Noname manuscript No.
(will be inserted by the editor)

Defining and Predicting Troll Vulnerability in Online Social Media

Paraskevas Tsantarliotis · Evaggelia Pitoura · Panayiotis Tsaparas

Received: date / Accepted: date

Abstract Trolling describes a range of antisocial online behaviors that aim at disrupting the normal operation of online social networks and media. Existing approaches to combating trolling rely on human-based or automatic mechanisms for identifying trolls and troll posts. In this paper, we take a novel approach to the problem: our goal is to identify troll vulnerable posts, that is, posts that are potential targets of trolls, so as to prevent trolling before it happens. To this end, we define three natural axioms that a troll vulnerability metric must satisfy and introduce metrics that satisfy them. We then define the troll vulnerability prediction problem, where given a post we aim at predicting whether it is vulnerable to trolling. 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 identifying a large fraction of the troll vulnerable posts.

P. Tsantarliotis
Department of Computer Science & Engineering,
University of Ioannina
Ioannina, Greece
E-mail: ptsantar@cs.uoi.gr

E. Pitoura
Department of Computer Science & Engineering,
University of Ioannina
Ioannina, Greece
E-mail: pitoura@cs.uoi.gr

P. Tsaparas
Department of Computer Science & Engineering,
University of Ioannina
Ioannina, Greece
E-mail: tsap@cs.uoi.gr

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 participate 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 dialog 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 dialog, 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. 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. 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 (see for example, Atwood (2011)). Even when successful, troll detection does not

fully address the problem. First, trolls are very good at working the system, for example, getting around bans by using different usernames, or masking the content of their postings, (see for example, Jeong (2014)). 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 paper, we take a different approach to addressing trolling. Instead of detecting trolls, we focus on identifying possible targets of trolls, that is, *troll vulnerable* posts. We first introduce *troll vulnerability* (TV) metrics that quantify the vulnerability of a post based on the amount of trolling and non-trolling activity that followed the post. We provide three natural axioms that a troll vulnerability metric must satisfy, and show that our metrics fulfill them. We then define the *troll vulnerability prediction task*, where, given a post, our goal is to predict whether it will be targeted by trolls. We build classification models for this task that use features of the post and its history to predict its vulnerability.

Our approach has several advantages, compared to traditional troll detection mechanisms. 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. Troll vulnerability metrics also offer a way to measure the severity of the troll activity with respect to a post. This is useful for monitoring the health of the system. It can also be applied for trolling posts for measuring and predicting trolling escalation. Finally, vulnerability prediction is a *pro-active* tool against the trolls. 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.

In summary, in this paper 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 *troll vulnerability metrics* for quantifying the vulnerability of a post to trolls. We define a set of axioms that we want our metrics to satisfy.
- 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 evaluate our approach using 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.

The rest of the paper is structured as follows. Section 2 reviews related work in trolling. In Section 3, we introduce the troll vulnerability metrics, while in Section 4, we define the classifier for predicting vulnerable posts. In Section 5, we present our experimental analysis, and in Section 6, we conclude the paper.

2 Related Work

The term *trolling* has been widely used to characterize different types of anti-social and disruptive online behavior. Some of it may be innocent, if not entertaining, but there are cases where it escalates to threatening and bullying behavior. As a result, trolling has become equivalent with online harassment. Previous work attempts to characterize such behavior and its various aspects, including the aspects of identity deception (see, Donath (1999)), and malicious impoliteness (see, Hardaker (2010)). There is also research on explaining the causes of trolling. For example, Suler (2004) shows that trolling is related to the (toxic) online disinhibition effect, while Buckels et al (2014) indicate that there is a relation between trolls and sadism.

Due to its critical importance, the problem of identifying malicious users and content 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 trolling 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. Related work along this line of research includes detecting vandalism (see, Adler et al (2011); Potthast et al (2008); Chin et al (2010)), and vandals (see, Kumar et al (2015)) in Wikipedia, bad behavior in multi-player online games (see, Blackburn and Kwak (2014)), and trolling comments in social new sites (see, Cambria et al (2010); Sood et al (2012); de-la-Peña-Sordo et al (2014)). In a recent study, Cheng et al (2015) analyze users who were banned from three large online discussion communities to identify the characteristics of their behavior and how this behavior changes through time. These characteristics were exploited to identify early the users who will be banned.

Another line of research in troll detection assumes the availability of a signed social graph among users where signs indicate positive and negative relations-

hips among users. Then, troll detection is modeled as a ranking problem in this graph. Related approaches use iterative algorithms that calculate centrality measures, (see, for example, Kumar et al (2014); Kunegis et al (2009)), or the trustworthiness of the user (see, for example Wu et al (2016); Ortega et al (2012)).

Lamba et al (2015) investigate how firestorms on Twitter affect the relationships between users. A firestorm is the event where a target (e.g. public figure) receives a large amount of negative attention. Firestorms are much different than trolling; firestorms may include trolls, but not all participants in firestorms are trolls. Thus, this problem is different than ours.

Our approach differs from these works. The key novelty is that we turn the spotlight on 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.

A poster of a preliminary version of this work appeared in Tsantarliotis et al (2016).

3 Modeling Troll Vulnerability

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

3.1 Preliminaries

In order to address the problem of troll vulnerability we need a definition of what constitutes trolling. We use the term *trolls* to refer to users that behave in a deceptive, destructive and disruptive manner in an online social setting. We use the term *trollings* to refer to the posts or messages generated by trolls that aim to hurt specific people or groups. In the following, we assume that we have some method for detecting trolls and trollings. 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.

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 paper, we will use Reddit, 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 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 1. The tree structure in posts is common to many social media. We note that our metrics are applicable to more general graph structures as well, as long as we can define a topological sorting of the nodes that corresponds to the temporal order of the posts. For simplicity, in the following we assume that the graph G corresponds to a discussion tree.

In order to define troll vulnerability, we need to assume that the graph G is *labeled*. That is, each node $v \in V$ is associated with a label $\ell_v \in \{T, NT\}$ indicating whether the node v is a trolling (T) or not (NT). Thus, we will think of the graph as a triplet $G = (V, E, L)$ where $L = \{l_v : v \in V\}$ is the set of labels of the nodes in V.

Our goal is to define metrics that quantify the vulnerability of a post to trolling attacks. Given a discussion tree $G = (V, E, L)$, a troll vulnerability metric is a function $TV : V \to \mathbf{R}$ that maps each post $p \in V$ in the discussion graph into a real number $TV(p)$, that captures the degree of vulnerability of the post p. Higher $TV(p)$ values indicate that the post is more vulnerable. We assume that the $TV(p)$ value is a function of the subtree $G_p = (V_p, E_p, L_p)$ rooted at node p, and thus, abusing the notation, we will sometimes write $TV(G_p)$. This is an important assumption. The vulnerability of a node depends solely on the subtree rooted at that node. Two nodes with the same subtrees, in terms of both structure and labels, will have the exact same vulnerability value.

3.2 Troll Vulnerability Axioms

Similarly to the definitions of distance or similarity between items, there are multiple ways to define vulnerability. The definition of a specific TV metric depends on the sensitivity and needs of the user of the metric (e.g., the administrator that is monitoring the health of a social networking system). However, as in the case of distance or similarity metrics, we want the TV metric

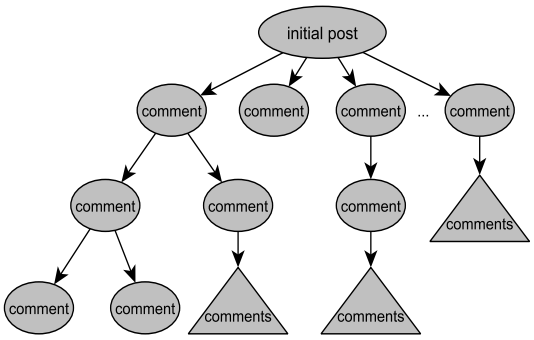


Fig. 1: An example of a conversation tree.

to satisfy certain properties, in order to be well defined. We thus define three axioms that any troll vulnerability metric must satisfy.

As we mentioned before, the TV value of a post p as a function of the subtree G_p rooted at node p. Therefore, we will talk about the TV value of a tree G_p, and study what happens to the TV metric when there are changes in the subtree G_p. As such, it makes sense to compare the TV values for posts that have subtrees with similar structure.

Axiom 1 (Trolling Monotonicity) *Let p be a post with subtree $G_p = (V_p, E_p, L_p)$, and let $G'_p = (V'_p, E'_p, L'_p)$ denote the subtree of p when we add a trolling node v to the subtree, that is, $V'_p = V_p \cup \{v\}$, $L'_p = L_p \cup \{\ell_v\}$ and $\ell'_v = T$, and $E'_p = E_p \cup \{(w, v)\}$ for some $w \in V_p$. Then, $TV(G_p) < TV(G'_p)$.*

Axiom 1 essentially says that the vulnerability metric of a post should increase with the addition of new trolling descendants. Equivalently, the TV metric should be a function that is strictly increasing with respect to the number of trolling nodes in the post subtree.

Axiom 2 (Non-Trolling Monotonicity) *Let p be a post with subtree $G_p = (V_p, E_p, L_p)$, and let $G'_p = (V'_p, E'_p, L'_p)$ denote the subtree of p when we add a non-trolling node v to the subtree, that is, $V'_p = V_p \cup \{v\}$, $L'_p = L_p \cup \{\ell_v\}$ and $\ell'_v = NT$, and $E'_p = E_p \cup \{(w, v)\}$ for some $w \in V_p$. Then, $TV(G_p) > TV(G'_p)$.*

Axiom 2 essentially says that the vulnerability metric of a post should decrease with the addition of new non-trolling descendants. Equivalently, the TV metric should be a function that is strictly decreasing with respect to the number of non-trolling nodes in the post subtree.

For the last axiom, we need to define the notion of distance in the discussion tree. Let p be a post, and let G_p the subtree of p, and v a descendant of p. The distance $d(p, v)$ between p and v is defined as the length of the path from p to v in the tree.

Axiom 3 (Troll Distance Monotonicity) *Let p be a post with subtree $G_p = (V_p, E_p, L_p)$, where there are nodes $u, v \in V_p$, such that $\ell_u = NT$ and $\ell_v = T$, and $d(p, u) < d(p, v)$. Let $G'_p = (V_p, E_p, L'_p)$ denote the subtree of p when we swap the labels of nodes u and v, that is, $\ell'_u = T$, and $\ell'_v = NT$. Then $TV(G_p) < TV(G'_p)$.*

Axiom 3 says that the vulnerability metric of a post should increase if we bring the trolling labels closer to the post. Essentially, it says that the TV-metric is monotonic with respect to the distances of the trollings to the root node.

Axioms 1 and 2 determine the effect of the volume of trolling and non-trolling posts in the post replies, while Axiom 3 refers to the relative position of the trollings in the post subtree. Intuitively, a post is more vulnerable if it is followed by many trollings compared to non-trollings, and the trollings appear close to the post in the discussion tree.

Note that our axioms are general enough to capture the effect of changing the label of a node in the tree. Intuitively, we would like the TV metric to increase if we change the label of a node from non-trolling to trolling, and to decrease in the opposite case. This follows from our axioms. We can emulate the change of a label using the addition and swap operations we described above. For example, to change the label of a node v from non-trolling to trolling, we can add a trolling node u to the subtree of v, swap the labels of v and u, and then remove the node u. From our axioms it follows that all these operations increase the TV metric value. We treat analogously the change of a label from trolling to non-trolling.

3.3 Troll Vulnerability Metrics

We now define three different troll vulnerability metrics. For the following recall that $G_p = (V_p, E_p, L_p)$ denotes the subtree rooted at node p in G. We also use $T_p = \{u \in G_p : \ell_u = T\}$ to denote the trolling descendants of p, and $NT_p = \{u \in G_p : \ell_u = NT\}$ to denote the non-trolling descendants of p.

3.3.1 The TVDiff metric

The idea behind the *TVDiff* metric is to use the difference between trolling and non-trolling descendants of a post p to define the vulnerability of the post. To give some additional control to the user of the metric, we introduce a parameter α that controls the relative importance between trolling and non-trolling descendants. For example, if $\alpha = 2$ this means that a trolling is twice more important than a non-trolling, and thus

the presence of a trolling counts as two non-trollings. Furthermore, in order to enforce Axiom 3 we weight the presence of a descendant v of the post p by a factor $\beta^{-d(p,v)}$. Therefore, posts that are further from the root post p contribute less to the metric. We thus define the *TVDiff* metric for post p as follows:

$$TVDiff(p) = \alpha \sum_{u \in T_p} \beta^{-d(p,u)} - \sum_{u \in NT_p} \beta^{-d(p,u)}$$

We can prove the following theorem.

Theorem 1 *The TVDiff metric satisfies Axioms 1, 2, and 3.*

The proof follows directly from the definition of the *TVDiff* metric. Adding a new trolling node v to the subtree of node p, will increase the $TVDiff(p)$ value by $\alpha\beta^{-d(p,v)}$, while adding a non-trolling node v will decrease the $TVDiff(p)$ value by $\beta^{-d(p,v)}$. Swapping the labels of non-trolling node u and trolling node v, at distances $d(p,u) < d(p,v)$, will increase the $TVDiff(p)$ value by $(\alpha+1)(\beta^{-d(p,u)} - \beta^{-d(p,v)})$.

3.3.2 The TVRatio metric

The idea behind the *TVRatio* metric is to use the fraction of the trolling descendants of a post to define its vulnerability. Again, we weight the presence of a descendant v of the post p by a factor $\beta^{-d(p,v)}$, so as to satisfy Axiom 3. However, there are extreme cases when our metric does not satisfy Axioms 1 and 2. If all descendants of p are trollings, then the addition of an additional trolling descendant does not change the ratio. Similarly, if all descendants of p are non-trollings, the addition of a non-trolling node has no effect. To alleviate this problem, we allocate weight ϵ to both the trolling and non-trolling classes, regardless of the number of descendants in each class. This acts as a smoothing factor, ensuring that the *TVRatio* can never become 1 or 0. In this way, the metric satisfies Axioms 1 and 2.

In conclusion, we define the *TVRatio* as follows:

$$TVRatio(p) = \frac{\sum_{u \in T_p} \beta^{-d(p,u)} + \epsilon}{\sum_{u \in V_p} \beta^{-d(p,u)} + 2\epsilon}$$

Note that in practice, we can make ϵ arbitrarily small, and this has essentially no effect in our experiments.

We can prove the following theorem.

Theorem 2 *The TVRatio metric satisfies Axioms 1, 2, and 3.*

Similar to before, the proof follows directly from the definition of the *TVRatio* metric. Adding a new trolling node v to the subtree of node p, increases the enumerator by a factor of $\beta^{-d(p,v)}$. The smoothing factor 2ϵ in the denominator guarantees that enumerator and denominator can never be equal, hence the addition of a trolling node will always lead to an increase. Adding a non-trolling node increases the denominator, while the enumerator stays the same, hence decreasing the *TVRatio* metric. Again, because of the smoothing factor ϵ, the enumerator can never become zero. Swapping the labels of non-trolling node u and trolling node v, at distances $d(p,u) < d(p,v)$, will again increase the enumerator, while the denominator stays the same, and hence increase the *TVRatio* value.

3.3.3 The TVRank metric

The *TVRank* is the more sophisticated of our three metrics. For this metric, we use Random Walks with Restarts (RWR) for the definition of the troll vulnerability of a post p. Intuitively, we relate the vulnerability of the node p with the probability that a random walk starting from p will visit a trolling descendant. The RWR takes place in the subtree G_p, where at each transition there is a chance β that the random walk restarts at p. For each descendant v of p it defines a probability $P_p(v)$ that the random walk is at node v after an infinite number of iterations. The *TVRank* metric is defined as follows:

$$TVRank(p) = \frac{\sum_{v \in T_p \setminus \{p\}} P_p(v)}{1 - P_p(p)}$$

$TVRank(p)$ is the probability that the RWR visits a trolling node, given that it is visiting a descendant of p.

Similar to the *TVRatio* metric, in the extreme case that all nodes are trollings (or non-trollings) the addition of a new trolling (non-trolling) node would have no effect on the *TVRank* value. To avoid such extreme cases, we add two "dummy" nodes t and n as children of every node $u \in V_p$ in the subtree of node p, with a fixed weight ϵ. The nodes are labeled as trolling, and non-trolling respectively. This way, no subtree in G_p can be "monochromatic", consisting of only trolling or non-trolling nodes, which alleviates the problem.

Practically, the vector of probabilities P_p is computed as follows.

$$P_p = (1-\beta)\, P_p\, A + \beta\, e_p,$$

where β is the restart probability, A is the row-stochastic transition matrix, and e_p is the restart vector, with $e_p(p) = 1$, and 0 otherwise. A is the normalized adjacency matrix of the graph G_p. In particular, for a node u, with set of children $C(u)$, we have that $A[u,n] = A[u,t] = \epsilon/(|C(u)| + 2\epsilon)$, and

$A[u,v] = 1/(|C(u)| + 2\epsilon)$ for all nodes $v \in C(u)$. For leaf nodes v, we set $A[u,n] = A[u,t] = \epsilon/(1+2\epsilon)$ and $A[v,p] = 1/(1+2\epsilon)$, that is, the random walk restarts at node p. The dummy nodes restart at node p. We note again that in practice, we can make ϵ arbitrarily small, and this has essentially no effect in our experiments.

We can prove the following theorem.

Theorem 3 *The TVRank metric satisfies Axioms 1, 2, and 3.*

Proof Axiom 3 is the easiest to prove. Note that due to the tree structure of the graph, there is a unique path from the root to some node v_i in the tree. Let $\{v_0, v_1, ...v_{i-1}, v_i\}$ denote that path, where v_0 is the root of the tree. Let also $d_0, d_1, ..., d_{i-1}, d_i$ denote the degrees of the nodes in the path. Let P_0 denote the probability of the root at the stationary distribution. Then the probability of node i is $P_i = (1-\beta)^i \frac{1}{d_0+2\epsilon}\frac{1}{d_1+2\epsilon}\cdots\frac{1}{d_{i-1}+2\epsilon}P_0$. Now, if we swap the labels of two nodes u and v, the probabilities of the nodes in the tree do not change, since the structure of the tree did not change. The swap causes the trolling label to come closer to the root, hence the probability of the newly labeled trolling node will increase, since it is an exponentially decreasing function of the distance to the root.

The proof of Axioms 1 and 2 is more technical, so we only give the sketch of the proof. Consider the addition of a node v in the tree (trolling, or non trolling). The effect of the addition is to decrease the probabilities of some of the remaining nodes in the tree. If node v is added as a child of a non-leaf-node v_i at depth i, with d_i children, this will cause the degree of node v_i to increase, and hence the probabilities of the nodes in the subtree rooted at v_i to decrease (the rest of the probabilities are not affected). Consider the case that node v is a trolling node. We know that node v_i has at least one non-trolling descendant (the "dummy" node n). The probability of these non-trolling descendants decreases, and hence the overall probability of the non-trolling label decreases, which means that the probability of the trolling label increases, and thus *TVRank* increases. The case of adding a non-trolling node is treated symmetrically.

In the case that the new node v is added to a leaf node, the probability of the root of the tree decreases. This causes the probability of all existing nodes in the tree to decrease. If v is a trolling node, this means that the probability of the non-trolling nodes decreases, and hence the probability of the trolling nodes increases, which results in the increase of the *TVRank* metric. The case of adding a non-trolling node is treated symmetrically.

Intuitively, when a new node v is added to the tree it "claims" some probability mass that it takes away from the rest of the nodes. If it is a trolling node, the probability mass goes to the trolling label, hence increasing the *TVRank* value. If it is a non-trolling node, the probability mass goes to the non-trolling label, hence decreasing the *TVRank* value.

RWRs have been widely used to define the strength of the relationship between two nodes in a graph, and they are the building blocks of many metrics including PageRank, (see, Lawrence et al (1998)), topic-sensitive PageRank, (see, Haveliwala (2002)), and SimRank, (see, Jeh and Widom (2002)). In this paper, we use RWRs to capture the relationship of a node with its trolling descendants.

3.4 Post Vulnerability

The TV metrics we defined provide a numerical value that quantifies the degree of vulnerability of a node. Using this value we can determine which nodes are vulnerable or not. In addition to a high TV value, for a post to be characterized as vulnerable, we impose a hard constraint that the node should have at least K descendants, for a chosen parameter value K. That is, in order for a post p to be considered as vulnerable it must satisfy that $|V_p| \geq K$.

The rationale behind this constraint is that a post must generate enough traffic in order to be of interest to moderators. If the responses to a post are few, even if they are trollings, they are essentially a failed attempt at trolling since they did not generate any additional discussion.

We are now ready to define the notion of post vulnerability.

Definition 1 (Post Vulnerability) *Given a troll vulnerability metric TV, and parameters $K > 0$ and $\theta \geq 0$, we define a post p to be vulnerable to trolls if $|V_p| \geq K$, and $TV(p) \geq \theta$.*

The definition is dependent on the exact metric that we use, and the parameters K and θ 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 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.

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, positive/negative score, negative to postive score ratio, #controversial comments, #positive/negative/zero scored posts,
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, positive/negative score, negative to postive score ratio, #controversial comments, #positive/negative/zero scored posts,

Table 1: The features of our prediction model.

4 Prediction of Troll Vulnerability

Given a post, our goal is 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 Definition 1.

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 1.

Content Features. Content features include features related to the text of the post. Previous research (e.g., Cheng et al (2015)) 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.

Author Features. Author features aim 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 users to express their preference, or opinion, for a post, (e.g., whether they like it or not, or 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.

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. The intuition is that posts that try to change the topic of the conversation may attract an unpleasant reaction by the community.

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

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

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

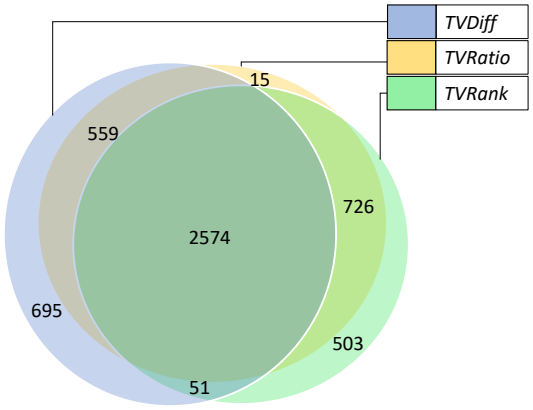


Fig. 2: Venn diagram that shows the agreement of the TV metrics in the dataset.

features can be thought of as describing the average user that participated in the previous posts.

Using these features, we build a classification model for predicting the troll-vulnerable posts. We can use any classification model for our task. In our experiments, we use a Logistic Regression classifier. We explore different classification models at the end of Section 5.

5 Experimental Results

In this section, we present results for the troll vulnerability metrics, and the troll vulnerability prediction task.

5.1 Experimental Setting

Dataset

Our dataset contains posts from the Reddit 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 then respond to these comments and conversation trees are formed whose roots are the submissions.

We consider 18 popular subreddits and retrieved 20 submissions from each of these subreddits, resulting in 555,332 comments.

Detection of Trollings

Although, identifying trollings is a problem orthogonal to our approach, to evaluate the performance of the troll vulnerability prediction task, we need a definition of trolling for our dataset. As we have already argued, the notion of trolling covers a wide range of behaviors, from innocent humor and misinformation to criminal activity. In our work, we focus on the anti-social part of trolls, i.e., we detect comments that contain offensive content.

The first step of our evaluation is to identify such offensive trolling posts in the Reddit comments. To this end, 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 platform[3]. The label is either 1 meaning an insulting comment, or 0 meaning a neutral (i.e., non-insulting) comment. The training set of the contest dataset contains 6,494 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 and the rest are neutral. The content of the comments is mostly in English with some occasional formatting.

We used a slightly modified version of a classifier used in the Kaggle contest Olariu (2013). The model consists of a neural network that takes as input the output of a few text classifiers[4]. The classifier takes as input the text content of the comments and assigns a score in [0,1] to each comment. In the contest dataset, the model achieved 76% accuracy, 84% precision, and 60% recall.

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 2,500 Reddit comments as trollings (i.e., insulting) or non-trollings (i.e., neutral). We selected these comments by splitting the $[0, 1]$ scores in five ranges and randomly selecting 500 comments from each range. These comments were assigned to three human judges along with the output of the classifier. The question was whether they agree with the decision of the classifier. The judges were given the following guidelines as to when to consider a post as being insulting:

- The author addresses insulting words or phrases to a person or a group of people.
- The author does not use insulting language, but clearly his goal is to insult a person or a group of people.
- The author expresses himself in a vulgar way.

We measured the pairwise agreement between the labelers using the Cohen's Kappa coefficient; they scored 0.64, 0.65 and 0.71, which means that they seem to agree in most cases. Considering the labels and the

[3] `https://www.kaggle.com/`
[4] `http://goo.gl/UL2VuE`

TVDiff				
$K \backslash \alpha$	*5*	*4*	*3*	2
2	6027	4325	3879	3233
3	4542	3351	2905	2259
5	2802	2070	1677	1237
8	1578	1144	863	593

TVRatio				
$K \backslash \theta$	*0.2*	*0.25*	*0.3*	*0.35*
2	5877	5142	3874	2869
3	4392	3657	2389	1895
5	2391	1823	1191	875
8	1129	802	516	379

TVRank				
$K \backslash \theta$	*0.2*	*0.25*	*0.3*	*0.35*
2	6271	4995	3858	3036
3	4786	3510	2373	1551
5	2321	1430	953	653
8	1098	671	434	281

Table 2: Number of vulnerable comments computed using the three TV metrics.

	TVDiff				*TVRatio*				*TVRank*			
Feature Group	A	P	R	AUC	A	P	R	AUC	A	P	R	AUC
Content (Con)	0.59	0.66	0.38	0.64	0.61	0.69	0.40	0.66	0.61	0.70	0.37	0.65
Author (Auth)	0.67	0.70	0.59	0.73	0.68	0.71	0.62	0.75	0.70	0.74	0.62	0.76
History (Hist)	0.65	0.70	0.51	0.69	0.68	0.74	0.56	0.72	0.69	0.75	0.57	0.73
Participants (Part)	0.68	0.74	0.57	0.73	0.72	0.77	0.61	0.76	0.71	0.76	0.61	0.76
Con + Auth	0.67	0.71	0.58	0.74	0.70	0.73	0.63	0.76	0.71	0.75	0.63	0.77
Con + Auth + Hist	0.69	0.72	0.62	0.76	0.72	0.75	0.65	0.79	0.74	0.77	0.66	0.80
Con + Auth + Hist + Part	0.72	0.76	0.64	0.79	0.74	0.78	0.67	0.82	0.77	0.81	0.68	0.83

Table 3: Prediction results for the various groups of features and combinations of the groups.

output of the classifier, we set the threshold for characterizing a comment as trolling at 0.5. In this setting, we achieved 82% accuracy, 75% precision and 80% recall in this set.

We applied this model to the full dataset. It characterized as trollings 9,541 comments which amounts to 1.7% of the total dataset.

5.2 Troll Vulnerability Metrics

In Section 3, we introduced three different TV metrics for quantifying the vulnerability of comments. In particular, for a post p to be considered vulnerable, its TV value should be larger than a threshold θ and p must be followed by at least K comments. Both θ and K act as filters that determine the number of comments that are vulnerable.

We set $K = 2$ as the default value, asking that a comment must be followed by at least 2 comments to be considered vulnerable. As argued, 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 tune θ and the value of the parameters of all three TV metrics so that we get around 3,800 comments characterized as vulnerable which amounts for about 2.5 trollings per vulnerable comment, on average, and use these as the default values.

In particular, for *TVRank*, we set $\theta = 0.3$ as the default value which results in 3,853 comments being characterized as vulnerable. We also set β, the restart probability, to 0.15 as in previous work, e.g., Lawrence et al (1998). We experimented also with different β values. This results in a small difference in the vulnerability rank of the nodes, however, it did not affect the performance of the classification model.

For *TVRatio*, we set $\theta = 0.3$ and $\beta = 0.5$. Using this setting, 3,874 comments are characterized as vulnerable

For *TVDiff*, we set θ equal to zero, so that in all cases at least one trolling descendant is present for characterizing a comment as vulnerable. We also set $\alpha = 3$ as the default value, which means that a trolling is considered three times more important than a trolling and $\beta = 0.25$. Using this setting results in 3,819 vulnerable comments.

We also experimented with different values of K for all TV metrics and with different θ values for *TVRatio* and *TVRank*. For *TVDiff*, we vary α instead of θ since this gives us better control on the relative importance of trolling comments. The results are shown in Table 2. Larger values of θ (resp. smaller values of α), reduce the sensitivity of the metrics, resulting in less comments being characterize as vulnerable. Similarly, larger values of K reduce the number of vulnerable comments.

Finally, in Figure 2 we compare the results of the three TV metrics using a Venn diagram to depict the extend at which the metrics agree in their characterization of comments as vulnerable. The Venn diagram includes all comments characterized as vulnerable by at least one TV metric, using the default values of the parameters. We can see that the majority of vulnerable comments are characterized as vulnerable by all three metrics.

5.3 Troll Vulnerability Prediction

We implemented classification models using the four group of features introduced in Section 4 to predict whether a comment is vulnerable or not.

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 and subsequently the number of the vulnerable comments is low as well. If we train the model without any preprocessing, the classifier will assign all the samples to the majority class, i.e., the non vulnerable comments. The minority class, i.e, the vulnerable comments, is so small that the classifier can simply ignore it.

In order to deal with this problem, we randomly undersample the majority class. We reduce the size of the non vulnerable class to be equal to the size of the vulnerable class. We repeat this procedure ten times and we report the average values of the evaluation metrics.

To understand the relative importance of each group, we first compare the performance of each group individually using Logistic Regression with 10-fold validation. We then incrementally combine the groups. Table 3 shows the classification results.

First, to put the numbers in perspective, a random assignment of labels to posts would result in a value 0.5 for all evaluation metrics. Therefore, all classifiers pass the sanity check of performing better than random predictions.

We then evaluate the different groups of features. We observe that in all metrics that 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: the author group, and the participants group (that includes information about the authors of the ancestor comments) perform the best. This indicates that the author of the comment as well as the authors of the preceding comments affect vulnerability more than the comments themselves. Combining features improves the prediction, with the classifier using features from all four groups being the best. In terms of TV metrics, *TVRank* and *TVRatio* perform almost the same and *TVDiff* performs slightly worse than the other two metrics.

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 4 shows the results of the single-feature classification for *TVRank*. The rest of the metrics perform similarly.

Feature Group	Feature	Accuracy	Precision	Recall	AUC
Content	#negative words	0.56	0.70	0.22	0.56
	#positive words	0.52	0.51	0.59	0.51
	whether is trolling	0.55	0.87	0.13	0.55
Author	#trollings	0.67	0.82	0.44	0.67
	sum positive score	0.57	0.68	0.27	0.57
	sum negative score	0.58	0.70	0.27	0.58
History	#zero scored comments	0.57	0.78	0.20	0.57
	#negative scored commments	0.66	0.80	0.43	0.66
	#trolling ancestors	0.58	0.80	0.22	0.58
Participants	#trollings	0.70	0.81	0.54	0.70
	#negative scored comments	0.65	0.70	0.53	0.65
	#zero scored comments	0.64	0.68	0.52	0.64

Table 4: Classification results using a single feature.

Most of the features perform poorly when used alone, only a few have recall measures larger than 50%. In terms of content, using strong opinion-words (positive or negative) 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 a strong signal. The same holds for the history of the ancestor comments and the authors of these comments.

Moreover, we checked the best features based on univariate statistical tests. In particular, we used the ANOVA F-test, which estimates the degree of linear dependency between two random variables. We scored the features using ANOVA F-value and we selected the highest scoring features. Table 5 shows the highest scoring features for the TV metrics. We can see the best features are almost the same in all metrics. This means that these features are important to the troll vulnerability prediction problem, regardless the metric we use.

Varying the Vulnerability Parameters

We also study the performance of the prediction model for different values of K, and when varying the parameters of TV metrics. In particular, we experiment with different θ values for the *TVRank* and *TVRatio* metrics, and different α values for the *TVDiff* metric.

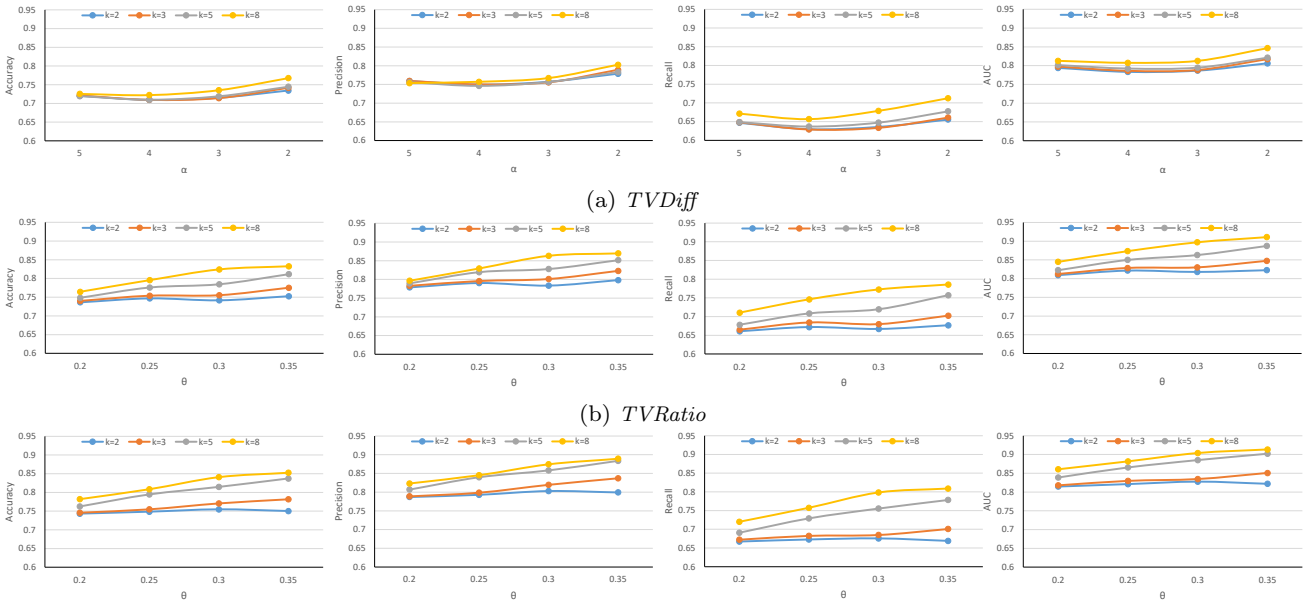
Figure 3 shows the results for the classifiers that include all features. Adjusting the parameters of the metrics (i.e., increasing θ and decreasing α) and increasing the constraint K increases the selectivity in the troll-vulnerability definition, resulting in fewer comments considered as vulnerable (Table 2). The performance of the classifier improves when the classes of vulnerable comments become more selective.

Other Datasets

In addition to our main dataset that consists of all available subreddits, we also tested our model on three

TVDiff		*TVRatio*		*TVRank*	
Feature Group	*Feature*	*Feature Group*	*Feature*	*Feature Group*	*Feature*
Participants	#trollings	Participants	#trollings	Participants	#trollings
Participants	#negative scored comments	Participants	#negative scored comments	Participants	sum positive score
History	#negative scored comments	History	#negative scored comments	Participants	#negative scored comments
Participants	sum negative score	Participants	sum negative score	History	#negative scored comments
Author	#positive scored comments	Participants	#zero scored comments	Participants	sum negative score
Participants	#zero scored comments	History	depth of the post	History	depth of the post
Author	#negative scored comments	Author	#trollings	Author	#positive scored comments
History	depth of the post	Author	#positive scored comments	Participants	#zero scored comments
Author	sum negative score	History	parent similarity	Author	#trollings
History	parent similarity	Author	#negative scored comments	Author	#negative scored comments

Table 5: The best features for each metric using statistical tests.



(a) *TVDiff*

(b) *TVRatio*

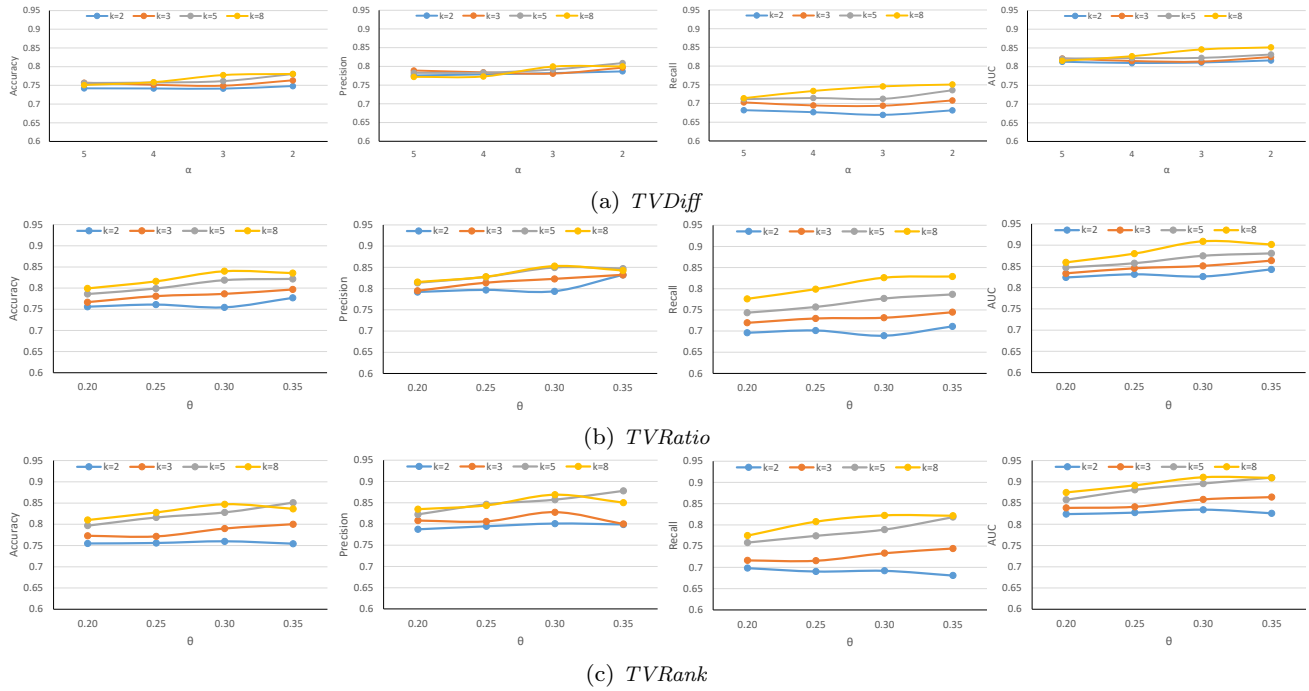
(c) *TVRank*

Fig. 3: Performance of the classification model with different values of K and of the parameters of the TV metrics.

smaller datasets, consisting of more homogeneous collections of posts. The first dataset is a subset of the main dataset that includes all comments posted in the subreddit "announcements". The subreddit "announcements" is the most popular subreddit in our dataset. It consists of 61,709 comments out of which 1,498 are trollings. For the second dataset, we used the same 18 subreddits as in our default dataset, but now we collect comments from the controversial submissions in these subreddits. A submission is controversial if it its comments have received the same number of positive and negative scores (i.e., upvotes and downvotes, respectively). This dataset has 270,144 comments out of which 5,805 are trollings. The third dataset contains comments from the "atheism" subreddit. This dataset contains 19,719 comments, out of which 438 (2.2%) are characterized as trollings. We selected this subreddit for the controversial nature of the topic, which makes it more likely to attract trolls. The evaluation results of the classifiers for the "announcements" dataset is shown in Figure 4, for the "controversial" dataset in Figure 5 and for the "atheism" dataset in Figure 6. The results are consistent over all datasets, demonstrating that thematic homogeneity of the comments has a small effect on the classifier performance. The slightly worse performance in the atheism dataset is due to the small size of the dataset. This is also the reason why we omitted the case $K = 8$ from our plots, since the number of vulnerable comments in this case was around 30 on average.

Varying Class Imbalance

We will now study how class imbalance affects the performance of the classifier. For this experiment we create different datasets by varying the ratio of the size of the negative (non-vulnerable) class (which is the majority)



(a) *TVDiff*

(b) *TVRatio*

(c) *TVRank*

Fig. 4: Performance of the classification model with different values of K and of the parameters of the TV metrics for the "announcements" subreddit.

to the positive (vulnerable) class (which is the minority). When training a classifier with unbalanced data, we assigned weights to the comments. 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. Therefore, if the classifier makes a wrong decision for a troll vulnerable comment, it is penalized more than making an error for a non-vulnerable comment. We also experimented with the Synthetic Minority Over-Sampling Technique (SMOTE) and Tomek links (Batista et al (2003)) method, but it performed slightly worse. For all datasets we perform 10-fold cross-validation to compute our results.

In Figure 7 we present results that show how the classifier performs as the ratio increases from 1 (perfectly balanced) to 25. Recall and AUC are not affected, however precision drops. Tables 6 and 7 show the confusion tables for the perfectly balanced case, and the full dataset respectively. For the table creation we have summed the numbers over the 10 folds. From the tables, it is clear that the reason for the precision drop is that many non vulnerable comments are incorrectly characterized as vulnerable (i.e., high number of false positives). Therefore, our classifiers err on the side of caution, putting more emphasis on capturing the vulnerable cases, as demonstrated by the high recall. One possible way to handle this in practice is to treat all comments characterized as vulnerable as warnings, and wait before taking more aggressive actions.

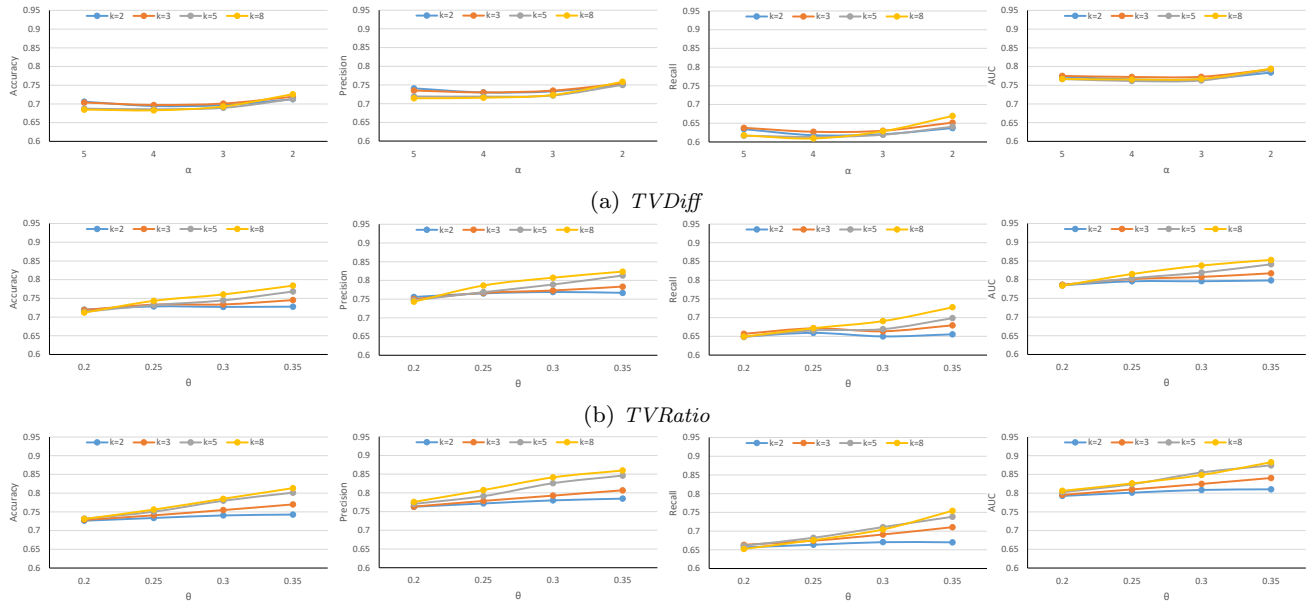
5.4 Trolling Escalation

We observed in our experiments that there is some correlation between trolling behavior and vulnerability. This is to some extend expected, as trollings invite reactions, causing trolling to escalate. We now study in detail the relationship between trollings and vulnerable comments.

A first question is whether all vulnerable comments are trollings. The percentage of the vulnerable comments that are trollings themselves in our dataset is 12.8%. This means that a comment does not have to be a trolling to attract trolls. In Figure 8a, 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 is much higher than the 0.6% probability of a non-trolling post to be vulnerable. However, clearly, 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



(a) *TVDiff*

(b) *TVRatio*

(c) *TVRank*

Fig. 5: Performance of the classification model with different values of K and of the parameters of the TV metrics for controversial submissions.

TVDiff			
		predicted	
		0	1
true	0	3.096	783
	1	1.419	2.460

TVRatio			
		predicted	
		0	1
true	0	3.191	683
	1	1.294	2.580

TVRank			
		predicted	
		0	1
true	0	3.206	652
	1	1.242	2.616

Table 6: Confusion matrices for balanced datasets.

TVDiff			
		predicted	
		0	1
true	0	429.808	121.645
	1	1.396	2.483

TVRatio			
		predicted	
		0	1
true	0	466.282	85.176
	1	1.240	2.634

TVRank			
		predicted	
		0	1
true	0	466.649	84.825
	1	1.235	2.623

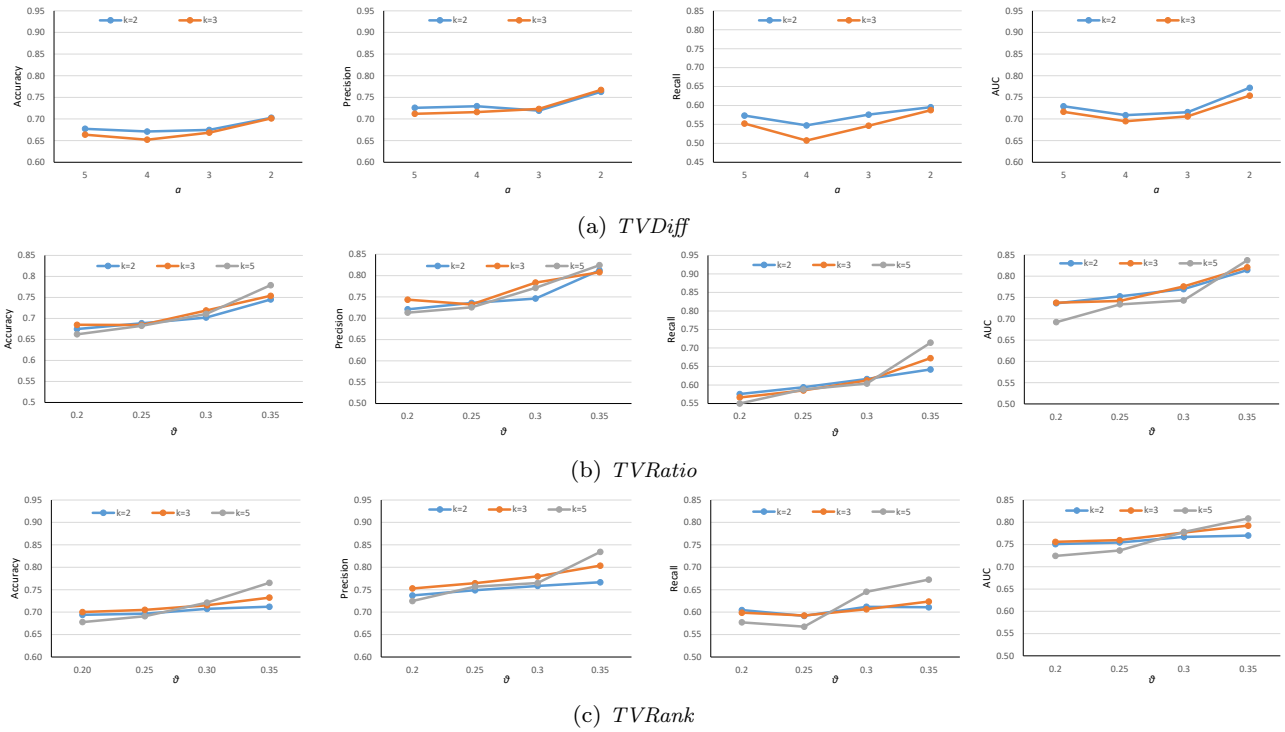
Table 7: Confusion matrices without balancing.

trolling comments and try to predict whether these trolling comments are vulnerable. Our classifier achieved AUC 70%, using the *TVRank* metric, 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 8b, 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 8c 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.

5.5 Additional Classifiers

Besides the logistic regression classifier, we also experimented with an SVM and a Random Forest classifier. The results for the three vulnerability metrics, with



(a) *TVDiff*

(b) *TVRatio*

(c) *TVRank*

Fig. 6: Performance of the classification model with different values of K and of the parameters of the TV metrics for the submissions in the subreddit "atheism".

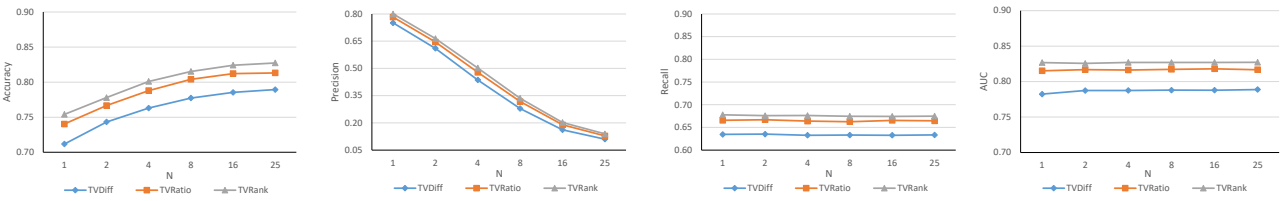


Fig. 7: Performance of the classification model as a function of the negative-to-positive class ratio.

the default parameters, are reported in Table 8. Performance is consistent across all three classifiers, with Random Forest slightly outperforming the other two. It appears that the specific classification model has only a small effect on the troll vulnerability prediction task.

6 Conclusions and Future Work

Understanding and detecting trolling behavior in social networks has attracted considerable attention. In this paper, 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 post is to trolls. We provide three measures of troll vulnerability with respect to both the volume and the proximity of the trollings associated with each post. We then 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 allows handling trolls proactively, by preventing them to appear instead of just detecting them, when they appear. 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 predicting troll vulnerability in other social networks beyond Reddit. In addition, our work creates interesting directions for future work towards studying vulnerability at different levels than that of a post. For example, instead of identifying posts that are likely to attract trolls, an interesting problem is identifying vulnerable *users*, e.g., users whose posts are

	TVDiff				TVRatio				TVRank			
Classifier	A	P	R	AUC	A	P	R	AUC	A	P	R	AUC
SVM	0.71	0.75	0.65	0.77	0.74	0.77	0.68	0.80	0.75	0.78	0.69	0.81
Logistic Regression	0.72	0.76	0.64	0.79	0.74	0.78	0.67	0.82	0.77	0.81	0.68	0.83
Random Forest	0.73	0.75	0.70	0.83	0.75	0.78	0.71	0.85	0.76	0.78	0.72	0.85

Table 8: Performance of additional classifiers.

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.

Fig. 8: Examples of trolling behavior.

vulnerable to trolls, or, vulnerable *topics*, e.g., topics that are likely to attract trolls.

References

Adler BT, De Alfaro L, Mola-Velasco SM, Rosso P, West AG (2011) Wikipedia vandalism detection: Combining natural language, metadata, and reputation features. In: Computational linguistics and intelligent text processing, Springer, pp 277–288

Atwood J (2011) Suspension, ban or hellban? `http://goo.gl/TxCGi7`

Batista GE, Bazzan AL, Monard MC (2003) Balancing training data for automated annotation of keywords: a case study. In: WOB, pp 10–18

Blackburn J, Kwak H (2014) Stfu noob!: predicting crowdsourced decisions on toxic behavior in online games. In: Proceedings of the 23rd international conference on World wide web, ACM, pp 877–888

Buckels EE, Trapnell PD, Paulhus DL (2014) Trolls just want to have fun. Personality and individual Differences 67:97–102

Cambria E, Chandra P, Sharma A, Hussain A (2010) Do not feel the trolls. In: Proceedings of the 3rd International Workshop on Social Data on the Web, ISWC

Cheng J, Danescu-Niculescu-Mizil C, Leskovec J (2015) Antisocial behavior in online discussion communities. In: Proceedings of ICWSM

Chin SC, Street WN, Srinivasan P, Eichmann D (2010) Detecting wikipedia vandalism with active learning and statistical language models. In: Proceedings of the 4th Workshop on Information Credibility, WICOW '10, pp 3–10

de-la-Peña-Sordo J, Pastor-López I, Ugarte-Pedrero X, Santos I, Bringas PG (2014) Anomalous user comment detection in social news websites. In: International Joint Conference SOCO'14-CISIS'14-ICEUTE'14 - Bilbao, Spain, June 25th-27th, 2014, Proceedings, pp 517–526

Donath JS (1999) Identity and deception in the virtual community. Communities in cyberspace 1996:29–59

Hardaker C (2010) Trolling in asynchronous computer-mediated communication: From user discussions to academic definitions. Journal of Politeness Research 6:215–242

Haveliwala TH (2002) Topic-sensitive pagerank. In: Proceedings of the Eleventh International World Wide Web Conference, WWW 2002, May 7-11, 2002, Honolulu, Hawaii, pp 517–526

Jeh G, Widom J (2002) 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

Jeong S (2014) Does twitter have a secret weapon for silencing trolls? `http://goo.gl/HcuL20`

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

Kumar S, Spezzano F, Subrahmanian VS (2015) 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

Kunegis J, Lommatzsch A, Bauckhage C (2009) The slashdot zoo: Mining a social network with negative edges. In: Proceedings of the 18th International Conference on World Wide Web, WWW '09, pp 741–750

Lamba H, Malik MM, Pfeffer J (2015) 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

Lawrence P, Sergey B, Motwani R, Winograd T (1998) The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University

Olariu A (2013) Repo for the insults detection challenge on kaggle.com. `https://github.com/andreiolariu/kaggle-insults/`

Ortega FJ, Troyano JA, Cruz FL, Vallejo CG, Enríquez F (2012) Propagation of trust and distrust for the detection of trolls in a social network. Computer Networks 56(12):2884 – 2895

Potthast M, Stein B, Gerling R (2008) Automatic vandalism detection in wikipedia. In: Advances in Information Retrieval , 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008, pp 663–668

Sood SO, Churchill EF, Antin J (2012) Automatic identification of personal insults on social news sites. J Am Soc Inf Sci Technol 63(2):270–285

Suler J (2004) The online disinhibition effect. Cyberpsychology & behavior 7(3):321–326

Tsantarliotis P, Pitoura E, Tsaparas P (2016) Troll vulnerability in online social networks. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016, San Francisco, CA, USA, August 18-21, 2016, pp 1394–1396

Wu Z, Aggarwal CC, Sun J (2016) The troll-trust model for ranking in signed networks. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, ACM, pp 447–456