# Accurately Detecting Trolls in Slashdot Zoo via Decluttering

Srijan Kumar, Francesca Spezzano, and V.S. Subrahmanian
Dept. of Computer Science & UMIACS
University of Maryland
College Park, MD 20742
Email: {srijan,vs}@cs.umd.edu, spezzano@umiacs.umd.edu

***Abstract*—Online social networks like Slashdot bring valuable information to millions of users - but their accuracy is based on the integrity of their user base. Unfortunately, there are many "trolls" on Slashdot who post misinformation and compromise system integrity. In this paper, we develop a general algorithm called TIA (short for Troll Identification Algorithm) to classify users of an online "signed" social network as malicious (e.g. trolls on Slashdot) or benign (i.e. normal honest users). Though applicable to many signed social networks, TIA has been tested on troll detection on Slashdot Zoo under a wide variety of parameter settings. Its running time is faster than many past algorithms and it is significantly more accurate than existing methods.**

## I. Introduction

A signed social network (SSN) is one in which a user $u$ can have a positive or negative relationship with another user $v$. There are many signed social networks in the real world. Even in small human populations (e.g. faculty in a computer science department), there will be individuals who like some individuals but dislike others. In some online networks like Slashdot, users may explicitly mark some users as friends and others as foes. On Wikipedia, an individual may "roll back" or "reverse" essential changes made by one person, while supporting and augmenting changes by another. An implicit negative opinion is conveyed in the first case and a positive opinion in the latter case. On Twitter, a user $u$ may frequently support what a user $v_1$ says while opposing or contradicting what another user $v_2$ says.

In this paper, we start with a "Signed Social Network (SSN)" $G = (V, E, W)$ where $V$ is a set of users, $E \subseteq V \times V$ is a set of edges, and $W : E \rightarrow [-1, +1]$ assigns a real valued weight from -1 to +1 indicating how positive or negative one user is to another. When $W(u, v) = 1$, $u$ considers $v$ to be a 100% friend, when $W(u, v) = -1$, he considers $v$ to be a 100% foe. While SSNs are explicitly present in Wikipedia and Slashdot Zoo, they can also be extracted via NLP techniques from Twitter. *We restrict this paper to network analysis and assume a signed network is given as input. Clearly, methods to extract signed networks from networks like Twitter is an important task - but is not addressed here.*

A *malicious user* in an SSN $G = (V, E, W)$ is a specially designated individual. On Slashdot, trolls are malicious users who post or spread misleading, offensive or nonsensical information on the network. Likewise Wikipedia describes a vandal as "an editor who intentionally makes unconstructive edits to Wikipedia's pages." Vandals may insert irrelevant information, nonsense, obscenity or crude humor to pages or entirely blank or delete pages. A *benign user* is one who is not malicious.

The goal of this paper is to present a single framework within which to identify malicious users. A major challenge in effectively identifying trolls is the fact that malicious users take a number of carefully designed steps that enable them to evade detection. We propose *5 graph decluttering operations* that help simplify a large, complicated SSN $G$ into a smaller and simpler SSN $G'$. Intuitively, the idea is to remove some "hay" from the "haystack" we are searching in order to present our TIA algorithm with a simpler signed graph, stripped of irrelevant edges, that enables TIA to operate more effectively. We tested all subsets of these 5 decluttering operations and found the combination that yields the best results.

Most CS work on signed networks have involved study of Slashdot, Epinions and Wikipedia administrator election networks. We focus only on Slashdot, as there is no ground truth present for malicious vs benign users on Epinions, and because NLP is needed to analyze Wikipedia.

The paper is organized as follows. Section II briefly defines Signed Social Networks (SSNs). Section III briefly looks at how centrality measures on SSNs have been used in the past in order to identify trolls on Slashdot Zoo and explains both these centrality measures as well as other centrality measures. Section IV presents our 5 decluttering operations to simplify a complex SSN into a smaller and simpler SSN - Section V presents the TIA algorithm that uses a subset of these decluttering operations to simplify an SSN. We present the Slashdot Zoo data set we used in Section VI. Section VII reports on the results of experiments to assess how well TIA works. We examine how the combination of decluttering and a Signed Eigenvector Centrality (SEC) measure together generate the best accuracy results on the Slashdot data set. We show that under appropriate settings,TIA has: (i) over 3 times the precision of the best existing algorithm to find trolls in Slashdot [1], (ii) retrieves over twice the number of trolls that [1] does, and (iii) does all this while running 25-50 times faster.

## II. Signed Social Networks

A *Signed Social Network (SSN)* is a directed, weighted graph $G = (V, E, W)$ where $V$ is a set of users, $E \subseteq V \times V$ is a set of edges, and $W : E \rightarrow [-1, +1]$ is a mapping. $W(u, v)$ can be thought of as assigning a "likes" or a "friendship" score describing how much user $u$ likes a user $v$.

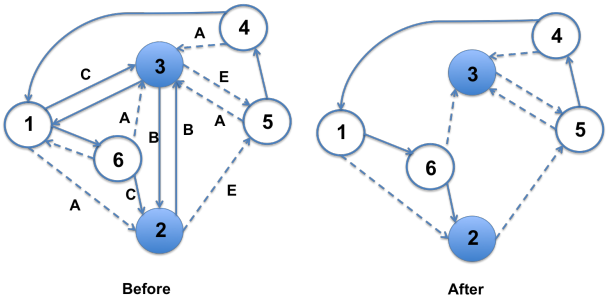



Fig. 1: (Left) Example of signed social network. Filled nodes are trolls, non-filled nodes are benign users. Solid (resp. dashed) edges mean positive (resp. negative) endorsements. Edges labels are the attack models used by trolls. (Right) resulting SSN after decluttering operations (a) and (d) using SEC.

**Slashdot:** On Slashdot, a user $u$ can explicitly mark a user $v$ as a foe (i.e. $W(u,v) = -1$) or as a friend (i.e. $W(u,v) = +1$).

**Wikipedia:** On Wikipedia, we can define $W(u,v)$ using NLP in many ways. One way is to set

$$W_{wi}(u,v) = \frac{supp(u,v) - rev(u,v)}{edits(v)}$$

where $edits(v)$ is the set of all edits made by $v$ during some fixed time window, $supp(u,v)$ is the number of edited documents in $edits(v)$ that were subsequently edited but not substantively reverted by $v$, and $rev(u,v)$ is the number of edits in $edits(v)$ that were subsequently reversed by $u$. A more sophisticated way to define $W(u,v)$ is given in [2].

**YouTube:** One way to derive an SSN from YouTube is to set:

$$W_{yt}(u,v) = \frac{thumbs\_up(u,v) - thumbs\_down(u,v)}{|posts(v)|}$$

where $posts(v)$ is the set of all videos posted (during some fixed time frame) by $v$ that were marked positively or negatively by $u$, $thumbs\_up(u,v)$ is the number of videos in $posts(v)$ marked with a "thumbs up" and "thumbs_down" is the number of videos in $posts(v)$ marked with a thumbs down.

**Twitter/Facebook:** Given an edge $(u,v)$ denoting that $u$ follows $v$ on Twitter (or friends $v$ on FB), we set:

$$W_{tw}(u,v) = \frac{pos(u,v) - neg(u,v)}{|tweets(v)|}$$

where $tweets(v)$ is the set of tweets posted by $v$ during a given time frame, $pos(u,v)$ is the subset of those tweets that are either retweeted by $u$ or essentially rephrased by $u$ afterwards, and $neg(u,v)$ is the subset of $tweets(v)$ that are the sets of tweets in $pos(u,v)$ whose content is contradicted by a subsequent post by $v$. Note that NLP techniques that use sentiment analysis [3], [4] can be used to estimate the sets $pos(u,v)$ and $neg(u,v)$.

**Stack Overflow:** Users of Stack Overflow can mark comments provided by other users as good $(+1)$ or not good $(-1)$.

A SSN can be represented as an adjacency matrix $A$ with values $A_{uv} = W(u,v) \in [-1,+1]$. We use $H$ to denote the transition matrix obtained from $A$ by dividing each non-zero element $A_{uv}$ by $\sum_w |A_{uw}|$.

*Example 1:* Figure 1 (left) shows a small toy social network which has two trolls — nodes 2 and 3 in it. □

## III. Related Work

Signed networks have been studied both in social science ([5]) and computer science ([6], [7]). Much work on reputation systems propagates trustworthiness of users on social networks – but this work has been done on unsigned social networks ([8], [9]). [10] studied the propagation of trust and distrust in social network for the purpose of edge sign prediction. In this section, we will look at the existing centrality measures, the axioms for good centrality measures and attack models.

### A. Centrality Measures for SSNs

Given a SSN $G = (V, E, W)$, a node centrality measure is a function $R : V \to \mathbb{R}$ that assigns a score to each user — the higher the score, the more important the user. Various authors have used different centrality measures to identify malicious users (e.g. [1] uses Negative Rank to identify trolls in Slashdot). We now review some of these measures.

**Freaks** The freak score of $u$ is the number of incoming negative edges [1]. For weighted networks, the *Freak Centrality* of $u$ is the sum of the weights of incoming negative edges.

$$freaks(u) = \sum_{v \in V | W(v,u) < 0} W(v,u)$$

According to this definition, node 3 in Figure 1 has freak centrality 3 because it has 3 incoming negative edges, while nodes 4 and 6 have freak centrality 0 because they have no incoming negative edges.

**Fans Minus Freaks (FMF)** The centrality of $u$ is the number of positive incoming edges (*fans*) minus the number of negative incoming edges (*freaks*) [1]. For weighted networks, we define *FMF* centrality as the total positive incoming weight minus the the total negative incoming weight.

$$FMF(u) = \sum_{v \in V | W(v,u) > 0} |W(v,u)| - \sum_{v \in V | W(v,u) < 0} |W(v,u)|$$

Node 3 in Figure 1 has FMF centrality -1 because it has 3 incoming negative edges and 2 incoming positive edge. Node 5 has an FMF centrality of -2. A similar measure, called *Prestige*, has been proposed in [11] and is obtained by dividing the FMF centrality by the sum of the absolute values of the incoming weights for each node.

**PageRank (PR)** PageRank ([12]) is defined for directed graphs with non-negative edge weights. It was originally developed for indexing web pages, and represents the likelihood that a person following links will arrive at a particular page. The PageRank of a node $u$ is defined as

$$PR(u) = \frac{1-\delta}{|V|} + \delta \sum_{v \in pred(u)} \frac{PR(v)}{|succ(v)|}$$

Here, $\delta$ is a "damping factor" (usually 0.85) which captures the probability that a user arrives at a web page by following links (as opposed to landing on the page via some other process). $pred(u)$ is the set of all vertices $v$ such that $(v,u) \in E$ and $succ(v)$ is the set of all vertices $v'$ such that $(v,v') \in E$. Node 3 in Figure 1 has a PageRank of 0.29, while node 5 has a PageRank of 0.18. A *Modified PageRank* (M-PR) has been proposed in [13] to take into account both positive and negative links. In particular, they apply PageRank separately on $A^+$ (sub-network with positive links) obtaining $PR^+$, and

on $A^-$ (sub-network with negative links) obtaining $PR^-$. The final rank vector M-PR is computed as M-PR $= PR^+ - PR^-$. Nodes 3 and 5 in Figure 1 have M-PR scores of $-0.09$ and $-0.41$, respectively.

**Signed Spectral Ranking (SSR)** Signed Spectral Ranking (SSR) [1] improves upon PageRank by taking edge signs into account. It is computed by taking the dominant left eigenvector of the signed matrix

$$G_S = \delta \cdot H_A + \frac{(1-\delta)}{|V|} \cdot J_{|V|\times|V|}$$

Positive edges correspond to endorsements, while negative edges to criticisms. Node 3 in Figure 1 has an SSR of $0.74$, while node 5 has an SSR of $-0.50$.

**Negative Ranking (NR)** An empirical evaluation of SSR and PR using Slashdot data was done in [1] who show that SSR and PR values were almost equivalent for benign users, but PR value for trolls was much more than their SSR value. They suggest a Negative Rank measure computed by subtracting PR from SSR, i.e. $NR(u) = SSR(u) - \beta \cdot PR(u)$, where $\beta$ is a parameter determining the influence of PageRank on the ranking. As [1] obtained their best results when $\beta = 1$, we use $\beta = 1$. Node 3 in Figure 1 has a NR of $0.45$, while node 5 has a NR of $-0.68$.

**Signed Eigenvector Centrality (SEC)** Eigenvector centrality (EC) was proposed by Bonacich [14] for networks with non-negative edge weights given by the dominant eigenvector of the adjacency matrix. As eigenvectors can be computed for any matrix, [5] suggests that this measure can also be computed for (weighted) signed networks. Thus, the signed eigenvector centrality of a vertex $v$ can be computed from the vector $x$ that satisfies the equation $Ax = \lambda x$, where $\lambda$ is the greatest eigenvalue. According to this definition, node 3 in Figure 1 has SEC of $0.68$, while node 5 has an SEC of $-0.55$.

**Modified HITS (M-HITS)** The HITS link analysis algorithm to rate Web pages [15] has been adapted for SSNs in [13] by iteratively computing the hub and authority scores separately on $A^+$ and $A^-$, using the equations:

$$\begin{cases} h^+(u) = \sum_{v \in succ^+(u)} a^+(v);\ a^+(u) = \sum_{v \in pred^+(u)} h^+(v) \\ h^-(u) = \sum_{v \in succ^-(u)} a^-(v);\ a^-(u) = \sum_{v \in pred^-(u)} h^-(v) \end{cases}$$

and by assigning, after convergence, the score $a(u) = a^+(u) - a^-(u)$ to each node $u$. $pred^+(u)$ (resp. $pred^-(u)$) denotes the set of nodes $v$ in $pred(u)$ s.t. $W(v,u) > 0$ (resp. $W(v,u) < 0$). Similarly for $succ^+(u)$ and $succ^-(u)$. For M-HITS, node 3 in Figure 1 has score of -0.92 and node 5 has score -$9\times10^{-9}$.

**Bias and Deserve (BAD)** In [16], a node $u$'s bias (BIAS) reflects the expected weight of an outgoing connection, while its deserve (DES) reflects the expected weight of an incoming connection from an unbiased node. Similarly to HITS, BIAS and DES are iteratively computed as:

$$\begin{cases} DES^{t+1}(u) = \frac{1}{|prec(u)|} \sum_{v \in prec(u)} [W(v,u)(1 - X^t(v,u))] \\ BIAS^{t+1}(u) = \frac{1}{2|succ(u)|} \sum_{v \in succ(u)} [W(u,v) - DES^t(v)] \end{cases}$$

where $X^t(v,u) = \max(0, BIAS^t(v)W(v,u))$. Finally, the scores of the nodes in the SSN are taken as the vector $DES$. In Figure 1, nodes 3 and 5 have BAD scores -0.16 and -1.0 respectively.

*Example 2:* Consider the SSN in Figure 1 (left) — nodes 2 and 3 are trolls. The following table (left part) shows how nodes are ranked according to the centrality measures. Here, the row "Freaks" should be read as: In the original network, the Freak centrality of node 3 is lowest, followed by 5, followed by 1 and 2 (with same freaks centrality) and 4,6 (with same freaks centrality). The rest of this row can be read similarly for the decluttered network (discussed later).

| Measure | Original network | | | | | | Decluttering with {a,b,d} | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Lowest | | | Highest | | | Lowest | | | Highest | | |
| Freaks | **3** | 5 | 1,2 | | 4,6 | | **3** | 5 | 1,2 | | 4,6 | |
| FMF | 5 | **3** | 1,2,4,6 | | | | 5 | **3** | 2,4,6 | | | 1 |
| Prestige | 5 | **3** | 1,2 | | 4,6 | | 5 | **3** | 2 | 1,4,6 | | |
| M-PR | 5 | **3** | 4 | 6 | 1 | 2 | 5 | **3** | 4 | 6 | 1 | 2 |
| SSR | 5 | 4 | 6 | 1 | 2 | 3 | **3** | **2** | 6 | 1 | 4 | 5 |
| NR | 5 | 4 | 1 | 6 | 2 | 3 | **3** | **2** | 6 | 1 | 4 | 5 |
| SEC | 5 | 4 | 6 | 1 | 2 | 3 | **3** | **2** | 6 | 1 | 4 | 5 |
| M-HITS | **3** | 5 | 4 | 6 | 1 | 2 | **3** | 5 | 4 | 6 | 1,2 | |
| BAD | 5 | **3** | 2 | 1 | 4,6 | | 5 | **3** | 2 | 1 | 4,6 | |

If we take the two lowest (as there are two trolls) scored nodes to be trolls, then Freaks, FMF, M-PR, Prestige and BAD identify one troll (node 3) correctly and incorrectly identify node 5, among the lowest two nodes. □

It is easy to construct cases where Freak Centrality cannot identify a node as a troll. For instance, consider a network with 1000 nodes including a node A which has 995 positive incoming edges and 4 negative edges. All other nodes have 5 incoming positive edges and either 0 or 1 incoming negative edges. Clearly, A would be designated as having the highest freaks centrality — but it is not likely to be a troll because the 995 positive incoming edges far outweigh the 4 negative incoming edges.

### *B. Requirements of a good scoring measure*

[5] proposes a set of axioms for SSNs that a good measure of centrality in SSNs must satisfy under the assumption that a set of nodes is benign (and the others are malicious).[1]

**Axiom 1**: *A positive edge from a benign node to a node $v$ should increase $v$'s centrality.* Intuitively, a positive edge from a benign node to another node means that the benign node also thinks the other node is benign — otherwise there is no reason for the benign node to implicitly endorse the other node.

**Axiom 2**: *A negative edge from a benign node to a node $v$ should decrease $v$'s centrality.* As in the previous axiom, benign nodes have an incentive to identify malicious nodes in order to preserve the integrity of the social network as a whole. As a consequence, when a benign node says a node is malicious, there is some chance that it actually is malicious.

**Axiom 3**: *A positive edge from a malicious node to a node $v$ should decrease $v$'s centrality.* The rationale behind this axiom is that malicious nodes have a strong incentive to endorse other malicious nodes so that an "army" of malicious nodes can collectively perform some task(s).

**Axiom 4**: *A negative edge from a malicious node to a node $v$ should increase $v$'s centrality.* As in the previous case, malicious nodes have an incentive to downgrade the centrality

[1] Our TIA algorithm will make such assignments initially and then iteratively modify these assignments in each iteration of a loop.

| *Ranking* | *Axiom 1* | *Axiom 2* | *Axiom 3* | *Axiom 4* |
|---|---|---|---|---|
| Freaks | No | Yes | No | No |
| FMF | Yes | Yes | No | No |
| Prestige | Yes | Yes | No | No |
| PR | Yes | No | No | Yes |
| M-PR | Yes | Yes | No | No |
| SSR | Cond-Yes | Cond-Yes | Cond-Yes | Cond-Yes |
| NR | Can't Say | Cond-Yes | Cond-Yes | Can't Say |
| SEC | Cond-Yes | Cond-Yes | Cond-Yes | Cond-Yes |
| M-HITS | Yes | Yes | No | No |
| BAD | Yes | Yes | No | No |

TABLE I: Table showing which axioms are satisfied by the centrality measures. Yes, No and Cond-yes mean that the axioms are satisfied, not satisfied and conditionally satisfied, respectively. Can't say means that nothing can be said in particular.

| *Measures* | *A* | *B* | *C* | *D* | *E* |
|---|---|---|---|---|---|
| Freaks | Yes | No | No | No | No |
| FMF | Yes | No | No | No | No |
| Prestige | Yes | No | No | No | No |
| PR | No | No | No | No | Yes |
| M-PR | Yes | No | No | No | No |
| SSR | Cond-Yes | Cond-Yes | Cond-No | Cond-No | Cond-Yes |
| NR | Cond-Yes | Cond-Yes | Cond-No | Cond-No | Can't say |
| SEC | Cond-Yes | Cond-Yes | Cond-No | Cond-No | Cond-Yes |
| M-HITS | Yes | No | No | No | No |
| BAD | Yes | No | No | No | No |

TABLE II: Table showing which centrality measure successfully prevents malicious users from using the attack models *A-E*. Yes and Cond-Yes means the attack is always and conditionally prevented, respectively, while No and Cond-No mean the opposite. Can't say means nothing can be said in general.

of benign nodes so that, in comparison, other malicious nodes get a high score. As a consequence, when a node is disliked by a malicious node, it probably means that the malicious node is trying to "bad mouth" a benign node.

Table I shows which centrality measures satisfy these axioms (in one iteration of computing the measure). A *Cond-Yes* means that under some reasonable conditions, the centrality measure satisfies the axiom in question. For instance, Freaks and FMF centrality measures do not distinguish between the origin of an edge and hence they do not satisfy some axioms — in both cases, each negative incoming edge decreases the centrality of a vertex irrespective of the origin of the edge (same thing happens with Prestige, M-PR, M-HITS and BAD). PageRank does not take the sign of an edge into account and so a negative incoming edge may increase a user's centrality. Signed Spectral Rank and Signed Eigenvector Centrality both satisfy all the four axioms as long as they assign positive and negative centrality scores to benign and malicious users, respectively. If this condition is not satisfied, then these measures violate the axioms. These two centrality measures take into consideration the centrality value of the edge generator and the sign of the edge. As Negative Rank depends on the relative increase in the values of PR and SSR, there can be some indecisive cases. Consider Axiom 1 – if the conditions stated above hold, then both Page Rank and Signed Spectral Rank would increase. The decision for Negative Rank depends on the relative increase in the centrality values and therefore, nothing can be said in general. Similarly for Axiom 4.

### C. Attack Models

In the real world, benign users can be tricked into endorsing malicious users (via positive outgoing edges). For example, benign users may endorse someone because they were endorsed by that user, not necessarily because they like that user. Malicious users may endorse a benign user (i.e. have positive edges to benign users) in the hope that the benign user reciprocates, which would increase their centrality. In such cases, past methods to identify malicious nodes in SSNs can lead to error as shown in Example 2 and the following discussion. Specific attacks described in [17] include:

*(A) Individual malicious peers.* Malicious users always present bad behavior, and hence receive negative links from good users. These are relatively stupid malicious users who should not be difficult to detect, say by using Freaks centrality.

*(B) Malicious collectives.* Malicious users endorse other malicious users. In this case, a malicious user's score may increase due to the presence of a bunch of positive incoming links.

*(C) Camouflage behind good transactions.* Malicious users can cheat some benign users to vote positively for them. This happens, for instance, when malicious users endorse a benign user who, out of courtesy, endorses them back.

*(D) Malicious spies.* There are two kinds of malicious users: some of them act as in threat models B and C, while the others (called *spies*) make benign users to vote positively for them, and assign positive value only to bad nodes.

*(E) Camouflage behind judgements.* The strategy in this case is to assign negative value to good users. Then, this can cause the decrease of rank for good peers, and, consequently, the increase of malicious user's rank.

A good scoring measure should robustly counter all these 5 attack models. Countering an attack model means preventing increase in a malicious user's centrality and decrease in a benign user's centrality. This way malicious nodes following these attack models would not be able to "game the system" and their scores would still be low. Figure 1 (left) shows how trolls in our toy example use the attack models (edge labels show the attack model used).

Table II depicts which centrality measures are resilient to which attacks. A Yes means the measure is able to counter the attack model. Since Freaks and FMF do not take the score of the origin of an edge into account, they both only disable attack model A (so is the case with Prestige, M-PR, M-HITS and BAD). PageRank ignores the sign of the edge, so it only counters attack model E as the centrality of a benign user would increase even when there are incoming negative edges. Signed Spectral Rank and Signed Eigenvector Centrality conditionally deflect ("Cond-Yes") attack models A, B, and E. The condition is that the centrality measure should assign a positive centrality score to benign users and a negative centrality score to malicious users. If the condition is satisfied, then the attack model is countered, otherwise it could be successful. For instance, if the centrality measure assigns a negative score to a malicious user, then malicious users using attack model B would fail as it would increase the centrality of a positive edge recipient. "Cond-No" means a conditional No. If the above condition is satisfied, then the attack model is successful, otherwise the attack model may fail. For instance, if the condition is satisfied and the centrality

measure assigns a positive score to a benign user, then the centrality of malicious users following attack model C would increase and the attack would succeed. Negative Rank deflects an the attack model only if the edge increases the PageRank of the user and decreases her Signed Spectral Rank. If both increase or decrease, then nothing can be said in general.

As no centrality measure successfully handles all the attack models described above, there is a critical need for a mechanism to prevent malicious users from increasing either their centrality or another malicious user's centrality and prevent the decrease of benign users' centrality.

## IV. Decluttering Operations

In order to handle the five attack models described above, we present 5 graph decluttering operations that help reduce the impact of such attacks. Our TIA algorithm (presented after the five operations) iteratively uses the score provided by a centrality measure to identify both benign and malicious users, and removes some edges between benign users, so that, at the end of the process, user scores enable us to better recognize malicious users. Given a centrality measure $\mathcal{C}$ and a threshold value $\tau$, benign users are those nodes $v$ in $V$ s.t. $\mathcal{C}(v) \geq \tau$ — everyone else is considered to be malicious. TIA takes as input, any centrality measure for SSNs, a corresponding $\tau$, as well as any set of decluttering operations.

Let $V$ be a set of nodes. We use $\mathcal{G}$ to denote the set of all possible SSNs $G = (V, E, W)$ over $V$. A decluttering operation is defined as follows.

*Definition 1 (Decluttering Operation):* A *decluttering operation* is an associative function $\rho : \mathcal{G} \to \mathcal{G}$ that transforms graphs into graphs such that for all $G = (V, E, W)$, if $\rho(G) = G' = (V', E', W')$, then $V = V'$, $E' \subseteq E$, and for all $e' \in E'$, $W'(e') = W(e')$.

Two decluttering operations $\rho_1, \rho_2$ can be composed. $\rho_2 \circ \rho_1(G)$ is defined as $\rho_2(\rho_1(G))$. We consider the following decluttering operations (see also Figure 2).

*DOP(a): Remove all positive pairs of edges between benign nodes.* Suppose $u, v$ are both benign and endorse each other. In this case, we do not know whether they are really endorsing each other or whether one is blindly reciprocating endorsements. For instance, if $u$ added $v$ as a friend in Slashdot, $v$ might reciprocate even if he has never read $v$'s posts. Moreover, removing positive pairs of edges between benign nodes helps to potentially alleviate the effects of attacks B,C, and D as it also removes positive loops between pairs of malicious nodes that may have initially been misclassified as being benign, thus reducing the scores of these malicious nodes. Consider a pair of malicious users who are mistakingly classified as benign nodes and both follow attack model B. On Slashdot, this means both add each other as friends. Removing the positive edge pair between them would prevent further increase in their centrality scores. Now, consider a single malicious user following attack model C. On Slashdot, one way to trick a non-malicious user into adding a troll as a friend is to add her as a friend first, which would prompt her to add the troll as a friend too. This would create a positive edge pair between them. By removing these edges, attack model C is countered. Since attack model D is a combination of models B and C, and both are countered by this operation, so is model D.

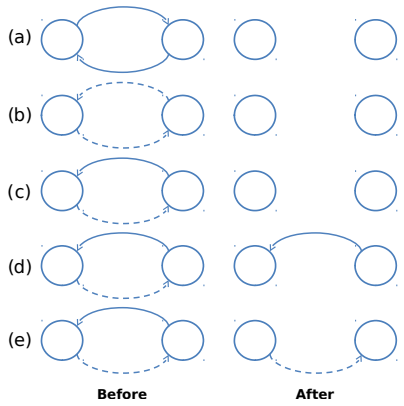



Fig. 2: Decluttering operations in TIA. All nodes are marked benign. Bold (dashed) edges denote a positive (negative) relationship.

*DOP(b): Remove all negative pairs of edges between benign nodes.* The removal of a negative pair of edges between two benign nodes is motivated by the fact that it might not actually mean that a user hates the other user's content. In Slashdot, a user $u$ might add $v$ as a foe just because $v$ adds $u$ as a foe. Removing negative edge pairs also counters attack models A and E. If a malicious user is mistakenly scored as a benign and she follows attack models A and E (a negative edge pair between her and the actual benign node), then removing this edge pair prevents a decrease in centrality of the benign user, and hence prevents increase of the malicious user's score.

*DOP(c): Remove all pairs of edges between benign nodes where one edge is positive and the other is negative.* The modifications related to positive-negative edge pair counters attack models C and E. Consider a malicious user, who is misidentified as a benign user, and having a negative edge to a benign user (model E) and the benign user has a positive edge to the malicious user (model C). Removing the positive edge counters attack model C, removing the negative edge counters attack model E and removing both the edges counters both the attack models in this case.

*DOP(d): In a positive/negative pair between two benign nodes, remove the negative edge.* Similar to the previous case.

*DOP(e): In a positive/negative pair between two benign nodes, remove the positive edge.* Similar rationale as for (c) above.

*Example 3:* Consider the SSN in Figure 1 (left) with SEC as centrality measure and $\tau = 0$. The SEC scores are as follows:

| Node | 1 | 2 | 3 | **4** | **5** | 6 |
|---|---|---|---|---|---|---|
| SEC | 0.16 | 0.33 | 0.68 | -0.30 | -0.55 | 0.09 |

Nodes 1, 2, 3, and 6 can be marked "benign" according to SEC, as they have a score greater than $\tau$. If we consider decluttering operations (a) and (d), we can remove the pair of positive edges between nodes 1-3 and 2-3, and the negative edge from 6 to 1. The resulting simplified network is shown in Figure 1 (right). Observe that the negative edge pair between nodes 3-5 is not removed because node 5 has score less than $\tau$. □

1: **Algorithm** TIA
2: **Input**: SSN $G = (V, E, W)$, centrality measure $\mathcal{C}$, a set $S = \{\rho_1, \ldots, \rho_m\}$ of decluttering operations, a threshold $\tau$
3: **Output**: A score for nodes in $V$
4: **do**
5: $G' = G$
6: C $\leftarrow$ compute $\mathcal{C}$ centrality of nodes in $V$ in graph $G$
7: $Benign = \{v \in V \mid centrality(C, v) \geq \tau\}$;
8: $Malicious = V - Benign$
9: $G = \rho_m \circ \cdots \circ \rho_1(G')$ %declutter graph
10: **while**$(G \neq G')$
11: Return C

Fig. 3: Troll Identification Algorithm (TIA).

In most online social networks, the number of malicious users is a small percentage of the total number of users. Hence, the number of interactions involving malicious users is much smaller than the number of interactions involving benign users. The removal of edges proposed in *DOPs* (a)-(e) reduces the effect benign users have on the network and magnify the actions of malicious users by removing the clutter of benign-benign user interactions. Our decluttering operations also counter attack models (A)-(E). For instance, consider the situation in Example 3: due to the decluttering operations we counter attack the attack model B used by node 2 and attack models B and C (resulting in the attack model D) followed by node 3.

## V. TIA Algorithm

In this section, we present the TIA algorithm (see pseudo-code in Figure 3). The algorithm takes as input, a signed social network $G = (V, E, W)$, together with any centrality measure $\mathcal{C}$ that applies to SSNs, as well as a centrality threshold $\tau$, and a set of decluttering operations selected from our 5 decluttering operations presented above. TIA operates iteratively and proceeds as follows.

• In the first iteration, it uses the original network to compute the centrality of all nodes in $V$ using the given centrality measure. Any node whose centrality is above the threshold $\tau$ is considered benign – all other nodes are considered malicious. It uses this initial labeling of nodes (which could be wrong) to declutter the graph using the selected decluttering operations. These operations transform the graph into a simpler graph.

• In the next iteration, we recompute the set of benign and malignant nodes using the decluttered graph and $\tau$. The updated set of benign and malignant nodes are used in conjunction with the decluttering operations to generate an even more simplified graph. This graph is the input to the next iteration.

• All iterations follow the same pattern as above — the iterations terminate when the decluttering in Step 9 of the algorithm leads to no change.

*Example 4:* Consider the toy SSN of Figure 1 (left). Suppose we run the TIA algorithm with NR, decluttering operations $S = \{a, b, d\}$, and a threshold $\tau = 0$. In the first iteration, shown in Figure 4(a), NR is computed on the original network, and nodes 2 and 3 are identified as benign users. So, the positive edge pair between these two nodes is removed resulting in the network shown in Figure 4(b). At this point, NR is computed again over this network and it identifies nodes 1, 3, and 6 as benign users. So, it removes the positive edge pair between nodes 1 and 3 and the negative edge from 6 to 1. This gives the network shown in Figure 4(c). Nodes $1, 4, 5$ and 6 are now marked as benign users by NR. Since no more decluttering operations can be further applied, the algorithm stops. At the end nodes 2 and 3 are correctly identified as trolls. In this process the same attack models that were counteracted in Example 3 are counteracted as well.

The table in Example 2 (right) shows the node rankings obtained by using TIA algorithm with the centrality measures considered in this paper and the set of decluttering operations $S = \{a, b, d\}$. Because the graph has been decluttered, SEC, SSR and NR are able to correctly identify both nodes 2 and 3 as trolls — something they could not do before (compare with the table in Example 2 left). □

We note that TIA terminates in at most $|E|$ iterations.

## VI. Slashdot Zoo Dataset Description

We tested our algorithm on a Slashdot Zoo data set.[2] This dataset is maintained by the authors of [1] and contains about 71.5K nodes and 490K edges — about 24% of the edges are negative. 96 nodes are marked as trolls by "No More Trolls" (an administrative Slashdot account). We treat these 96 trolls as the ground truth. This is the same setting followed by [1].

We evaluated TIA's performance in conjunction with various centrality measures and with various sets of decluttering operations. For each experimental setting, we also created various subsets of the entire Slashdot Zoo network. The subsets were created by randomly removing 5%, 10%, 15%, 20% and 25% of the nodes and their corresponding edges from the entire network. For each setting, we randomly generated 50 subgraphs of the Slashdot Zoo dataset by removing the appropriate percentage of edges from the network.

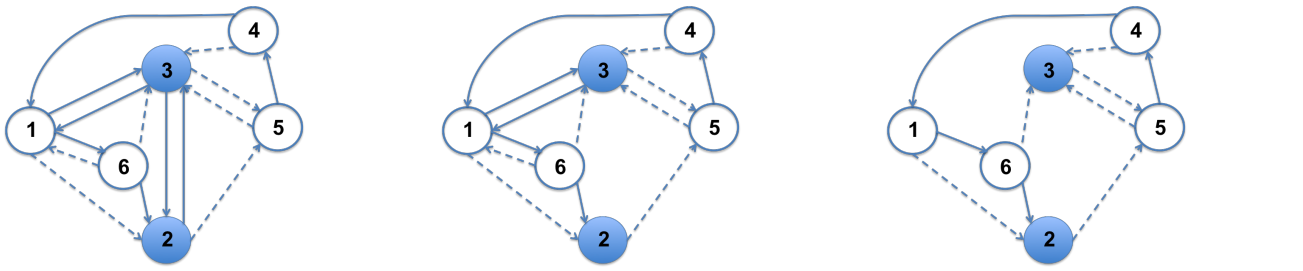
## VII. Experiments

We implemented TIA algorithm as well as all centrality measures and decluttering operations in this paper in about 1000 lines of Java code and ran them on a Intel Xeon @ 2.3 GHz, 24GB RAM Linux machine. We computed the score given by each centrality measure without any decluttering operations and with all possible subsets of decluttering operations (note that *DOPs* $d$ and $e$ together make *DOP* $c$, so there are 15 subsets). We use average precision and mean average precision for comparison [18] from IR (where they are used to measure goodness of document search algorithms). For systems that return a ranked sequence of documents, the measure considers the order in which the retrieved documents are presented. We use the same measure to compare the user scoring methods. Since our aim is to find malicious users, malicious and benign users are the analog of relevant and non-relevant documents, respectively.The average precision is defined as:

$$AveP = \frac{\sum_{k=1}^{n}(P(k) \times Mal(k))}{\text{Number Of Trolls}}$$

Here, *P(k)* is the precision at cut-off *k* (i.e. users ranked in the top $k$ for being malicious) and *Mal(k)* is 1 if the $k^{th}$ user is malicious, and 0 otherwise. $n$ is the total number of users.

[2] http://konect.uni-koblenz.de/networks/slashdot-zoo. Note that this is not the same Slashdot network used in [1], as that is not available.




| Nodes | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| NR | -0.06 | 0.13 | 0.45 | -0.40 | -0.68 | -0.01 |

(a)

| Nodes | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| NR | 0.02 | -0.15 | 0.48 | -0.49 | -0.72 | 0.04 |

(b)

| Nodes | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| NR | 0.13 | -0.11 | -0.85 | 0.21 | 0.34 | 0.07 |

(c)

Fig. 4: TIA algorithm iterations by using Negative Rank.

For the experiments on the subset networks, we find the mean of the AveP values of the 50 individual sub-networks and this is reported as the Mean Average Precision (MAP).

In freaks centrality, we took the mean of the maximum and minimum value of the centrality in each round as the threshold $\tau$ for computing the sets $Malicioius$ and $Benign$ in Steps 7 and 8 of the TIA algorithm. For others, zero is the threshold. We don't include PageRank as it is not intended to identify malicious users.

Tables III and IV show the average precision and the running time (in seconds), respectively, of the TIA algorithm using various subsets of decluttering operations and different SSN centrality measures applied to the Slashdot Zoo data set. In these tables, the *None* column shows results when only the centrality measure is used with no decluttering. For some sets of decluttering operations, Signed Spectral Rank does not converge in over 100,000 iterations - due to this, Negative Rank is also not found. Cells with - depict this.

Table V shows the number of trolls found among the lowest ranked 96 users by the TIA algorithm using various subsets of decluttering operations and different SSN centrality measures. Table VI shows the Mean Average Precision and the running time of the nine centrality measures and TIA algorithm using SEC with the top 2 settings that gave the best result on the original network, averaged over the 50 versions each for 95%, 90%, 85%, 80% and 75% randomly selected nodes from the Slashdot network.

*Best Settings.* The best results are obtained when TIA algorithm uses the Signed Eigenvector Centrality with decluttering operations *a* and *e*. In this setting, we retrieve more than twice as many trolls as Negative Rank does in the bottom 96 ranked users. The running time of this algorithm is less than 2 minutes, which is quite reasonable – and more than 27 times faster on the original network and 35-50 times faster on the sub-networks as compared to using the best centrality measure (Negative Rank) in [1].

## VIII. Conclusion

The problem of detecting trolls in online environments like Slashdot and other signed social networks is increasingly important as open source, collaboratively edited information becomes used more widely. Ensuring the integrity of this information is important for users while posing a technical challenge as many entities have strong incentives to compromise the accuracy of such information.

In this paper, we have shown that we can significantly improve on past works in the detection of trolls on Slashdot Zoo using a suite of decluttering operations that simplify a signed social network by removing confusing or irrelevant edges from the network. We proposed the TIA algorithm that takes any centrality measure and any set of decluttering operations as input parameters, and uses them to iteratively identify the trolls in the social network. Using the standard Average Precision measure to capture accuracy of our TIA algorithm, we show that TIA using Signed Eigenvector Centrality and decluttering operations *a* and *e* gives us the best result of 51.04%, significantly exceeding the 15.07% when no decluttering operations are performed. Compared to the best existing results on troll detection in Slashdot [3], our algorithm runs 27 times faster on the original network and 35-50 times faster on the sub-networks. Moreover it is able to retrieve about twice as many trolls with a Mean Average Precision which is more than three times as good as [1]. *The final message is simple: decluttering SSNs helps expose a clearer picture to our TIA algorithm which enables it to achieve much higher precision as well as identify far more trolls — all while running faster.*

There is much future work to be done. In this paper, we have not combined the power of natural language processing methods and network analysis methods to find trolls. Clearly, looking at the content of posts on Twitter or Facebook would help find troll like individuals in Facebook or Twitter – on Wikipedia, finding vandals involves NLP as we need to understand the relationship between changes made by a user and the previous content in order to check whether the edits reverted or contradicted what had previously been said.

**Acknowledgements.** This work was supported by US Army Research Office grant W911NF0910206.

## References

[1] J. Kunegis, A. Lommatzsch, and C. Bauckhage, "The slashdot zoo: mining a social network with negative edges," in *WWW*, 2009, pp. 741–750.

[2] S. Maniu, B. Cautis, and T. Abdessalem, "Building a signed network from interactions in wikipedia," in *DBSocial*, 2011, pp. 19–24.

[3] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!" in *ICWSM*, 2011.

| *Centrality* | *None* | *a* | *b* | *c* | *d* | *e* | *a,b* | *a,c* | *a,d* | *a,e* | *b,c* | *b,d* | *b,e* | *a,b,c* | *a,b,d* | *a,b,e* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Freaks | 15.07 | 15.22 | 14.92 | 14.77 | 14.77 | 15.22 | 14.92 | 14.77 | 14.77 | 15.22 | 14.46 | 14.46 | 14.92 | 14.46 | 14.46 | 14.92 |
| FMF | 3.13 | 4.6 | 3.13 | 3.12 | 3.13 | 3.13 | 4.2 | 4.35 | 4.33 | 4.64 | 3.13 | 3.12 | 3.13 | 4.01 | 3.96 | 4.25 |
| Prestige | 0.18 | 0.2 | 0.18 | 0.17 | 0.17 | 0.18 | 0.2 | 0.2 | 0.2 | 0.2 | 0.17 | 0.17 | 0.18 | 0.2 | 0.2 | 0.2 |
| M-PR | 1.25 | 0.99 | 1.26 | 1.21 | 1.28 | 1.19 | 0.96 | 0.94 | 0.84 | 1.12 | 1.23 | 1.28 | 1.19 | 0.76 | 0.83 | 0.9 |
| SSR | 10.27 | - | 10.44 | 10.05 | 10.34 | 10.03 | - | - | - | - | 10.17 | 10.46 | 10.16 | - | - | - |
| NR | 13.9 | - | 13.94 | 13.69 | 13.87 | 13.70 | - | - | - | - | 13.67 | 13.91 | 13.70 | - | - | - |
| SEC | 3.42 | 49.79 | 3.38 | 3.25 | 3.33 | 3.34 | 48.84 | 50.96 | 50.02 | 51.04 | 3.25 | 3.30 | 3.32 | 50.42 | 48.97 | 50.38 |
| M-HITS | 13.38 | 15.87 | 13.37 | 13.4 | 13.4 | 13.38 | 15.78 | 15.79 | 15.79 | 15.88 | 13.39 | 13.39 | 13.37 | 15.71 | 15.71 | 15.79 |
| BAD | 0.18 | 0.19 | 0.18 | 0.18 | 0.18 | 0.18 | 0.19 | 0.19 | 0.18 | 0.19 | 0.18 | 0.18 | 0.18 | 0.18 | 0.18 | 0.19 |

TABLE III: Table comparing Average Precision (in %) using TIA algorithm with different centrality measures and decluttering operations on Slashdot network.

| *Centrality* | *None* | *a* | *b* | *c* | *d* | *e* | *a,b* | *a,c* | *a,d* | *a,e* | *b,c* | *b,d* | *b,e* | *a,b,c* | *a,b,d* | *a,b,e* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Freaks | 0.25 | 2.89 | 2.48 | 2.02 | 1.89 | 2.69 | 4.57 | 4.82 | 4.72 | 4.62 | 4.53 | 4.55 | 4.53 | 6.05 | 6.98 | 6.99 |
| FMF | 0.41 | 2.31 | 1.74 | 2.12 | 2.2 | 2.85 | 3.4 | 2.99 | 2.99 | 2.98 | 4.6 | 3.09 | 3.07 | 2.84 | 2.83 | 2.83 |
| Prestige | 0.59 | 2.58 | 2.0 | 2.36 | 2.29 | 2.26 | 3.05 | 3.13 | 3.16 | 3.14 | 4.66 | 3.36 | 3.34 | 2.88 | 3.01 | 3.09 |
| M-PR | 13.8k | 65.6k | 38.4k | 39.2k | 26.4k | 50.6k | 73.1k | 63.8k | 59.9k | 67.5k | 39.7k | 49.7k | 51.9k | 73.3k | 78.6k | 76.6k |
| SSR | 2.5k | - | 7.5k | 10.3k | 7.6k | 7.6k | - | - | - | - | 9.9k | 6.1k | 7.5k | - | - | - |
| NR | 3.2k | - | 8.5k | 8.7k | 8.4k | 8.6k | - | - | - | - | 8.1k | 8.5k | 8.5k | - | - | - |
| SEC | 8.74 | 121.93 | 41.31 | 49.68 | 52.44 | 49.07 | 107.96 | 118.21 | 123.96 | 114.49 | 56.48 | 58.51 | 57.37 | 87.75 | 88.12 | 88.64 |
| M-HITS | 28.58 | 106.99 | 89.73 | 58.02 | 58.29 | 58.7 | 114.13 | 101.89 | 105.13 | 100.84 | 90.21 | 82.71 | 89.97 | 105.32 | 103.26 | 103.77 |
| BAD | 35.72 | 110.52 | 108.67 | 71.38 | 105.2 | 71.74 | 101.79 | 100.08 | 99.15 | 99.16 | 108.58 | 110.49 | 111.86 | 104.76 | 105.29 | 100.74 |

TABLE IV: Table comparing running time (in sec.) using TIA algorithm with different centrality measures and decluttering operations on Slashdot network.

| *Centrality* | *None* | *a* | *b* | *c* | *d* | *e* | *a,b* | *a,c* | *a,d* | *a,e* | *b,c* | *b,d* | *b,e* | *a,b,c* | *a,b,d* | *a,b,e* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Freaks | 17 | 17 | 16 | 16 | 16 | 17 | 16 | 16 | 16 | 17 | 15 | 15 | 16 | 15 | 15 | 16 |
| FMF | 10 | 10 | 10 | 10 | 10 | 10 | 9 | 10 | 10 | 10 | 10 | 10 | 10 | 9 | 10 | 9 |
| Prestige | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| M-PR | 6 | 6 | 5 | 5 | 6 | 5 | 6 | 6 | 5 | 7 | 5 | 5 | 5 | 5 | 6 | 6 |
| SSR | 19 | - | 20 | 19 | 19 | 19 | - | - | - | - | 20 | 20 | 20 | - | - | - |
| NR | 24 | - | 24 | 23 | 24 | 23 | - | - | - | - | 23 | 25 | 23 | - | - | - |
| SEC | 7 | 50 | 6 | 5 | 6 | 6 | 50 | 51 | 51 | 51 | 5 | 5 | 6 | 51 | 50 | 50 |
| M-HITS | 16 | 19 | 16 | 17 | 17 | 16 | 19 | 19 | 19 | 19 | 17 | 17 | 16 | 19 | 19 | 19 |
| BAD | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

TABLE V: Number of trolls found among the lowest ranked 96 users (the number of trolls in Slasdot is 96) by using TIA algorithm with different centrality measures and decluttering operations on Slashdot network.

| | 95% | | 90% | | 85% | | 80% | | 75% | |
|---|---|---|---|---|---|---|---|---|---|---|
| *Measure* | MAP | Runtime | MAP | Runtime | MAP | Runtime | MAP | Runtime | MAP | Runtime |
| Freaks | 15.35% | 0.17 | 15.22% | 0.17 | 15.12% | 0.16 | 15.46% | 0.15 | 15.62% | 0.14 |
| FMF | 3.23% | 0.24 | 3.16% | 0.26 | 3.2% | 0.23 | 3.52% | 0.21 | 3.42% | 0.19 |
| Prestige | 0.18% | 0.34 | 0.18% | 0.36 | 0.18% | 0.31 | 0.19% | 0.29 | 0.19% | 0.26 |
| M-PR | 1.31% | 12.6k | 1.3% | 10.9k | 1.43% | 8.9k | 1.67% | 8.3k | 1.6% | 7.6 |
| SSR | 10.34% | 1.7k | 10.27% | 1.6k | 10.21% | 1.4k | 9.95% | 1.2k | 10.05% | 1.1k |
| NR | 13.66% | 2.2k | 13.45% | 1.9k | 13.38% | 1.7k | 13.08% | 1.6k | 13.21% | 1.4k |
| SEC | 3.27% | 5.21 | 3.3% | 4.75 | 3.27% | 4.29 | 3.56% | 3.97 | 3.27% | 3.6 |
| M-HITS | 13.65% | 27.96 | 13.17% | 25.84 | 13.29% | 24.37 | 13.73% | 23.71 | 14.66% | 22.09 |
| BAD | 0.18% | 32.55 | 0.18% | 29.97 | 0.19% | 27.11 | 0.19% | 24.15 | 0.2% | 21.9 |
| SEC + *a,c* | 51.14% | 47.75 | 51.33% | 43.79 | 51.02% | 43.53 | 52.14% | 35.33 | 51.14% | 39.64 |
| SEC + *a,e* | 51.24% | 46.87 | 51.4% | 42.9 | 51.12% | 42.8 | 52.22% | 33.12 | 51.24% | 37.68 |

TABLE VI: Table showing Mean Average Precision (MAP) and runtime of the five centrality measures and TIA with SEC and the top 2 decluttering operations, averaged over 50 different versions each for 95%, 90%, 85%, 80% and 75% randomly selected nodes from the Slashdot network.


[4] R. Feldman, "Techniques and applications for sentiment analysis," *Commun. ACM*, vol. 56, no. 4, pp. 82–89, 2013.

[5] P. Bonacich and P. Lloyd, "Calculating status with negative relations," *Social Networks*, vol. 26, no. 4, pp. 331 – 338, 2004.

[6] J. Leskovec, D. P. Huttenlocher, and J. M. Kleinberg, "Predicting positive and negative links in online social networks," in *WWW*, 2010, pp. 641–650.

[7] ——, "Signed networks in social media," in *CHI*, 2010, pp. 1361–1370.

[8] J. A. Golbeck, "Computing and applying trust in web-based social networks," Ph.D. dissertation, University of Maryland, USA, 2005.

[9] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *WWW*, 2003, pp. 640–651.

[10] R. V. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust," in *WWW*, 2004, pp. 403–412.

[11] A. A. Zolfaghar K., "Mining trust and distrust relationships in social web applications," in *IEEE ICCP*, 2010, p. 7380.

[12] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks*, vol. 30, no. 1-7, pp. 107–117, 1998.

[13] M. Shahriari and M. Jalili, "Ranking nodes in signed social networks," *Social Netw. Analys. Mining*, vol. 4, no. 1, 2014.

[14] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *Journal of Mathematical Sociology*, vol. 2, no. 1, pp. 113–120, 1972.

[15] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[16] A. Mishra and A. Bhattacharya, "Finding the bias and prestige of nodes in networks based on trust scores," in *WWW*, 2011, pp. 567–576.

[17] D. Donato, S. Leonardi, and M. Paniccia, "Combining transitive trust and negative opinions for better reputation management in social networks," in *SNA KDD*, 2008.

[18] M. Najork, H. Zaragoza, and M. J. Taylor, "Hits on the web: how does it compare?" in *SIGIR*, 2007, pp. 471–478.