadata		
#mentions	number of mentions in a tweet	Numerical
#hashtags	number of hashtags in a tweet	Numerical
# badwords	number of "swear words" found [a]	Numerical
# retweets	number of times a message has been retweeted	Numerical
is_retweet	whether a message is re-post of another message or not	Categorical
is_reply	whether a message is a reply to a previous message or not	Categorical
possibly_sensitive_content	whether a message contains a link to an external URL	Categorial
messaging-graph ($\mathcal{G}_m$) metadata		
# mentions/# messages	fraction of messages that contain mentions in messages of sender	Numerical
# messages/age of account	fraction of messages from sender on a per day basis	Numerical
# mentions/age of account	fraction of mentions from sender on a per day basis	Numerical
replies over users	fraction of replies to tweets a user has authored	Numerical
tweet reciprocity	true if sender subscribed to receiver and receiver subscribed to sender (mutual link)[b]	Categorical
social-graph ($\mathcal{G}_f$) metadata		
# subscribers	number of subscribers to public feed of the sender	Numerical
# subscriptions	number of subscriptions of the sender	Numerical
# subscribers/age of account	ratio of follows received since account creation	Numerical
# subscriptions/age of account	ratio of follows sent since account creation	Numerical
# subscriptions/# subscribers	ratio of subscriptions to subscribers of sender account	Numerical
# subscribers/# subscriptions	ratio of subscribers to subscriptions of sender account	Numerical
similarity social-graph ($\mathcal{G}_f$) metadata		
$\mathcal{J}$ (subscriptions, subscriptions)	Jaccard similarity index among subscriptions of sender and receiver	Numerical
$\mathcal{J}$ (subscribers, subscribers)	Jaccard similarity index among subscribers of sender and receiver	Numerical
$\mathcal{J}$ (subscribers^s, subscriptions^r)	Jaccard similarity index among subscribers of sender and subscriptions of receiver	Numerical
$\mathcal{J}$ (subscriptions^s, subscribers^r)	Jaccard similarity index among subscriptions of sender and subscribers of receiver	Numerical

Table 4.1: Feature set

[a] `http://fffff.at/googles-official-list-of-bad-words/`

[b] `https://en.wikipedia.org/wiki/Reciprocity_(network_science)`

Classifier	Metric	Overall	Acceptable	Abusive
HB	Precision	0.88 ± 0.08	0.97 ± 0.02	**0.78 ± 0.13**
	Recall	0.79 ± 0.21	0.99 ± 0.20	**0.59 ± 0.21**
	F-score	0.83 ± 0.10	0.98 ± 0.04	**0.67 ± 0.16**
BL	Precision	0.48 ± 0.00	0.958 ± 0.006	0.000 ± 0.000
	Recall	0.50 ± 0.00	1.000 ± 0.000	0.000 ± 0.000
	F-score	0.49 ± 0.00	0.979 ± 0.003	0.000 ± 0.000
DT	Precision	0.61 ± 0.07	0.98 ± 0.01	0.24 ± 0.14
	Recall	0.73 ± 0.14	0.92 ± 0.05	0.55 ± 0.23
	F-score	0.64 ± 0.09	0.95 ± 0.02	0.33 ± 0.16
RF	Precision	0.75 ± 0.20	0.98 ± 0.01	0.52 ± 0.40
	Recall	0.76 ± 0.14	0.97 ± 0.04	0.55 ± 0.23
	F-score	0.74 ± 0.11	0.98 ± 0.01	0.51 ± 0.21
ET	Precision	0.74 ± 0.15	0.98 ± 0.01	0.45 ± 0.32
	Recall	0.78 ± 0.11	0.97 ± 0.04	0.59 ± 0.26
	F-score	0.74 ± 0.07	0.97 ± 0.01	0.51 ± 0.12
GB	Precision	0.76 ± 0.14	0.98 ± 0.01	0.55 ± 0.27
	Recall	0.75 ± 0.09	0.98 ± 0.01	0.51 ± 0.18
	F-score	0.75 ± 0.09	0.98 ± 0.01	0.52 ± 0.18
AB	Precision	0.85 ± 0.27	0.98 ± 0.01	**0.72 ± 0.55**
	Recall	0.77 ± 0.15	0.98 ± 0.03	0.55 ± 0.23
	F-score	0.78 ± 0.14	0.98 ± 0.01	**0.58 ± 0.26**
SVM	Precision	0.65 ± 0.28	0.976 ± 0.048	0.324 ± 0.535
	Recall	0.68 ± 0.31	0.666 ± 0.738	**0.700 ± 0.389**
	F-score	0.54 ± 0.53	0.725 ± 0.611	0.349 ± 0.464
Voting	Precision	0.80 ± 0.13	0.98 ± 0.01	0.62 ± 0.27
	Recall	0.75 ± 0.17	0.98 ± 0.19	0.51 ± 0.34
	F-score	0.76 ± 0.10	0.98 ± 0.01	0.54 ± 0.20

Table 4.2: Evaluation of classifiers trained using 5-fold cross validation and Trollslayer *ground truth*

Classifier	Metric	Overall	Acceptable	Abusive
HB	Precision	0.86 ± 0.24	0.94 ± 0.12	**0.77 ± 0.36**
	Recall	0.86 ± 0.24	0.97 ± 0.10	**0.75 ± 0.39**
	F-score	0.86 ± 0.24	0.95 ± 0.11	**0.76 ± 0.37**
BL	Precision	0.46 ± 0.00	0.911 ± 0.000	0.000 ± 0.000
	Recall	0.50 ± 0.00	1.000 ± 0.000	0.000 ± 0.000
	F-score	0.48 ± 0.000	0.953 ± 0.000	0.000 ± 0.000
DT	Precision	0.58 ± 0.04	0.954 ± 0.034	0.212 ± 0.084
	Recall	0.68 ± 0.08	0.764 ± 0.210	**0.600 ± 0.329**
	F-score	0.57 ± 0.09	0.843 ± 0.129	0.305 ± 0.073
RF	Precision	0.68 ± 0.07	0.936 ± 0.009	0.428 ± 0.139
	Recall	0.64 ± 0.05	0.955 ± 0.025	0.333 ± 0.094
	F-score	0.66 ± 0.06	0.945 ± 0.015	0.373 ± 0.102
ET	Precision	0.65 ± 0.06	0.951 ± 0.010	0.342 ± 0.035
	Recall	0.71 ± 0.05	0.894 ± 0.066	0.533 ± 0.111
	F-score	0.67 ± 0.06	0.922 ± 0.035	**0.412 ± 0.084**
GB	Precision	0.71 ± 0.10	0.934 ± 0.013	**0.483 ± 0.195**
	Recall	0.64 ± 0.07	0.967 ± 0.022	0.304 ± 0.144
	F-score	0.66 ± 0.07	0.950 ± 0.012	0.367 ± 0.140
AB	Precision	0.69 ± 0.07	0.927 ± 0.006	0.450 ± 0.140
	Recall	0.59 ± 0.04	0.974 ± 0.012	0.215 ± 0.073
	F-score	0.62 ± 0.04	0.215 ± 0.073	0.289 ± 0.085
SVM	Precision	0.62 ± 0.08	0.930 ± 0.011	0.316 ± 0.153
	Recall	0.61 ± 0.06	0.938 ± 0.026	0.281 ± 0.111
	F-score	0.62 ± 0.07	0.934 ± 0.017	0.296 ± 0.120
Voting	Precision	0.69 ± 0.09	0.935 ± 0.011	0.447 ± 0.177
	Recall	0.64 ± 0.07	0.962 ± 0.019	0.311± 0.120
	F-score	0.66 ± 0.07	0.948 ± 0.013	0.366 ± 0.137

Table 4.3: Evaluation of classifiers trained using 5-fold cross validation and Crowdflower *ground truth*

Classifier	Metric	Overall	Acceptable	Abusive
HB	Precision	0.87 ± 0.23	0.96 ± 0.10	**0.77 ± 0.36**
	Recall	0.87 ± 0.24	0.97 ± 0.10	**0.77 ± 0.37**
	F-score	0.87 ± 0.24	0.96 ± 0.1	**0.77 ± 0.37**
BL	Precision	0.47 ± 0.00	0.94 ± 0.002	0.000 ± 0.000
	Recall	0.50 ± 0.00	1.000 ± 0.000	0.000 ± 0.000
	F-score	0.48 ± 0.00	0.967 ± 0.001	0.000 ± 0.000
DT	Precision	0.60 ± 0.07	0.964 ± 0.019	0.238 ± 0.124
	Recall	0.70 ± 0.14	0.892 ± 0.030	0.500 ± 0.270
	F-score	0.62 ± 0.09	0.926 ± 0.019	0.321 ± 0.162
RF	Precision	0.71 ± 0.03	0.959 ± 0.016	0.453 ± 0.050
	Recall	0.67 ± 0.11	0.970 ± 0.018	0.377 ± 0.240
	F-score	0.68 ± 0.09	0.964 ± 0.003	**0.400 ± 0.184**
ET	Precision	0.65 ± 0.06	0.965 ± 0.016	0.333 ± 0.110
	Recall	0.72 ± 0.12	0.933 ± 0.015	0.500 ± 0.235
	F-score	0.67 ± 0.08	0.949 ± 0.011	0.399 ± 0.150
GB	Precision	0.72 ± 0.08	0.952 ± 0.007	0.487 ± 0.163
	Recall	0.63 ± 0.04	0.980 ± 0.014	0.272 ± 0.086
	F-score	0.65 ± 0.04	0.966 ± 0.006	0.344 ± 0.082
AB	Precision	0.76 ± 0.17	0.954 ± 0.007	**0.573 ± 0.330**
	Recall	0.64 ± 0.07	**0.983 ± 0.019**	0.301 ± 0.126
	F-score	0.68 ± 0.09	0.969 ± 0.011	0.391 ± 0.173
SVM	Precision	0.52 ± 0.02	**0.975 ± 0.036**	0.069 ± 0.007
	Recall	0.55 ± 0.04	0.153 ± 0.033	**0.937 ± 0.103**
	F-score	0.20 ± 0.02	0.264 ± 0.049	0.129 ± 0.013
Voting	Precision	0.71 ± 0.03	0.957 ± 0.010	0.466 ± 0.064
	Recall	0.66 ± 0.07	0.973 ± 0.011	0.345 ± 0.149
	F-score	0.68 ± 0.05	0.965 ± 0.003	0.392 ± 0.101

Table 4.4: Evaluation of classifiers trained using 5-fold cross validation with Both *ground truth*

5

Abuse detection in decentralized Online Social Networks

Contents

This chapter assesses the design and development of abuse detection systems for future decentralized OSN. To aid in the detection of abusive behavior while respecting privacy of participants, we use both features that comprise local knowledge (e.g., available in the local view of the network by the user) and a data minimization approach for those that provide neighborhood knowledge (e.g., social graph structure) when required. Our approach provides an analysis of each of the input features used in our supervised machine learning classification framework for the Twitter use case. We therefore imagine this culture of timeline construction will carry over to future decentralized versions of these platforms.

5.1 Introduction

Today abuse is a pervasive issue in OSN. Even in decentralized OSN settings, some claim abuse is even more prominent here due to the anonymity such platforms provide. Therefore, decentralized abuse detection algorithms may also be required in the future as they are today in centralized OSN platforms.

Related work using social graph metadata tries to hide away the participating nodes' identity. For that several techniques have been proposed but it is also well known that an attacker can reconstruct the social graph topology by means of mere anonymous identifiers [8]. On the other hand, novel techniques have taken forward the security aspect of Peer-to-Peer networks [33], but those do not consider analyzing and proving countermeasures for possible abusive behavior in such decentralized OSN setting.

Our goal is to protect the social graph topology as best as we can while detecting abuse. To than end, we design our features that involve several types of metadata, analyze their abuse distribution and apply classic supervised learning techniques to realize if it is feasible to detect abuse with a limited training set (the main constraint in a decentralized setting). While these supervised learning techniques are not novel, the use of a set of privacy-preserving features shows to be reasonably good in the detection of a abuse (without a larger feature set).

5.2 Learning with privacy in OSN

Assumptions and Adversarial model We assume that each participant in a directed OSN network is free to subscribe to other participants, despite of what others do. We want to enable two parties to compute the similarity metric of their neighborhood in the social graph, namely subscribers, subscriptions, but without disclosing additional metadata about those.

We consider two different types of attackers or adversaries: participants in the OSN and external agents. We dot not discuss the latter type of adversary in detail in this thesis. For internal participants, we have to assume their operational mode is driven by the functionality of the OSN provider (e.g., send messages, add subscription, remove subscription, etc) and that they can deviate from the use of such functionality in order to change the topology of the social graph and gain more visibility or value in the network. For instance, an adaptive user may want to show as a subscriber of an honest, popular account as to attempt to gain more subscribers or even gain the trust (in the form a reciprocal subscription) from such a popular account.

We consider how to adapt the abuse detection algorithm to a decentralized privacy-preserving OSN, where we face an adaptive adversary who will change his behavior to evade detection. In this setting, we need to consider how to obtain the numeric value in a way that respects the privacy constraints, and how to make it difficult for an attacker to forge or falsify the value of a given feature.

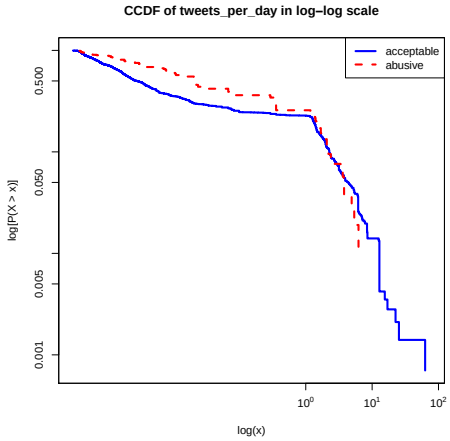


Figure 5.1: CCDF of messages/day.

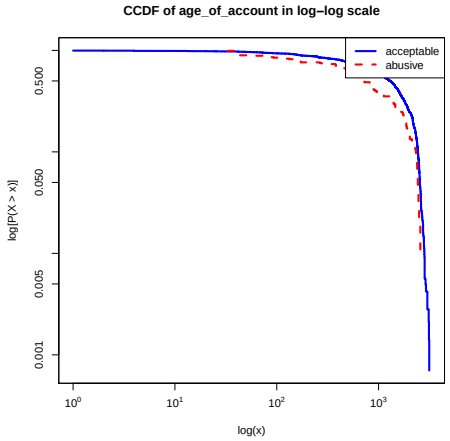


Figure 5.2: CCDF of age of account.

5.2.1 Account properties

Various features reflect properties of the sender's account that are entirely under the control of the sender. This includes the number of lists the user has created, number of favorited tweets and the age of the account. Given an adaptive adversary who knows how the abuse detection algorithm uses these features, we have to assume that the adversary can freely adapt these properties and thus deliberately manipulates all such features. This analysis also holds for features like the number of "retweets" and "favorited messages".

Age of account The "age of account" feature considers how many days ago the account was created. The classifiers generally assume that older accounts are less likely to exhibit abusive behavior (which is supported by the CCDF in Figure 5.2). Thus, an adversary has an interest in making his accounts look old. Using the age of an account is not privacy sensitive, as it hardly can be considered to be sensitive personal information about the user.

In a fully decentralized network, a time-stamping service [71] will have to be enforced to prevent malicious or adaptive participants from backdating the time at which their account was created. Naturally, a time-stamping service does not prevent an adversary from creating dormant accounts to be used at a later time for attacks. However, time-stamping raises the bar in terms of required planning, and is thus unlikely to be defeated by non-professional trolls.

5.2.2 Messaging properties

This feature simply counts the number of times a message contains some of the special functions available in existing OSN, such mentioning users (`@user`) or highlighting a topic (`#hashtags`) in Twitter. These two are examples of message properties that are

trivial to evaluate locally. The first one (mentions) seem to have negative implications for privacy when the computation is performed by the receiver, while the latter does not.

In case of mentions, adaptive adversaries may again shape their messages as to avoid a true positive in abuse classification, but possibly at the expense of being less effective at hurting the victim (e.g., not being able to mention her, thus not disrupting). We already shown CCDFs for these in previous Chapters.

5.2.3 Messaging graph

The feature "message reciprocity" is a predicate that is true if sender and receiver of the message are mutual subscribers, that is both the sender subscribes to the receiver, and the receiver subscribes to the sender. If either party is not subscribed to the other, the message is considered "invasive". Table 5.1 shows that messages that are invasive are more likely to be abusive.

	acceptable	abusive
invasive	2875	209
non-invasive	669	9

Table 5.1: Relationship between abusive behavior and invasiveness.

The predicate is trivial to evaluate locally, as both parties know their subscriptions and subscribers. While an attacker can easily subscribe to the victim, it would be hard to convince a victim to subscribe to the attacker's feed.

5.2.4 Messages per day

The feature "messages over age" represents the number of public messages sent of average by a user to all of its subscribers each day. The CCDF shows no clear trend as to whether abusive users in our data set send fewer or more messages per day (Figure 5.1). To establish this value securely, a user could subscribe to the public feed and observe the message stream. As these are public messages, there is no privacy concern. Subscribing would—with some delay—provide an accurate count of the number of messages made per day.

By supporting anonymous subscriptions and gossip-based message distribution, an OSN could make it difficult for an adversary to give the victim an inaccurate view of the public message stream of the adversary.

Naturally, the adversary may be able to adapt by sending fewer or more messages, but this may have an adverse and indirect impact into other features, particularly the adversary subscriber base. A similar analysis holds for features like "mentions over tweets", "mentions per day" and "subscriptions/subscribers per day".

5.2.5 Social features

5.2.5.1 Number of subscribers and subscriptions

In this category are features as the "# subscribers" to the sender account and the "# subscriptions" to other accounts. The first for instance represents the number of

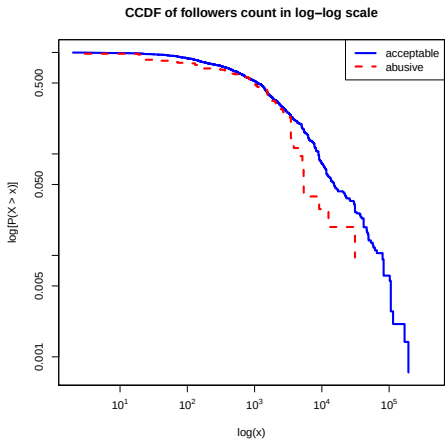


Figure 5.3: CCDF of # of subscribers.

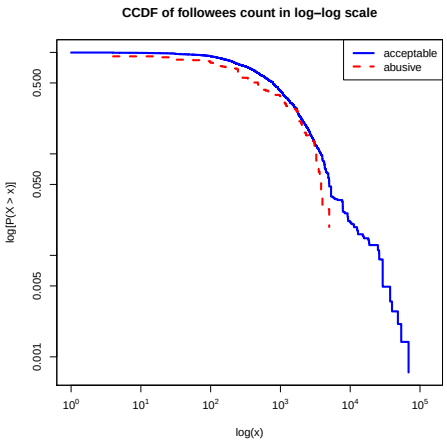


Figure 5.4: CCDF of # of subscriptions.

subscribers of the user sending a message. Figures 5.3,5.4 show that there is no clear trend in our data set between abusive and non-abusive senders. It is conceivable that this is because the feature is trivial to manipulate: creating new accounts is generally relatively cheap, and there are even existing blackmarkets for Twitter [118].

Assuming that abusive accounts do need to artificially inflate their subscriber base, one could use proof-of-work based group size estimation methods [59] to increase the cost of faking a large subscriber base. However, the network size estimation method presented in [59] would reveal the public keys of some of the subscribers. Still, this is easily mitigated by having each subscriber use a fresh pseudonym for each subscription, limiting the use of this special pseudonym to the group size estimation protocol. This has the drawback that the proof-of-work computation would have to be performed again for each subscription.

In any case, we do not expect such methods to work particularly well: an adversary can typically be expected to be willing to spend significant energy to create fake accounts. As a result, preventing fake accounts from being created by increasing the complexity is likely to deter normal users from using the system long before this would become an effective deterrent for a determined adversary.

5.2.6 Similarity features

5.2.6.1 Subscriptions^s ∩ subscriptions^r

The "Subscriptions^s ∩ subscriptions^r" feature is measuring the Jaccard similarity index among two sets of subscriptions, of the sender and the receiver in relation; it normalizes dividing by the sum of the number of subscriptions of the receiver and the sum of subscriptions of the sender, minus the intersection of the two. Subscriptions should be private information, and thus neither sender nor receiver can be expected to simply

provide this information in a privacy-preserving OSN set up. In our data set, the resulting number of this feature is substantially less for messages classified as abusive (Figure 5.5), thus an adversary would attempt to increase the value. This requires the adversary to guess which subscriptions the victim may have, and then to create (or pretend to have made) the same subscriptions. We expect this to be costly, but not computationally hard: by watching the victim's public activity, it is likely possible to deduce quite a bit of information about the victim's subscriptions.

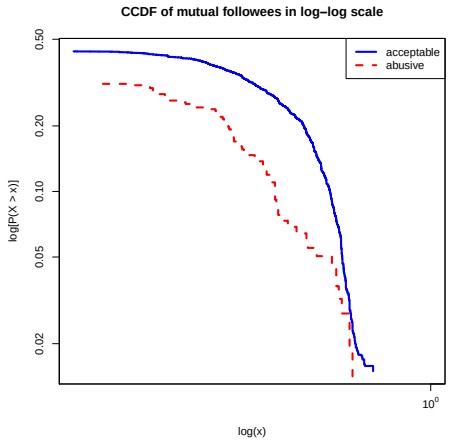


Figure 5.5: CCDF of subscription intersection.

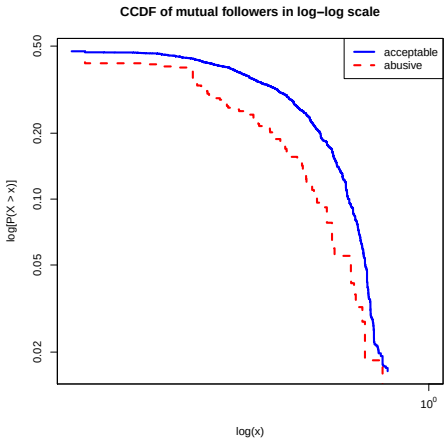


Figure 5.6: CCDF of subscribers intersection.

5.2.6.2 Subscribers^s ∩ subscribers^r

The "Subscribers^s ∩ subscribers^r" feature is measuring the Jaccard similarity index of the intersection among the set of subscribers of the sender and the receiver; it is again normalized accordingly to the definition of Jaccard index. Unlike their subscription set, a user cannot freely determine the set of their subscribers: A user needs to actually convince other users that they should subscribe to their public channel. We assume the channel owner knows its subscribers, and that the subscribers are willing to cryptographically sign a message saying that they are subscribed to the user's channel.

Later in chapter 6, we show how to create a protocol which uses signatures that allow Bob to prove to Alice that his input consists really of his subscribers. The tricky part there is that the identities of the subscribers are still sensitive private information, so we had to find particular signature scheme for our privacy-preserving computation of the overlap in subscriber sets. This is explained in Section 6.4 of chapter 6, where we introduce the use of BLS signatures for our privacy-preserving protocol. A key difference of our protocol when compared to using a CA for private set intersection (PSI-CA) [51] is that in our proposal the subscribers and not the CA provide the necessary signatures. This makes our protocol more efficient and we are also able to amortize the cost of these offline operations during later data minimization in chapter 7.

Assessment In our data set, the size of the subscriber set intersection is again substantially lower for messages classified as abusive (Figure 5.6), thus an adversary would attempt to increase the value. It is hard for an adversary to try to get the subscribers of the victim to subscribe to the adversary's feed, especially given that the subscribers are typically unknown to the adversary as subscriptions are private information.

It is again possible for the adversary to create fake accounts which subscribe to both the adversary and the victim. While these accounts may be relatively new, the "age of account" feature only considers the age of the sender's account, not the age of the accounts of subscribers. As with the "subscribers count" feature, proof-of-work techniques may increase the cost of this attack.

5.2.6.3 Subscribers^s ∩ Subscriptions^r

Finally, we consider the size of the Jaccard similarity index among the set intersection of subscribers of the sender and the subscriptions of the receiver, "subscribers^s ∩ Subscriptions^r". Figure 5.7 shows that, an adversary would try to increase the intersection of their subscribers (subscribers^s) with the subscriptions of the receiving victim (subscriptions^r). This feature is particularly interesting, as the sending attacker cannot easily influence set of subscriptions of the receiver, and will similarly have a hard time obtaining subscriptions from the user's to whom the victim is subscribed to. Unlike "subscribers ∩ subscribers", creating fake accounts is ineffective unless the receiver subscribes to these fake accounts.

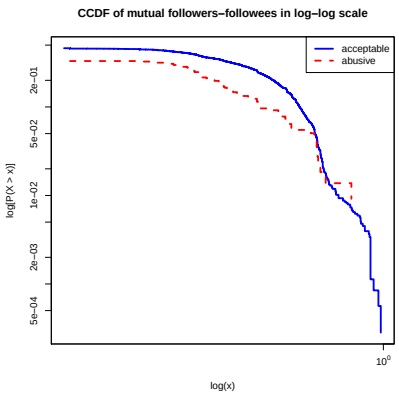


Figure 5.7: CCDF of subscribers^s-subscriptions^r intersection.

5.2.6.4 Subscriptions^s ∩ Subscribers^r

Naturally, computing the "subscriptions^s-subscribers^r" overlap is again dependent on privacy-sensitive information. However, the protocol from the previous section can be trivially adapted to the situation where Alice uses her set of subscriptions instead of her set of subscribers.

5.3 Evaluation

We have shown how to obtain some of the key features from our original abuse detection heuristic even in a privacy-preserving decentralized OSN. While many of the features can be inherently manipulated by a sophisticated adversary, others can be made robust even against strong and adaptive attacks.

We now evaluate the abuse detection system in the context of an adaptive adversary. In particular, we assume that the adversary can trivially adapt all of the account

properties of the sender's account, possibly create fake accounts (Sybils) and fake subscriptions, and is willing to make costly behavioral adaptations, e.g. by adapting the text of messages to avoid message properties as mentions' 5.2.2 and the frequency at which messages of any type are sent (Table 5.2). However, the adversary is unable to manipulate the age of accounts (by breaking the timeline service) or to break the cryptographic primitives used in the protocols presented in this paper.

Given this adversary model, only three features remain: the age of the account, the subscribers^s $\cap$ subscriptions^r Jaccard similarity index, and the "reciprocity" predicate. All other features need to be excluded from the classification algorithm's inputs, as we have to assume that the adversary will adapt to provide the worst-case input, thereby making abusive messages seem more benign.

We compare the results of the supervised learning classifiers presented in chapter 4 with a privacy-preserving, limited number of features as shown in Table 5.2. Tables 5.3 to 5.5 summarize results obtained using the same classifiers. The AB and SVM classifiers generally perform best in precision and recall than DT and RF respectively using Both ground truth datasets; however, the high variance means that this comparison may not generalize. The reduced feature set largely impacts the precision for abusive messages, cutting it by a bit more than a third in the best case scenario (AB), and

	Feature	Falsification/Adaptation	Crypto helps?
5.2.1	age of account	**hard**	yes
	# lists	trivial	n/a
	# favorited messages	costly	n/a
5.2.2	# mentions	costly	n/a
	# hashtags	costly	n/a
	# retweets	costly	n/a
	# mentions/# messages	n/a	
5.2.3	message reciprocity	**hard**	n/a
5.2.4	$\frac{\#\text{ messages}}{\text{age of account}}$	costly	yes
	$\frac{\#\text{ subscriptions}}{\text{age of account}}$	trivial	n/a
	$\frac{\#\text{ subscribers}}{\text{age of account}}$	possible	minimally
5.2.5	# subscribers	possible	minimally
	# subscriptions	trivial	n/a
	$\frac{\#\text{ subscriptions}}{\#\text{subscribers}}$	trivial	n/a
	$\frac{\#\text{subscribers}}{\#\text{subscriptions}}$	trivial	n/a
5.2.6.1	$\mathcal{J}$(subscriptions, subscriptions)	costly	w. privacy
5.2.6.2	$\mathcal{J}$(subscribers, subscribers)	possible	w. privacy
5.2.6.3	$\mathcal{J}$(subscribers^s, subscriptions^r)	**very hard**	yes
5.2.6.4	$\mathcal{J}$(subscriptions^s, subscribers^r)	possible	w. privacy

Table 5.2: Summary of how difficult it would be for an adversary to manipulate features.

Classifier	Metric	Overall	Only Acceptable	Only Abusive
DT	Precision	0.64 ± 0.09	0.98 ± 0.01	0.31 ± 0.19
	Recall	0.76 ± 0.14	0.94 ± 0.05	0.60 ± 0.31
	F-score	0.68 ± 0.10	0.96 ± 0.02	0.40 ± 0.18
RF	Precision	0.65 ± 0.12	0.98 ± 0.01	0.32 ± 0.24
	Recall	0.76 ± 0.08	0.94 ± 0.09	0.58 ± 0.19
	F-score	0.68 ± 0.12	0.96 ± 0.02	0.40 ± 0.22
ET	Precision	0.59 ± 0.09	0.98 ± 0.01	0.21 ± 0.17
	Recall	0.73 ± 0.15	0.89 ± 0.09	0.58 ± 0.31
	F-score	0.62 ± 0.12	0.93 ± 0.05	0.30 ± 0.20
GB	Precision	0.75 ± 0.13	0.97 ± 0.01	0.45 ± 0.20
	Recall	0.71 ± 0.12	0.98 ± 0.02	0.45 ± 0.25
	F-score	0.72 ± 0.10	0.98 ± 0.01	**0.47 ± 0.19**
AdaBoost	Precision	0.82 ± 0.30	0.97 ± 0.02	**0.67 ± 0.60**
	Recall	0.67 ± 0.18	0.99 ± 0.02	0.35 ± 0.37
	F-score	0.69 ± 0.16	0.35 ± 0.37	0.41 ± 0.32
SVM	Precision	0.60 ± 0.15	0.981 ± 0.026	0.218 ± 0.297
	Recall	0.70 ± 0.23	0.709 ± 0.567	**0.700 ± 0.389**
	F-score	0.54 ± 0.37	0.787 ± 0.424	0.288 ± 0.328
Ensemble	Precision	0.66 ± 0.14	0.981 ± 0.012	0.341 ± 0.283
	Recall	0.76 ± 0.15	0.942 ± 0.053	0.580 ± 0.306
	F-score	0.69 ± 0.14	0.961 ± 0.027	0.415 ± 0.253

Table 5.3: Classifiers trained with 5-fold cross validation and hard to forge features using Trollslayer *ground truth only*

Classifier	Metric	Overall	Only Acceptable	Only Abusive
DT	Precision	0.53 ± 0.02	0.931 ± 0.02	0.134 ± 0.028
	Recall	0.58 ± 0.07	0.714 ± 0.148	**0.452 ± 0.258**
	F-score	0.50 ± 0.04	0.806 ± 0.094	0.204 ± 0.050
RF	Precision	0.57 ± 0.04	0.930 ± 0.011	0.217 ± 0.073
	Recall	0.60 ± 0.06	0.883 ± 0.046	0.326 ± 0.109
	F-score	0.58 ± 0.05	0.906 ± 0.026	**0.259 ± 0.079**
ET	Precision	0.52 ± 0.03	0.923 ± 0.016	0.120 ± 0.041
	Recall	0.55 ± 0.07	0.716 ± 0.042	0.393 ± 0.111
	F-score	0.49 ± 0.04	0.806 ± 0.032	0.184 ± 0.060
GB	Precision	0.59 ± 0.10	0.919 ± 0.01	**0.253 ± 0.19**
	Recall	0.55 ± 0.06	0.957 ± 0.022	0.141 ± 0.098
	F-score	0.56 ± 0.07	0.938 ± 0.014	0.179 ± 0.123
AdaBoost	Precision	0.58 ± 0.25	0.91 ± 0.004	0.24 ± 0.50
	Recall	0.51 ± 0.03	0.995 ± 0.01	0.03 ± 0.05
	F-score	0.50 ± 0.05	0.95 ± 0.005	0.05 ± 0.10
SVM	Precision	0.55 ± 0.03	0.930 ± 0.010	0.160 ± 0.047
	Recall	0.59 ± 0.05	0.800 ± 0.044	0.385 ± 0.076
	F-score	0.54 ± 0.04	0.860 ± 0.028	0.226 ± 0.058
Ensemble	Precision	0.57 ± 0.04	0.930 ± 0.008	0.217 ± 0.080
	Recall	0.60 ± 0.04	0.885 ± 0.035	0.319 ± 0.059
	F-score	0.58 ± 0.05	0.907 ± 0.022	0.257 ± 0.076

Table 5.4: Classifiers trained with 5-fold cross validation and hard to forge features with Crowdflower *ground truth*

Classifier	Metric	Overall	Only Acceptable	Only Abusive
DT	Precision	0.55 ± 0.02	0.959 ± 0.012	0.132 ± 0.032
	Recall	0.64 ± 0.07	0.775 ± 0.040	0.511 ± 0.155
	F-score	0.53 ± 0.03	0.857 ± 0.024	0.210 ± 0.053
RF	Precision	0.59 ± 0.07	0.954 ± 0.008	0.235 ± 0.130
	Recall	0.63 ± 0.07	0.924 ± 0.033	0.333 ± 0.116
	F-score	0.61 ± 0.07	0.938 ± 0.020	0.273 ± 0.121
ET	Precision	0.54 ± 0.03	0.954 ± 0.013	0.122 ± 0.040
	Recall	0.61 ± 0.08	0.789 ± 0.014	0.437 ± 0.169
	F-score	0.53 ± 0.03	0.864 ± 0.010	0.191 ± 0.065
GB	Precision	0.61 ± 0.10	0.946 ± 0.012	0.274 ± 0.194
	Recall	0.58 ± 0.08	0.967 ± 0.020	0.189 ± 0.159
	F-score	0.59 ± 0.09	0.957 ± 0.010	0.220 ± 0.167
AdaBoost	Precision	0.67 ± 0.23	0.940 ± 0.007	**0.390 ± 0.459**
	Recall	0.53 ± 0.04	0.993 ± 0.011	0.063 ± 0.079
	F-score	0.54 ± 0.06	0.063 ± 0.079	0.105 ± 0.125
SVM	Precision	0.51 ± 0.04	0.957 ± 0.046	0.072 ± 0.026
	Recall	0.55 ± 0.13	0.344 ± 0.061	**0.758 ± 0.318**
	F-score	0.32 ± 0.02	0.505 ± 0.058	0.131 ± 0.049
Ensemble	Precision	0.60 ± 0.06	0.955 ± 0.008	0.250 ± 0.106
	Recall	0.64 ± 0.06	0.927 ± 0.025	0.354 ± 0.115
	F-score	0.62 ± 0.06	0.941 ± 0.016	**0.292 ± 0.109**

Table 5.5: Classifiers trained with 5-fold cross validation and hard to forge features with Both *as ground truth*

approximately half in the worst (e.g., ET) when comparing Table 4.4 with Table 5.5. However, even with this strong adaptive adversary, the GB classifier performs at slightly more than half the recall and nearly the same precision for abusive messages than before. Note that Table 5.5 shows the result on running the same set of features with annotations from both, Trollslayer and Crowdflower, which shows consistent results among classifiers. Figures 5.8 to 5.11 provide the ROC curve, precision-recall (P-R) curves and the confusion matrix (CM).

5.3.1 Feature relative importance

In terms of relative importance (RI), the age of account has always the highest importance (DT: 0.64%, RF: 0.59%, ET: 0.44%, GB: 0.80%) and the boolean feature of "reciprocity" ranks pretty low in importance (DT: 0.00%, RF: 0.07%, ET: 0.27%, GB: 0.01%).

We see in Figure 5.14 that compared to learning without privacy, some of the graph-metadata based features we first presented in chapter 4 are still useful and even rank high in some of our classifiers, meaning they can potentially be able to aid in the problem of privacy-preserving abuse detection if we are apply the privacy-preserving protocol of chapter 6 such that no sensitive information about users or their social graphs is required to the classifiers.

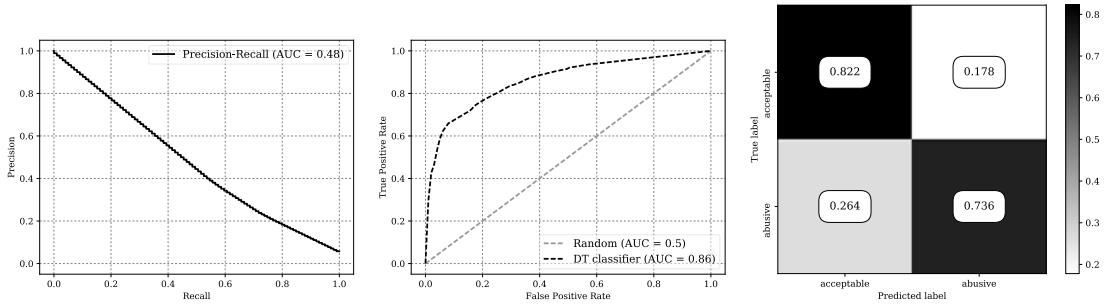
The graph-metadata based feature "mutual_followers_followees", which is 5.2.6.3 in Table 5.2 of privacy-preserving features, ranks consistently as second best among all classifiers except the extra trees (ET). This result indicates that such metadata-based graph features can play a promising role when using supervised machine learning methods for Sybil-defenses in OSN models as Twitter.

5.4 Summary of results

Our results show how to combine local knowledge with privacy-preserving protocols to detect abuse in future decentralized online social networks. Given an adaptive adversary that would be able to manipulate most features we propose in our supervised learning approach, it is surprising that with just three features resistant to adversarial manipulation, the algorithms still provide useful classifications classifications for timeline construction.

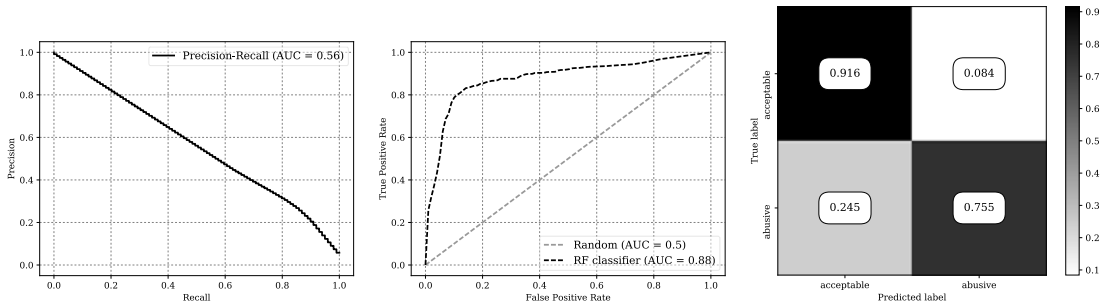
Many of the features we originally considered could not be effectively secured against an adversary creating fake accounts and fake subscriptions. It might be possible to use some of these features if we additionally considered the age of the accounts: given a time-stamping service, the adversary may be able to create fake accounts, but it would be very hard to back-date them. Combining timestamped public keys with the privacy-preserving set intersection protocols is thus an interesting open problem for future work.

That said, even if we included some of these features that could be secured, the performance of the privacy-preserving classifiers did not significantly improve. The more substantial gains seem to depend on features involving basic account properties



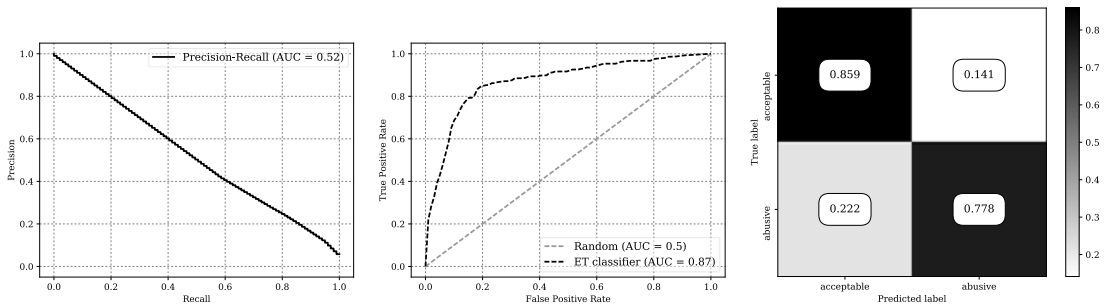
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 5.8: Evaluation for decision trees (with strong adaptive adversary)



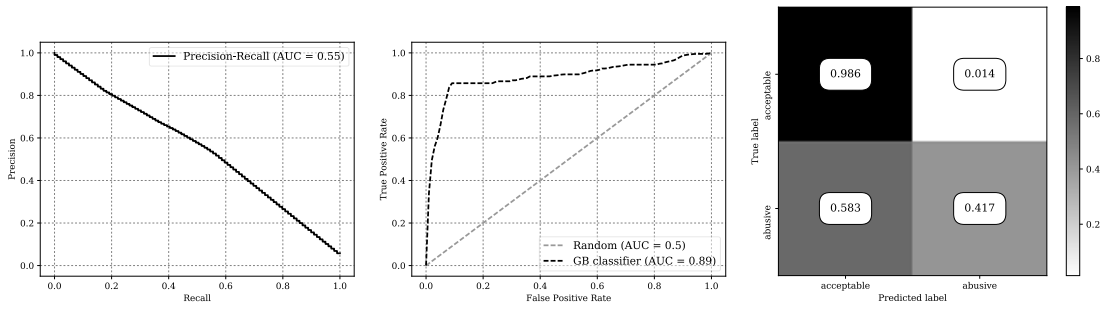
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 5.9: Evaluation for random forest (with strong adaptive adversary)



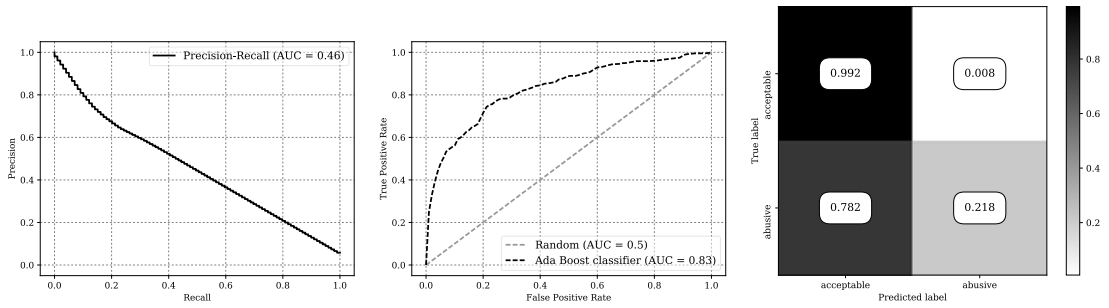
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 5.10: Evaluation for extra trees (with strong adaptive adversary)



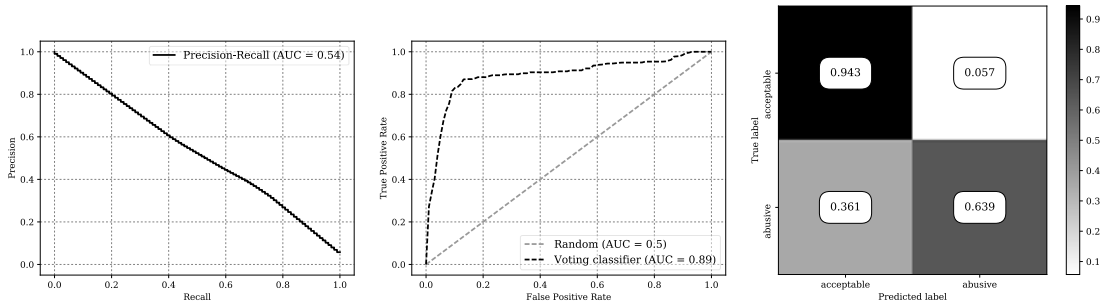
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 5.11: Evaluation for gradient boosting (with strong adaptive adversary)



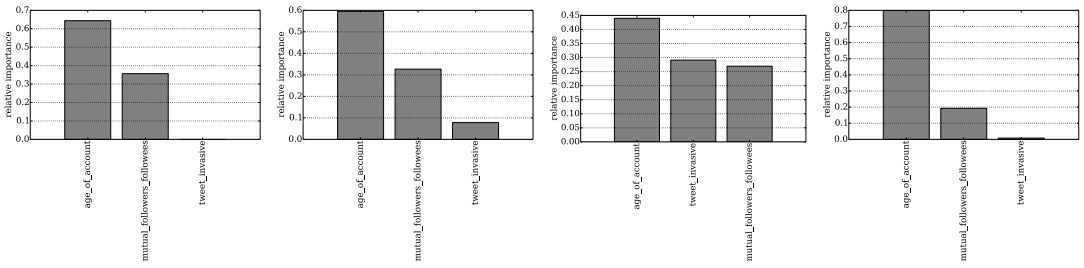
(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 5.12: Evaluation for adaBoost (with strong adaptive adversary)



(a) P-R curve *(b) ROC curve* *(c) CM*

Figure 5.13: Evaluation for ensemble voting(with strong adaptive adversary)



(a) RI of DT *(b) RI of RF* *(c) RI of ET* *(d) RI of GB*

Figure 5.14: RI of classifiers with privacy

and sender behavior which fundamentally cannot be secured against an adaptive adversary as they are under full control of the adversary. Real-world deployments will thus have to figure out whether including those features would help (because real-world adversaries are not that adaptive) or hurt (because real-world adversaries would adapt to use these features to their advantage).

We envision that future decentralized privacy-preserving OSN will use the sort of abuse classifiers discussed here as part of ranking messages in the user's timeline, not for binary filtering of messages for an inbox. By timeline, we mean any interface that displays short message summaries ordered so that users never feel the desire to read all listed messages. After browsing only a brief portion of their timeline, a user should firstly feel they have skimmed enough summaries to be up to date on any topics about which they consult the application, and secondly not have spent time on matters they might later regret, such as responding to abusive messages.

We have treated abuse as a binary classification problem in this article, but actually one would prefer the different features to report back a numerical risk score for timeline construction. As a result, the concerns around bias one encounters with binary classifiers [95] seem unnecessary here. Instead, actual timeline constructions requires integrating an array of features with both positive and negative aspects.

In terms of concrete deployments, we envision that future OSN would include a decision tree baked into the code and not expect users to train their own classifier. This will simplify the deployed software, improve usability and avoid users running expensive training algorithms.

6

Privacy-preserving set intersection cardinality protocol

Contents

This chapter presents the design of a privacy-preserving protocol we employ for calculating the cardinality of set intersection based features in our abuse classifier; namely the features that involve the combination of subscriptions and subscribers of two parties to asess the common number of contacts among them.

The chapter first introduces the basic straw-man design of the protocol in 6.1.1, secondly presents a cut & choose variation of the protocol that prevents a malicious participant cheating 6.3 and finally concludes with for a privacy-preserving scheme based in BLS signatures, which we argue is compatible with the blinding operations we provide in the initial two phases of protocol design and hence enable the use bilinear maps into our protocol 6.4.

6.1 Introduction

Data privacy is known to be an increasing concern for users, as several works already show a conflicting trade-off among privacy and data ownership [112, 79, 129, 47]. In particular, in [103] authors investigated the relationship among data privacy and utility, namely maximum variance. For doing so, they extract statistical graph utility measurements with techniques such as degree-based statistics (number of edges, average degree, maximum degree, degree variance and power laws), shortest-path bases statistics (average distance, effective diameter, connectivity length, diameter) and clustering coefficient. Their approach claims to outperforms(k,E)-obfuscation when evaluated against some datasets from the SNAP (Stanford Large Network Dataset Collection)[1] repository, in particular they used Youtube, DBLP, and Amazon.

In particular, the use for privacy-preserving protocols is common in many research areas, from healthcare to data analytics. Such protocols are the basis of sensitive systems or tasks that need to preserve the privacy of individuals, whether is medical records [109], advanced passenger traveling information, etc. In these settings, the goal is to have privacy-preserving techniques working without having to trade with anything like ownership, effectively providing satisfactory results for the task at hand.

However, with pervasive increasing levels of surveillance [2] and censorship [3] in social media, privacy is a pressing issue for citizens, who see how their metadata is owned by social media providers and other actors, ranging from authorities to malicious participants [66].

In this work we design a privacy-preserving protocol that aids to preserve the privacy of social graph metadata based features when detecting abuse, thus enabling its use in decentralized settings where the potential victim (recipient) may need to perform machine assisted classification of abusive content locally. The protocol is useful in decentralized OSN settings, but a limitation of this work is that it is not applicable on undirected social networks such as Facebook.

6.1.1 Straw-man

First we present a naive, straw-man design of our protocol. We show the exchange of messages in the straw-man version of the protocol in Figure 6.1 that we later evolve into a final version that uses signatures for subscription verification. Obviously, this first version of the protocol is not robust against confirmation attacks (addressed by the cut and choose version) where a malicious participant tries to guess some of the elements in the set of the other party or tampering attacks (addressed by the signatures version) where a party is able to place sock-puppet accounts among the subscriptions or subscribers list of the other party to compute a higher intersection value.

Suppose each user has a private key c_i and the corresponding public key is in $C_i := g^{c_i}$, where g is some generator. Let $\mathcal{A}$ be the set of public keys representing

[1] `http://snap.stanford.edu/data/`
[2] `https://theintercept.com/2016/10/23/endace-mass-surveillance-gchq-governments/`
[3] `https://ooni.torproject.org/`

Alice's subscriptions and $\mathcal{B}$ be the set of keys representing Bob's subscriptions. Fix a cryptographic hash function h. For any set Z, define $Z' := \{h(x)|x \in Z\}$. We also assume a fixed system security parameter $\kappa \geq 1$ has been agreed upon.

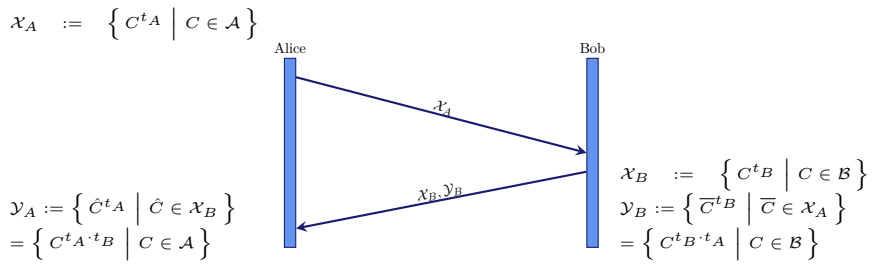


Figure 6.1: The protocol part 1

Diffie-Hellman key exchange uses commutativity of exponentiation: $A^b = (g^a)^b = (g^b)^a = B^a (\mathrm{mod} p)$

Alice can get $|\mathcal{Y}_A \cap \mathcal{Y}_B|$ within linear cost

6.2 Background

6.2.1 Private Set Intersection protocols

Private Set Intersection protocols aim to compute the cardinality of the intersection without revealing the elements of each party involved in for such result. This provides a number of applications that are beyond just the field of computer science, and that span many real use cases in the literature.

Authorized PSI protocols rely on a non-malicious, authorized (signed) and mutually trusted authority by both parties [54]. Our research follows previous works in PSI [51] that use a Diffie-Hellman (DH) key exchange but unlike ours, assumes a Random Oracle model in the honest but curious adversary model.

6.2.2 The Boneh-Lynn-Shacham (BLS) signature scheme

We use BLS signatures in the second part of the protocol because they are compatible with the blinding we perform and offer the properties we want in the protocol. Note the bilinear maps should not be symmetric in our case, as otherwise other types of attacks are presumably possible in the scheme we present even though we do not explor in detail provable security implications due to such protocol design.

We first outline the BLS signature scheme [23], which begins with a Gap co-Diffie-Hellman group pair (G_1, G_2) of order p with an efficiently-computable bilinear map $e\colon G_1 \times G_2 \to G_T$, a generator g_2 of G_2, and a cryptographic hash function $H : \{0,1\}^* \to G_1$.

In the BLS scheme, a private key consists of a scalar $c \in \mathbb{Z}/p\mathbb{Z}$, while the corresponding public key is $C := g_2^c$, and a signature on a message m by C is $\sigma := H(m)^c$.

A signature σ is verified by checking that $e(H(m), C) = e(\sigma, g_2)$. If $\sigma = H(m)^c$ then this holds by bilinearity of e.

6.3 Set intersection cardinality with privacy

We provide a new privacy-preserving protocol to compute the size of a set intersection, which is a variation of the PSI-CA protocol of [51]

Suppose Alice wishes to know $n := |\mathcal{A} \cap \mathcal{B}|$. First, she generates an ephemeral private scalar $t_A \in \mathbb{Z}/p\mathbb{Z}$ and sends Bob the following,

$$\mathcal{X}_A := \mathtt{sort}\left[C^{t_A} \;\middle|\; C \in \mathcal{A} \right] \tag{6.1}$$

Second, Bob picks ephemeral private scalars $t_{B,j} \in \mathbb{Z}/p\mathbb{Z}$ for $j \in 1, \ldots, \kappa$ and computes

$$\mathcal{X}_{B,j} := \mathtt{sort}\left[C^{t_{B,j}} \;\middle|\; C \in \mathcal{B} \right] \tag{6.2}$$

$$\mathcal{Y}_{B,j} := \mathtt{sort}\left[\overline{C}^{t_{B,j}} \;\middle|\; \overline{C} \in \mathcal{X}_A \right] \tag{6.3}$$

He then sends commitments $\mathcal{Y}'_{B,i}$ for $i \in 1, \ldots, \kappa$ to Alice. Third, Alice picks a non-empty random $J \subseteq \{1, \ldots, \kappa\}$ and sends J to Bob.

Fourth, Bob sends Alice his scalar $t_{B,j}$ for $j \notin J$, as well as $\mathcal{X}_{B,j}$ for $j \in J$. Fifth, Alice checks the $t_{B,j}$ matches the commitment $\mathcal{Y}'_{B,j}$ for $j \notin J$. She also verifies the commitment to $\mathcal{X}_{B,j}$ for $j \in J$. She then computes for $j \in J$:

$$\mathcal{Y}_{A,j} := \left\{ \hat{C}^{t_A} \;\middle|\; \hat{C} \in \mathcal{X}_{B,j} \right\} \tag{6.4}$$

Finally, and for $j \in J$, Alice computes the result of $|\mathcal{Y}'_{A,j} \cap \mathcal{Y}'_{B,j}|$, checking that all $|J| \geq 1$ values agree.

We have presented the above protocol using standard cut and choose terminology, partially for clarity, but also because one standard optimizations seems like a too pessimistic in our case. As usual, one could eliminate the middle round trip using standard techniques.

Instead of steps two and three, Bob computes J to be the hash of Alice's initial message and the commitments he would send in step two. Now Bob sends his data from rounds two and four in one round. Alice verifies his choice of J using his commitments and verifies his commitments as before. Assuming one chooses κ to provide cryptographic security, Bob cannot realistically choose malicious values that survive verification.

We observe however that κ need not be chosen to provide cryptographic security because an attacker gains relatively little value from falsifying a larger intersection. We therefore envision using the four trip version with κ large enough so that faking an introduction is expensive but not impractical.

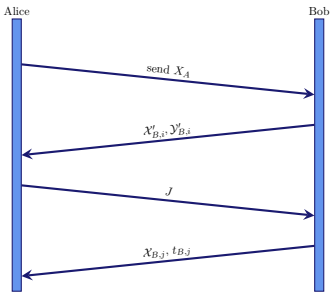


Figure 6.2: The protocol part 2

We note that this first part of our privacy-preserving protocol also applies for computing the overlap between the sender's subscriptions and the receiver's subscribers. In this version it is easier for a malicious adversary to manipulate the feature value, as the adversary can simply create fake accounts to subscribe to the victim, and it is trivial for the adversary to subscribe to these fake accounts. As a result, the adversary can increase the overlap for the "-subscriptions^s-subscribers^r" feature limited only by the number of fake accounts. As with the "number of subscribers", this attack can again be slightly mitigated by making account creation expensive. The cut-and-choose version of the protocol is depicted in Figure 6.2.

6.4 Set intersection cardinality with privacy and signatures

We again define $Z' := \{h(x) | x \in Z\}$ whenever Z is some set under discussion, and assume a fixed system security parameter $\kappa \geq 1$ has been agreed upon. Each participant is identified by a public key pair $C = g_2^c$ for the BLS signature scheme. Each participant N has a set of subscribers consisting of tuples $(C, \sigma_{N,C})$ where $\sigma_{N,C} := H(N, \texttt{date})^c$ is a BLS signature affirming that $C = g_2^c$ was subscribed to N until some expiration `date`, the specifics of which depend on the application. We envision these signatures being provided in advance so that subscribers of a participant need not to be online when the other one is willing to run the protocol to compute the set intersection features.

Suppose Alice wishes to know $n := |\mathcal{A} \cap \mathcal{B}|$. First, she generates an ephemeral private scalar $t_A \in \mathbb{Z}/p\mathbb{Z}$ and sends Bob,

$$\mathcal{X}_A := \texttt{sort}\left[C^{t_A} \;\middle|\; (C, \sigma_{A,C}) \in \mathcal{A} \right] \tag{6.5}$$

Second, Bob picks ephemeral private scalars $t_{B,j} \in \mathbb{Z}/p\mathbb{Z}$ for $j \in 1, \ldots, \kappa$ and computes,

$$\mathcal{X}_{B,j} := \texttt{sort}\left[(C^{t_{B,j}} \;\middle|\; (C, \sigma_{B,C}) \in \mathcal{B} \right] \tag{6.6}$$

$$\mathcal{Y}_{B,j} := \texttt{sort}\left[\overline{C}^{t_{B,j}} \;\middle|\; \overline{C} \in \mathcal{X}_A \right] \tag{6.7}$$

For $j \in 1, \ldots, \kappa$ and $(C, \cdot) \in \mathcal{A}$, Bob picks more ephemeral private scalars $s_{j,C} \in \mathbb{Z}/p\mathbb{Z}$ and computes $S_{j,C} := g_2^{s_{j,C}}$ and $\sigma_{B,S_{j,C}} := H_1(B||\texttt{date})^{s_{j,C}}$. Let $\pi_{B,j}$ denote the permutation applied by the `sort` for $\mathcal{X}_{B,j}$. Bob also computes

$$\mathcal{U}_{B,j} := \pi_{B,j} \left[C^{t_{B,j}} S_{j,C} \;\middle|\; (C, \sigma_{B,C}) \in \mathcal{B} \right] \tag{6.8}$$

$$\mathcal{V}_{B,j} := \pi_{B,j} \left[\sigma_{B,C}^{t_{B,j}} \sigma_{B,S_{j,C}} \;\middle|\; (C, \sigma_{B,C}) \in \mathcal{B} \right] \tag{6.9}$$

In the second message exchange, Bob sends to Alice the commitments

$$\mathcal{Y}'_{B,i}, \mathcal{V}'_{B,i}, \mathcal{U}_{B,i} \;\forall\; i \in 1, \ldots, \kappa \tag{6.10}$$

Third, Alice picks a non-empty random $J \subseteq \{1, \ldots, \kappa\}$ and sends J to Bob.

Fourth, Bob sends Alice his scalar $t_{B,j}$ and $\mathcal{V}_{B,j}$ for $j \notin J$, as well as $\mathcal{X}_{B,j}$ and his scalars $\pi_{B,j}[s_{j,C} | (C, \sigma_{B,C}) \in \mathcal{B}]$ for $j \in J$.

Fifth, Alice verifies Bob's commitments for $j \notin J$ as follows,

- that the $t_{B,j}$ matches the commitment $\mathcal{Y}'_{B,j}$ by computing $\mathcal{Y}_{B,j}$ herself.
- that $\mathcal{V}_{B,j}$ matches the $\mathcal{V}'_{B,j}$.
- and verifies the signatures in $\mathcal{V}_{B,j}$ using $\mathcal{U}_{B,j}$ as public keys.

These signatures validate because we employ the BLS pairing based signature scheme where:

$$\begin{aligned} e(H_1(B||\texttt{date}), C^{t_{B,j}} S_{j,C}) &= e(H_1(B||\texttt{date}), g_2)^{t_{B,j}c + s_{j,C}} \\ &= e(\sigma_{B,C}^{t_{B,j}} \sigma_{B,S_{j,C}}, g_2) \end{aligned}$$

For $j \in J$, Alice computes

$$\mathcal{Y}_{A,j} := \left\{ \hat{C}^{t_A} \;\middle|\; \hat{C} \in \mathcal{X}_{B,j} \right\} \tag{6.11}$$

Finally, she obtains the result from $|\mathcal{Y}'_{A,j} \cap \mathcal{Y}'_{B,j}| = n$ for $j \in J$, checking that all $|J| \geq 1$ values agree.

As in the version without signatures, there is a two round-trip version of this protocol in which both parties compute J from Alice's initial message and Bob's commitments. We do not envision using it though because doing so requires κ to be of cryptographic size.

A high level view of the protocol flow is depicted in Figure 6.3.

An attack on this blinded signature scheme translates into an attack on the underlying BLS signature scheme. If Bob tries to manipulate the overlap to increase it, the cut-and-choose part detects this with probability $1 : 2^\kappa$, thus terminating.

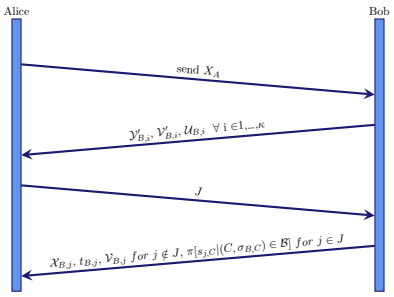


Figure 6.3: Protocol message exchange

6.5 Protocol security

We briefly give an intuition behind the design of our protocols in Section 6.3 and Section 6.4 for computing the number of common subscribers and subscriptions respectively.

As above, we suppose each user has a private key c_i and the corresponding public key is $C_i := g^{c_i}$ where g is the group generator. Let $\mathcal{A}$ be the set of public keys representing Alice's subscriptions and $\mathcal{B}$ be the set of keys representing Bob's subscriptions.

We first consider the simple straw-man protocol: Alice begins by creating an ephemeral private scalar $t_A \in \mathbb{Z}/p\mathbb{Z}$ and sending Bob her blinded friends lists

$$\mathcal{X}_{\text{Alice}} := \left[C^{t_A} \;\middle|\; C \in \mathcal{A} \right] \tag{6.12}$$

Second, Bob replies by creating an ephemeral private scalar $t_B \in \mathbb{Z}/p\mathbb{Z}$ and sending Alice his blinded friends lists and her list reblinded

$$\mathcal{X}_{\text{Bob}} := \left[C^{t_B} \;\middle|\; C \in \mathcal{B} \right] \tag{6.13}$$

$$\mathcal{Y}_{\text{Bob}} := \left[\overline{C}^{t_B} \;\middle|\; \overline{C} \in \mathcal{X}_{\text{Alice}} \right] \tag{6.14}$$

Alice then computes

$$\mathcal{Y}_{\text{Alice}} := \left\{ \hat{C}^{x_A} \;\middle|\; \hat{C} \in \mathcal{X}_{\text{Bob}} \right\} \tag{6.15}$$

, finally obtaining $|\mathcal{Y}_A \cap \mathcal{Y}_B| = n$.

6.5.1 Attacks

We address several types attacks from our protocol, mainly confirmation attacks in the protocol version of Section 6.3. First, suppose Bob creates sock puppets accounts in pairs S_i and S_i^k where Bob knows only k. Then he also knows

$$\{S_i^k\} = \mathcal{X}_{\text{Alice}} \cap \left\{ \overline{C}^k \;\middle|\; \overline{C} \in \mathcal{X}_{\text{Alice}} \right\} \tag{6.16}$$

As he knows these elements' locations in $\mathcal{X}_{\text{Alice}}$, he can deduce the number of Alice's contacts C with $S_i < C < S_{i+1}$ for all i assuming Alice's original contact list $\mathcal{A}$ is sorted by some ordering function $<$. This would allow Bob to obtain information about Alice's contact list. Alice defeats this by sorting her list after blinding, which produces a random ordering by our discrete logarithm assumptions. For the same reasons, Bob must sort $\mathcal{X}_{\text{Bob}}$ as well.

Second, assume Alice or Bob can place sock puppet accounts among the other's subscriptions or subscribers lists. If say Alice inserts C^k in Bob's list $op\mathcal{B}$, then she can identify if C lies in $\mathcal{X}_{\text{Bob}}$.

$$C \in \mathcal{B} \iff \mathcal{X}_{\text{Bob}} \cap \left\{ \overline{C}^k \;\middle|\; \overline{C} \in \mathcal{X}_{\text{Bob}} \right\} \neq \emptyset \tag{6.17}$$

We block this attack this by requiring a signature from contacts joining subscription or subscriber lists. Alice and Bob can still place sock puppets, but only ones for whom they know the private key, so then this attack no longer yields information.

Third, suppose Alice places C and a sock puppet C^k into her own list where she only knows the marker k. Given $\mathcal{Y}_{\text{Bob}}$ she could then identify the pair of elements $\overline{C}, \overline{C}^k \in \mathcal{Y}_{\text{Bob}}$, and then check if $\overline{C} \in \mathcal{Y}_{\text{Alice}}$ to test if $C \in \mathcal{B}$. We therefore only send the hashed $\mathcal{Y}'_{\text{Bob}}$, never $\mathcal{Y}_{\text{Bob}}$ itself.

Forth, suppose Bob correctly guesses the C corresponding to any $\overline{C} = C^{t_A}$ from $\mathcal{X}_{\text{Alice}}$, then Bob can choose $K \subset \mathbb{Z}/p\mathbb{Z}$ and send fakes using $\mathcal{X} = [C^k | k \in K] \cup \{\cdots\}$ and $\mathcal{Y} = [\overline{C}^k | k \in K] \cup \{\cdots\}$, so that Alice computes $n = |K|$. To defend against successful guessing, Alice and Bob can pad their lists if they are short. At first blush, we might ask if such attacks could be detected using padding with tag contacts similar to the first attack, but our restriction from sending $\mathcal{Y}_{\text{Bob}}$ prohibits this.

Finally, we encounter a new form of attack in the protocol due to the use of pairing based cryptography, where there is no decisional Diffie-Hellman (DDH) assumption because $e(g^a, g^b) = e(g^{ab}, g)$. Hence, only the far weaker computational Diffie-Hellman assumption remains tenable. Note that even if DDH does not hold in the source group, 2-DLIN (Decision Linear Assumption) has been proposed in [22].

It follows that, if Alice knows both σ and σ^{t_A} for some σ, such as our BLS signatures, then she can compute $e(C^t, \sigma) = e(C', \sigma^t)$ as C' runs over all known contacts, thereby revealing Bob's contact list. Again, we avoid this with standard cut and choose in our protocol.

6.6 Protocol efficiency

This section describes the computational complexity and communication overhead of our protocol. Our protocol design assumes users can collect public keys and possibly signatures during subscription operations. Assuming that is possible and such information is available, we thus avoid protocol participants having to engage into more costly communication at protocol runtime. Therefore, computing a privacy-preserving set intersection using our protocol only involves two round trips, or four message ex-

changes. This is an important aspect that characterizes the computational complexity and bandwidth overhead of our protocol.

Each of these only consumes $\mathcal{O}(\kappa \cdot n)$ computation time and bandwidth where n is the number of contacts, and κ is our security paramater, except for choosing J which is constant, and the initiation step, which consumes only $\mathcal{O}(n)$ computation time and bandwidth. We feel this cost is reasonable because κ need not to be chosen to have cryptographic size, as noted above, and the constant terms hidden by the Big $\mathcal{O}$ notation consists of a few public key operations. There remains some freedom for optimizing bandwidth usage during the commitment phase however.

Computational complexity Alice needs to blind her contact list to send it to Bob, namely $\mathcal{X}_A$. Such operation only incurs linear cost, which is directly proportional to the number of contacts $|A|$ in her list, namely the public keys $C_i \in \mathcal{A}$. The cost of this operation is $\mathcal{O}(pk \cdot |A|)$, where pk is the actual size of the keys being blinded.

Likewise, Bob uses his private scalar $t_{B,j}$ for blinding his list $\mathcal{A}$ and re-blinding the list he receives from Alice. Therefore it is trivial to see that the magnitude of such computations is equal for both, Bob and Alice.

Assuming v and w represent the # of contacts in Alice's and Bob's sets respectively, the final computational complexity cost is $\mathcal{O}(v + w)$ for Alice and $\mathcal{O}(w + v)$ for Bob.

Communication complexity We also show the bandwidth complexity for each party, Alice and Bob. First step for Alice simply involves sending her blinded list of contacts $\mathcal{X}_A$. When Bob receives Alice's blinded list he responds with commitments $\mathcal{Y}'_{B,i}$, $\mathcal{V}'_{B,i}$ for $i \in 1, \ldots, \kappa$, along with $\mathcal{U}_{B,i}$. Next Alice picks the non-empty random J and sends it to Bob, who replies to Alice with his blinded list of contacts $\mathcal{X}_{B,j}$, his scalars $t_{B,j}$ and $\pi_{B,j} := \pi[s_{j,C}|(C, \sigma_{B,C}) \in \mathcal{B}]$ for $j \in J$ as well as the commitments $\mathcal{V}_{B,j}$ for $j \notin J$.

Assuming v and w represent the # of contacts in Alice's and Bob's sets respectively, the final bandwidth overhead or cost is, $\mathcal{O}(v+w)$ for Alice and $\mathcal{O}(w+v)$ for Bob, which remains of linear order then.

Comparing existing PSI protocols Finally, in Table 6.1, we showcase the complexity and requirements of each of the PSI protocols that exist in the literature. Most of the contents are taken from Table 2 (performance comparison of PSI and APSI protocols) in [54], so that we can compare our PSI protocol with theirs.

6.7 Summary of results

Our protocol, similarly to the PSI version of DeCristofaro differs from APSI protocols [52] in that it does not operate in or need an RSA setting. On the other hand, unlike DeCristofaro we do not use product of hashes to defend against malicious opponents.

Protocol	**Model**	**Adv**	**Com**	**Server ops**	**Client ops**	**Mod**
PSI in [76]	Std	Mal	$\mathcal{O}(v+w)$	$\mathcal{O}(v+w(loglogv))$	$\mathcal{O}(v+v)$	1024
PSI in [85]	Std	Mal	$\mathcal{O}(v+w)$	$\mathcal{O}(wv)$	$\mathcal{O}(wv)$	2048
PSI in [82]	Std	HbC	$\mathcal{O}(v+w)$	$\mathcal{O}(v)$	$\mathcal{O}(v)$	1024/2048
PSI in [83]	ROM	Mal	$\mathcal{O}(v+w)$	$\mathcal{O}(v)$	$\mathcal{O}(v)$	1024
PSI in Fig.3 of [54]	ROM	HbC	$\mathcal{O}(v+w)$	$\mathcal{O}(v)$	$\mathcal{O}(v)$	1024
PSI in Fig.4 of [54]	ROM	HbC	$\mathcal{O}(v+w)$	$\mathcal{O}(v)$	$\mathcal{O}(v)$	1024
Our PSI protocol in [69]	Std	Mal	$\mathcal{O}(v+w)$	$\mathcal{O}(v)$	$\mathcal{O}(v)$	256

Table 6.1: Performance comparison of PSI protocols

In terms of optimizations, by obviating an RSA setting DeCristofaro [54] reduces the size of the mod value in his assessment of PSI protocol complexity due to the use of shorter exponents. We do not see any immediate advantage from paying attention to such detail for our complexity analysis, as in any case their protocol remains of linear order in bandwidth and computation.

Unlike in APSI protocols, the inputs in a PSI protocol might be arbitrarily chosen. To mitigate malicious adversaries, DeCristofaro [54] mentions the use of product of hashes, and the need for a security proof in that regard. We protect our protocol against malicious adversaries by using a signature scheme as BLS, which together with our blinding and cut & choose ensures each party involved in the protocol proves statements made about a third party, namely proving the subscriptions and/or subscribers.

7 Data minimization

Contents

This chapter applies a data minimization technique that can compute our set of features involving neighborhood metadata among two parties in a more efficient manner. The method is compatible with our privacy-preserving protocol in the previous chapter, thus rendering its implementation much more efficient in practice.

To the best of our knowlege, this is the first work focused on efficient, privacy-preserving computation of social graph features for assessing abusive behavior in OSN deployments. In addition, we benefit from a design that is compatible with decentralized platforms where a limited amount of metadata is available and privacy of feature computation is desirable. The final results show that we can obtain the same classification results by approximating similarity features based in social graph metadata, which will greatly improve time requirements of our privacy-preserving protocol in Big Data settings.

7.1 Introduction

Artificial intelligence (AI) describes the term rational as "maximizing" an expected utility. In SNAM, the concept of rationality is expressed in terms of the amount of data available. That is not strictly necessary, but widely accepted. For instance, to analyze the latest social trends in a modern OSN, common wisdom says we can make better decisions with more data (a.k.a. as Big Data). However, that unfolds the issue of exposing too much personal or sensitive data to private parties as well as public agencies and governments. The goal of this chapter is to analyze the cost of obtaining such metadata in a centralized OSN and integrate our privacy-preserving protocol from the previous chapter with an scheme that computes a fingerprint to represent social graph metadata, often the most previous to privacy sensitive users.

Data "minimization" consists in disclosing or opening only the necessary amount of information to the public or user in order to still have utility from it. We will explore the inter-relationship among the level of privacy in data (anonymity) and its utility predicting abusive behavior as compared to a ground-truth database. That will support verification and performance of the abuse detection algorithm when changing privacy requirements. Furthermore, this may be provide useful insights applying such technique not only to Twitter, but other decentralized OSNs, where the level of privacy is supposed to be inherently higher.

7.2 The cost of crawling Twitter

Early in the thesis we obtained social graph metadata performing several BFS crawls over Twitter. However, this is a time consuming process, especially when it comes to social graph metadata. This is because Twitter needs to throttle requests to their API (1% sample) in order to avoid aggressive crawling or issues such as Distributed Denial of Service attacks (DDoS). Limiting the crawling speed, Twitter significantly reduces the amount of metadata that can be collected for research experiments based in OSN; thus having to wait long periods of time to crawl an small portion of the Twitter social graph. Given that their API limits, see Table 7.1, are based in requests per minute, this cost can be modeled as a function of the items being crawled. Note that time complexity in BFS is at worst $\mathcal{O}(|V| + |E|)$, as we do not have the input graph in advance so it is not possible to determine the number of edges to be discovered along the graph during a real BFS crawl. From Algorithm 3.1 in Chapter 3, and because we use a maximum depth parameter to limit the crawl, this will take $\mathcal{O}(b^{d+1}))$, being b the average out-degree or branching factor, and d our *maxdepth*.

The second reason to perform such cost analysis is due to the fact that "time is money" [1]. Some alternatives include using whitelisted machines provided by Twitter [64] or use of proxy servers to avoid enforcement of Twitter limits. The former is not possible anymore to the best of our knowledge in the particular case of Twitter, and the latter does not respect Twitter's policy and usage guidelines, which is also more

[1] `http://www.cogsci.ucsd.edu/~coulson/Courses/101c/analogy3.pdf`

difficult as of today due to Twitter restrictions that apply per IP an user account (e.g., multiple user accounts must be created on Twitter to act as service accounts at the time of crawling). Therefore, researchers using these platforms are forced to come up with creative solutions that aim to crawl Twitter in parallel but quantitative results have not been yet reported. In this situation, where it is necessary to build custom tools to pull a dataset, it is nevertheless necessary to obtain enough metadata for reproducibility and dissemination of experiments to the research community, thus advancing the state of the art. We find that listing the tweet identifiers of our dataset is the best way of trying to open our experiments to the research community. Therefore, we do not break Twitter usage guidelines. However, researchers will need to replay our crawl and obtain the missing metadata using their own custom tool or crawling infrastructure.

We envision reducing the cost of crawling thanks to the use of sampling at each node when doing the graph traversal of the BFS. In particular, this will reduce the number of API calls required to collect data from the Twitter social graph.

7.2.1 User metadata cost

User metadata involves the first twelve features in Table 7.1. They largely rely on the call to a single endpoint of the Twitter API, therefore a single call to "get/users/show " can provide all the required metadata. This also makes sense in decentralized settings, whereas we consider such information "local knowledge" as it does not involve information about the state of the network, more specifically the OSN.

7.2.2 Tweet metadata cost

Message metadata is also "local knowledge" and cheap in terms of API costs, namely a single call to the endpoint "statuses/show/:id".

7.2.3 Messaging graph metadata cost

Messaging graph metadata involves knowing the patterns of the messaging system in place in the OSN. Since the messages form a sort of multi-graph, we would need to iterate over the limits of Twitter API timeline crawling for obtaining a complete view of such multi-graph. This is highly expensive and anyway Twitter only allows crawling of the last 3200 messages in a user's timeline. To do so, we use the endpoint "get/statuses/lookup", which provides custom configuration settings to perform the crawl in batches.

7.2.4 Social graph metadata cost

The features involving social graph are the most expensive to obtain in Twitter, especially for users with large number of subscriptions and/or subscribers. However an adjacency matrix can be built using only these API calls: "get/followers/ids ", "get/friends/ids ".

Table 7.1: API costs in Twitter

Feature	API endpoint	Items returned
age_of_account	get/users/show	1
verified	get/users/show	1
account_lists	users/show	1
favorite_count	get/users/show	1
tweets_per_day	get/users/show	1
recent_account	get/users/show	1
followers_count	get/users/show	1
friends_count	get/users/show	1
friends_to_followers	get/users/show	1
followers_to_friends	get/users/show	1
ratio_follows_sent	get/users/show	1
ratio_follows_received	get/users/show	1
mention_count	statuses/show/:id	1
hashtag_count	statuses/show/:id	1
badwords	statuses/show/:id	1
retweet_count	statuses/show/:id	1
retweeted	statuses/show/:id	1
is_reply	statuses/show/:id	1
sensitive	statuses/show/:id	1
mentions_over_tweets	get/statuses/lookup	100
mentions_per_day	get/statuses/lookup	100
replies_over_users	get/statuses/lookup	100
mutual_followees	get/friends/ids + get/friends/ids	5000 x 2
mutual_followers	get/followers/ids + get/followers/ids	5000 x 2
mutual_followers_followees	get/followers/ids + get/friends/ids	5000 x 2
mutual_followees_followers	get/friends/ids + get/followers/ids	5000 x 2
tweet reciprocity	get/friendships/show	1

For the feature about reciprocity, note we can obtain its boolean value with a single call the new "get/friendships/show" endpoint in Twitter.

Finally, we group our features into "local" and "neighborhood knowledge". The latter is not to say not to say it is representative of the network state but mostly about the information not available to a participant directly in a decentralized setting. The feature sub-division is depicted in Table 7.3.

7.3 Efficient privacy-preserving protocols for abuse detection

In chapter 6 we presented a protocol that provides a privacy-preserving signed PSI, using BLS signatures that protect the identity of subscribers. Here we present a "novel"

Features	Timing (ms)	# of hash. func. (k)	Error bound
All (using $\mathcal{J}$ index)	3,018,632.98	–	–
All (using approx. $\mathcal{J}$ index)	2,626,971.92	64	$\mathcal{O}(1/\sqrt{k})$
All (using approx. $\mathcal{J}$ index)	2,642,225.02	128	$\mathcal{O}(1/\sqrt{k})$

Table 7.2: Average time over 5 runs approximating Jaccard with k hash functions and using baseline (no approximation)

approach using approximated PSI for abuse detection, which in turn proves to be compatible with such a protocol and reduces its computational overhead in practice for a future deployment in a DOSN.

Estimation techniques that approximate Jaccard index often rely on theory behind Min-Wise Independent Permutations [32] but due to the cost of implementing a perfect hashing function, practical approaches use random number generators to minimize collisions (e.g., here we use Mersenne prime[2]). The idea is to apply hashing to elements in the sets and find the probability that each of the minimums in a pair of values among the sets match; thus yielding the Jaccard.

MinHashes has been previously studied for efficient estimation of network metrics as triangle count [9]. In MinHashes given a real set S, define a representation of it (k), which is the size of the representation or sampling factor over the real set S when using several hash functions. Let F be a family of hash functions mapping elements from a set U to distinct r-bit integers. We select k different hash functions from F, namely $h_1(\cdot)$,...,$h_k(\cdot)$. For any subset of elements in U, let $h^i_{min}(S)$ be the lowest value in the subset. A MinHash representation $h_k(S)$ of the set S is a vector of elements from i to k. In turn, we can estimate the Jaccard, $\mathcal{J}(\mathcal{A},\mathcal{B})$, by counting the number of indexes (i) that satisfy $h^{\mathcal{A}}_{min}(\cdot) = h^{\mathcal{B}}_{min}(\cdot)$, more formally described as equation:

$$\frac{|h_k(\mathcal{A}_i) \cap h_k(\mathcal{B}_i)|}{k} \tag{7.1}$$

Table 7.2 shows the computation times for the complete list of features we provide early in the thesis, which is our baseline. Here we provide computation times for their original group of features and show that using MinHashes for approximating PSI based features ($\mathcal{J}$) is feasible. Benchmarking is done in a MacBook Pro laptop with 2.6 GHz Intel Core i7 and 16 GB 1600 MHz DDR3.

7.4 Impact of data minimization into abuse classifications

In order evaluate how approximated PSI features compare in abuse classifications, we again, we compare decision trees (DT) [29] random forest (RF) [28] , extra trees (ET) [70] , gradient boosting (GB) [27], AdaBoost (AB) and support vector machines (SVM). Additionally we combine all previous ones in an ensemble of classifiers, namely voting classifier.

[2]http://en.wikipedia.org/wiki/Mersenne_prime

	Metadata	Feature	Description
Local knowledge	**Message**	# lists	number of lists of sender.
		# mentions	mentions count in tweet.
		# hashtags	hashtag count in the tweet.
		# retweets	times a message has been reposted.
		is_retweet (true/false)	message is a repost.
		is_reply (true/false)	message is a reply.
		sensitive	message links to external URL.
		#badwords	number of badwords from Google list.
		verified (true/false)	sender account is verified by Twitter.
		# favorited messages	sender # of messages favorited.
neighborhood knowledge	**Sender**	age of account	days since creation of account.
		recent account	check if age of sender account is <= 30 days
		# $messages/_{age}$	tweets per day.
		# $mentions/_{age}$	average mentions per day.
		# $mentions/_{\#\ tweets}$	average mentions per tweet.
		# $replies/_{user\ tweets}$	fraction of tweets that are replies.
	Social	# $subscriptions^s$	followee count from public feed of sender
		# $subscribers^s$	follower count to public feed of sender
		# $subscribers/_{age}$	ratio of subscribers to age of sender account.
		# $subscriptions/_{age}$	ratio of subscriptions to age of sender account.
		# $subscriptions/_{\#\ subscribers}$	ratio of subscriptions to subscribers of sender.
		# $subscribers/_{\#\ subscriptions}$	ratio of subscribers to subscriptions of sender.
		reciprocity (true/false)	relationship of sender and receiver in social graph.
	PSI	$\mathcal{J}$ ($subscriptions^s$, $subscriptions^r$)	$\mathcal{J}$ of sender & receiver subscriptions.
		$\mathcal{J}$ ($subscribers^s$, $subscribers^r$)	$\mathcal{J}$ of sender & receiver subscribers.
		$\mathcal{J}$ ($subscriptions^s$, $subscribers^r$)	$\mathcal{J}$ of subscriptions of sender & subscribers of receiver.
		$\mathcal{J}$ ($subscribers^s$, $subscriptions^r$)	$\mathcal{J}$ of subscribers of sender & subscriptions of receiver.

Table 7.3: Subsets of features by category

The first observation is that using a larger ground truth (here we aggregate Trollslayer and Crowdflower) provides more balanced classifications in terms of precision-recall. Generally, results show a reduced gap among false positives/negatives. That is, missing more cases of abuse but reducing the cases of acceptable messages being classified as abusive. The voting classifier provides the most balanced result, with a moderate ratio of false-positives in both cases, whether it is using local of neighborhood knowledge. In addition SVM detects abuse better when using neighborhood knowledge with approximated PSI features, which means an adversary trying to subvert the system will have a harder time to evade detection here.

The human baseline we employ here (ground truth from Both crowdsourcing platforms) as ground truth shows a P of 0.77 for abusive and 0.96 for acceptable content. R is 0.77 as well for abusive, and 0.97 for acceptable; thus having a ground truth of with a balanced ratio of false positives/negatives among human raters. However, it highlights the difficulty for humans to agree on rating abuse.

7.5 Summary of results

Adversarial learning to privately detect abuse in DOSN is possible. Our benchmarking indicates that the data minimization technique obtains features 13% faster while providing similar or, in the case of SVM classifier, even better abuse detection rates with just approximated neighborhood knowledge in supervised learning 7.4. This is an advantage in decentralized settings, where we can benefit from the decoupled nature of a given OSN network.

In terms of classifications, employing approximated PSI features, we obtain similar detection rates in supervised learning learning algorithms. However, classification

Classifier	Metric	Features: local		Features: neighborhood		Features: neighborhood approx.		Features: all	
		Acceptable	Abusive	Acceptable	Abusive	Acceptable	Abusive	Acceptable	Abusive
DT	Precision	0.970 ± 0.013	0.223 ± 0.152	0.963 ± 0.020	0.143 ± 0.105	0.956 ± 0.020	0.140 ± 0.091	0.960 ± 0.015	0.231 ± 0.111
	Recall	0.849 ± 0.135	**0.562 ± 0.146**	0.745 ± 0.123	**0.583 ± 0.200**	0.800 ± 0.155	**0.449 ± 0.292**	0.900 ± 0.033	**0.438 ± 0.219**
	F-score	0.903 ± 0.084	0.315 ± 0.175	0.839 ± 0.083	0.228 ± 0.145	0.869 ± 0.091	0.207 ± 0.116	0.929 ± 0.018	0.300 ± 0.134
RF	Precision	0.958 ± 0.014	0.377 ± 0.176	0.954 ± 0.002	0.323 ± 0.072	0.956 ± 0.016	0.343 ± 0.118	0.955 ± 0.015	0.445 ± 0.174
	Recall	0.956 ± 0.029	**0.376 ± 0.210**	0.955 ± 0.014	**0.313 ± 0.013**	0.957 ± 0.018	**0.345 ± 0.232**	0.973 ± 0.019	**0.314 ± 0.215**
	F-score	0.957 ± 0.015	0.370 ± 0.173	0.954 ± 0.007	0.316 ± 0.030	0.956 ± 0.008	0.339 ± 0.171	0.964 ± 0.009	0.359 ± 0.191
ET	Precision	0.973 ± 0.014	0.279 ± 0.139	0.957 ± 0.013	0.218 ± 0.058	0.956 ± 0.015	0.230 ± 0.090	0.964 ± 0.014	0.368 ± 0.106
	Recall	0.884 ± 0.053	**0.636 ± 0.184**	0.905 ± 0.035	**0.396 ± 0.198**	0.915 ± 0.038	**0.376 ± 0.221**	0.945 ± 0.008	**0.479 ± 0.216**
	F-score	0.926 ± 0.034	0.386 ± 0.163	0.930 ± 0.015	0.279 ± 0.087	0.935 ± 0.019	0.282 ± 0.125	0.955 ± 0.007	0.415 ± 0.145
GB	Precision	0.955 ± 0.002	0.448 ± 0.216	0.948 ± 0.011	0.419 ± 0.109	0.949 ± 0.015	0.422 ± 0.259	0.953 ± 0.006	0.459 ± 0.108
	Recall	0.972 ± 0.021	**0.312 ± 0.056**	0.980 ± 0.021	**0.199 ± 0.157**	0.982 ± 0.011	**0.209 ± 0.233**	0.976 ± 0.012	**0.292 ± 0.089**
	F-score	0.963 ± 0.010	0.363 ± 0.090	0.963 ± 0.006	0.254 ± 0.166	0.965 ± 0.009	0.275 ± 0.263	0.965 ± 0.004	0.353 ± 0.062
AB	Precision	0.953 ± 0.011	0.568 ± 0.202	0.951 ± 0.005	0.674 ± 0.390	0.951 ± 0.014	0.530 ± 0.312	0.954 ± 0.007	0.564 ± 0.289
	Recall	0.987 ± 0.007	**0.269 ± 0.197**	0.990 ± 0.014	**0.250 ± 0.078**	0.987 ± 0.008	**0.241 ± 0.209**	0.983 ± 0.016	**0.302 ± 0.116**
	F-score	0.969 ± 0.005	0.360 ± 0.200	0.970 ± 0.007	0.359 ± 0.110	0.968 ± 0.010	0.329 ± 0.257	0.969 ± 0.010	0.391 ± 0.161
SVM	Precision	0.955 ± 0.004	0.371 ± 0.062	0.975 ± 0.039	0.072 ± 0.011	0.957 ± 0.014	0.352 ± 0.098	0.975 ± 0.036	0.069 ± 0.007
	Recall	0.963 ± 0.008	**0.323 ± 0.047**	0.208 ± 0.041	**0.916 ± 0.143**	0.956 ± 0.010	**0.366 ± 0.205**	0.153 ± 0.033	**0.937 ± 0.103**
	F-score	0.959 ± 0.005	0.345 ± 0.048	0.343 ± 0.055	0.134 ± 0.020	0.957 ± 0.006	0.356 ± 0.149	0.264 ± 0.049	0.129 ± 0.013
Voting	Precision	0.958 ± 0.013	0.404 ± 0.326	0.955 ± 0.006	0.344 ± 0.103	0.954 ± 0.015	0.332 ± 0.101	0.956 ± 0.011	0.452 ± 0.099
	Recall	0.956 ± 0.040	**0.385 ± 0.181**	0.956 ± 0.021	**0.334 ± 0.088**	0.959 ± 0.013	**0.314 ± 0.215**	0.973 ± 0.012	**0.334 ± 0.163**
	F-score	0.957 ± 0.025	0.388 ± 0.238	0.955 ± 0.010	0.335 ± 0.063	0.957 ± 0.004	0.319 ± 0.163	0.964 ± 0.004	0.379 ± 0.121

Table 7.4: Evaluation of features by category in scikit-learn with 5-cross validation

results are naturally bounded by our human baseline, in this case Both, so we note our approximation of PSI features is generally close or slightly better than when using no approximation. and even though we have not extensively tested such fact, these approximation can prove to be more resilient in the presence of malicious adversaries (e.g, attackers obfuscating the feature values.

In addition, using MinHashes for approximation can effectively minimize the amount of social graph metadata that needs to be computed and sent over a network. Because our features are largely based on metadata available in the proximity of the user performing the detection, this method can provide a self-defense mechanism compatible with decentralized efforts implementing secure multicast at the network level (e.g., `secushare.org`).

To the best of our knowledge, this is the first work that shows useful insights about the efficiency of privacy-preserving protocols aimed to to detect abuse and resist adaptive adversaries. This is while providing a reduced computational overhead for protocol computations at the network level. Incidentally, by approximating our PSI features as input for supervised learning algorithms, we find out that in some cases this approach can actually better mitigate the effect of adversarial samples into classifications outcomes.

Conclusion and Future work

In this thesis we apply supervised machine learning algorithms and Private Set Intersection protocols to designing a privacy-preserving abuse classification framework for future DOSNs, where participants may be more prone to abusing others due to the anonymous or semi-anonymous nature of the platform. For that we first collect data from Twitter in order to first understand what metadata we can extract from a modern OSN, and see if it is useful to detect abuse. Once we have designed a number of data features we characterise by being made of, metadata from the Twitter messaging multigraph, social graph, user metadata, tweet metadata, etc. In addition we analyse how relevant each of the individual features is to the abuse classifiers. Then we look into how to translate such useful features into privacy-preserving features for future DOSN. We achieve that by means of a PSI protocol which uses BLS signatures together with hashing and standard DH techniques to be resistant against malicious attackers.

Regarding abuse, we see it is not easy to build automated supervised classification methods, as they require a high-quality ground-truth in order to obtain sound predictive models.

The privacy-preserving protocol we have designed is based in the analysis of data features we have extracted from Twitter data collected using our own custom breadth-first-search (BFS) crawler. The privacy-preserving protocol is mainly intended for future decentralised DOSNs, where for example a user may want to set different privacy policies based in the value of a mutual friends set intersections that has with another participant in the network.

Given that we have designed a privacy-preserving signed PSI protocol, the next step would be to implement it. A prototype of such a protocol can help us to test its performance in a DOSN deployment (simulated or real). Future decentralised DOSN as SecuShare or Peergos can be a good testbed for that purpose.

A new direction also emerged during a PhD internship at the Systems Group (NetOS) in the Cambridge Computer Laboratory. The future work will require some privacy input regarding the ongoing work they are doing with estimation of network metrics in large graphs. This can also prove to be useful to test our abuse detection methodology in large graphs where we may be able to detect the volume of malicious nodes involved in sending abusive messages. However, it is possible that we may still need to annotate abusive content manually or with the help of OSN providers.

A

Database code

The database code and datasets for “Trollslayer” can be found in [68].

B

Summary in English

Today, modern Online Social Networks (OSNs) have become a vast source of information about citizens and their lifes. These platforms have a large user-base and global reach that allows providers to monetize participant metadata through advertising, marketing campaigns, or even building recommendation engines targeted to predicting user activity patterns in the platform (e.g., increase of corporate revenue streams). Their business model is thus based in having access to large amounts of user metadata that they use as a product, with the corresponding implications for the privacy of those users. In this context, large Internet corporations such as Google or OSNs as Facebook have successfully attracted massive user-generated content to their platforms in exchange of providing a free service to users. If we make the analogy with free software [1], unfortunately their idea of "free" service to users does not equal "free as in freedom", but more "free as in beer" (note Google allows users to delete all their metadata from their Google+ OSN). Such control over user metadata is an instrument of unjust power as the Free Software Foundation advocates, and it is driving our digital society and the Internet into a data monopoly. Given the value of such user-generated content, companies can exploit it for building or letting third parties build data analytics and products that provide insightful knowledge for critical decision making businesses and retail. In other words, such valuable data provides unparalleled levels of "power" to whoever controls it. This data monopoly has the ability of influencing our life style and vital choices to the point that even government are interested in the control of such resource. This creates several issues, including ethical or legal ones. Besides, such loss of privacy can lead to discrimination based in gender, age or any other social status of a citizen. A recent controversial application of Social Network Analysis (SNA) to real data from citizens involves the analysis of customer social graph metadata in the Facebook OSN

[1] `https://www.gnu.org/philosophy/free-sw.en.html`

in order to discover if such client is more or less likely to return a loan to the bank in the future [2]. While the startup behind the idea, Lenddo[3], states: "we don't share any of the data, only a score. Full stop.", this seems rather naive if we consider that they also mention the following: "by last summer, we finally had enough data to show that the score worked.". This is definitely not the approach we envision to best protect users privacy. In that line, we need to ask ourselves what would happen to our data gets lost, into the wrong hands or just hacked by specialised teams of criminals, even governments or public agencies with enough resources to tap into the data with or without consent. Isn't user data collection a first step towards total loss of privacy for users? Data collection by OSNs needs to be revised and we definitely need to ensure users control their own information, rather than giving away this control to the OSN provider who will not just exploit their metadata but put their privacy at serious risk. In the meantime, we will continue to see more products and companies that rely on user metadata and create another away of data discrimination and privacy loss to users. In the case applying machine learning data analytics to credit and verification of customers, stating that thanks to data analytics banks can now give loans to people they would usually reject shall be carefully put into context to make users aware of the amount of metadata the product requires about themselves. A similar issue has been investigated as direct of source discrimination in online purchasing web portal, where geo-location metadata impacts prices differently in distinct geo-graphical areas. Such discrimination is highlighted by researchers from the Data Transparency Lab [99]. To refer back to OSNs, they collect all such metadata about participants, including activity patterns such as messaging, social relationships, etc. Therefore user data is exposed to the risk of misuse by either network participants, the OSN provider or a third-party. In this context, application of data analytics to user generated content is an arguably ethical business model that OSN providers have built over years of user data collection. Firstly, this so called "Big Brother" model goes against democratic values in societies where citizens themselves are expected to provide transparency and accountability for their actions, but also have the fundamental human right to privacy according to the United Nations Declaration of Human rights. Such Big Data approach consistently violates any reasonable privacy expectations of citizens, who lose their right to be forgotten if data is not regulated carefully. Meanwhile, private companies owning these data silos have become critical keystone for the military-industrial espionage complex. Examples include project PRISM[4], a programme from the NSA (National Security Agency) in United States for cyber-espionnage. To solve it, decentralised OSNs [5] are a growing alternative to traditional, centralised OSNs, as a means for user's to regain control over their private communications. Due to the lack of a centralised server, detecting and reigning in abuse it can be even more challenging for these systems. This is because in a decentralised setting we assume better privacy assurances are provided, which are likely to unhinge abusive participants. In this context, we see a conflict among detecting

[2] `https://www.wired.com/2015/01/banks-handing-loans-people-normally-shun`
[3] `https://www.lenddo.com/about.html`
[4] `http://en.wikipedia.org/wiki/PRISM_(surveillance_program)`
[5] `https://peergos.org/`

abuse and preserving the privacy requirements of the users in the platform.

Regarding abuse, it is worth noting that the free-speech nature of OSNs, which naturally allows users to debate and often create constructive discussions on certain topics as politics, sports, etc, can also become a serious problem when misused by OSN participants. For example, and taking Twitter as example, the micro-blogging site provides functionality that allows a single topic or user to be the focus of discussion. One way to achieve that is by means of hashtags, a tweet prefixed by the hash symbol (#) to remark the topic of a certain discussion or message. Another, is to take the attention of participants by addressing them with the @ symbol followed by the user's screen name in the network. This is obviously prone to abuse when allowed to be used at free will, and participants often even include more than just a hashtag or mention in their messages. In this context, it is not difficult to encounter users that exploit such functionality to suit their own needs, whether they are a political agenda, marketing campaigns or similar. In summary, the functionality Twitter provides is a very fragile tool that when abused can lead to the manipulation of information diffusion which in turn gives raise to a new generation of security issues in OSNs. These include but are not limited to astroturfing [110] or the spreading of misinformation and abuse [111], hate-speech [113], cyber-bullying [78] and even cyber-terrorism [105].

In all these cases above, attackers are presumably lured by the low or marginal cost of operations that the infrastructure provides them with. So much so, that many of these attackers can range from very well organized criminals trying to push their own propaganda to just teenagers in schools that use the infrastructure of the platform to discredit or bully their fellow colleagues. Real cases of such attacks include but are not limited to social engineering techniques using Twitter, as a failed attempt to manipulate public opinion and create chaos (e.g., spreading news about a hoax chemical explosion in Columbia, US) dated back in 2014 and reportedly authored by russian trolls from St. Petersburg [6]). This modus operandi is somehow similar to spam, but with a much richer toolset at the disposal of attackers, so much so that it is well on the way to become a replacement or contemporary version of traditional spam. The issue is becoming so pressing to users and OSN providers, that OSN infrastructure providers as Twitter are currently combating the problem both legally and technically. Twitter has announced an upcoming battle against abuse and have introduced changes in their user policy [97]. To address such abuse, other OSN providers as Tuenti in Spain have traditionally employed personnel staff who searches for incidents and other privacy related issues in their platforms. This manual search and filtering effort is often too costly and slow at large scales. Therefore, intelligent systems are built to support the task. Large OSNs as Facebook have security teams that deal with the issue and build systems as the Facebook immune system(FIS) [115] to aid humans in detecting abuse at large scale in the platform. For instance, the FIS information from user activity logs to automatically detect and act upon suspicious behaviors in the OSN. This makes more difficult for malicious users to perform such attacks or create a number of fake identities in order to deny, disrupt, deceive and degrade participants in these platforms, to name

[6] `http://www.nytimes.com/2015/06/07/magazine/the-agency.html`

a few. The problem is that while abuse detection is something that has received a great deal of attention from the security research community, to the best of our knowledge little or less effort has been put into making the analysis and detection of such abuse privacy-preserving.

In this thesis, we consider as use case a modern platform for social micro-blogging, Twitter. The most characteristic feature of Twitter is that users write public messages in no longer than 140 characters, in which they have the ability to mention other users by referring to their public identifier, create hashtags to tag a message with a topic, or even post media content and links to external information sources.

Unlike in a public, centralised OSN, we envision less metadata available for analysis of the participants' behavior when in a decentralised OSN. For instance, a social graph is likely not to be made available to the rest of the network participants through the OSN interface or developer API as it is the case in Twitter. Therefore, collecting such metadata may be more challenging or impossible to collect due to application design choices. Not to mention that the application of machine learning classification algorithms that aim to detect abuse can also be more difficult.

Our approach is to first develop a set of features that detect abusive behavior in a centralised, public OSN. Then we investigate the suitability of porting each of our features into a future decentralised, privacy-preserving OSN where such metadata may be impossible to collect of insufficiently privacy-preserving. Therefore, we try to asses the impact of each feature from the point of view of privacy in order to realize which data will be more sensitive, important, etc and how to be able to display or collect it in a privacy-preserving manner for abuse classification.

We will use private-set intersection techniques to asses the suitability of using social graph data for abuse detection. Having such protocols, may prove useful for classifiers to obtain a viable privacy-preserving result that reasonably detects abuse. To arrive to that conclusion, we analyse how secure and resilient each of the features we have developed is against strong adversaries in the OSN. Such analysis can aid developers of future decentralised OSNs to make design choices that correctly balance the trade-off among privacy and abuse detection, eventually discouraging such misconduct.

A summary of our research, and the main objective of this thesis is to investigate privacy-preserving protocols that only need local metadata or privacy-preserving metadata to detect abusive behavior in the OSN. By applying Social Network Analysis (SNA) techniques that reduce the amount of sensitive metadata we obtain acceptable results when compared to non privacy-preserving. Therefore, we plan to draw a series of recommendations on how to build future decentralised online social networking applications that by default discourage abusive behavior while respecting all.

Eventually, we hope our research will inspire future work into the area of privacy-preserving, decentralised online social networking systems that protect user metadata by design. Several candidates are in our list, among of which SecuShare[7] and Peergos offer the most comprehensive yet promising infrastructure on which to develop and scale such systems. SecuShare is being developed at Inria on top of the GNUnet platform,

[7] `http://secushare.org/`

while Peergos is built on top of IPFS (the Inter-Planetary File System) [11] and independently developed by researchers in the United Kingdom. We have been in contact with both groups, and the idea of integrating our protocol in their network stack seems a very appealing proposal, especially because privacy-preserving contact discovery is yet an open problem on these type of systems. Previous research in that direction has investigated the suitability of well-known data structures such as bloom-filters, private information retrieval techniques (PIR), and of course the private-set intersection cardinality literature. We expect this area of research to continue growing in interest by the research community in privacy-enhancing technologies (PETs). In the long term, we would like to implement a working version of our protocol, bechmark it and verify how well the statements we make during its theoretical analysis hold in practise. For that, we will need to build a test-harness system that can help us to improve the efficiency and security guarantees of the protocol in a future testbed decentralised OSN deployment.

C

Résumé en Français

De nos jours, les réseaux sociaux sont devenus une source d'information considérable notamment concernant la vie des citoyens. Ces plateformes disposent de millions d'utilisateurs répartis sur l'ensemble de la planète, ce qui permet aux fournisseurs de service de collecter puis de monétiser les métadonnées associées. La monétisation est assurée par la publicité pour l'exemple le plus direct mais peut également prendre la forme de moteurs de recommandations pour prédire l'activité des utilisateurs (e.g., augmentation des revenus). Leur modèle économique est basé sur la masse presque inépuisable de métadonnées considérée comme un produit usuel, avec les implications que cela engendre concernant la protection de la vie privée. Dans ce contexte, les multinationales de l'Internet comme Google ou les réseaux sociaux avec Facebook ont parfaitement réussies à attirer massivement les internautes en fournissant un service gratuit pour l'utilisateur.

Si nous faisons l'analogie entre les logiciels libres et les logiciels gratuits[1], une différence subtile existe entre les réseaux sociaux decentralises et Facebook, Twitter, etc. Dans le deuxième cas le logiciel contrôle l'utilisateur donc il ne sont pas libres, même si il sont gratuit. Ce contrôle des données est un véritable outil de pouvoir tel qu'il est décrit par la Free Software Foundation, le monde digital et Internet sont ainsi dirigés par le contrôle des données. Les entreprises ont conscience de la valeur du contenu généré par les utilisateurs, c'est pourquoi elles exploitent ces données. Pour cela elle peuvent faire appel à des entreprises spécialisées dans l'analyse de données, la conception de produit mais aussi de l'aide à la décision. En d'autre mots, ces données donne un véritable pouvoir à celui qui en dispose. Le contrôle des données peut influencer nos vies ainsi que nos décisions les plus personnelles, les gouvernements sont eux aussi particulièrement intéressé par le contrôle de cette ressource. Cela génère de nombreux

[1] `https://www.gnu.org/philosophy/free-sw.en.html`

problèmes, ont peut citer les considérations éthiques ou légale. Par ailleurs, l'accès à la vie privée des utilisateurs peut engendrer des discriminations basées sur le genre, l'âge ou le statu social. Une controverse a récemment éclatée concernant une application basée sur l'analyse des réseaux sociaux, en effet à partir des métadonnées associées au graphe d'un client dans Facebook il est possible de déterminer si un individu est susceptible de rembourser un prêt ou non [2]. La startup à l'origine de cette idée [3] déclare : "Nous ne partageons aucunes données, seulement un score". On peut tout de même douter de leur bonne foi car ils ont aussi déclaré : "l'été dernier nous avons collecté suffisamment de donnée pour montrer que notre méthode de score fonctionne". Ce n'est certainement pas la meilleur approche à envisager pour protéger la vie privée des utilisateurs. Nous devons nous demander qu'elles seraient les répercussions si nos données étaient perdues, tombées entre des mains peu scrupuleuses, piratées par des criminels ou à la disposition d'agence de renseignement avec ou sans l'accord des utilisateurs. La collecte des données utilisateurs n'est-elle pas un premier pas vers une perte totale de vie privée ? L'accumulation des données par les réseaux en ligne doit être revisitée afin d'assurer aux utilisateurs un contrôle sur leur information plutôt que de donner un "chèque en blanc" aux fournisseurs de services qui vont exploiter les métadonnées sans tenir compte de la vie privée.

Dans le même temps, nous continuons de voir toujours plus de produits et d'entreprises qui comptent sur les métadonnées utilisateurs afin de créer de nouveaux moyens de discriminer les données sans aucun respect de la vie privée. Il est possible d'utiliser de l'apprentissage automatique pour vérifier la solvabilité d'une personne, les banques commencent à développer de telles techniques. Grâce à l'analyse des données les banques peuvent ainsi accorder des prêts à des gens qui jusqu'à lors étaient rejeté par le système traditionnel. Des questions restent alors en suspend, comment une banque peut-elle aller à l'encontre de son système historique et favoriser les données collectées sur l'utilisateur ? Celle-ci doit disposer en effet d'une masse d'information non négligeable concernant la vie privée de l'utilisateur et ses relations sociales, en obtenant ainsi le comportement du client pour s'assurer du remboursement. Un problème similaire a été constaté comme un moyen clair de discrimination sur les sites web marchands, en effet en fonction de la géolocalisation du client les métadonnées associées font varier le prix des produits. De telles discriminations sont mise en lumières par les chercheurs du Data Transparency Lab[99]. Pour revenir aux réseaux sociaux en ligne, ils collectent un maximum de métadonnées sur leur clients comme par exemple leur comportement sur la messagerie instantanée ou leur relations sociales etc. De plus ces données peuvent être source d'erreur pour les participants, les réseaux sociaux en ligne mais aussi un ou des tiers. Dans ce contexte l'analyse des données générées par les utilisateurs eux-mêmes est indiscutablement le modèle économique que les réseaux sociaux en ligne on construit ces dernières années. On peut faire une analogie entre ce modèle et "Big Brother". En effet les citoyens se doivent de respecter les valeurs démocratiques de la société en étant transparent et en assumant leur actions, mais ils doivent également disposer grâce à la

[2] https ://www.wired.com/2015/01/banks-handing-loans-people-normally-shun
[3] https ://www.lenddo.com/about.html

déclaration des droits de l'homme à un respect de leur vie privée. Cette approche basée sur les données de masse viole les droits des citoyens, ceux-ci perdent ainsi le droit à l'oubli si leur données ne sont pas utilisées et réglementées avec attention. Parallèlement, les entreprises privées qui possèdent les données sont devenues une clef de voûte essentielle pour le complexe militaire-industriel ainsi que les agences de renseignement. L'exemple le plus notable est le projet PRISM [4], dirigé par la NSA (Agence National de Sécurité aux États-Unis) pour le *cyber-espionnage*. Pour résoudre une partie du problème, les réseaux sociaux décentralisés [5] sont une alternative intéressante pour les utilisateurs afin de retrouver un certain contrôle sur leur communication privée. En raison du manque de serveur centralisé, la détection des abus est plus difficile pour les systèmes décentralisés. La conception d'un système dans un contexte décentralisé par convention assure un meilleur respect de la vie privée et est susceptible de déstabiliser les participants abusifs. Dans ce contexte, un conflit oppose d'une part la détection d'abus et d'autre part le respect de la vie privée des utilisateurs sur la plateforme.

Les utilisateurs disposent d'une grande liberté sur les réseaux sociaux, ce qui permet de nombreux débats constructifs sur des sujets très variés (politique, santé, sport). Cette liberté peut poser de sérieux problème quand des abus sont présent. Par exemple Twitter qui est un site de micro-blogging, fournit une fonctionnalité qui permet à un seul sujet ou utilisateur d'être au centre de la discussion. Une façon d'y parvenir est d' utiliser des hashtags, il s'agit d'un mot préfixé par le symbole (#). On peut également attirer l'attention de participants en faisant référence à une personne avec le symbole @ suivi par l'alias de l'utilisateur sur le réseau. Il est clair que ces fonctionnalités peuvent être sujette à de nombreux abus lorsqu'elles sont utilisées de manière malicieuse. De plus les messages postés par les participants contiennent souvent plus qu'un simple hashtag ou qu'une seul référence vers d'autres utilisateurs. Basé sur ce constat il n'est pas difficile de rencontrer des utilisateurs qui exploitent ces fonctionnalités pour parvenir à leur fin (agenda politique, campagne marketing, etc...). En résumé, les fonctionnalités que Twitter fournit sont fragile, si celles-ci sont abusées cela peut conduire à de la manipulation d'information. Cela peut donc engendrer une nouvelle génération de problèmes de sécurité dans les réseaux sociaux. On peut compter parmi elles "l'astroturfing" [110] ou la divulgation de fausse information et d'abus [111], discours haineux[113], cyber-intimidation[78] et même le cyber-terrorisme[105].

Dans tous les exemples ci-dessus les attaquants sont attirés par la facilité d'utilisation du service ainsi que son faible coût. Tant et si bien que beaucoup de ces attaquants peuvent ensuite essayer de propager leur propre propagande aux utilisateurs qui utilisent ces infrastructures pour discréditer ou intimider leur collègues. Des cas réels de telles attaques incluent, sans s'y limiter, des techniques d'ingénierie sociale utilisant Twitter. Comme par exemple une tentative de manipulation de l'opinion publique afin de créer le chaos qui remonte à 2014 et aurait été rédigé par des trolls russes de Saint-Petersbourg (une explosion chimique aux États-Unis) [6]. Ce modus operandi est similaire au *spam*, mais avec un ensemble d'outils à la disposition des attaquants

[4] `https://en.wikipedia.org/wiki/PRISM_(surveillance_program)`
[5] `https://peergos.org`
[6] `http://www.nytimes.com/2015/06/07/magazine/the-agency.html`

bien plus riche, à tel point qu'il est en passe de devenir une version contemporaine du spam traditionnel. La question devient si inquiétante pour les utilisateurs et les réseaux sociaux que ceux-ci tentent de combattre le problème à la fois légalement mais aussi techniquement. Twitter a annoncé une bataille à venir contre les abus et a introduit des changements dans sa charte utilisateur. Pour adresser ces abus d'autre réseaux sociaux comme Tuenti en Espagne utilise de manière plus traditionnelle des employés qui recherchent spécifiquement ces type incidents (vie privée, propagande, etc ...). Cette recherche manuelle est beaucoup trop couteuse et lente lorsque l'on doit traiter une masse considérable de données. De plus des systèmes intelligents existent afin d'assurer ce besoin. Facebook par exemple dispose d'une équipe de sécurité qui traite ce genre de problème tout en étant guidé par un système nommé *Facebook Immune System*(FIS) [115], ce qui permet de gérer le flux considérable de données. Pour le moment le système FIS collecte les informations correspondant au comportement d'un utilisateur et agit de manière approprié en cas d'activité suspecte. Les attaques menées par les attaquants depuis de faux comptes deviennent alors plus compliquées. En effet ceux-ci ont pour habitude de créer de fausse identité pour nier, perturber, tromper et dégrader certain utilisateur de ces plateformes. La détection d'abus à reçu beaucoup d'attention de la part de la communauté de la recherche en sécurité. Malheureusement à ma connaissance peu d'effort on était fournit pour permettre l'analyse et la détection de ces abus tout en respectant la vie privée des utilisateurs.

Dans cette thèse, nous considérons comme cas d'usage une plate-forme moderne dédiée au microblogging, Twitter. Le trait le plus caractéristique de Twitter est que les utilisateurs écrivent des messages publics contenant au maximum 140 caractères, dans lesquels ils ont la capacité de mentionner d'autres utilisateurs en se référant à leur identifiant public, de créer des hashtags pour étiqueter un message par rapport à un sujet ou même de publier du contenu multimédia et des liens vers d'autre sources d'information.

Contrairement à un réseaux social public centralisé, nous envisageons de disposer de moins de métadonnées pour l'analyse du comportement des participants lorsqu'un réseaux social est décentralisé. Par exemple, un graphe social ne sera probablement pas mis à la disposition du reste des participants grâce à une API public pour les développeurs comme c'est le cas dans Twitter. De plus, la collecte de ces métadonnées peut s'avérer difficile voire impossible à réaliser en raison des choix de conception de l'application. Sans oublier que l'application d'algorithmes de classification par apprentissage automatique visant à détecter les abus peut également s'avérer plus difficile.

Notre approche est dans un premier temps de développer un ensemble de fonctionnalités qui détecte les comportements malveillants dans un réseaux social centralisé et public. Ensuite, nous évaluons la possibilité de porter chacune de nos caractéristiques dans un réseau social décentralisé qui respecterait la vie privée de ces utilisateurs et où ces métadonnées peuvent être impossibles à collecter. Par ailleurs, nous essayons d'évaluer l'impact de chaque caractéristique du point de vue de la vie privée afin de prendre conscience qu'elles sont les données les plus sensibles, les plus importantes, etc., et comment nous serions capable de les afficher ou de les recueillir de manière à protéger la vie privée des utilisateurs pour classifier des abus.

Nous utiliserons la technique “private set intersection” pour évaluer si le graphe social est adéquat pour la détection d'abus. Disposer de tels protocoles peut s'avérer utile pour les classificateurs, afin d'obtenir un résultat qui détecte raisonnablement les abus tout en respectant la vie privée. Pour arriver à cette conclusion, nous analyserons la sécurité et la résilience de chacune des caractéristiques que nous avons développé contre des adversaires “forts” dans les réseaux sociaux. De telles analyses peuvent aider les futurs développeurs de réseaux sociaux décentralisés à faire les bon choix de conception. Ceux-ci doivent trouver un compromis entre la protection de la vie privée et la détection des abus, ce qui décourage finalement une telle inconduite.

Le principal objectif de cette thèse est d'évaluer les protocoles qui prennent en considération la protection de la vie privée et qui nécessitent seulement des métadonnées locales ou des métadonnées non discriminantes pour détecter les comportements malveillants sur les réseaux sociaux décentralisés. En appliquant des techniques d'analyse de réseaux sociaux qui réduisent la quantité de métadonnées sensibles obtenue, nous obtenons des résultats acceptables comparé aux techniques qui ne préservent pas la vie privée. De plus, nous prévoyons d'élaborer une série de recommandations sur la manière de construire de futurs réseaux sociaux décentralisés qui découragent par défaut les comportements abusifs tout en respectant la vie privée de chacun.

Finalement, nous espérons que nos recherches inspireront de futurs travaux dans le domaine de la protection de la vie privée et de la décentralisation des réseaux sociaux en ligne afin de protéger les métadonnées des utilisateurs par choix de conception. Nous pensons à plusieurs candidats, parmi lesquels SecuShare ou Peergos, qui offre chacun des infrastructures complètes et prometteuses sur lesquelles il est possible de développer et de tester à grande échelle de tels systèmes. SecuShare est développé à INRIA et basé sur la plateforme GNUnet alors que Peergos est construit sur IPFS (the Inter-Planetary File System) [11]. Nous sommes en contact avec les deux groupes ci-dessus, et l'idée d'intégrer notre protocole dans leur pile réseau semble les intéresser, notamment parce que la découverte de contacts protégeant la vie privée est encore un problème ouvert sur ces types de systèmes. Des recherches ont déjà étaient menées dans ce sens, celles-ci évaluent l'adéquation de structure de données connue comme les filtres de bloom, d'autres proposent des techniques de récupération d'informations privées, et bien sûr la littérature sur le “private set intersection” que nous avons utilisé dans notre travail.

Nous nous attendons à un intérêt croissant pour ce domaine de la part de la communauté de la recherche intéressée par la protection de la vie privée. A plus long terme, nous aimerions implémenter une version fonctionnelle de notre protocole et mener des expérimentations pour vérifier les hypothèses que nous avons formulées. Pour cela nous devrons construire un système de test réaliste qui nous guidera pour améliorer la conception et les garanties de sécurité du protocole tout en prévoyant un déploiement chez un réseaux social décentralisé.

Bibliography

[1] Detecting insults in social commentary. https://www.kaggle.com/c/detecting-insults-in-social-commentary/data

[2] Facebook data use policy. https://www.facebook.com/policy.php

[3] Building businesses and potential threats with online social networks. http://la.trendmicro.com/media/misc/spotlight-building-businesses-and-potential-threats-en.pdf April 2010.

[4] Caution on twitter urged as tourists barred from US. http://www.bbc.com/news/technology-16810312 March 2012.

[5] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. Tensorflow : A system for large-scale machine learning. Technical report, Google Brain, 2016. arXiv preprint.

[6] P. Alves and P. Ferreira. Anonylikes : Anonymous quantitative feedback on social networks. In ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing, pages 466–484. Springer, 2013.

[7] M. Atallah, M. Bykova, J. Li, K. Frikken, and M. Topkara. Private collaborative forecasting and benchmarking. In Proceedings of the 2004 ACM workshop on Privacy in the electronic society, WPES '04, pages 103–114, New York, NY, USA, 2004. ACM.

[8] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x ? : anonymized social networks, hidden patterns, and structural steganography. Proceedings of the 16th International Conference on the World Wide Web (WWW), 54 :181–190, 2007.

[9] L. Becchetti, P. Boldi, C. Castillo, and A. Gionis. Efficient semi-streaming algorithms for local triangle counting in massive graphs. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 16–24. ACM, 2008.

[10] A. Ben-David, N. Nisan, and B. Pinkas. FairplayMP : a system for secure multi-party computation. In Proceedings of the 15th ACM conference on Computer and communications security, CCS '08, pages 257–266. ACM, 2008.

[11] J. Benet. Ipfs-content addressed, versioned, p2p file system. arXiv preprint arXiv :1407.3561, 2014.

[12] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In Collaboration, electronic messaging, anti-abuse and spam conference (CEAS), volume 6, page 12, 2010.

[13] M. Bertier, D. Frey, R. Guerraoui, A.-M. Kermarrec, and V. Leroy. The gossple anonymous social network. In Proceedings of the ACM/IFIP/USENIX 11th International Conference on Middleware, Middleware '10, pages 191–211, Berlin, Heidelberg, 2010. Springer-Verlag.

[14] A. Bessi, M. Coletto, G. A. Davidescu, A. Scala, G. Caldarelli, and W. Quattrociocchi. Science vs Conspiracy : collective narratives in the age of (mis)information. CoRR, abs/1408.1, 2014.

[15] A. Beutel, W. Xu, V. Guruswami, C. Palow, and C. Faloutsos. Copycatch : Stopping group attacks by spotting lockstep behavior in social networks. In Proceedings of the 22Nd International Conference on World Wide Web, WWW '13, pages 119–130, New York, NY, USA, 2013. ACM.

[16] C. M. Bishop. Pattern Recognition and Machine Learning. 1613-9011. Springer-Verlag New York, 1 edition, 2006.

[17] C. Blundo, E. De Cristofaro, and P. Gasti. Espresso : efficient privacy-preserving evaluation of sample set similarity. In Data Privacy Management and Autonomous Spontaneous Security, pages 89–103. Springer, 2013.

[18] P. Bogetoft, D. L. Christensen, I. Damgård, M. Geisler, T. Jakobsen, M. Krøigaard, J. D. Nielsen, J. B. Nielsen, K. Nielsen, J. Pagter, M. Schwartzbach, and T. Toft. Financial cryptography and data security. chapter Secure Multiparty Computation Goes Live, pages 325–343. Springer-Verlag, Berlin, Heidelberg, 2009.

[19] P. Boldi, F. Bonchi, A. Gionis, and T. Tassa. Injecting Uncertainty in Graphs for Identity Obfuscation. Proc. VLDB Endow., 5(11) :1376–1387, July 2012.

[20] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. Journal of Computational Science, 2(1) :1–8, 2011. http ://dx.doi.org/10.1016/j.jocs.2010.12.007.

[21] F. Bonchi, A. Gionis, and T. Tassa. Identity obfuscation in graphs through the information theoretic lens. Information Sciences, 275(0) :232–256, 2014.

[22] D. Boneh, X. Boyen, and H. Shacham. Short Group Signatures, pages 41–55. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[23] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. In Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security : Advances in Cryptology, ASIACRYPT '01, pages 514–532, London, UK, UK, 2001. Springer-Verlag.

[24] Y. Boshmaf, D. Logothetis, G. Siganos, J. Lería, J. Lorenzo, M. Ripeanu, and K. Beznosov. Integro : Leveraging victim prediction for robust fake account detection in osns. In NDSS, volume 15, pages 8–11. Citeseer, 2015.

[25] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. The socialbot network : When bots socialize for fame and money. In Proceedings of the 27th Annual Computer Security Applications Conference, ACSAC '11, pages 93–102, New York, NY, USA, 2011. ACM.

[26] Y. Boshmaf, M. Ripeanu, K. Beznosov, and E. Santos-Neto. Thwarting fake osn accounts by predicting their victims. In Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security (AI-Sec), 2015.

[27] L. Breiman. Arcing the edge. Technical report, Technical Report 486, Statistics Department, University of California at Berkeley, 1997.

[28] L. Breiman. Random forests. Machine learning, 45(1) :5–32, 2001.

[29] L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and regression trees. 1984.

[30] R. L. Brennan and D. J. Prediger. Coefficient kappa : Some uses, misuses, and alternatives. Educational and psychological measurement, 41(3) :687–699, 1981.

[31] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. Computer networks, 33(1) :309–320, 2000.

[32] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher. Min-wise independent permutations. In Proceedings of the thirtieth annual ACM symposium on Theory of computing, pages 327–336. ACM, 1998.

[33] S. Buchegger, D. Schiöberg, L.-H. Vu, and A. Datta. Peerson : P2p social networking : Early experiences and insights. In Proceedings of the Second ACM EuroSys Workshop on Social Network Systems, SNS '09, pages 46–52. ACM, March 2009.

[34] E. E. Buckels, P. D. Trapnell, and D. L. Paulhus. Trolls just want to have fun. Personality and individual Differences, 67 :97–102, 2014.

[35] M. Burkhart, M. Strasser, D. Many, and X. Dimitropoulos. SEPIA : privacy-preserving aggregation of multi-domain network events and statistics. In Proceedings of the 19th USENIX conference on Security, USENIX Security'10, pages 15–15, Berkeley, CA, USA, 2010. USENIX Association.

[36] E. Cambria, P. Chandra, A. Sharma, and A. Hussain. Do not feel the trolls. In CEUR Workshop Proceedings, volume 664, 2010.

[37] A. Campan and T. M. Truta. A clustering approach for data and structural anonymity in social networks. In ACM SIGKDD International Workshop on Privacy, Security, and Trust (PinKDD), pages 1–10, 2008.

[38] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro. 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, pages 15–15, Berkeley, CA, USA, 2012. USENIX Association.

[39] Q. Cao, X. Yang, J. Yu, and C. Palow. Uncovering large groups of active malicious accounts in online social networks. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, CCS '14, pages 477–488, New York, NY, USA, 2014. ACM.

[40] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi. Measuring user influence in twitter : The million follower fallacy. ICWSM, 10(10-17) :30, 2010.

[41] C.-C. Chang and C.-J. Lin. Libsvm : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2(3) :27, 2011.

[42] A. Chen. The agency. `http ://www .nytimes .com/2015/06/07/magazine/the-agency .html ?_r=2`, June 2015.

[43] J. Cheng, A. W.-c. Fu, and J. Liu. K-isomorphism : privacy preserving network publication against structural attacks. International Conference on Management of Data (SIGMOD), pages 459—-470, 2010.

[44] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu. Tools for privacy preserving distributed data mining. SIGKDD Explor. Newsl., 4(2) :28–34, Dec. 2002.

[45] R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. European transactions on Telecommunications, 8(5) :481–490, 1997.

[46] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi. Fame for sale : efficient detection of fake twitter followers. Decision Support Systems, 80 :56–71, 2015.

[47] E. D. Cristofaro, C. Soriente, G. Tsudik, and A. Williams. Tweeting with Hummingbird : Privacy in Large-Scale Micro-Blogging OSNs. IEEE Data Eng. Bull., 35 :93–100, 2012.

[48] M. Dadvar and F. De Jong. Cyberbullying detection : a step toward a safer internet yard. In Proceedings of the 21st international conference companion on World Wide Web, pages 121–126. ACM, 2012.

[49] G. Danezis and P. Mittal. Sybilinfer : Detecting sybil nodes using social networks. In NDSS. San Diego, CA, 2009.

[50] M. De Choudhury, Y.-R. Lin, H. Sundaram, K. S. Candan, L. Xie, A. Kelliher, et al. How does the data sampling strategy impact the discovery of information diffusion in social media ? ICWSM, 10 :34–41, 2010.

[51] E. De Cristofaro, P. Gasti, and G. Tsudik. Fast and private computation of cardinality of set intersection and union. In Cryptology and Network Security : 11th International Conference, CANS, pages 218–231, Berlin, Heidelberg, 2012. Springer.

[52] E. De Cristofaro, S. Jarecki, J. Kim, and G. Tsudik. Privacy-preserving policy-based information transfer. In International Symposium on Privacy Enhancing Technologies Symposium, pages 164–184. Springer Berlin Heidelberg, 2009.

[53] E. De Cristofaro, C. Soriente, G. Tsudik, and A. Williams. Hummingbird : Privacy at the time of Twitter. In Proceedings - IEEE Symposium on Security and Privacy, pages 285–299, 2012.

[54] E. De Cristofaro and G. Tsudik. Practical private set intersection protocols with linear computational and bandwidth complexity. In Financial Cryptography and Data Security, pages 143–159. http ://eprint.iacr.org/2009/491.pdf, 2010.

[55] K. Dinakar, R. Reichart, and H. Lieberman. Modeling the Detection of Textual Cyberbullying. Association for the Advancement of Artificial Intelligence, pages 11–17, 2011.

[56] I. Dlala, D. Attiaoui, A. Martin, and B. Ben Yaghlane. Trolls identification within an uncertain framework. In Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on, pages 1011–1015, Nov 2014.

[57] J. R. Douceur. The sybil attack. In Revised Papers from the First International Workshop on Peer-to-Peer Systems, IPTPS '01, pages 251–260, London, UK, UK, 2002. Springer-Verlag.

[58] Y. Eudes and C. Grothoff. Skynet, le programme ultra-secret de la nsa créé pour tuer. Le Monde, (20.10.2015), January 2015.

[59] N. S. Evans, B. Polot, and C. Grothoff. Efficient and secure decentralized network size estimation. IFIP International Conferences on Networking, 2012.

[60] A. Fisher. How jihadist networks maintain a persistent online presence. Perspectives on Terrorism, 9(3), 2015.

[61] P. W. Fong, M. Anwar, and Z. Zhao. A privacy preservation model for facebook-style social network systems. In European Symposium on Research in Computer Security, pages 303–320. Springer, 2009.

[62] M. Freitas. Twister, peer-to-peer microblogging.

[63] Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In European conference on computational learning theory, pages 23–37. Springer, 1995.

[64] M. Gabielkov and A. Legout. The complete picture of the twitter social graph. In Proceedings of the 2012 ACM Conference on CoNEXT Student Workshop, CoNEXT Student '12, pages 19–20, New York, NY, USA, 2012. ACM.

[65] P. Galán-García, J. G. de la Puerta, C. L. Gómez, I. Santos, and P. Bringas. Supervised machine learning for the detection of troll profiles in twitter social network : Application to a real case of cyberbullying. In International Joint Conference SOCO'13-CISIS'13-ICEUTE'13, volume 239 of Advances in Intelligent Systems and Computing, pages 419–428. Springer International Publishing, 2014.

[66] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and characterizing social spam campaigns. In Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, IMC '10, pages 35–47, New York, NY, USA, 2010. ACM.

[67] A. García-Recuero. Discouraging abusive behavior in privacy-preserving online social networking applications. In Proceedings of the 25th International Conference Companion on World Wide Web, WWW '16 Companion, pages 305–309, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.

[68] Á. García-Recuero. Trollslayer : a framework to annotate twitter abuse. `https : //doi .org/10 .5281/zenodo .800741` May 2017.

[69] Á. García-Recuero, J. Burdges, and C. Grothoff. Privacy-preserving abuse detection in future decentralised online social networks. In 11th International ESORICS Workshop in Data Privacy Management, DPM 2016. Springer Lecture Notes in Computer Science, 2016.

[70] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. Machine Learning, 63(1) :3–42, 2006.

[71] B. Gipp, N. Meuschke, and A. Gernandt. Decentralized trusted timestamping using the crypto currency bitcoin. In iConference. iSchools, 2015.

[72] B. Greschbach, G. Kreitz, and S. Buchegger. The devil is in the metadata—new privacy challenges in decentralised online social networks. In Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on, pages 333–339. IEEE, 2012.

[73] C. Grothoff and J. M. Porup. The NSA's SKYNET program may be killing thousands of innocent people. ARS Technica UK, 2016.

[74] J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. Radiology, 143(1) :29–36, 1982.

[75] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. Anonymizing social networks. Computer Science Department Faculty Publication Series, page 180, 2007.

[76] C. Hazay and K. Nissim. Efficient set operations in the presence of malicious adversaries. In International Workshop on Public Key Cryptography, pages 312–331. Springer, 2010.

[77] S. Hinduja and J. W. Patchin. Bullying, cyberbullying and suicide. Archives of Suicide Research, 14(3), 2010.

[78] H. Hosseinmardi, S. A. Mattson, R. I. Rafiq, R. Han, Q. Lv, and S. Mishra. Analyzing labeled cyberbullying incidents on the instagram social network. In International Conference on Social Informatics, pages 49–66. Springer, 2015.

[79] H. Hu, G.-J. Ahn, and J. Jorgensen. Detecting and resolving privacy conflicts for collaborative data sharing in online social networks. In Proceedings of the 27th Annual Computer Security Applications Conference on - ACSAC '11, page 103, 2011.

[80] P. Jaccard. The distribution of the flora in the alpine zone. New phytologist, 11(2) :37–50, 1912.

[81] T. F. A. Jamie Wilkinson and T. (F.A.T.). Google's official list of bad words. http ://fffff.at/googles-official-list-of-bad-words/, July 2011.

[82] S. Jarecki and X. Liu. Efficient oblivious pseudorandom function with applications to adaptive ot and secure computation of set intersection. In Theory of Cryptography Conference, pages 577–594. Springer, 2009.

[83] S. Jarecki and X. Liu. Fast secure computation of set intersection. In International Conference on Security and Cryptography for Networks, pages 418–435. Springer, 2010.

[84] T. Khazaei, L. Xiao, R. Mercer, and A. Khan. Privacy behaviour and profile configuration in twitter. In Proceedings of the 25th International Conference Companion on World Wide Web, pages 575–580. International World Wide Web Conferences Steering Committee, 2016.

[85] L. Kissner and D. Song. Privacy-preserving set operations. In Annual International Cryptology Conference, pages 241–257. Springer, 2005.

[86] A. Kramer, J. Guillory, and J. Hancock. Experimental evidence of massive-scale emotional contagion through social networks. Proceedings of the National Academy of Sciences of the United States of America, 2013.

[87] E. Kross, P. Verduyn, E. Demiralp, J. Park, D. S. Lee, N. Lin, H. Shablack, J. Jonides, and O. Ybarra. Facebook use predicts declines in subjective well-being in young adults. PloS one, 8(8) :e69841, 2013.

[88] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tompkins, and E. Upfal. The web as a graph. In Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pages 1–10. ACM, 2000.

[89] J. Kunegis, A. Lommatzsch, and C. Bauckhage. The slashdot zoo : Mining a social network with negative edges. In Proceedings of the 18th International Conference on World Wide Web, WWW '09, pages 741–750, New York, NY, USA, 2009. ACM.

[90] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media ? In Proceedings of the 19th International Conference on World Wide Web, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM.

[91] C. Langos. Cyberbullying : The Challenge to Define, volume 15. 2012. http ://dx.doi.org/10.1089/cyber.2011.0588.

[92] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers : social honeypots+ machine learning. In SIGIR'10, July 19–23, 2010, Geneva, Switzerland. Copyright, pages 435–442, 2010.

[93] C. Lesniewski-Lass and M. F. Kaashoek. Whanau : A sybil-proof distributed hash table. NSDI, 2010.

[94] J. Lin and A. Kolcz. Large-scale machine learning at twitter. In Proceedings of the 2012 international conference on Management of Data SIGMOD 12, pages 793–804, 2012.

[95] V. López, A. Fernández, S. García, V. Palade, and F. Herrera. An insight into classification with imbalanced data : Empirical results and current trends on using data intrinsic characteristics. Information Sciences, 250 :113–141, 2013.

[96] D. Luxton, J. June, and J. Fairall. Social media and suicide : A public health perspective. American Journal of Public Health, 102 :195–200, May 2012.

[97] W. magazine. Twitter makes changes to stem online abuse. http ://www.wired.co.uk/news/archive/2015-02/27/twitter-abuse.

[98] P. Mandeep K. Dhami. Behavioural science support for jtrig's (joint threat research and intelligence group's) effects and online humint operations. https ://firstlook.org/theintercept/2015/06/22/controversial-gchq-unit-domestic-law-enforcement-propaganda/, March 2011 2011.

[99] J. Mikians, L. Gyarmati, V. Erramilli, and N. Laoutaris. Detecting price and search discrimination on the internet. In Proceedings of the 11th ACM Workshop on Hot Topics in Networks, HotNets-XI, pages 79–84, New York, NY, USA, 2012. ACM.

[100] J. Mitchell-Wong, R. Kowalczyk, A. Roshelova, B. Joy, and H. Tsai. OpenSocial : From social networks to social ecosystem. In Proceedings of the 2007 Inaugural IEEE-IES Digital EcoSystems and Technologies Conference, DEST 2007, pages 361–366, 2007.

[101] A. Mohaisen, A. Yun, and Y. Kim. Measuring the mixing time of social graphs. In Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, IMC '10, pages 383–389, New York, NY, USA, 2010. ACM.

[102] A. Nazir, S. Raza, and C.-N. Chuah. Unveiling facebook : A measurement study of social network based applications. In Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement, IMC '08, pages 43–56, New York, NY, USA, 2008. ACM.

[103] H. H. Nguyen, A. Imine, and M. Rusinowitch. A Maximum Variance Approach for Graph Anonymization. In The 7th International Symposium on Foundations & Practice of Security FPS'2014. Inria, 2014.

[104] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang. Abusive language detection in online user content. In Proceedings of the 25th International Conference on World Wide Web, pages 145–153. International World Wide Web Conferences Steering Committee, 2016.

[105] D. O'Callaghan, N. Prucha, D. Greene, M. Conway, J. Carthy, and P. Cunningham. Online social media in the syria conflict : Encompassing the extremes and the in-betweens. In Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on, pages 409–416. IEEE, 2014.

[106] A. Odlyzko. The glorious promise of the post-truth world. Ubiquity, 2017(March) :2 :1–2 :7, Mar. 2017.

[107] F. J. Ortega. Detection of dishonest behaviors in on-line networks using graph-based ranking techniques. AI Communications, 26 :327–329, 2013.

[108] F. J. Ortega, J. A. Troyano, F. L. Cruz, C. G. Vallejo, and F. Enríquez. Propagation of trust and distrust for the detection of trolls in a social network. Computer Networks, 56(12) :2884 – 2895, 2012.

[109] Y. Peng, G. Kou, Y. Shi, and Z. Chen. Privacy-preserving data mining for medical data : Application of data partition methods. In Communications and Discoveries from Multidisciplinary Data, pages 331–340. Springer, 2008.

[110] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, S. Patil, A. Flammini, and F. Menczer. Truthy : mapping the spread of astroturf in microblog streams. In Proceedings of the 20th international conference companion on World wide web, pages 249–252. ACM, 2011.

[111] J. Ratkiewicz, M. D. Conover, M. Meiss, B. Gonc, A. Flammini, F. Menczer, B. Gonçalves, A. Flammini, and F. Menczer. Detecting and Tracking Political Abuse in Social Media. In ICWSM, pages 297–304, 2011.

[112] A. Shakimov, A. Varshavsky, L. P. Cox, and R. Cáceres. Privacy, cost, and availability tradeoffs in decentralized OSNs. In Proceedings of the 2nd ACM workshop on Online social networks - WOSN '09, page 13, 2009.

[113] L. Silva, M. Mondal, D. Correa, F. Benevenuto, and I. Weber. Analyzing the targets of hate in online social media. arXiv preprint arXiv :1603.07709, 2016.

[114] C. Soghoian. Why google won't protect you from big brother [Online]. TEDxSanJose, CA, May 2012.

[115] T. Stein, E. Chen, and K. Mangla. Facebook immune system. In Proceedings of the 4th Workshop on Social Network Systems, page 8. ACM, 2011.

[116] Techopedia. Definition - what does troll mean ? http ://www.techopedia.com/definition/429/troll.

[117] K. Thomas, C. Grier, D. Song, and V. Paxson. Suspended accounts in retrospect : An analysis of twitter spam. In Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, IMC '11, pages 243–258, New York, NY, USA, 2011. ACM.

[118] K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson. Trafficking fraudulent accounts : the role of the underground market in twitter spam and abuse. In USENIX Security Symposium, 2013.

[119] H. Vandebosch and K. Van Cleemput. Defining cyberbullying : a qualitative research into the perceptions of youngsters. Cyberpsychology & behavior : the impact of the Internet, multimedia and virtual reality on behavior and society, 11 :499–503, 2008.

[120] T. Vanhove, P. Leroux, W. Tim, and D. T. Filip. Towards the design of a platform for abuse detection in OSNs using multimedial data analysis. In IFIP/IEEE IM2013Workshop : 5th InternationalWorkshop onManagement of the Future Internet (ManFI), pages 1195–1198, 2013.

[121] G. Vermeulen. Article 8, eu gdpr, conditions applicable to child's consent in relation to information society services. Essential Texts on International and European Criminal Law 9th edition, January 2017.

[122] G. Wang, M. Mohanlal, C. Wilson, X. Wang, M. Metzger, H. Zheng, and B. Y. Zhao. Social turing tests : Crowdsourcing sybil detection. arXiv preprint arXiv :1205.3856, 2012.

[123] S. D. Warren and L. D. Brandeis. The right to privacy. Harvard Law Review, 4(5) :pp. 193–220, 1890.

[124] M. J. Warrens. Inequalities between multi-rater kappas. Advances in Data Analysis and Classification, 4(4) :271–286, 2010.

[125] A. C. Yao. Protocols for secure computations. In Proceedings of the 23rd Annual Symposium on Foundations of Computer Science, SFCS '82, pages 160–164, Washington, DC, USA, 1982. IEEE Computer Society.

[126] D. Yin, Z. Xue, L. Hong, B. D. Davison, A. Kontostathis, and L. Edwards. Detection of Harassment on Web 2.0. In In Proceedings of the Content Analysis in the WEB 2.0 (CAW2.0) Workshop at WWW2009, ., 2009.

[127] H. Yu, P. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit : A near-optimal social network defense against sybil attacks. Networking, IEEE/ACM Transactions on, 18(3) :885–898, June 2010.

[128] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybil guard : defending against sybil attacks via social networks. In ACM SIGCOMM Computer Communication Review, volume 36, pages 267–278. ACM, 2006.

[129] C. Zhang, J. Sun, X. Zhu, and Y. Fang. Privacy and security for online social networks : Challenges and opportunities. IEEE Network, 24 :13–18, 2010.

[130] E. Zheleva and L. Getoor. Preserving the privacy of sensitive relationships in graph data. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4890 LNCS :153–171, 2008.

[131] J. Zhu, H. Zou, S. Rosset, and T. Hastie. Multi-class adaboost. Statistics and its Interface, 2(3) :349–360, 2009.