Identification of Troll Vulnerable Targets in Online Social Networks

A Thesis

submitted to the designated
by the General Assembly of Special Composition
of the Department of Computer Science and Engineering
Examination Committee

by

Paraskevas Tsantarliotis

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

WITH SPECIALIZATION
IN SOFTWARE

University of Ioannina

July 2016

Dedication

Dedicated to my grandmother Melpomeni Iliopoulou.

Acknowledgements

First and foremost, I would like to thank my advisor Professor Evaggelia Pitoura and my co-advisor Associate Professor Panayiotis Tsaparas. Their guidance and continuous support helped me throughout this thesis and the entire graduate program.

Besides my advisors, I would like to thank the last member of my thesis committee Associate Professor Panos Vassiliadis for his encouragement and insightful comments. Special thanks are given to my fellow labmates in *D.A.T.A. Lab* for the stimulating discussions and for all the fun we had the last two years.

I must express my sincere gratitude to my family, their support and encouragement worth more than I can express on paper. It is a pleasure to thank my friends Kostas, Vasso, Giannis and Kostas for the wonderful times we shared.

Last but not least, I would like to thank my girlfriend Maroussa for all her love and support.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Abstract

Paraskevas Tsantarliotis, M.Sc. in Computer Science, Department of Computer Science and Engineering, University of Ioannina, Greece, July 2016.
Identification of Troll Vulnerable Targets in Online Social Networks.
Advisor: Evaggelia Pitoura, Professor.

"Trolling" describes a range of antisocial online behaviors that aim at disrupting the normal operation of online social networks and media. Combating trolling is an important problem in the online world. Existing approaches rely on human-based or automatic mechanisms for identifying trolls and troll posts. In this work, we take a novel approach to the trolling problem: our goal is to identify the targets of the trolls, so as to prevent trolling before it happens. We thus define the troll vulnerability prediction problem, where given a post we aim at predicting whether it is vulnerable to trolling. Towards this end, we define a novel troll vulnerability metric of how likely a post is to be attacked by trolls, and we construct models that use features from the content and the history of the post for the prediction. Our experiments with real data from Reddit demonstrate that our approach is successful in recalling a large fraction of the troll-vulnerable posts.

Εκτεταμενη Περιληψη

Παρασκευάς Τσανταρλιώτης, Μ.Δ.Ε. στην Πληροφορική, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Ιωαννίνων, Ιούλιος 2016.
Αναγνώρηση Ευπαθών Στόχων σε Κακόβουλες Επιθέσεις (Trolls) σε Διαδικτυακά Κοινωνικά Δίκτυα.
Επιβλέπων: Ευαγγελία Πιτουρά, Καθηγήτρια.

Τα κοινωνικά μέσα παίζουν σημαντικό ρόλο στις μέρες μας. Καθημερινα, δισεκατομύρια από χρήστες από διάφορες χώρες συμμετέχουν σε κοινωνικά δίκτυα, φόρουμς και υπηρεσίες micro-blogging. Θα μπορούσαμε να πούμε ότι συμμετέχουν σε εικονικούς διαλόγους, που διεξάγονται σε παγκόσμια κλίμακα, όπου μπορουν να συζητούν και να ανταλλάσουν απόψεις με ανθρώπους από όλο τον κόσμο. Ωστόσο, υπάρχουν κάποιοι χρήστες που έχουν διαφορετικά κίνητρα για τέτοιες εικονικές συζητήσεις. Η συνισφορά τους σε τέτοιεις συζητήσεις δεν είναι θετική, αντίθετα προσπαθούν να προκαλέσουν αναστάτωση και να διασπάσουν τη συζήτηση. Τέτοιοι χρήστες συνήθως αποκαλούνται διαδικτυακά *τρολλς*. Τα τρολλς αποτελούν ένα σημαντικό πρόβλημα στα κοινωνικά μέσα γιατί υπονομεύουν την ομαλή τους λειτουργία.

Η έννοια του τρολλ έχει χρησιμοποιηθεί για να χαρακτηρίσει μια ευρεία γκάμα συμπεριφορών σε εικονικές συζητήσεις. όπως αστεϊσμός εκτός θέματος και υβριστική συμπεριφορά. Χαρακτηριστικά παραδείγματα συπεριφοράς ειναι ο χλευασμός και η απαξίωση των συνομιλιτών του και η διακίνηση ψευδών πληροφοριών ή ειδήσεων. Επίσης, πολλές φορές παρουσιάζουν πιο επιθετική συμπεριφορά που θα μπορούσε να χαρακτηριστεί εγκληματική. Οι στόχοι των τρολλς δεν είναι εμφανείς, ωστόσο φαίνεται οτι αρέσκονται στο να δημιουργούν σύγχυση και να εκνευρίζουν τους συνομιλητές τους. Τα διαδικτυακά τρολλς συμμετέχουν στις συζητήσεις παριστάνοντας τους κανονικούς χρήστες και προσπαθούν να επιτεύξουν τους στόχους τους.

Είναι εμφανές ότι το διαδικτυακό τρολλάρισμα είναι ένα σημαντικό πρόβλημα στα κοινωνικά μέσα. Πολλά κοινωνικά δίκτυα έχουν αναπτύξει διάφορους μηχανισμούς για να αντιμετωπίσουν αυτό το φαινόμενο. Τα τελευταία χρόνια το πρόβλημα αυτό έχει κεντρίσει το ενδιαφέρον της επιστημονικής κοινότητας. Οι περισσότερες εργασίες έχουν επικεντρωθεί στην ανίχνευση μηνυμάτων με ακατάλληλο περιεχόμενο και των χρηστών που δημοσιεύουν τέτοια μηνύματα. Ωστόσο, το πρόβλημα δεν φαίνεται να έχει λυθεί.

Σε αυτή τη δουλειά, προσεγγίζουμε το πρόβλημα από μία διαφορετική γωνία. Αντί να προσπαθήσουμε να ανιχνεύσουμε τρολλς και τα μηνύματα τους, προσπαθούμε να αναγνωρίσουμε πιθανούς στόχους των τρολλς. Πιο συγκεκριμένα, δεδομένου ενός μηνύματος, προσπαθούμε να προβλέψουμε αν θα προσελκύσει τρολλς, δηλαδή αν είναι ευάλωτο από τρολλς. Προκειμένου να το επιτύχουμε αυτό χρειαζόμαστε μία μετρική που να ποσοτικοποιεί την επικινδυνότητα του μηνύματος.

Αρχικά, ορίζουμε τρεις βασικές ιδιότητες που πρέπει να πληροί αυτή η μετρική και στη συνέχεια βασιζόμενοι σε αυτές τις ιδιότητες ορίζουμε τον *Βαθμό Ευπάθειας από Τρολλς (Troll Vulnerablility Rank)*, την οποία ονομάζουμε *TVRank*. Η μετρική αυτή βασίζεται στην ποσότητα των τρολλ μηνυμάτων που ακολουθούν ένα συγκεκριμένο μήνυμα.

Χρησιμοποιώντας την μετρική *TVRank*, ορίζουμε το *πρόβλημα της πρόβλεψης ευπαθών μηνυμάτων από τρολλς*. Στόχος μας είναι να προβλέψουμε σχόλια τα οποία θα αποκτήσουν μεγάλες τιμές της μετρικής *TVRank*. Χρησιμοποιούμε μοντέλα τα οποία εκπαιδεύουμε με ιστορικά δεδομένα και τα εφαρμόζουμε σε νέα μηνύματα. Η αξιολόγηση των μοντέλων μας γίνεται σε πραγματικά δεδομένα από το Reddit και έδειξε ότι τα μοντέλα μας μπορούν να αναγνωρίσουν ένα μεγάλο ποσοστό από ευπαθή μηνύματα.

Chapter 1

Introduction

1.1 Introduction

Online social media and networks have emerged as the principal forum for the public discourse. Billions of users from diverse cultures and backgrounds gather in online social networks (e.g., Facebook), microblogging services (e.g., Twitter), or discussion forums (e.g., Reddit), where they engage in discussions and exchange opinions on all possible topics, creating a dialogue at a global scale. However, this open global forum is threatened by users that actively try to undermine its operation. Such users engage in discussions without the intention of constructively contributing to the dialogue, but rather to disrupt it. They act as agents of chaos on the Internet, and they are commonly referred to as *trolls*.

Trolling is an inclusive term that characterizes different types of disruptive online behavior ranging from off-topic joking comments to offensive and threatening behavior. Trolls enter online social networks and media as ordinary users, and cause havoc, disrupting the normal conversation and flow of information. They propagate misinformation, obfuscate the issues and at times threaten and bully other internet users. Different from spammers, trolls do not aim at a financial gain; creating disarray is actually a goal in itself. Typical examples of trolling behavior include mocking

and discrediting discussion participants, inciting and escalating arguments, and impersonating expert users while spreading bad advice and false information.

Trolling is a serious issue that undermines the operation of social networks and media and their role as a global channel of communication. Thus, combating trolls is a top priority for all major user engagement portals, such as news portals, social networks, and social media sites, and it is a problem that requires immediate and effective solutions. The common practice against trolls is to simply ignore them, hoping that the lack of attention will drive them away (also known as "Do Not Feed The Trolls"[1]). Some of the largest social networks have deployed user-driven mechanisms to detect trolling behavior, where users report abusive behavior to the system, and moderators suspend, ban or remove the perpetrators from the community [1]. Given the importance of early detection, considerable effort has been dedicated in devising algorithms for automatically detecting trolls and trolling behavior in both research and practice [2, 3, 4, 5].

Even when successful, troll detection does not fully address the problem. First, trolls are very good at working the system, getting around bans by using different usernames, or masking the content of their postings [6]. More importantly, all these measures are *reactive*: they are usually applied after a defamatory, threatening, or misleading comment has already been posted. In many cases this is too late; the damage is already done.

In this work, we take a different approach to the trolling problem. Instead of detecting trolls, we focus on identifying the possible targets of trolls. Given a post, we ask whether it is likely to attract trolls in the future, that is, how *vulnerable* the post is to trolls. To estimate the vulnerability of a post, we define the *Troll Vulnerability Rank*, called *TVRank*, based on the amount of trolling activity that followed that post. Using the *TVRank*, we can define the *Troll Vulnerability prediction problem*, where the goal is to predict which posts will acquire high *TVRank* value. Using historical data, we train models for the prediction task and apply them to new posts.

Our approach has several advantages, compared to traditional troll detection mechanisms:

- It is *pro-active*. Rather than detecting and removing trolls after they occur, we try to anticipate the troll activity and take preventive actions to eliminate it before it appears. Vulnerability prediction is a useful tool to both social media

[1] https://en.wikipedia.org/wiki/Internet_troll

moderators, who can be on guard for high-risk posts, as well as for ordinary users, who can have an advance warning if their posts are vulnerable.

- Modeling troll vulnerability offers valuable insights into what makes a post susceptible to trolling behavior. Although the characteristics of trolls have been studied in detail, there is little understanding about what makes a post a troll target.

- The *TVRank* value is a useful metric in itself. Disruptive behavior is likely to occur in social media. If it consists of sparse isolated incidents, then it can be absorbed by the normal operation of the system. It becomes a problem when it is targeted and intense. The *TVRank* offers a way to quantify the severity of the troll activity with respect to a post. Depending on the system sensitivity we can identify the post as being under troll attack.

- Troll vulnerability can be computed for troll posts as well. In this case, it provides a metric for *trolling escalation*, that is, it provides a measure of the degree to which a troll post will generate further trollings. Measuring and predicting troll escalation is again important in monitoring the behavior of a system.

In summary, in this work we make the following contributions.

- We define the novel problem of troll vulnerability prediction, where we want to predict if a post is likely to become the victim of a troll attack. To the best of our knowledge, we are the first to consider this problem.

- We propose *TVRank*, a metric for quantifying the vulnerability of a post to trolls. We define a set of properties that we want our metric to satisfy, and based on these properties we compute *TVRank* using a random walk with restarts.

- We build classification models for predicting troll vulnerability. Our models explore features that use the content of the post, the properties of the user that posted the content, as well as the history of the post in the discussion tree. We investigate the importance of the different features in the prediction task.

- We experiment with a real dataset from Reddit. We demonstrate that our model is able to recall a large fraction of the vulnerable posts with overall high accuracy.

1.2 Structure of the Dissertation

The rest of the thesis is structured as follows. Chapter 2 reviews the related work in trolling. In Chapter 3, we define the concept of Troll Vulnerability and the *TVRank*. In Chapter 4, we describe the dataset, the generation of the ground truth and we provide some statistical information about the dataset. In Chapter 5, we describe the classifier for predicting vulnerable posts, and in Chapter 6, we conclude our work.

Chapter 2

Related Work

In this chapter, we review previous works related to ours. The term *troll* has been widely used to characterize different types of anti-social and disruptive online behavior. Such behavior may range from off-topic joking to offensive and threatening behavior. Trolls have also attracted much negative attention from the media in the past few years and because of this, trolling has become equivalent with online harassment.

We categorize the previous works into two major categories. The first group includes works that study the antisocial behavior on a theoretical basis. The other group includes works that identify malicious behavior in online social settings. In particular, we review works that detect vandalism in Wikipedia[1], malicious users and inappropriate content in online social settings. Each of these groups is described in detail later in this chapter.

[1] https://www.wikipedia.org/

2.1 Antisocial Behavior

This group includes works that do not belong exclusively in the field of computer science; but also from other fields like psychology and sociology. Such works provide insights about the behavior of the users in online communities and factors that exacerbate antisocial behavior.

The author in [7] discusses the online disinhibition effect and defines six factors that affect it, including anonymity and invisibility. The online disinhibition effect is a loosening or complete abandonment of social restrictions and inhibitions that would otherwise be present in normal face-to-face interaction during interactions with others on the Internet. This disinhibition can affect the users in two opposing ways. Some users exhibit heartwarming tendencies and become more willing to open up to others (benign disnhibition). On the other hand, other users show a "darker" version of themselves by behaving inappropriately, without the fear of a meaningful punishment. This is called the toxic disnhibition, which can be related to trolls.

The online dishibition effect seems to align with the findings of the studies presented in [8, 9]. According to [9], the anonymity in the online social settings seems to encourage users to be impolite to other users. In addition, the authors in [8] tried to identify the motivations of posting benevolent and malicious comments online. The results showed that users post benevolent comments to encourage and help each other, whereas users post malicious comments to express anger, resolve feelings of dissatisfaction, etc.

In [10], the author studies the issue of the reliability and accountability of online personae. The purpose of this paper is to understand how identity is established in an online community and to examine the effects of identity deception and the conditions that cause it. According the author, a close examination of the user's identity (e.g., account name, language, signature, etc.) can reveal a great deal about the users and their credibility within the community. In addition, four types of identity deception are identified within text-based virtual communities, such as Usenet[2], including trolling.

Furthermore, the results in [11] indicate that there is a relation between trolls and sadism. The authors conducted two online studies with over 1200 participants, who took personality test regarding their internet commenting behavior. They found that Dark Tetrad[3] scores were highest among people who said trolling was included in

[2] https://en.wikipedia.org/wiki/Usenet

[3] https://en.wikipedia.org/wiki/Dark_triad#Dark_tetrad

their internet activities. The dark tetrad is a subject in psychology that focuses on four personality traits: sadism, narcissism, Machiavellianism and psychopathy. Use of the term ”dark” implies that people possessing these traits have malevolent qualities. Of all personality traits, sadism showed the most robust associations with trolling behavior.

2.2 Detection of Malicious Behavior

Due to its critical importance, the problem of identifying malicious behavior in online social settings has received considerable attention. Most existing techniques extract a variety of features from the available data and use them to create models to detect such behavior. Commonly used features include textual, topic and sentiment characteristics of the posts, activity related metrics, such as post frequency, feedback from the participants, such as upvotes or likes, and moderator features, when available.

2.2.1 Detecting Vandalism on Wikipedia

Wikipedia is a free online encyclopedia which its users can mostly edit any article. Even though its openness is the key to success, it can also cause some trouble. Most of the edits in Wikipedia are constructive, however some edits are done in bad faith. Vandalism is defined as any edit that changes content in a way that deliberately compromises the integrity of Wikipedia. The most common and obvious types of vandalism include insertion of obscenities, crude humour and spam. Detecting vandalism on Wikipedia can be considered one of the earliest attempts to identify online malicious behavior.

The community has deployed several bots in order to detect and revert malicious edits. Such bots, initially, were simple, but over time they evolved to more complex systems. There, however, exists room for improvement. Many researchers have proposed methods to detect vandalism automatically.

In [12], the problem is seen as a binary classification problem. The authors manually studied vandalism cases to inspire a feature set based on meta-data and content–level properties. Their logistic regression classifier was able to outperform the best performing bots. Chin et al. [13] address the problem using by natural language processing techniques. They constructed statistical language models of an article from its

revision history and used them as features. In addition, they categorized vandalism into seven major types that are based on a basic taxonomy of Wikipedia actions. Their models outperformed the baseline approaches and excelled in detecting specific types of vandalism.

Another interesting approach is presented in [14]. The authors have built an automated system to detect vandalism on Wikipedia, using features that have been proposed in the bibliography. These features include natural language processing features [15], reputation features [16] and spatio-temporal features [17] extracted from the revision metadata. This feature combination performed better than all previous methods and established a new baseline for Wikipedia vandalism detection.

A more recent approach is presented in [18], where the authors focus on detecting vandals in Wikipedia. At first, the authors conduct an analysis of user behaviors and identify similarities and differences between benign users and vandals. Using the insights of the analysis, a model that can detect vandals is proposed. The model performs better than previous approaches. In addition, combining their model with previous approaches, the authors manage to achieve better performance and identify vandals faster than before.

2.2.2 Detection of Malicious Users

Related work in this line of research includes detection of malicious users in online communities. The purpose of these works is to identify users that exhibit inappropriate behavior and take actions against them.

In [19], the authors are trying to detect trolls in an online social network. The authors make the hypothesis that every troll profile is followed by the real profile of the user behind the fake one. They extract features from the user's profile, e.g., writing style and connections. The goal is to link a troll profile to the corresponding real profile using machine learning algorithms. They also provide a real life case in which their methodology was applied to detect and stop a cyberbullying situation in a real elementary school.

The authors of [20] detect trolls in Question Answering Communities (Q&AC). Their method is based on the belief function theory, which is used to solve problems with uncertain, incomplete or even missing data. They define a conflict measure that is used to measure, at first, the conflict between messages between different users and

eventually the conflict between the users. After calculating the conflict between the users, they applied the k-means method in order to distinguish trolls from the other users. The results of their approach in different simulated data prove its feasibility for detecting malicious users.

A more recent work is described in [5]. The authors focus on detecting users that exhibit antisocial behavior in online discussion communities. Antisocial users are users that were banned from these communities by the moderators. Studying comments from three different news communities, the authors claim that antisocial users write worse than other users over time and they become increasingly less tolerated by the community. In addition, they were able to identify the characteristics of the behavior of antisocial users and how their behavior changes through time. Using these insights they managed to built a classifier able to detect antisocial users early on, by observing only a few of their posts, with high accuracy.

Multi-player games is one of the most popular online activities that is also targeted by malicious users. The authors in [21] try to address bad behavior in online gaming, which is usually called *toxic* in such communities. Toxic players seem to have great impact in such communities. For instance, a quarter of customer support calls to online game companies are complaints about toxic players[4]. The purpose of their work is to predict whether users that have potentially exhibit toxic behavior will eventually punished by the community. Their results are very promising and provide opportunities for further research in toxic behavior

Troll Detection using Signed Social Networks

Another line of research in detecting malicious users assumes the availability of a signed social graph among users. A SSN is defined as $G = (V, E, W)$ where V is a set of users, $E \rightarrow V \times V$ is a set of edges, and $W : E \rightarrow [-1, +1]$ assigns a real valued weight from -1 to $+1$, indicating positive and negative relationships among users. Then, troll detection is modeled as a ranking problem in this graph.

While signed social networks are explicitly present in some social networks, e.g., Slashdot[5], they can also be extracted from other social networks, such as Facebook or Twitter. For instance, consider two users, u and v on YouTube; we could assign an edge from u to v based on how many videos of v were marked positively/negatively

[4]https://www.theguardian.com/technology/2006/jun/15/games.guardianweeklytechnologysection2

[5]`https://slashdot.org/`

by u. More complex techniques can also be considered by elaborating text content and natural language (NLP) techniques.

Related approaches use that use centrality measures to detect troll are presented in [22, 23]. The key idea is that users with low centrality are more likely to be malicious. In particular, in [22], the authors propose an iterative algorithm that calculates the centrality. In each step, the algorithm performs a set of user-defined graph transformations, called decluttering operations, and then recalculates the centrality. The algorithm terminates when the decluttering operations lead to no change. The results outperform previous works and also the algorithm is much faster than previous approaches.

Furthermore, there some approaches that use trust propagation to detect trolls in online signed social networks [24, 25]. The novelty in these approaches is that their method propagates both positive and negative trust in the network. This is important because negative opinions are as determining as the positive ones (or even more). The goal of this approach is to the users according to their trustworthiness, denoting the users who present a dishonest behavior, i.e., trolls, in the system. According to the authors of [25] their model can be easily modified in order to be used for other applications, such as link prediction.

2.2.3 Detection of Inappropriate Content

The works presented so far are used to detect vandalism on Wikipedia and malicious users. However, there are some works that focus on detecting inappropriate user content in online communities. Most of these works focus on characterizing whether posts or comments are trolling or not.

The authors of [2] propose a system that its goal is to detect and filter trolling posts. They use a technique, called sentic computing[6], to measure the "trollness" of a post. Sentic computing is an opinion mining and sentiment analysis paradigm to analyse the texts. At first, they identify the concepts most commonly used by trolls and then expanding the resulting knowledge base with semantically related concepts. The trollness of a post is defined as the concept-based similarity of the concepts contained in the post and the known concepts used by trolls. The post is characterized as trolling if the similarity trollness exceeds a certain threshold.

[6] http://www.sentic.net

A similar approach to [2] is presented in [4]. The authors focus on detecting trolling posts in Meneame[7] social news website, using an anomaly detection approach. To this end, they extract three different types of features from the comments; statistical, syntactic and opinion features. Then, considering a group of troll posts (control group) they classify each comment as trolling or non-trolling based on the deviation, i.e., the distance, of the comment from the control group. They experimented with different settings regarding the distance and compared their results with other supervised machine learning techniques.

The study in [3] detects personal insults on Yahoo!Buz[8] social news site. The retrieved comments were tagged them as insulting or not insulting, using the Amazon Mechanical Turk[9]. The authors built a model that uses a multi-step classifier that utilizes valence and relevance analysis, as well as two classifiers to detect insults and the object of the insults. Their experiment show good performance on detecting insults and the object of the insults, outperforming previous works.

Our approach differs from these works. The key novelty is that we turn the spotlight to the side of the trolling victim, aiming at characterizing her vulnerabilities, and estimating the risk of becoming a target of trolling. There is no previous work, to our knowledge, studying the problem of troll vulnerability of potential targets.

An interesting approach is presented in [26]. The authors investigate how *firestorms* on Twitter affect the relationships between users. Firestorm is called an event where a target (e.g. public figure, organization) receives a large amount of negative attention. We have to point out that firestorms is much different than trolling. Firestorms may include trolls, but not all participants in firestorms are trolls. Thus, this problem is much different than ours.

[7] https://www.meneame.net/

[8] https://en.wikipedia.org/wiki/Yahoo!_Buzz

[9] https://www.mturk.com/mturk/

Chapter 3

Model of Troll Vulnerability

In this section, we introduce the concept of troll vulnerability, and we define a metric to quantify it.

3.1 Preliminaries

Trolls pose a serious threat to online social media, since they undermine their normal operation. For example, it is common in social networks for trolls to cause havoc in the comments of an initial post. In order to address the problem of troll vulnerability we need to have some definition of what constitutes trolling. In the following we will use the following two informal definitions to characterize trolling behavior.

Definition 1. *Trolls are people that behave in a deceptive, destructive and disruptive manner in a social setting on the Internet, such as a social network. Their goal is to provoke other users and lure them in pointless conversations in order to emotionally compromise them.*

Definition 2. *Trollings are the posts/comments that are coming from trolls and aim to hurt specific people or groups.*

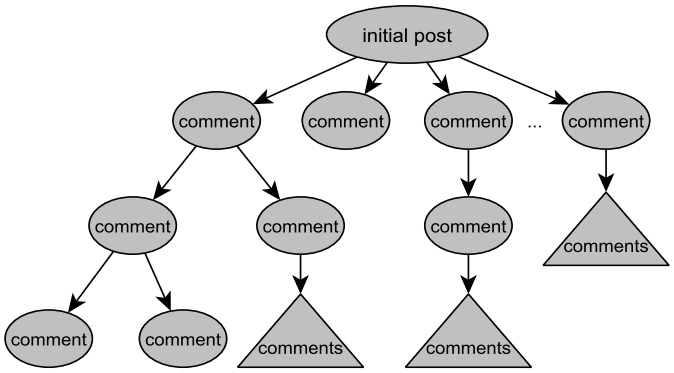


Figure 3.1: An example of a conversation tree.

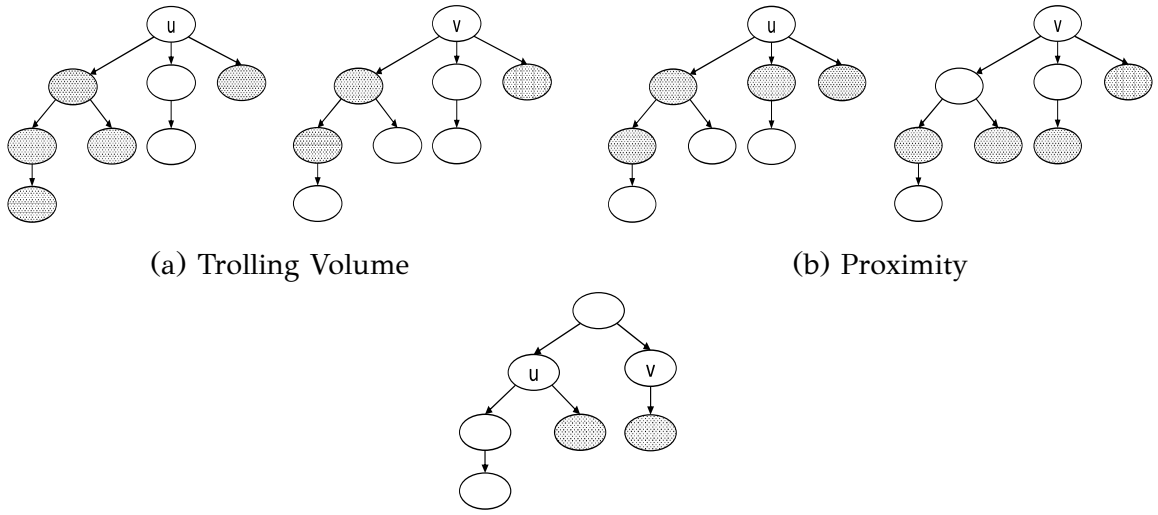
As we have already discussed, there is no consensus on what constitutes trolling behavior. We intentionally use a general definition, in order to capture different notions of trolling. We note that our definition of vulnerability is independent of the exact definition of trolling; depending on the specific application one could use the appropriate trolling definition.

Although, there has been previous research on detecting trolls and their posts, the problem of understanding which users, or what kind of published content are likely to become the target of trolls is vastly unexplored. In this work, we focus on characterizing and identifying posts that are vulnerable to trolls, that is, posts that are likely to attract trolls and generate trollings. Identifying potential trolling targets is of critical importance in predicting, and preventing or deflecting troll attacks. For example, a vulnerable post could raise some flags, informing the user or the administrators for the imminent troll attack.

We assume that trolling occurs within an online user-engagement ecosystem, such as a social network, a micro-blogging system, or a discussion forum. Users contribute content in the form of posts, and they interact with each other, creating discussions. We model interactions between posts as a directed graph $G = (V, E)$, where nodes $u \in V$ correspond to posts and there is an edge (u, v), from post u to post v, if v is a reply to u. For example, in Twitter, nodes may correspond to tweets and there is an edge from a tweet (node) u to all tweets (if any) that this tweet refers to. Similarly, in Facebook, nodes may correspond to comments on user posts.

In this work, we will use Reddit[1], a popular online discussion forum, as our running example. In this case, the conversation graph of the posts defines a tree. The root of the tree corresponds to the initial post (message) that generated the

[1]https://www.reddit.com/



(a) Trolling Volume

(b) Proximity

(c) Popularity

Figure 3.2: Examples of the properties that a good troll vulnerability measure should satisfy. Shaded nodes correspond to trollings and non-shaded ones to non-trollings. Note that in Figure 3.2c, the activity after the comments u and v do not have to be necessarily trolling.

discussion. Each node of the tree, other than the root, has a unique parent, and there is a directed edge from the parent-comment node to the child-comment node, indicating that the child comment is a reply to the parent comment. A comment may have multiple replies (children), but each comment replies to a single previous comment (the parent). An example of a discussion tree is shown in Figure 3.1. The tree structure in posts is common to many social media. We note that our metrics are applicable to general graph structures as well.

3.2 Troll Vulnerability Rank

Our goal is to define a metric that quantifies the vulnerability of a post to trolling attacks. Such metric can be a useful weapon against the troll phenomenon. We first describe some intuitive properties that such a metric must satisfy.

First, clearly, posts that attract a large number of trollings must have high vulnerability.

Property 1 (Trolling Volume). *The vulnerability rank of a post should increase with the number of its descendants that are trollings.*

Figure 3.2a shows an example of a discussion tree, where the shaded nodes are trollings. We consider node u to be more vulnerable than node v, since u has more trolling descendants than v.

Second, the proximity of trolling descendants should also be accounted for in the definition of troll vulnerability.

Property 2 (Proximity). *The vulnerability rank of a post should increase with its proximity to trollings.*

For example, in Figure 3.2b, nodes u and v have the same number of trolling descendants. However, we consider node u to be more vulnerable, because node u is closer to its trolling descendants than node v.

To capture trolling volume and proximity, we use Random Walks with Restarts (RWR) for the definition of troll vulnerability. Intuitively, we relate the vulnerability of a node u with the probability that a random walk starting from u will visit a trolling. The RWR takes place in the subtree rooted at u, where at each transition there is a chance α that the random walk restarts at u. For each descendant v of u it defines a probability $p_u(v)$ that the random walk, that starts from node u, is at node v after an infinite number of transitions. We compute the vector of probabilities p_u as follows.

$$p_u = (1 - \alpha)\, p_u\, A + \alpha\, e_u, \tag{3.1}$$

where α is the restart probability, A is the row-stochastic transition matrix, and e_u is the restart vector, with $e_u(u) = 1$, and 0 otherwise. A is the normalized adjacency matrix of graph G. In particular, for sink nodes v (e.g., leaves in the case of trees), we set $A[v, u] = 1$, and 0 otherwise, that is, the random walk restarts at node u. For all other nodes, we set $A[u, v] = 1/|out\_degree(u)|$, if $(u, v) \in E$ and 0 otherwise.

RWRs have been widely used to define the strength of the relationship between two nodes in a graph and are the building blocks of many metrics including PageRank [27] topic-sensitive PageRank [28] and SimRank [29]. In this work, we use RWRs to capture the relationship of a node with its trolling descendants.

We now define the troll vulnerability rank of a node as follows.

Definition 3 (Troll Vulnerability Rank). *The Troll Vulnerability Rank (TVRank) of a post u is defined as:*

$$\textit{TVRank}(u) = \sum_{\substack{v \text{ is a trolling and} \\ \text{a descendant of } u}} \frac{p_u(v)}{1 - p_u(u)}, \tag{3.2}$$

Intuitively, the *TVRank*(u) value is the probability that the RWR visits a trolling, given that it is visiting a descendant of u. The higher the *TVRank* value of a post, the more vulnerable the post is. Note that the random walk will assign a probability to node u as well, which represents the probability to be in node u after an infinite number of transitions. We do not sum this probability to the *TVRank* value, even if the node is a trolling. In the contrary, we distribute it to the descendants of u by dividing with the probability not to be in the node u after infinite number of transitions, so that that the sum of the descendant's probabilities sum to one.

Our definition naturally incorporates the desired properties. According to Definition 3, for a given node u we sum the probabilities of being in a descendant that is a trolling after infinite steps. Thus, the larger fraction of its descendants that are trollings the higher its vulnerability, satisfying the first property. Furthermore, since the restart vector e_u is not uniform, the random walk is biased towards the nodes that are close to u and due to the restarts the shorter paths are more important. Therefore, distant trolling descendants have a smaller effect on the *TVRank*(u) than closer ones, satisfying the second property.

We use *TVRank* to detect vulnerable posts. In addition to having a high *TVRank* value, for a post to be characterized as vulnerable, we ask that it also satisfies the following property.

Property 3 (Popularity). *To be considered as vulnerable, a node must have a large enough number of descendants (not necessarily trollings).*

The popularity property requires for a post to generate enough traffic in order to be of interest to moderators. For example, in Figure 3.2c, nodes u and v have the same number of trolling descendants, but v has just one descendant in total. Even though this is a trolling response, there is no additional interaction and no further responses, so this is clearly a failed attempt at trolling.

Definition 4 (Post Vulnerability). *A post u is considered vulnerable to trolls if it has at least K, $K > 0$, descendants and TVRank$(u) \geq \theta$, $0 \leq \theta \leq 1$, where K and θ are parameters that control the sensitivity of post vulnerability.*

The θ value determines the intensity of trolling activity that a post needs to generate for the post to be considered vulnerable. When moderation needs to be strict (for instance, to avoid insults in a social media where kids participate), a lower θ value allows prompt notification for potential trolling behavior. The threshold value K determines the minimum number of responses that a post needs to generate for the post to be considered important enough to be characterized as vulnerable.

In Appendix B, we include a few examples of conversation subtrees and their *TVRank* values.

3.3 Discussion

In this section, we discuss the advantages of our vulnerability measure and compare it with two other measures that we also considered. As we discussed in previous section, the *TVRank* value of a comment u is the probability that the random walk visits a trolling, given that it is visiting a descendant of u. Thus, it is an intuitive measure (it is a probability) and it satisfies the properties defined in Section 3.2. We move on describing the alternative measures that we considered.

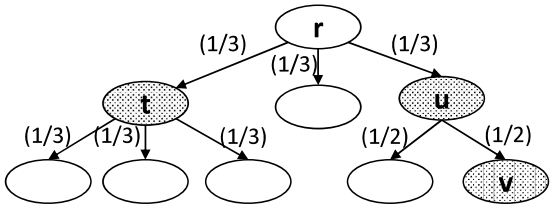


Figure 3.3: Example of a converation subtree. The values in the parenthesis correspond to edge weights used by an alternative vulnerability measure. Shadowed nodes correspond to trollings.

The first alternative measure is the proportion of trolling descendants of a comment, we name it *Trolling Ratio* (*T-Ratio*). Given a comment u, its *T-Ratio* is the ratio of the number of posts that are both descendants of u and trollings to the total number of descendants of u. It is a simple and intuitive measure, it ranges in $[0, 1]$ and the

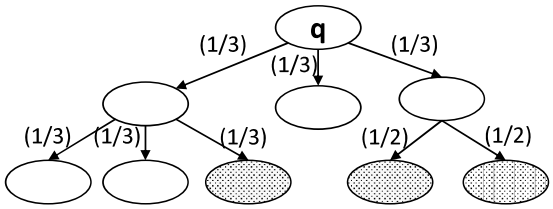


Figure 3.4: Example of a converation subtree. The values in the parenthesis correspond to edge weights used by an alternative vulnerability measure. Shadowed nodes correspond to trollings.

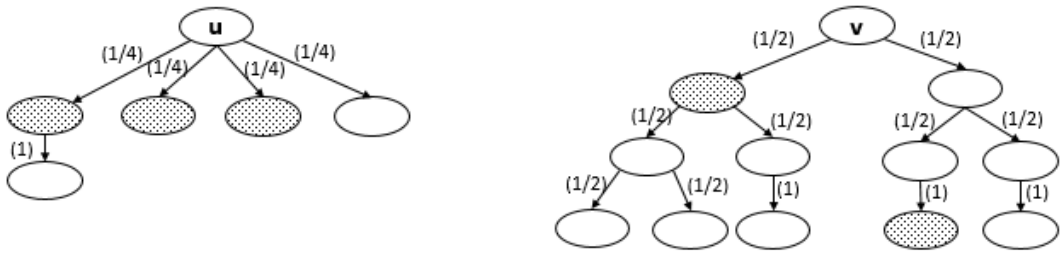
higher the value the more vulnerable the comment is to trolls. Actually, it can be considered as the probability a random descendant of comment u to be a trolling. Figure 3.3 shows an example of a comment and its descendants. According to this measure the vulnerability of comment r is *T-Ratio*$(r) = 3/8 = 0.375$.

The second alternative measure is a score that involves weight assignment in the edges of the conversation tree. We assign weight w on each outgoing edge of a node u, $w = 1/|out\_degree(u)|$ if u has any outgoing edges and 0 otherwise. Then, we sum the path probabilities of walking from the root node to any trolling descendant. We call this measure *Path Score* (*P-Score*). Note that, because we sum the probabilities the resulting score is not a probability, it is possible that the *P-Score* of a comment to be larger than one. The score of node r in Figure 3.3 is calculated as:

$$\begin{aligned} \textit{P-Score}(r) &= pathToNode(t) + pathToNode(u) + pathToNode(v) \\ &= \frac{1}{3} + \frac{1}{3} + \frac{1}{3} * \frac{1}{2} = \frac{1}{3} + \frac{1}{3} + \frac{1}{6} = \frac{5}{6} = 0.833 \end{aligned}$$

The first question we have to ask is if these measures satisfy the property that we described in Section 3.2. According Figure 3.3, the vulnerability measures of the comment r are *TVRank*$(r) = 0.514$, *T-Ratio*$(r) = 0.375$ and *P-Score*$(r) = 0.835$. Clearly, both the alternative measures satisfy the first property, regarding the trolling volume, by definition. The more the trollings that follow the comment, the higher the vulnerability of the comment.

In order to test if the measures satisfy the second properties we use the example in Figure 3.4. It is an example of a conversation subtree with the same structure as the example in Figure 3.3, but we have changed the position of the trollings. The vulnerability measures for the comment q in this example are the following *TVRank*$(q) = 0.248$, *T-Ratio*$(q) = 0.375$ and *P-Score*$(q) = 0.444$. The *T-Ratio* does not satisfy the second property, because the value remained unchanged in both examples.



(a) (b)

Figure 3.5: Example of two converation subtrees to compare two vulnerability measures, *P-Score* and *TVRank*. The values in the parenthesis correspond to edge weights used by *P-Score*. Shadowed nodes correspond to trollings.

Actually, the *T-Ratio* will remain the same in any example that has the exact same numbers of descendants and trolling descendants. Thus, we can safely say that the *T-Ratio* is not a good vulnerability measure. The *P-Score* seems to satisfy the second property. This is expected because while we calculate the path probabilities we have to multiply with values less than one, penalizing the longer paths.

However, the *P-Score* measure has a serious disadvantage; its values are not intuitive. It is not clear what its values mean for the troll vulnerability of a comment. Figure 3.5 shows two examples where the nodes u and v have the exact *P-Score* values in two completely different situations. In Figure 3.5a, the node u has *P-Score* equal to 0.750 and *TVRank* equal to 0.619. The comment u seems to be susceptible to trolls. In Figure3.5b, the comment v has the same *P-Score* as in previous example (0.75) and *TVRank* equal to 0.264. However, in this case the comment is not as susceptible as in the previous example. The *P-Score* measure fails to provide a clear picture about the vulnerability of the comment. On the other hand, the *TVRank* measure performs better in these two examples; assigning much lower value to the comment that is not vulnerable to trolls.

Chapter 4

Reddit Dataset

In this chapter, we summarize preliminary results of our work. At first we describe the dataset that we retrieved for the purposes of this thesis. Then, we describe how we annotate the dataset, i.e., whether the comments are trollings and/or vulnerable. Finally, we provide an analysis regarding the vulnerability of the comments.

4.1 Dataset

Our dataset contains posts from the Reddit[1] social network website. The site is a collection of entries, called *submissions*, posted by registered users. Submissions are organized into categories, called *subreddits*. Once a user posts a submission to a subreddit, users post comments on this submission. Users are also able to respond to these comments. Thus, conversation trees are formed whose roots are the submissions.

Reddit also enables us to sort and retrieve submissions, and the corresponding comments, based on their popularity or controversy over a certain period of time.

[1] https://www.reddit.com/

Popular (also known as "top") submissions are the the submissions with significantly more upvotes than downvotes and controversial are the submissions which have even amounts of upvotes and downvotes. We retrieved submissions[2] from each of 18 subreddits based on their popularity and their controversy. In particular, we retrieved 40 submissions from each subreddit, 20 top and 20 controversial submissions, resulting in 555,332 and 270,144 comments from top and controversial submissions, respectively.

The subreddits that we used are:

1. **world news**: major news from around the world except US-internal news
2. **news**: factual, objective articles covering recent news
3. **sports**: discussion around popular sport events
4. **science**: discussion of various fields of science (moderated subreddit)
5. **politics**: current and explicitly political U.S.news
6. **space**: dedicated to the discussion of outer space
7. **movies**: news, questions and discussions about movies
8. **OutOfTheLoop**: discussion of recent trends and news
9. **NotTheOnion**: discussion of eal news stories that sound like they're Onion[3] articles, but aren't
10. **history**: a place for discussions about history
11. **atheism**: topics related to atheism, agnosticism and secular living
12. **funny**: humor posts
13. **gadget**: discussion about gadgets
14. **announcements**: official announcements from the reddit admins
15. **Dota2**: a subreddit for Dota 2, an action RTS game developed by Valve Corporation

[2]The dataset was crawled using the Python Reddit API Wrapper (PRAW) package for Python programming language. It is available at `https://github.com/praw-dev/praw`

[3]`http://www.theonion.com/`

16. **leagueoflegends**: a subreddit for content and discussion about League of Legends, a game created by Riot Games.

17. **dataisbeautiful**: a place for visual representations of data: Graphs, charts, maps, etc.

18. **gaming**: a subreddit for (almost) anything related to games - video games, board games, card games, etc. (but not sports).

Figure 4.1 shows the subreddits and the corresponding fraction of comments that belong to each subreddit.

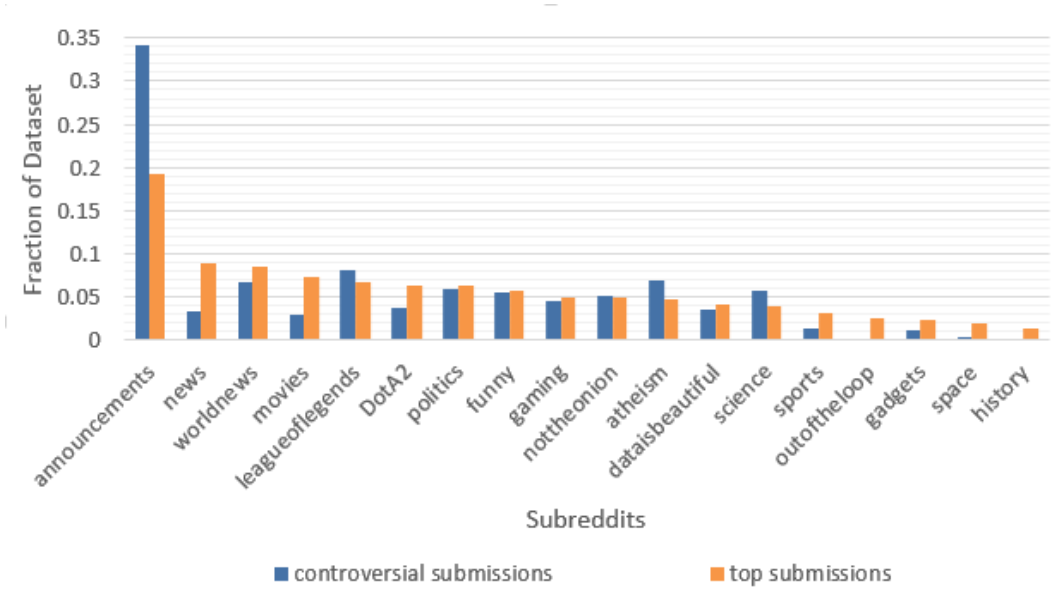


Figure 4.1: The fraction of comments that belong to each subreddit.

The main drawback of our dataset is the lack of the ground truth. There is no indication about which comments are actually trollings and which are not. The same goes for the troll vulnerable comments. In Section 4.2, we describe a method to detect trollings and in Section 4.3 we describe how we annotate the troll vulnerable comments.

4.2 Annotation of Trollings

Although, identifying trollings is a problem orthogonal to our approach, to evaluate the performance of the troll vulnerability prediction task, we need first to detect

trollings in our dataset.

Thus, as a first step of our evaluation, we identify trollings among the Reddit comments. The notion of trolling covers a wide range of behaviors, from innocent humor and misinformation to criminal activity. We focus on the anti-social part of trolls, i.e., we detect comments that contain offensive content.

Specifically, to classify trollings, we build a classifier that detects insulting content using only text features. To train the classifier, we used a labeled dataset from an online contest in the Kaggle[4].

4.2.1 Kaggle Dataset

The dataset consists of a label column followed by two attribute fields. This is a single-class classification problem. The label is either 0 meaning a neutral comment, or 1 meaning an insulting comment (neutral can be considered as not belonging to the insult class). The first attribute is the time at which the comment was made. It is sometimes blank, meaning an accurate timestamp is not possible. It is in the form "YYYYMMDDhhmmss" and then the Z character. It is on a 24 hour clock and corresponds to the localtime at which the comment was originally made. The second attribute is the unicode-escaped text of the content, surrounded by double-quotes. The content is mostly English language comments, with some occasional formatting.

Note that this dataset is different from the dataset that we use in troll vulnerability analysis. The only purpose of this dataset is to train a classifier to detect trollings.

The dataset consists of two subsets; the first one is used to for the training of the models and the second one is used for the evaluation of the models. The training set contains 6,594 comments, from which 1,743 comments are insulting and the rest are neutral. The test set contains 2,236 comments, from which 1,077 comments are insulting. There is a small amount of noise in the labels as they have not been meticulously cleaned. However, the error in the training and testing data is less than 1%.

4.2.2 Model

We used a slightly modified version of a classifier used in the Kaggle contest [30]. This is the proposed solution of the third winner of the contest. Note that all the

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

top proposed solutions in the contest were very close to each other. For example, the difference between the best solution and this one is less than 0.4% in the AUC metric and the top-5 approaches differ less than 1%.

The model consists of three basic phases. At first, the text of the comments were pre-processed. The preprocessing includes

- removing links, html entities, html code and non-ASCII characters
- removing tabs, new lines and duplicate spaces
- removing dots inside words or grouping together sequences of one-letter words (e.g. *"l a l a"* or *"l.a.l.a"* $\rightarrow$ *"lala"*)
- adding special tokens in texts for groups of characters such as: "#$%#$", "?!???", "!!!!!!"

In the second phase, three different classifiers are trained. The first classifier is an SVM classifier and it takes as input n-grams, with $n \in [1, 4]$. The second classifier is also an SVM classifier, but this time the input is character n-grams, with $n \in [4, 10]$. Similar classifiers, char/word n-grams, were very common among the contestants. Finally, the third classifier was a custom build dictionary based classifier. It used a curse words dictionary. This classifier just looked if the text had words from the dictionary and also words like "you", "your", "yourself". Then, it computed a simple score based on the distances between the curse words and the "you"-words.

The last phase of the model includes the training of a neural network; a multilayer perceptron with a hidden layer of 3 neurons. The neural network combines the previous classifiers and also uses some additional features as input:

- the ratio of curse words
- the text length
- the ratio of *, ! or ?
- the ratio of letters in capital

The classifier takes as input the text content of the comments and assigns a score in [0,1] to each comment. Comments that have high score are more likely to be insulting and vice versa.

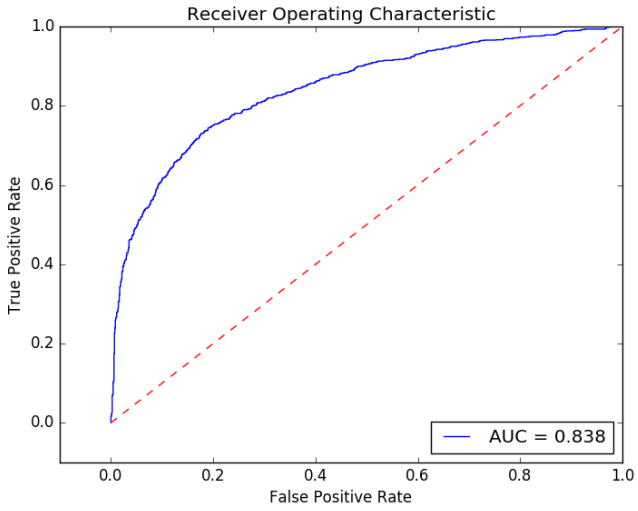


Figure 4.2: The performance evaluation of the neural network.

4.2.3 Results

In order to evaluate the results, we have considered two methods. At first, we followed the same settings like the contest, i.e. we trained and tested the models with the same train and test sets, in order to be able to compare our result with other participants. The model performed pretty well on the dataset provided for the contest. To be more accurate, it achieved 83.8% area under the ROC curve (AUC) –Figure 4.2–, its accuracy is almost 76%, with precision and recall 84% and 61% respectively .

We also performed a 5-fold cross-validation along the whole dataset, i.e., both the training and test sets. Cross-validation will provide a better indication about the performance of the model and how it will generalize to an independent dataset. The model achieved 91% AUC, as we can see in Figure 4.3. Its accuracy, precision and recall are 85%, 80%, 71% respectively. We notice that there is significant difference between the two tests. This difference can be attributed to the different fraction of trollings in the two sets. The test set contains more than 40% insulting comments, whereas the training set contains only 21%. Note that these analogies of trolling and non-trollings do not correspond to the real world. As we will discuss later the trollings are much more scarce, in general.

Furthermore, we want to understand the relative importance of the features. To this direction, we distinguished the features in three groups; n-grams, character n-grams and hand-crafted features, which includes the rest of the features. Then, we compare the performance of each group individually, using a logistic regression clas-

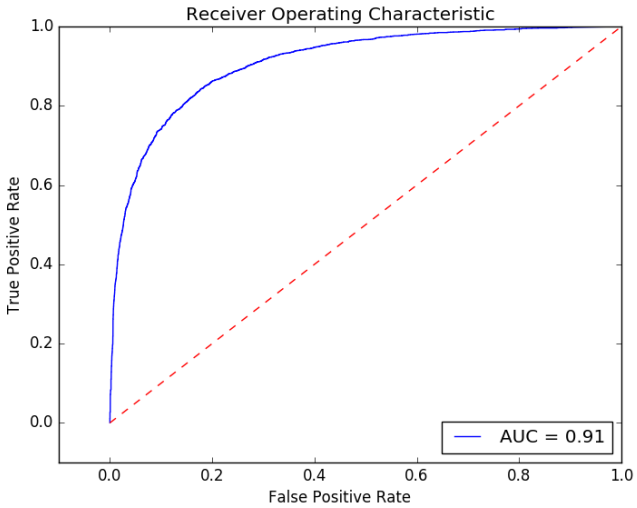


Figure 4.3: The performance evaluation of the neural network in performing 5-fold cross validation in training and test sets.

Feature Group	Accuracy	Precision	Recall	AUC
hand-crafted	0.71	0.54	0.62	0.72
n-grams	0.79	0.69	0.72	0.85
character n-grams	0.82	0.71	0.70	0.87
combination of all groups	0.83	0.74	0.73	0.90

Table 4.1: Evaluation of the features in trolling detection.

sifier. The results are summarized in Table 4.1.

This model performed slightly different than the model described in Section 4.2.2. The hand-crafted features performed worse than the other features, its precision and recall metrics are 54% and 62% respectively. The most important feature of the group is the ratio of the curse words in the comment. The character n-grams and the n-grams perform similarly, with the first to be slightly better than the latter.

Diving deeper into the models, we can see which n-grams and character n-grams are important in the decision in the classification by checking the coefficient assigned my the logistic regression classifier. As expected, the classifier assign larger coefficients to curse words and phrases. For example, some examples are *"id**t"*, *"mor**n"*, *"d*mb"*, *"piece of sh*t"*, *"f*ck you"*, etc. Note that in the case of character n-grams, it may contain prefixes or suffixes of the curse words/phrases. An important observa-

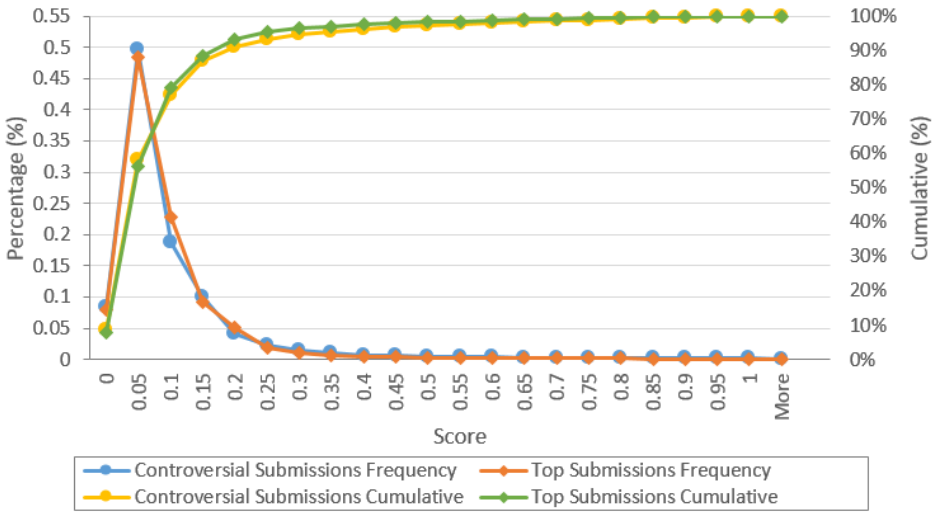


Figure 4.4: The distribution of trolling scores in the comments of top and controversial submissions.

tion is the second-person pronoun ”you” and its derivatives seem to play important role in the decision. We speculate this is caused because people use second-person pronouns to address the insults to other other people.

We used this model to build a classifier for detecting trollings in the Reddit dataset. To evaluate the performance of the classifier in the Reddit dataset, we manually labeled 2500 Reddit comments as trollings (i.e., insulting) or no trollings (i.e., neutral). At first, we stratified the the $[0, 1]$ scores in five different non-overlapping ranges; $[0, 0.1]$, $(0.1, 0.3]$, $(0.30, 0.50]$, $(0.50, 0.70]$ and $(0.70, 1]$. Then, we randomly selected 500 comments from each range. We set the threshold for characterizing a comment as trolling at 0.5. We achieved 82% accuracy, 75% precision and 78% recall in this set. We also experimented with other threshold values but with no significant improvement.

Table A.1 contains a few comments and they are annotated by the model described above.

Using this model we were able to detect 15,346 trollings in our dataset, which amounts to 1.8% of the total dataset. The top submissions contain 9,541 (1,7%) trollings and the controversial submissions contain 5,805 trollings (2.1%). Figure 4.4 shows the distribution of trolling score in the comments for top and controversial submissions. As we can see the curves follow the same trends, which indicates that the distribution of trollings is similar to the top and controversial submissions. We

also notice that 90% of the comments scored less than 0.2 and only 2% scored more than 0.5. Table 4.2 summarizes our dataset statistics.

Description of the dataset		
Number of subreddits	18	
Number of submissions	360	
Submission Type:	Top	Controversial
Number of posts	555,332	270,144
Number of trollings	9,541	5,805

Table 4.2: Dataset statistics.

4.3 Annotation of Troll Vulnerable Comments

In this section, we describe how we decide whether a comment in our dataset is troll vulnerable or not. We discuss about the parameters of Definition 4 and we describe the annotation of the comments.

4.3.1 Parameter Calibration

Our definition of troll vulnerability includes two parameters (θ and K) that control the sensitivity to trolling behavior. In particular, for a comment c to be vulnerable, $\mathit{TVRank}(c) \geq \theta$ and c must be followed by at least K comments. Both parameters also determine the number of comments that are vulnerable, i.e., the size of our class.

There is also another parameter hidden in Definition 3, the restart probability a. We experimented also with different α values. This results in a small difference in the vulnerability rank of the nodes, however, it does not affect the performance of the prediction model described in Chapter 5. Thus, we set α, the restart probability, equal to 0.15 as in previous work, e.g., [27].

We set $K = 2$ as a default, asking that a comment must be followed by at least 2 comments to be considered vulnerable. As argued in Section 3.2, if $K = 1$, then even if the following comment is a trolling, it is a failed one, since it did not generate any additional discussion. We experimented with larger values of K as well which result

in a smaller positive class of vulnerable comments and we present related results.

As a default value, we set θ = 0.30, which means that a comment should have at least 30% probability to visit a trolling descendant after infinite number of steps. We experimented with other values of θ and report results. Thus, a comment c should be have at least $K = 2$ descendants and $\mathit{TVRank}(c) \geq 0.30$, unless stated otherwise. Table 4.3 summarizes the default values of the parameters of the model.

Parameter	Default	Range
TVRank threshold (θ)	0.30	$0.15, 0.2, 0.25, 0.3, 0.35$
Popularity threshold (K)	2	$2, 3, 5, 8$

Table 4.3: Troll vulnerability parameters.

Algorithm 1 Procedure that calculates the Troll Vulnerability Rank

procedure GETTVR(c, G, a) ▷ The comment c, the conversation-tree G and parameter a
  $A \leftarrow create\_trans\_matrix(c, G)$ ▷ Create the transition matrix of the subtree
  $e_u \leftarrow create\_restart\_vector(c, A)$ ▷ Create the restart vector
  $p_u \leftarrow [1, 0, \ldots, 0]$ ▷ Initialize the probability vector
  $\delta \leftarrow 1$
  while $\delta \geqslant 0.0001$ **do** ▷ Calculate the probability vector
    $p_{temp} \leftarrow p_u$
    $p_u \leftarrow (1 - a)p_{temp}A + ae_u$
    $\delta \leftarrow |p_u - p_{temp}|$
  end while
  $tvr_c \leftarrow 0$ ▷ Initialize the *TVRank* for the comment c
  for each $v \in V$ and $v \neq c$ and $isTrolling(v)$ **do** ▷ For comments that are both descendants of c and trollings
    $tvr \leftarrow tvr + \frac{p_u(v)}{1-p_u(c)}$ ▷ Update the *TVRank* value of comment c
  end for
  return tvr_c ▷ Return the vulnerability of the comment.
end procedure

4.3.2 Annotation

In order to annotate the comments as vulnerable or non-vulnerable, we have to calculate the *TVRank* for each comment. Algorithm 1 summarizes the procedure that calculates the *TVRank* for a comment.

Considering a comment c in the conversation tree, we perform a Random Walk with Restarts on the (sub-)tree rooted at the comment c. Algorithm 1 the comment c, the conversation subtree and the parameters K, θ and a. We also have to calculate the transition matrix for the subtree and the restart vector (always restart at c). The result of the random walk is a vector of probabilities p_u as described in Equation 3.1. Then, we calculate the *TVRank* for the comment c according Equation 3.2.

Finally, the comment c is characterized as troll vulnerable or not, according to Algorithm 2. If the *TVRank* value of the comments is larger than θ, then it is troll vulnerable and not troll vulnerable otherwise. Note that if the comment has less than K comments then it is not considered troll vulnerable.

Algorithm 2 Algorithm to decide whether a comment is troll vulnerable

```
1:  procedure IsTrollVulnerable(c, G, K, θ, a)
2:      if K ⩽ get_descendants_count(c) then        ▷ Check the #descendants of c
3:          return FALSE                            ▷ The comment is not vulnerable
4:      end if
5:      tvr_c ← getTVR(c, G, a)                     ▷ Calculate the TVRank for the comment c
6:      if tvr ⩾ θ then                             ▷ If the TVRank value of c is larger than threshold θ
7:          return TRUE                             ▷ The comment is vulnerable
8:      else
9:          return FALSE                            ▷ The comment is not vulnerable
10:     end if
11: end procedure
```

4.3.3 Results

We annotate all the comments of the dataset using Algorithm 2. Table 4.4 shows the number of vulnerable comments in both popular and controversial submissions, for different combinations of K and θ. Considering the default values for the parameters results in 3,858 comments being characterized as troll vulnerable in top submissions,

which amounts for about 2.5 trollings per vulnerable comment, on average. In controversial submissions, 2,875 comments are characterized as troll vulnerable, which amounts for about 2 trollings per vulnerable comment, on average. The percentage of troll vulnerable comments in the dataset is very low; 0.7% for the top submissions and 1.1% for the controversial submissions.

This means that the vast majority of the comments do not attract trolls. Most of the users seem to act in good faith. However, this can cause a few problems in our work. The troll vulnerable comments are limited to only a small percent of the dataset, which means that it can be very harsh to identify them. In addition, this imbalance between the troll vulnerable and the non-vulnerable comments affects the prediction task, that we describe in Section 5.

Top Submissions						
		θ				
		0.15	0.2	0.25	0.3	0.35
K	2	7,943	6,271	4,995	3,858	3,036
	3	6,458	4,786	3,510	2,373	1,551
	5	3,774	2,321	1,430	953	653
	8	1,884	1,098	671	434	281

Controversial Submissions						
		θ				
		0.15	0.2	0.25	0.3	0.35
K	2	5,631	4,545	3,695	2,875	2,286
	3	4,506	3,420	2,570	1,750	1,161
	5	2,541	1,619	1,036	696	485
	8	1,155	694	449	285	193

Table 4.4: Number of vulnerable comments for different values of K and θ in popular (left) and controversial (right) submissions.

4.4 Statistical Analysis of the Dataset

In this section, we provide an analysis regarding the troll vulnerable comments in our dataset. Our goal is to study the behavior of vulnerable comments and find any factors that affect them.

4.4.1 Distribution of Troll Vulnerable Comments in the Dataset

We want to study how the *TVRank* values are distributed in the dataset. The *TVRank* values correspond to the values assigned to the comments according the procedure described in Section 4.3. In Figure 4.5 we can see the distribution of *TVRank* in the comments of top and controversial submissions using the default parameters. Note

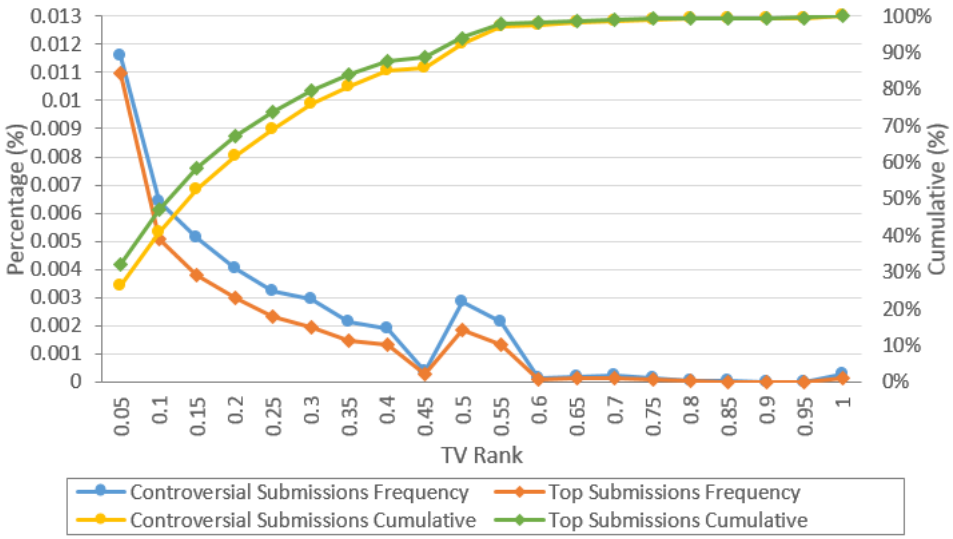
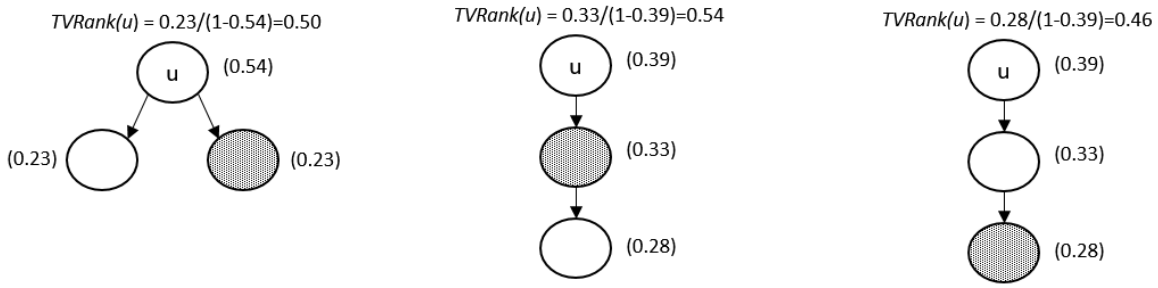


Figure 4.5: The distribution of *TVRank* in the comments of top and controversial submissions. We skipped the percentage of comments that their *TVRank* equals 0. Due to their high percentage, more than 98%, they would dominate the figure.

that more than 98% of the comments have zero *TVRank*, thus we removed this part from the plot in order to have a better visual of the distribution. Again, we can see that there is no significant difference between the curves of controversial and top submissions. It seems that the vulnerability of the comments follows a power-law distribution, which is very common on a broad array of user-generated websites [31]. Most of the comments have low *TVRank* values and only a few comments have high values. This means that only a few comments are vulnerable to trolls.

An interesting observation in Figure 4.5 is the "bump" that appears in the curve around $[0.45, 0.55]$. We investigated this anomaly and found out that it is caused by frequent small subtrees that contain three specific sequences of trollings and non-trollings. We shall call such frequent sequences, *attack patterns*. The most common *TVRank* values in $[0.45, 0.55]$ are 0.46, 0.50 and 0.54. These values correspond to the subtrees in Figure 4.6.

Note that the pattern in Figure 4.6a can be extended to two more patterns that could make node u to have $\mathit{TVRank}(u) = 0.5$. The simplest one is node u to have 4 immediate descendants, i.e., children, and two of them should be trollings. The other one requires that each of the two immediate descendants of node u have an additional descendant. Then, one child and one grandchild of u should be trollings. However, these additional attacks occur only a few times and the majority of the nodes that



(a) (b) (c)

Figure 4.6: Three attack patterns that occur frequently in the dataset. Shaded nodes correspond to trollings and the values in the parenthesis correspond to the values assigned by the random walk.

their *TVRank* equals 0.5 match the sequence described in Figure 4.6a. Similarly, the attack patterns described in Figures 4.6b and 4.6c represent the majority of the attack patterns with the corresponding, however there are a few other patterns that occur only a few times.

As we can see in Figure 4.7, these frequent attack patterns amount for almost 38% of vulnerable comments in top submissions. Note that the curve that represents the dataset in general, only includes comments that have at least two descendants. Furthermore, we notice that there is a difference in the descendants of the vulnerable comments in the top submissions. In particular, vulnerable comments seem to have less descendants that the comments in general in the top submissions.

However, this is not the case for the vulnerable comments in the controversial submissions. As we can see in Figure 4.8, the vulnerable comments in the controversial submissions seem to have almost the same descendants as the other comments. The vulnerable comments still have less descendants, however the difference is very small, about 1-2%.

It seems that people who participate in controversial submissions are more eager to continue the conversation in a ”hostile” environment. They are willing to support and defend their opinions against other users, who may not be well-behaved. On the other hand, people that post comments in top submissions do not put up with other people’s inappropriate behaviors. Most of the conversations that contain trollings do not last long enough.

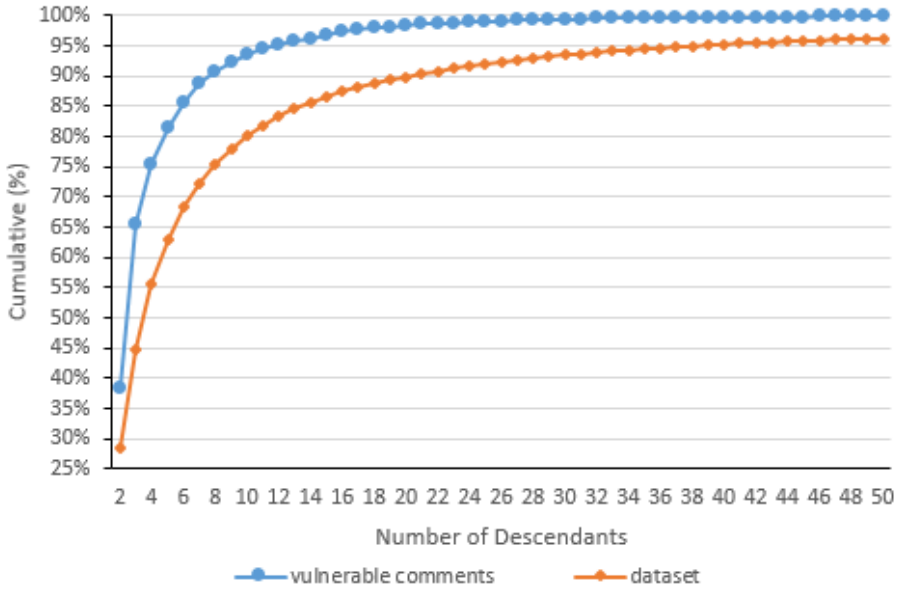


Figure 4.7: The cumulative distribution function of descendants in vulnerable comments and in the dataset in general for top submissions. Vulnerable comments by definitions must have at least two descendants, thus the curve that represents the dataset in general does not include comments that have less than two descendantsd.

4.4.2 Analysis of Troll Vulnerability per Subreddit

Furthermore, we would like to study whether there is a relationship between vulnerable comments and subreddits or submissions. It is obvious that in order to have a lot of vulnerable comments, we should have a lot of vulnerable comments. Table 4.5 show five subreddits with the most and five with the least number of troll vulnerable comments for top and controversial submissions. The table indicates that some subreddits have more troll vulnerable comments. However, this is not definitive. As we can see, the ranking in the top submissions is different than the ranking in the controversial submissions. In addition, the subereddits for the top submissions do not fully match the subreddits for the controversial submissions.

An interesting observation in Table 4.5 is that controversial submissions seem to have higher percentage of trollings and troll vulnerable comments than the top submissions. As we saw in Section 4.3, the controversial submissions had marginally larger fraction on average (about 0.3%) of troll vulnerable comments than the top submissions. This difference seems to be larger in some subreddits. For instance, the top submissions of the ”news” subreddit contains 1% troll vulnerable comments,

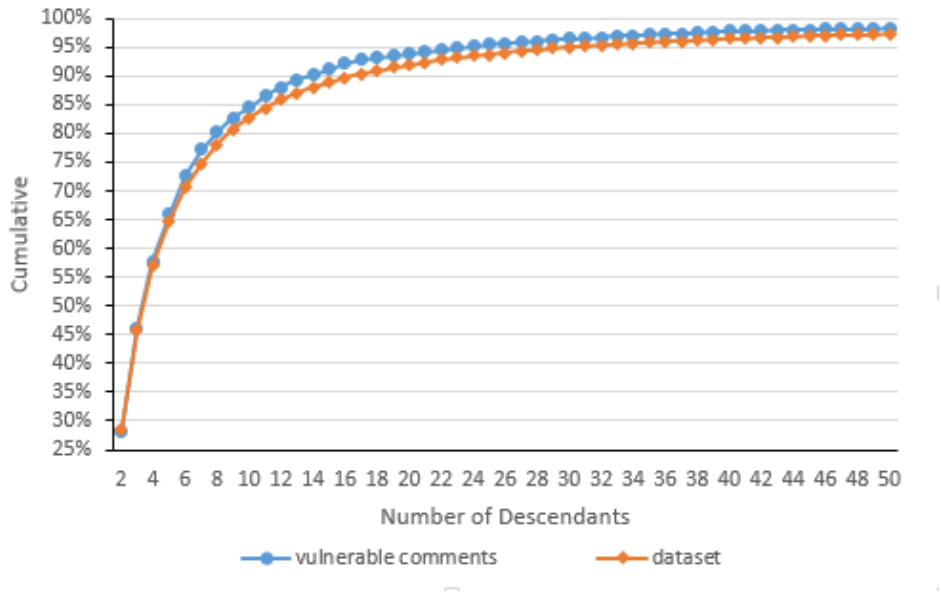


Figure 4.8: The cumulative distribution function of descendants in vulnerable comments and in the dataset in general for controversial submissions. Vulnerable comments by definitions must have at least two descendants, thus the curve that represents the dataset in general does not include comments that have less than two descendants.

whereas the corresponding percentage in controversial submissions is 1.8%, which is much larger.

Furthermore, we want to investigate whether individual submissions are more likely to attract more trolls than others, i.e., if there are submissions with more vulnerable comments than other submissions. In Figure 4.9, we can see the distribution of the vulnerability in the submissions for both top and controversial submissions. The horizontar axis reports the percentage of troll vulnerable comments that a submission contains and vertical axis the number of submissions that contain a specific percentage of troll vulnerable comments. We can see that most of the submissions have low percentage of troll vulnerable comments, whereas only a few have higher percentage of vulnerable comments. Note that there are a few submissions with only a few tens of comments that exhibit very high percentage of troll vulnerable comments (larger than 10%). Thus, the submissions seem to have a relationship with the troll vulnerability. However, it is not clear how they affect the troll vulnerable comments. We speculate that submissions that refer to a sensitive, polarized or controversial subject may attract more trolls, but this requires additional research.

subreddit	total comments	#trollings	#troll vulnerable
announcements	61709	1498 (2.4%)	668 (1.0%)
news	62712	1534 (2.4%)	670 (1.0%)
atheism	19719	438 (2.2%)	182 (0.9%)
outoftheloop	20481	425 (2.1%)	172 (0.8%)
nottheonion	24903	490 (2.0%)	205 (0.8%)
gaming	26589	361 (1.3%)	117 (0.4%)
movies	50093	629 (1.2%)	217 (0.4%)
space	15285	103 (0.6%)	35 (0.2%)
history	10460	60 (0.5%)	23 (0.2%)
science	16644	81 (0.5%)	27 (0.1%)

subreddit	comment count	#trollings	#troll vulnerable
news	9034	314 (3.5%)	162 (1.8%)
DotA2	10386	444 (4.2%)	167 (1.6%)
atheism	18625	554 (2.9%)	291 (1.5%)
funny	15136	395 (2.6%)	183 (1.2%)
announcements	92243	3282 (3.5%)	1064 (1.1%)
space	1258	22 (1.7%)	6 (0.5%)
movies	8246	88 (1.0%)	31 (0.4%)
gadgets	2967	31 (1.0%)	9 (0.3%)
science	15704	97 (0.6%)	43 (0.2%)
history	621	2 (0.3%)	1 (0.1%)

Table 4.5: Subreddits with the most and the least number of troll vulnerable comments for top (left) and controversial (right) submissions. The subreddits are sorted by the percentage of troll vulnerable comments that they contain.

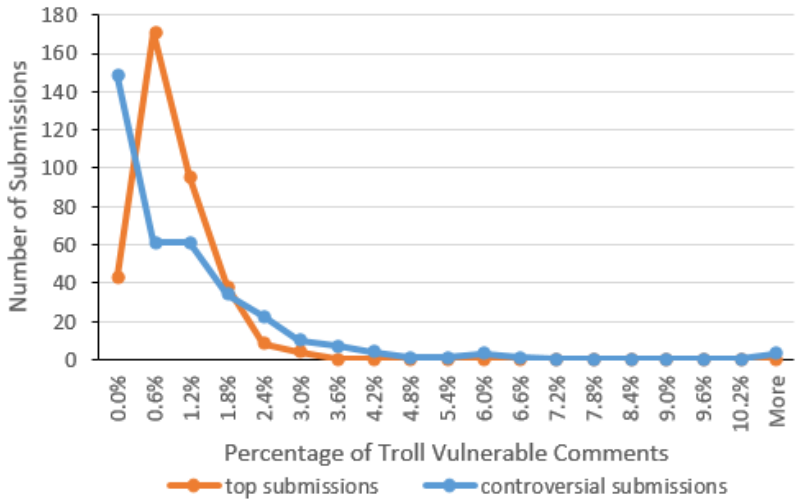


Figure 4.9: The distribution of vulnerability of in the submissions.

4.4.3 Analysis of Troll Vulnerability per User

Another question we would like to answer is whether users that post more trollings are more likely to be targeted by trolls, i.e., receive more trollings. To this direction, we measure the correlation between the trollness of a user posts with the vulnerability of his/her comments. We used two correlation coefficients; Spearman's *rho* [32] (ρ) and Kendall's *tau* [33] (τ).

In particular, we conducted two experiments. The first experiment measured the correlation between the number of trollings that a user posts with the number of his/her troll vulnerable comments. In the second experiment, we measured the correlation between the number of trollings that a user posts with the average *TVRank*

of his/her comments. Note that comments that are not troll vulnerable may have *TVRank* larger than zero (and smaller that θ).

Table 4.6 shows the results of our analysis. The correlations seems to be weak when we include all the users that participated in our dataset, i.e., posted at least one comment. However, we notice that filtering out users, who posted only less than a certain number of comments, revealed a stronger relation. The idea is that we do not have much information for the users that post only a few comments, thus they may induce noise in our analysis. According to Spearman's rho[5], the strength of the relationship is weak when we include users that posted ten or less posts, whereas when we include users that posted at least fifteen comments the strength of the relationship is moderate. Therefore, users that post more trollings have increased risk of encountering troll attacks.

	users with at least n comments	#vulnerable comments		average vulnerability of the comments	
	n	τ	ρ	τ	ρ
number of trollings	1	0.2697	0.2706	0.2667	0.2714
	5	0.3355	0.3426	0.3325	0.3579
	10	0.3616	0.3772	0.3600	0.4045
	15	0.3774	0.4009	0.3692	0.4267
	20	0.4029	0.4347	0.3855	0.4576
	25	0.4215	0.4624	0.3966	0.4803
	30	0.4125	0.4588	0.3884	0.4788
	35	0.4275	0.4798	0.4059	0.5058
	40	0.4508	0.5104	0.4306	0.5402

Table 4.6: Correlation between the number of trollings a user posts with the vulnerability of his/her comments. The vulnerability of a users posts is calculated with two ways; the number of the vulnerable comments and the average vulnerability of the comments.

[5] http://www.statstutor.ac.uk/resources/uploaded/spearmans.pdf

Chapter 5

Prediction of Troll Vulnerability

In this chapter, we build a model that detect troll vulnerable comments and evaluate its performance. Our goal is for a given a post to predict whether the post will be vulnerable to trolls or not. We treat the problem as a two class classification problem, with the positive class corresponding to the vulnerable posts and the negative class to the non-vulnerable posts and build a classification model. For defining the positive class (i.e., the set of vulnerable posts), we use Algorithm 1. At first we describe the features used to describe the comments. Then, we present the evaluation of the model using the Reddit dataset.

5.1 Feature Space

We design features that capture various aspects of the post and its past. Note that we only consider ancestors of the post, since we want to decide on its vulnerability, before

the post receives any replies (i.e., acquires any descendants). We group features in four categories, namely, content, author, history and participants. The features we used are summarized in Table 5.1.

5.1.1 Content Features

Content features include features related to the text of the post. Previous research (e.g., [5]) shows that the comments that were written by provocative users tend to be less readable than those written by other users. Thus, we include a number of readability-related features (e.g., the number of words written in capital letters, which is considered rude in online chatting) as well as the automated readability index[1] (ARI). We also count the number of positive and negative words, using an opinion lexicon[2]. The motivation is that opinionated comments are more likely to attract trollings. We also include a feature indicating whether the post itself is a trolling.

Furthermore, we tested n-grams and character n-grams classifiers, like we did in Section 4.2 for the trolling detection. However, they performed really poorly. Such classifiers try to find groups of words or characters that have a special relationship with the classes. For instance, in the trolling detection curse words and phrase played important role in the decision of the classifier. It seems that in troll vulnerability prediction problem there are no word or phrases that carry strong signal that can distinguish the two classes.

5.1.2 Author Features

Author features try to capture the behavior of the author of the post in the social setting. Features related to the activity of the author include the number of her posts, the number of trollings in them and the average replies per post. We also include the largest number of posts that a users posted in a single conversation tree, since it may be more likely for users that are very active in conversations to engage in a debate with trolls.

Additionally, we consider features related to how the other users in the community perceive the author and her comments. Most social networks provide mechanisms for

[1] https://en.wikipedia.org/wiki/Automated_readability_index

[2] https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon

users to express their preference, or opinion, for a post, (e.g., whether they like it or not, find it useful or not) by rating them. In Reddit, this rating is a score: 1 (upvote) if the users like the comment, -1 (downvote) if they do not. We use score-related features (such as the average score, the average of the absolute score values, number of comments that are scored positively, etc.) to help us to capture the perception of the user from the rest of the community.

5.1.3 History Features

History-related features are extracted from the conversation tree of the post. We consider the depth of the post in the tree and also information about the ancestors of the post. Information about the ancestors includes a number of score related features, such as the average and absolute score, as well as the number of posts that have negative, positive and zero score and the number of trollings. The motivation is that posts whose preceding posts do not include trollings and have positive scores are less likely to be targeted by trolls. This group also includes the similarity of the post with the previous three posts, by calculating the cosine similarity of the words used in these posts, since posts that try to change the topic of the conversation may attract an unpleasant reaction by the community.

5.1.4 Participant Features

Finally, the features related to the participants in a discussion contain information about the authors of the previous comments. In particular, we average the features in the second group for all the users that participate in the ancestor posts. These features can be thought of as describing the average user that participated in the previous posts.

5.2 Troll Vulnerability Prediction

We have built a model for predicting whether a comment is vulnerable to trolls or not using combinations of the features described in Section 5.1. We use a Logistic Regression classifier with a 5-fold cross validation. We also used additional classifiers, like Random Forest and SVM, however, Logistic Regression outperforms both. The

Feature Group	Features
Content (9)	#char, #words, #sentences, #quotes, #words in capital, A.R.I, #negative/positive words, whether is trolling
Author (13)	#posts, #trollings, max posts in single conversation tree, avg replies per post, #avg score per post, #avg absolute score, sum positive/negative score, #controversial comments, #positive/negative/zero scored posts, negative to postive score ratio
History (10)	depth of the post, parent similarity, zero/positive/negative scored posts, sum score, sum absolute score, sum negative/positive score, ancestors that are trollings
Participants (13)	#posts, #trollings, max posts in single conversation tree, avg replies per post, #avg score per post, #avg absolute score, sum positive/negative score, #controversial comments, #positive/negative/zero scored posts, negative to postive score ratio

Table 5.1: The features of our prediction model.

results presented here correspond to the popular submissions. The results for the controversial submissions are shown in Appendix C.

5.2.1 Class Imbalance

An important problem that we had to address is the class-imbalance of the dataset. The number of trollings in the dataset is very low. This means that the number of the vulnerable comments would be low as well and it can affect the performance of the classifier.

We experimented with two methods trying to balance the dataset. The first assigns weights to each class to balance the dataset. The weights are adjusted inversely proportional to class frequencies in the training set; higher weight means the classifier puts more emphasis on the class during the training phase. For instance, during the training phase if the classifier makes a wrong decision for a troll vulnerable comment it would be ”punished” more than making an error for a non-vulnerable comment.

The other method is a combination of two balancing methods; the Synthetic Minority Over-Sampling Technique [34] (SMOTE) and the Tomek links [35] method. This combination was first used in [36]. The key idea is that we over-sample the minority class using the SMOTE approach and under-sample the majority class using the Tomek links method.

Both methods are applied only during the training phase and only on the train-

ing dataset; we do not alter the test dataset. They both manage to improve the performance of the model. However, the SMOTE and Tomek links method has two important drawbacks. It is not scalable for large amount of data and it can be sensitive to noise. Therefore, we report results of the weighting method.

In addition, we tried another technique to overpass the imbalance of the dataset. We used an one-class classification model to predict troll vulnerable comments. Such models are different from and more difficult than the traditional classification problem. Instead of using a training set that contains comments from both classes, the model learns from a training set containing only the troll vulnerable comments. Comments that are not troll vulnerable are not used in the training phase. Then, the model classifies the incoming comments based on their distance (or similarity) from the known training set. In particular, we used One-Class SVM classifier [37] with a radial basis function (rbf) kernel. Unfortunately, the classifier did not performed well in our dataset.

Feature Group	Accuracy	Precision	Recall	AUC
Content	0.83	0.02	0.38	0.65
Author	0.78	0.02	0.62	0.77
History	0.80	0.02	0.57	0.73
Participants	0.82	0.01	0.61	0.76
Content + Author	0.78	0.02	0.64	0.77
Content + Author + History	0.80	0.02	0.66	0.80
Content + Author + History + Participants	0.83 *(0.78)*	0.03 *(0.67)*	0.68 *(0.68)*	0.82 *(0.82)*
Random Prediction	0.54	0.01	0.45	0.50
Random Biased Prediction	0.89	0.01	0.09	0.50

Table 5.2: Prediction results for the various groups of features and combinations of the groups. In the parenthesis we include the performance of the model when the dataset was randomly balanced in both training and test phases.

5.2.2 Troll Vulnerability Results

We implemented classification models using the four group of features introduced in Section 5.1. To understand the relative importance of each group, we compare the performance of each group individually using logistic regression. We then incrementally combine the groups. In addition to the Logistic Regression classifiers,

we consider two random classifiers as baselines. The first one is unbiased, i.e. each comment has a 50% probability to be vulnerable. The second classifier is biased to the proportion of the classes. Table 5.2 shows the classification results.

All classifiers outperform random predictions. Accuracy is very high in most cases, but this is basically due to the fact that the classes are highly unbalanced. Precision is low again due to the imbalance of the dataset. Thus, recall and the area under the ROC curve (AUC) are the most interesting measures.

Note that we also experimented with reducing the size of the majority class both in training and testing phase of the classifier. In particular, we randomly (under)sampled the majority class reducing its size to twice the size of the minority class. This resulted to drastic improvement of precision, up to 67%, the recall remained the same (68%) and the accuracy was slightly reduced to 77%. However, this is not a realistic scenario because the analogy of troll vulnerable and non-vulnerable comments is completely different.

Content features are the weakest of the four groups of features, followed by the history group that includes features related to the ancestor comments. Features related to the users that post the comments seem to carry a stronger signal, since both the author group and the participants group (that includes information about the authors of the ancestor comments) work better. This indicates that the author of the comment as well as the authors of the preceding comments affect vulnerability more that the comments themselves.

Combining features improves the prediction, with the classifier using features from all four groups being the best.

Individual Features

We also investigate the relative importance of individual features. To this end, we selected from each of the four groups the three features with the highest (in absolute value) logistic regression coefficients and build the corresponding single-feature classifiers. Table 5.3 shows the results of the single-feature classification. In terms of content, using positive and negative words in a comment affects troll vulnerability. In terms of the author of the comment, the fact that the author has previously posted trollings or is negatively perceived by the community is, as expected, a strong signal. The same holds for the history of the ancestor comments and the authors of these comments.

Feature Group	Feature	Accuracy	Recall	AUC
Content	#negative words	0.89	0.21	0.56
	#positive words	0.42	0.59	0.51
	whether is trolling	0.98	0.13	0.56
Author	#trollings	0.90	0.44	0.67
	sum positive score	0.87	0.27	0.57
	sum negative score	0.88	0.27	0.58
History	#zero scored comments	0.94	0.20	0.57
	#negative scored commments	0.88	0.43	0.66
	#trolling ancestors	0.95	0.22	0.58
Participants	#trollings	0.88	0.55	0.71
	#negative scored comments	0.77	0.53	0.66
	#zero scored comments	0.76	0.52	0.64

Table 5.3: Classification results using a single feature.

		θ																			
		0.15				0.20				0.25				0.30				0.35			
		A	P	R	AUC	A	P	R	AUC	A	P	R	AUC	A	P	R	AUC	A	P	R	AUC
K	2	0.81	0.05	0.67	0.81	0.81	0.04	0.67	0.81	0.82	0.03	0.67	0.82	0.83	0.03	0.68	0.82	0.84	0.02	0.67	0.82
	3	0.81	0.04	0.68	0.81	0.82	0.03	0.67	0.82	0.83	0.02	0.68	0.83	0.85	0.02	0.69	0.84	0.86	0.01	0.70	0.85
	5	0.81	0.02	0.69	0.83	0.83	0.02	0.70	0.84	0.86	0.01	0.73	0.87	0.88	0.01	0.76	0.89	0.90	0.01	0.79	0.90
	8	0.82	0.01	0.72	0.86	0.84	0.01	0.74	0.88	0.86	0.01	0.78	0.90	0.89	0.01	0.80	0.91	0.90	0.01	0.80	0.92

Table 5.4: Performance of the model with different values of K and θ. A, P, R, AUC stand for accuracy, precision, recall and AUC, respectively.

Varying the Vulnerability Parameters

In addition to the previous experiments, we also study the performance of the model for different values of K and θ. The results are shown in Table 5.4 for classifiers that include all features. We can see that both parameters act like filters on the vulnerable comments. Larger values of K and θ increase the selectivity in the troll-vulnerability definition, resulting in fewer comments considered as vulnerable (Table 4.4). The performance of the classifier improves when the classes of vulnerable comments become more selective. The improvement is not always that significant, but surely is notable.

Split Setting	Accuracy	Precision	Recall	AUC
50%-50%	0.83	0.03	0.68	0.76
75%-25%	0.85	0.03	0.69	0.77
100%-100%	0.93	0.08	0.92	0.92

Table 5.5: The performance of the classifier including the user vulnerability as a feature.

5.3 User Vulnerability

As we discussed earlier, it seems that the number of trollings that a user's posts is important to the decision of the classifier. We also wanted to investigate if the number of vulnerable comments of a user can improve the performance of the classifier. Thus, we trained the classifier including as feature the number of previous vulnerable comments of the user.

To this direction, we had to split the dataset in two parts. The first part is used to create a history of the users, i.e., count the number of vulnerable comments of the users, and the second part is used as input to the classifier. We tested three different settings on splitting the dataset; 50%-50%, 75%-25% and 100%-100%. The split is done randomly, because our dataset has no time continuity. The submissions may have been created months between with each other. In the last setting, the vulnerability of the user is extracted using the whole dataset and the the classifier is tested using also the whole dataset.

Table 5.5 shows the performance of the model using the user vulnerability as feature. The feature does not make any difference in the first setting, whereas in the second setting we notice a small improvement (around 1%). However, the performance of the model could also be affected by size reduction of the dataset. In the last setting, we notice a significant increase in the recall of the experiments (around 90%). These results are not reliable because the classifier seems to indirectly know the truth. There are a lot of users that posted only one comment, thus the vulnerability of the user must be 1 if the comment is troll vulnerable and 0 otherwise.

Maybe our dataset is not appropriate for this task. As we discussed, the dataset has no time continuity. In addition, we would like to contain a longer history of the users (including the vulnerability of their comments). An idea is to crawl subreddits that

have common users from their inception. Such dataset would provide a long history for a lot of users. Another thought for future work is to define user vulnerability differently, e.g. elaborating information from the social graph.

5.4 Trolling Escalation

We also study the relationship between trollings and vulnerable comments.

In both top and controversial submissions, a trolling comment has a 5% probability to be vulnerable, whereas a non-trolling comments has a 0,6% probability to be vulnerable. Thus as expected, a trolling comment is more likely to attract trolls than a non-trolling one.

A first question is whether all vulnerable comments are trollings. The percentage of the vulnerable comments that are trollings themselves in top submissions is 12.8% and in controversial 13.8%. This means that a comment does not have to be a trolling to attract trolls. Thus, early detection of such comments is a useful tool to moderators. In Figure 5.1a, we see an example of a benign vulnerable comment with a high *TVRank*.

Another question is whether all trollings are vulnerable to trolls. The percentage of trollings that are vulnerable to trollings in our dataset is 5%. This means that not all trollings generate additional trollings. Some of the trollings escalate, but others do not.

Thus, we ask whether we can use our classifiers to predict whether a trolling comment will escalate or not. To this end, we use our classifier with input only the trolling comments and try to predict whether these trolling comments are vulnerable. Our classifier achieved 64% recall and 70% area under the ROC curve indicating that such a prediction task is possible. This would be a useful tool for distinguishing between trollings that will end-up causing havoc and trollings that will have only a limited effect.

In Figure 5.1b, we see an example of a trolling that escalates. The content of the initial comment is abusive and the comments that follow it are also abusive. Figure 5.1c shows a trolling that did not escalate. User B quotes a phrase (from a movie), that contains inappropriate content, but there is no trolling reaction.

Submission: Why was /r/fatpeoplehate, along with several other communities just banned?
-D: God, the announcement thread is a nightmare to read. [...] and honestly, this felt like the only safe place here on Reddit right now.
-A: F**k off you fat cancer.
-D: Don't bring your FPH toxic mentality here.
-A: LOL [...]I just wanted to end your nonsense "safe place." Go kill yourself, no one wants you alive.
[...]

(a) An example of a vulnerable comment.

Submission: Pakistan Is Arresting People Who Refuse to Vaccinate Their Kids Against Polio
-F: *Good. I hope they throw away the key and let these wa***s rot. [...] your right to be an ignorant f****ard ends at the point [...].*
-S: *FAIL. Not. Even. Close. [...] You are the ignorant f-**rd. It's really getting tiring listening to uninformed blowhards like you. [...]*
-F: Unfortunately the thing about diseases, smart guy, [...] be a stubbornly ignorant s***head [...] and stick it up you're a**e [..] your petulant stupidity.

(b) An example of a trolling that escalated.

Submission: A biotech startup has managed to 3-D print fake rhino [...] undercutting the price poachers can get and forcing them out eventually.
[...] **-B:** *You're not wrong, Walter, you're just an a*****e.*
-R: *Saw that coming.*

(c) An example of a trolling that did not escalate.

Figure 5.1: Examples of trolling behavior.

5.5 Trolling Detection

Another interesting idea, is to use the model that we used for the prediction of vulnerable comments to detect trollings. Using the exact same settings as in the vulnerability prediction, the classifier yielded very high accuracy and recall in both top and controversial submissions. In top submissions the classifier achieved about 92% accuracy and 89% recall and in controversial submissions the same metrics were 92% and 87% respectively. However, we included in the features the number of trollings that the user has posted. Similarly with user vulnerability, the classifier is cheating because of the users that posted only a few comments. Thus, we repeated the experiment excluding the features related to the number of trolling posts a user has posted. We

notice that the performance of the classifier decreased; 83% accuracy and 85% recall for the top submissions and 84% accuracy and 66% recall for the controversial submissions.

In addition, it want to investigate whether the troll vulnerability of the comment can improve the performance of this classifier. Thus, we included in the experiment the *TVRank* of the comment as a feature in the classifier. The performance of the classifier remained unchanged, which indicates that the troll vulnerability of the comment is not a strong feature.

Chapter 6

Conclusions & Future Work

Understanding and detecting trolling behavior in social networks has attracted considerable attention. In this work, we take a different approach shifting the focus from the trolls to their victims. In particular, we introduce the novel concept of troll vulnerability to characterize how susceptible a target is to trolls. We provide an intuitive measure of troll vulnerability, termed *TVRank*. This measure uses random walks to account for both the volume and the proximity of the trolling activity associated with each target.

We apply this measure in user posts from the Reddit website. Intuitively, the *TVRank* value of a given comment c is the probability that the random walk visits a trolling, given that it is visiting a descendant of c. Therefore, the higher the TVRank value of a post, the more vulnerable to trolls the post is. Using this measure, we distinguish the comments into two categories; troll vulnerable and not troll vulnerable.

We also address the troll vulnerability prediction problem: given a post how to predict whether this post will attract trolls in the future. Predicting the vulnerability of a post can be a powerful weapon against the trolling phenomenon, because it can allow handling trolls proactively. For instance, such predictions can enable administrators of online communities to take precautionary measures to prevent trollings to appear instead of just detecting them, when they appear. However, this problem proved to be very hard. The dataset is heavily imbalanced, the troll vulnerable comments constitute about 1% of the dataset. In addition, the factors that make a comment vulnerable to trolls are not completely clear. We have built a classifier that combines features

related to the post and its history (i.e., the posts preceding it and their authors) to identify vulnerable posts. Our initial results using the Reddit dataset are promising, suggesting that a proactive treatment of trolls is feasible.

In the future, we plan to extend our evaluation, applying our classifier to predict troll vulnerability in larger datasets, including other social networks in addition to Reddit. An essential prerequisite for the datasets is the accurate annotation of the posts as trollings or not. In this work, we used a model that is not fully accurate and can introduce noise into our analysis. Additional features can also be considered in the future. Including semantic features in collaboration with sentiment features in the classifier may improve the performance of the prediction task. For example, a post referring to a public figure in negative manner may be more likely to attract trolls.

In addition, our work creates interesting directions for future work towards studying vulnerability at different levels than that of a post. Studying the vulnerability of the users seems an interesting direction. In Section 5.3, we used a simple definition of user vulnerability as feature in our classifier. However, more ways to measure the user vulnerability can be considered. For example, we could build graph-based models to measure the vulnerability of the user, using information from the social graph. Another interesting question for the user vulnerability is how it evolves through time.

Furthermore, identifying topics or concepts that are more likely to attract trolls seems to be an interesting problem. As we discussed in Section 4.4.2, it seems that there are subreddits and submissions that attract more trolls. Thus, it would be interesting to study if there is any relationship between subreddits and submissions and topics.

Bibliography

[1] J. Atwood, "Suspension, ban or hellban?." http://goo.gl/TxCGi7, 2011. Accessed: 2016-06-11.

[2] E. Cambria, P. Chandra, A. Sharma, and A. Hussain, "Do not feel the trolls," in *Proceedings of the 3rd International Workshop on Social Data on the Web*, *ISWC*, 2010.

[3] S. O. Sood, E. F. Churchill, and J. Antin, "Automatic identification of personal insults on social news sites," *Journal of the American Society for Information Science and Technology*, vol. 63, no. 2, pp. 270–285, 2012.

[4] J. de-la Peña-Sordo, I. Pastor-López, X. Ugarte-Pedrero, I. Santos, and P. G. Bringas, "Anomalous user comment detection in social news websites," in *International Joint Conference SOCO'14-CISIS'14-ICEUTE'14*, pp. 517–526, Springer, 2014.

[5] J. Cheng, C. Danescu-Niculescu-Mizil, and J. Leskovec, "Antisocial behavior in online discussion communities," in *Proceedings of ICWSM*, 2015.

[6] S. Jeong, "Does twitter have a secret weapon for silencing trolls?." http://goo.gl/HcuL20, 2014. Accessed: 2016-06-11.

[7] J. Suler, "The online disinhibition effect," *Cyberpsychology & behavior*, vol. 7, no. 3, pp. 321–326, 2004.

[8] S.-H. Lee and H.-W. Kim, "Why people post benevolent and malicious comments online," *Commun. ACM*, vol. 58, pp. 74–79, Oct. 2015.

[9] C. Hardaker, "Trolling in asynchronous computer-mediated communication: From user discussions to academic definitions," *Journal of Politeness Research*, vol. 6, pp. 215–242, 2010.

[10] J. S. Donath, "Identity and deception in the virtual community," *Communities in cyberspace*, vol. 1996, pp. 29–59, 1999.

[11] E. E. Buckels, P. D. Trapnell, and D. L. Paulhus, "Trolls just want to have fun," *Personality and individual Differences*, vol. 67, pp. 97–102, 2014.

[12] M. Potthast, B. Stein, and R. Gerling, "Automatic vandalism detection in wikipedia," in *Advances in Information Retrieval*, pp. 663–668, Springer, 2008.

[13] S.-C. Chin, W. N. Street, P. Srinivasan, and D. Eichmann, "Detecting wikipedia vandalism with active learning and statistical language models," in *Proceedings of the 4th workshop on Information credibility*, pp. 3–10, ACM, 2010.

[14] B. T. Adler, L. De Alfaro, S. M. Mola-Velasco, P. Rosso, and A. G. West, "Wikipedia vandalism detection: Combining natural language, metadata, and reputation features," in *Computational linguistics and intelligent text processing*, pp. 277–288, Springer, 2011.

[15] S. Mola Velasco, "Wikipedia Vandalism Detection Through Machine Learning: Feature Review and New Proposals: Lab Report for PAN at CLEF 2010," in *Notebook Papers of CLEF 2010 Labs and Workshops, 22-23 September, Padua, Italy* (M. Braschler, D. Harman, E. Pianta, and E. Pianta, eds.), Sept. 2010.

[16] B. Adler, L. de Alfaro, and I. Pye, "Detecting Wikipedia Vandalism using WikiTrust? Lab Report for PAN at CLEF 2010," in *Notebook Papers of CLEF 2010 Labs and Workshops, 22-23 September, Padua, Italy* (M. Braschler, D. Harman, E. Pianta, and E. Pianta, eds.), Sept. 2010.

[17] A. G. West, S. Kannan, and I. Lee, "Detecting wikipedia vandalism via spatio-temporal analysis of revision metadata?," in *Proceedings of the Third European Workshop on System Security*, EUROSEC '10, pp. 22–28, ACM, 2010.

[18] S. Kumar, F. Spezzano, and V. S. Subrahmanian, "VEWS: A wikipedia vandal early warning system," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 607–616, 2015.

[19] P. Galán-García, J. G. de la Puerta, C. L. Gómez, I. Santos, and P. G. Bringas, "Supervised machine learning for the detection of troll profiles in twitter so-

cial network: Application to a real case of cyberbullying," in *International Joint Conference SOCO'13-CISIS'13-ICEUTE'13*, pp. 419–428, Springer, 2014.

[20] I. O. Dlala, D. Attiaoui, A. Martin, and B. Ben Yaghlane, "Trolls identification within an uncertain framework," in *Proceedings of the 26th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 1011–1015, IEEE, 2014.

[21] J. Blackburn and H. Kwak, "Stfu noob!: predicting crowdsourced decisions on toxic behavior in online games," in *Proceedings of the 23rd international conference on World wide web*, pp. 877–888, ACM, 2014.

[22] S. Kumar, F. Spezzano, and V. Subrahmanian, "Accurately detecting trolls in slashdot zoo via decluttering," in *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, pp. 188–195, IEEE, 2014.

[23] 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*, pp. 741–750, ACM, 2009.

[24] 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*, vol. 56, no. 12, pp. 2884–2895, 2012.

[25] Z. Wu, C. C. Aggarwal, and J. Sun, "The troll-trust model for ranking in signed networks," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pp. 447–456, ACM, 2016.

[26] H. Lamba, M. M. Malik, and J. Pfeffer, "A tempest in a teacup? analyzing firestorms on twitter," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, *ASONAM*, pp. 17–24, 2015.

[27] P. Lawrence, B. Sergey, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," technical report, Stanford University, 1998.

[28] T. H. Haveliwala, "Topic-sensitive pagerank," in *Proceedings of the Eleventh International World Wide Web Conference*, *WWW 2002*, *May 7-11*, *2002*, *Honolulu*, *Hawaii*, pp. 517–526, 2002.

[29] G. Jeh and J. Widom, "Simrank: a measure of structural-context similarity," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, pp. 538–543, 2002.

[30] A. Olariu, "Repo for the insults detection challenge on kaggle.com." `https://github.com/andreiolariu/kaggle-insults/`, 2013. Accessed: 2016-06-11.

[31] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," *SIGCOMM Comput. Commun. Rev.*, vol. 29, pp. 251–262, Aug. 1999.

[32] C. Spearman, "The proof and measurement of association between two things," *The American journal of psychology*, vol. 15, no. 1, pp. 72–101, 1904.

[33] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.

[34] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, pp. 321–357, 2002.

[35] I. Tomek, "Two modifications of cnn," *IEEE Trans. Syst. Man Cybern.*, vol. 6, pp. 769–772, 1976.

[36] G. E. Batista, A. L. Bazzan, and M. C. Monard, "Balancing training data for automated annotation of keywords: a case study.," in *WOB*, pp. 10–18, 2003.

[37] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, J. C. Platt, *et al.*, "Support vector method for novelty detection.," *NIPS*, vol. 12, pp. 582–588, 1999.

Appendix A

Examples of Trolling Annotation

Score Range	Text
(0.70, 1]	*"How dare you spew your hateful speech here. You deserve every single downvote you piece of shit."* *"You're an idiot. Shaming and death threats are only expected by idiots."* *"Screw you very much SJWs admins. Eat a bag of dicks. I have faith that fph will revive and be stronger than ever. FAT CUNTS"* *"Go fuck yourself. Your feminist CEO has butchered reddit. The place is gutted. And you want to piss moan about misogyny and men's rights? You SJW twats have won. Reddit is yours now. Everyone who disagrees with your stupid hypersensitive opinions will be banned. So you won. Give it a fucking rest already. Soon reddit will only be people exactly just like you. A new Tumblr. Enjoy."*
(0.50, 0.70]	*"NO ONE ASKED FOR YOUR CHANGES. LISTEN TO YOUR FUCKING USERS!"* *"Welp, fuck this shit. See all you shitlords on Voat."* *"Lol you're out of your fucking mind. Overall those catches had more meaning, but skill wise this is the best catch ever."* *"Dear diary, today OP was a faggot,"*

Score Range	Text
(0.30, 0.50]	*"Did you catch /u/kickme444's TED talk about redditgifts? Apparently it's more fun to give away a pittance than to spend it on yourself."* *"Well said. You really did sum it up exceptionally."* *"Shit happens is my favorite life story. More specifically:Shit happens. You learn to roll with it or throw it around."* *"Not all Jews are Zionists. Learn the difference and quit your baseless accusations of racism."*
(0.10, 0.30]	*"Okay PUBLIC stats could potentially hurt the game. Not allowing you to see your own, or what competitive players are doing, is poor design imo."* *"Well to be completely fair in the first few levels if you fight more perfectly you'll use less food, thus you'll have Swallow before you need to heal a lot."* *"you wouldn't be angry to know that 40% of the money you (most likely) worked pretty damn hard for would disappear when you died?"* *"Dude, you're totally missing the point."*
[0, 0.10]	*"Sure, so you might as well believe I'm superman. Can't prove me wrong."* *"you really think no other game has a problem with toxicity? play CSGO, WoW or any Xbox Live game for 5 minutes rofl"* *"I was really hoping for a 5 game series for both finals. I'm a bit disappointed about how one sided this series was."* *"Yup because Dominoes, Footlocker and JD Sports were to blame for police brutality. We sure showed the man! "It was a sad state of affairs.""*

Table A.1: Examples of the annotation of comments in our dataset using the model described in Section 4.2.2.

Appendix B

Troll Vulnerability Rank Examples

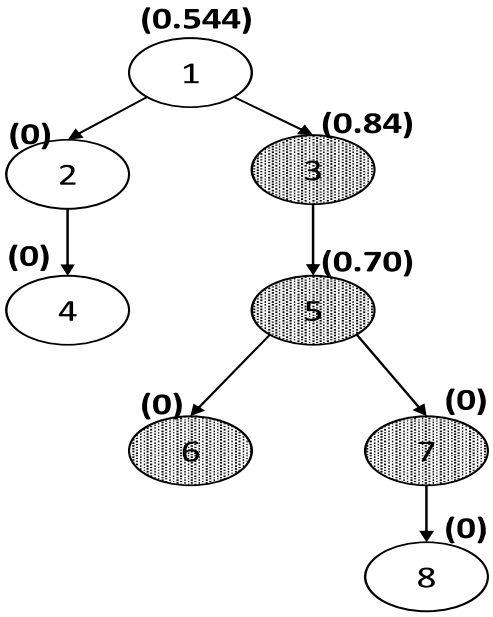


Figure B.1: An example of a conversation subtree. Values in bold correspond to the *TVRank* values of the nodes. Shadowed nodes correspond to trollings.

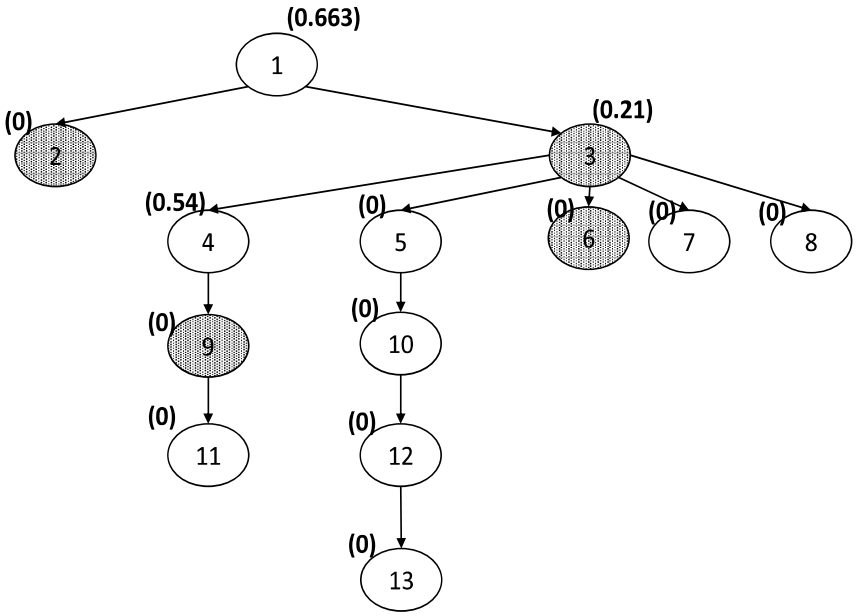


Figure B.2: An example of a conversation subtree. Values in bold correspond to the *TVRank* values of the nodes. Shadowed nodes correspond to trollings.

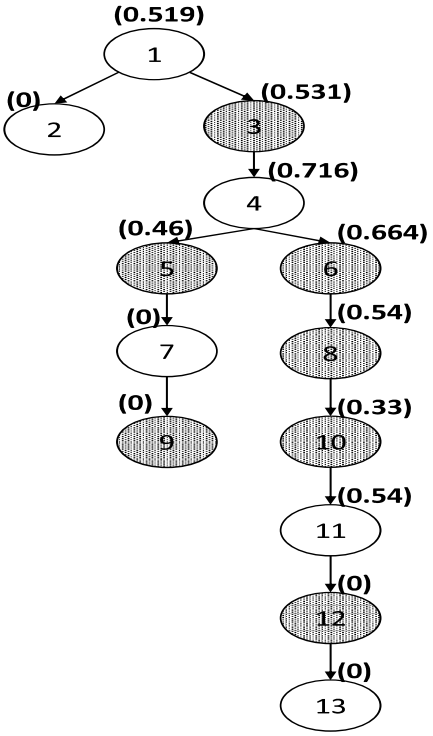


Figure B.3: An example of a conversation subtree. Values in bold correspond to the *TVRank* values of the nodes. Shadowed nodes correspond to trollings.

Appendix C

Troll Vulnerability Predictions Results for the Controvesial Submissions

Feature Group	Accuracy	Precision	Recall	AUC
Content	0.79	0.02	0.39	0.63
Author	0.72	0.02	0.62	0.68
History	0.73	0.02	0.58	0.69
Participants	0.82	0.03	0.57	0.73
Content + Author	0.72	0.03	0.63	0.76
Content + Author + History	0.77	0.03	0.65	0.78
Content + Author + History + Participants	0.86 *(0.76)*	0.01 *(0.64)*	0.67 *(0.67)*	0.81 *(0.81)*
Random Prediction	0.54	0.02	0.45	0.50
Random Biased Prediction	0.89	0.02	0.09	0.50

Table C.1: Prediction results for the various groups of features and combinations of the groups.

Feature Group	Feature	Accuracy	Recall	AUC
Content	#negative words	0.88	0.24	0.56
	#positive words	0.42	0.59	0.50
	whether is trolling	0.96	0.14	0.56
Author	#trollings	0.88	0.44	0.66
	sum positive score	0.83	0.33	0.58
	sum negative score	0.84	0.33	0.58
History	#zero scored comments	0.91	0.19	0.55
	#negative scored commments	0.78	0.49	0.64
	#trolling ancestors	0.94	0.23	0.58
Participants	#trollings	0.87	0.52	0.70
	#negative scored comments	0.73	0.52	0.62
	#zero scored comments	0.70	0.48	0.59

Table C.2: Classification results using a single feature.

		θ																			
		0.15				0.20				0.25				0.30				0.35			
		A	P	R	AUC	A	P	R	AUC	A	P	R	AUC	A	P	R	AUC	A	P	R	AUC
K	2	0.78	0.06	0.65	0.78	0.79	0.05	0.66	0.79	0.80	0.04	0.66	0.80	0.81	0.04	0.67	0.81	0.81	0.03	0.67	0.81
	3	0.78	0.05	0.66	0.79	0.79	0.04	0.67	0.80	0.80	0.03	0.68	0.81	0.82	0.02	0.69	0.83	0.83	0.02	0.71	0.84
	5	0.78	0.03	0.66	0.79	0.80	0.02	0.68	0.81	0.82	0.01	0.69	0.83	0.85	0.01	0.72	0.86	0.86	0.01	0.76	0.88
	8	0.80	0.01	0.67	0.83	0.83	0.01	0.69	0.84	0.87	0.01	0.70	0.85	0.87	0.01	0.73	0.87	0.88	0.01	0.76	0.88

Table C.3: Performance of the model with different values of K and θ. A, P, R, AUC stand for accuracy, precision, recall and AUC, respectively.

Split Setting	Accuracy	Precision	Recall	AUC
50%-50%	0.83	0.03	0.68	0.76
75%-25%	0.85	0.03	0.69	0.77
100%-100%	0.93	0.08	0.92	0.92

Table C.4: The performance of the classifier including the user vulnerability as a feature.

Author's Publications

1. P. Tsantarliotis, and E. Pitoura. "Topic Detection Using a Critical Term Graph on News-Related Tweets." *In the Workshop Proceedings of EDBT/ ICDT Joint Conference*. pp. 177-182, 2015.

2. P. Tsantarliotis, E. Pitoura, and P. Tsaparas. "Troll Vulnerability in Online Social Networks." *In the Proceedings of International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2016

Short Biography

Paraskevas Tsantarliotis was born in Agrinio, Greece in 1989. Currently, he is a Graduate Student in the Department of Computer Science and Engineering of the University of Ioannina. He received his B.Sc. degree from the same institution in 2014. He is a member of *Data, Algorithms, Technologies and Architectures (D.A.T.A.) Research Lab*. His research interests include Data Mining, Social Network Analysis and Software Engineering.