Identifying Vulnerabilities in Trust and Reputation Systems

Taha D. Güneş , **Long Tran-Thanh** and **Timothy J. Norman**
Electronics and Computer Science, University of Southampton, UK
{t.d.gunes, l.tran-thanh, t.j.norman}@soton.ac.uk

Abstract

Online communities use trust and reputation systems to assist their users in evaluating other parties. Due to the preponderance of these systems, malicious entities have a strong incentive to attempt to influence them, and strategies employed are increasingly sophisticated. Current practice is to evaluate trust and reputation systems against known attacks, and hence are heavily reliant on expert analysts. We present a novel method for automatically identifying vulnerabilities in such systems by formulating the problem as a derivative-free optimisation problem and applying efficient sampling methods. We illustrate the application of this method for attacks that involve the injection of false evidence, and identify vulnerabilities in existing trust models. In this way, we provide reliable and objective means to assess how robust trust and reputation systems are to different kinds of attacks.

1 Introduction

Trust and reputation systems (TRSs) are widely adopted by companies that host online communities for their customers. Users are encouraged to provide feedback on services and goods that are visible to others in aggregate, with commentaries often associated with individual ratings. The underlying philosophy is to drive increased service quality and to increase users' confidence in positive outcomes of future transactions. Companies such as Amazon, Ebay and Airbnb incorporate this idea within they own infrastructure, while companies such as TrustPilot, TripAdvisor, Yelp and Google offer TRS services to other organisations. Where users have a choice among a range of similar options, *relative* ratings can have a big impact on decisions. This, of course, introduces a strong incentive for companies and individual services/goods providers to game the system.

A wide variety of strategies have been reported, some simplistic such as injecting negative reviews for rival service providers, so called bad mouthing, or purchasing good "reviews". In response, TRS owners introduce controls; for example, only to permit reviews from confirmed customers. This has led to more sophisticated attacks, such as those reported recently by the Wall Street Journal [Emont and Bürge, 2018], where items are purchased and then returned in order to qualify to inject negative reviews. Users may report such incidents, but the moderation process is manual, time consuming, and may be equally used by dishonest sellers.

The complexity of attacks is expected to increase and, according to Brundage *et. al.* [2018], will soon exceed human capabilities through the malicious use of AI algorithms. It would be realistic to predict that TRSs can be influenced by sophisticated algorithms by, for example, automating the process of finding effective combinations of attacks. The current means by which TRSs are evaluated is by assessing the accuracy of predictions across a population of simulated agents, or through the use of data sets collected from rating sites. Robustness to certain kinds of known attacks such as whitewashing (exchanging a poor reputation for a default via a new identity) have been explored [Burnett *et al.*, 2010; Liu *et al.*, 2009]. While these methods are offer important benefits, they focus on a simple attack by a single actor, eschewing the possibility of strategic attacks.

Strategic attacks may involve multiple actors with coordinated objectives. Devising defensive strategies against them would require a different approach than assessing the vulnerabilities of a system to a set of known weaknesses. Alternative methods have been proposed in information security, however, including *fuzz testing* where the space of possible inputs to a system is searched to identify insecure states [Godefroid *et al.*, 2008]. We use a similar approach here: we start with a description of the space of possible ways to manipulate a TRS, and propose methods for efficiently searching this space driven by the objective of increasing the relative ranking of some target agent. In this way, we identify effective strategies that represent TRS vulnerabilities.

The contributions we present are threefold. First, we model coordinated, strategic attacks with a specific objective as a derivative-free optimization problem. We then propose two search methods for efficiently identifying coordinated attacks in complex attack spaces through sampling-based optimization. Finally, we use this novel method to analyze a selection of existing trust models, providing evidence for the kinds of complex attacks they are vulnerable to. The primary contribution is our new method for rigorously assessing trust and reputation systems, but before presenting this, we place our work in the context of existing research.

2 Related Work

Investigating Trust and Reputation Systems (TRSs) attacks is a key driver in the development of contemporary algorithms for assessing the trustworthiness of actors in online environments. Numerous kinds of attacks and defence strategies have been explored [Hoffman *et al.*, 2009], but robustness analyses of individual TRSs tend to consider relatively simple attack profiles. The Beta Reputation System with filtering [Whitby *et al.*, 2004], for example, focusses on identifying and excluding attackers who provide unfair feedback by badmouthing or ballot-stuffing. TRAVOS [Teacy *et al.*, 2006] takes a similar approach, but discounts outlying ratings in making trust assessments (cf. Muller *et al.* [2015]). The HABIT [Teacy *et al.*, 2012] model uses hierarchical Bayesian model to identify participants with various profiles of reliability, and factor this into aggregated ratings. In this way, evidence from unreliable participants are not simply filtered out, but their biases taken into account. These and other models, in essence, only consider the fact that participants may provide unreliable ratings.

There are a few reported studies that analyze robustness of TRSs against realistic attacks [Ruan and Durresi, 2016]. In general, the approach taken is to first identify the types of vulnerability of interest. A strategy (or set of strategies) to exploit the vulnerability is then devised, and the candidate model is assessed (either theoretically or empirically) against them. Kerr & Cohen [2009], for example, explore what they refer to as a *reputation lag* vulnerability, where the attacker delays their actions within a transaction (e.g. shipping items) to postpone negative feedback. They define a sequence of predefined actions for how an attacker may exploit this vulnerability. This attacker-profiling approach is a common method applied across a range of security contexts. These studies are valuable in identifying specific attack strategies, but a key challenge is in understanding how vulnerable TRSs are to an intelligent, adaptive attacker.

One of the most common classes of attack on TRSs centres on the injection of false evidence [Jøsang and Golbeck, 2009]. These kinds of attack include the misleading feedback attack, unfair rating attack, bad mouthing and ballot stuffing [Wang *et al.*, 2014], and are referred to by Hoffman *et al.* [2009] as self-promotion and slandering. Muller *et al.* [2016] provide advice for the design of TRSs to mitigate these kinds of attack, but not the means to identify vulnerabilities from such attackers. Wang *et al.* [2015] show how users and services can hide their true behaviour using these strategies. Various attacks including injecting false evidence and whitewashing are considered by Bidgoly & Ladani [2016], where these are modelled as primitive actions in a planning mechanism (POMDP) that learns effective attack strategies through trial and error. The use of a partially observable MDP is relevant in designing a single attacker attempting to exploit an unknown TRS, where the ordering of the attacker's actions influences the outcome. In practice, however, the search space for even a single attacker is substantial, making a POMDP-based method infeasible. Further, the focus of this paper is on identifying vulnerabilities to coordinated attacks on TRSs from multiple actors, which is an important emerging threat to contemporary systems.

3 A Trust Environment

We are agnostic about the specific nature of the trust model being employed, and so we characterise the trust assessment problem in relatively abstract terms. We assume a set of agents, $\mathcal{A} = \{a_1, ..., a_n\}$, consisting of (potentially overlapping) sets of consumers, $\mathcal{C} = \{c_1, ..., c_l\}$ and service providers $\mathcal{P} = \{p_1, ..., p_m\}$. Some consumers may also act as witnesses $\mathcal{W} \subseteq \mathcal{C}$, and we identify a specific agent, $\delta \in \mathcal{A}$ as the decision maker. The series of direct (or reported) observations made by a consumer (or witness), c_i, of the performance of a provider, p_j, up to time t is $O^{0:t}_{c_i \to p_j}$. We assume that observations are discrete, and the number of possible values that an observation may have is bounded: $O^{t}_{c_i \to p_j} = 0, ..., k$ where $k \geq 2$. All information that is, in principle, available to form a prediction of the future behaviour of an agent (i.e. a trust assessment) at time t is, therefore, $\mathcal{E} = \left\{ O^{0:t}_{c_i \to p_j} \mid c_i \in \mathcal{C}, p_j \in \mathcal{P} \right\}$. The goal of a statistical trust model is to use such evidence to make assessments of future performance; i.e. the aim is to compute, for c_i interested in the future performance of p_j, the expectation of $\Pr\left(O^{t+1}_{c_i \to p_j} \mid \mathcal{E} \right)$.

In recommender systems it is reasonable to assume that all *reported* evidence from witnesses is available to a decision-maker/aggregator. In the multi-agent context this is not the case, and hence we consider situations in which a single decision-maker, δ, has a partial view of the evidence available, $\mathcal{E}^{\delta} \subset \mathcal{E}$. In *both* recommender systems and multi-agent systems, however, evidence may be misleading; i.e. a reported observation may differ from the actual experience of the consumer concerned. Furthermore, in the multi-agent context, the veracity of each reported observation may differ for each agent collating its own viewpoint on the evidence of past interactions; i.e. $O^{t}_{c_i \to p_j}$ may vary among agents because $c_i \in \mathcal{W}$ provided very different witness reports. Throughout the paper we consider the perspective of the agent that is the target of the attack (the decision maker, δ), and so $O^{0:t}_{c_i \to p_j}$ is always understood to be the observations *reported by* c_i about p_j to δ; actual observations may be missing and inaccurate ones may be added. We refer to the set of evidence available to agent δ on the basis of reported observations from other agents and its own direct experience as $\mathcal{E}^{\delta}$.

Given the evidence available, a decision maker needs to make assessments of the relative trustworthiness of potential providers, and use these to decide whom to trust. For simplicity and ease of evaluation, we consider only the relative ranking of potential providers, which is the typical output of a trust assessment mechanism. We do, however, consider adversarial witnesses that can inject spurious evidence into the system. The challenge for an adversary (or a set of adversaries), therefore, is to find types of attack that significantly influence the decision maker. The challenge for a trust and reputation system, in contrast, is how to interpret evidence in a manner that is robust to the possibility of adversaries searching for means to exploit the system. Addressing both challenges is necessary to develop a generalised attacker model for trust and reputation systems.

3.1 The Attack Space

We define an attack as an alteration of the evidence available to a decision maker. A successful attack is one for which the relative trustworthiness of the provider agents is significantly changed from the viewpoint of the decision maker, δ. If we assume that the evidence available to δ prior to the attack is $\mathcal{E}^{\delta}$, an attack is the introduction of $\mathcal{E}'$ so that $\tilde{\mathcal{E}}^{\delta} = \mathcal{E}' + \mathcal{E}^{\delta}$, where $\mathcal{E}'$ contains our misleading/fake reviews. We make no assumptions about the new evidence, $\mathcal{E}'$. It may be from multiple witnesses, either because it is a collaborative attack, or because an attacker can, in some way, control the generation of these reports. Identifying the most rewarding attack in some context is, clearly, a highly complex problem.

In reality, an attacker will be restricted by the number of witness reports it can affect, and there will be limits to the number of additional observations that it can inject into the system. We, therefore, investigate cases in which an attacker is limited by: (1) its *power*, or the number of observations that it can add through the attack ($\rho = |\mathcal{E}'|$); and (2) its *control* over the witnesses ($\mathcal{W}' \subseteq \mathcal{W}$). The space of possible attacks is $\mathcal{X}$, such that:

$$|\mathcal{X}| = \binom{\rho + k \cdot \left|\left\{O^{0:t}_{w_i \to p_j} \mid w_i \in \mathcal{W}', p_j \in \mathcal{P}\right\}\right| - 1}{k \cdot \left|\left\{O^{0:t}_{w_i \to p_j} \mid w_i \in \mathcal{W}', p_j \in \mathcal{P}\right\}\right|} \quad (1)$$

The space of possible attacks is then the weak compositions of ρ into the space in which the selected witnesses are controlled by the attacker to provide new reports. When $|\mathcal{W}'|$ is reasonably large, it is not feasible to sample even a small percentage of this space. For this reason, we explore a restriction on strategies that reduces this large attack space, while avoiding the imposition of designed-in attacks as is done in related research. The aim here is to retain the challenge for the attacker, where in any realistic scenario its search would be limited to the selection of witnesses to use in an attack, because using a witness may be costly in some context (e.g. cost of spoofing or bribing the witness.).

The space of attacks is defined in terms of:

1. The number of witnesses to be used, s; and
2. The distribution of the attack power, ρ, across these selected witnesses, considering those they can report on:
   (a) All restricted partitions of ρ into s ($D = RP_s(\rho)$) and their permutations without repetition: P_s^D
   (b) The distribution of these permutations to each witness-provider pair, such that the number of possible distributions is $(|\mathcal{P}| \cdot k)^s$

The number of attacks in this reduced space is, therefore:

$$|\mathcal{X}| = \binom{|\mathcal{W}'|}{s} D \cdot P_s^D \cdot (|\mathcal{P}| \cdot k)^s \quad (2)$$

where restricted partitions of ρ into s parts is: $RP_s(\rho) = RP_s(\rho - s) + RP_{s-1}(\rho - 1)$, $RP_0(0) = 1$ and $RP_s(\rho) = 0$ if $\rho \leq 0$ or $s \leq 0$. The number, ρ, of additional reported observations from witnesses is distributed across all partitions, restricted by the number of selected witnesses and the number of providers. By this reduction, each witness can provide a portion of the malicious reviews to a single selected provider.

3.2 An Example Attack

To illustrate the kinds of attack within this space, and the potential effect of an attack from the perspective of the target, δ, consider the example illustrated in Figure 1. Here, we have five providers, $\{p_1 \ldots p_5\} \in \mathcal{P}$ and five witnesses, $\{c_1 \ldots c_5\} \in \mathcal{W}$, the attacker has power, $\rho = 5$, and it has control over (and/or has chosen) witnesses c_1, c_3 and c_4 through which to target its attack. The aim is to improve the relative position of provider p_1 from the perspective of the decision maker, δ.

In Figure 1, we show the ranking of each provider, $r(p_i)$, before and after the attack, where this ranking is based on the trustworthiness of each provider computed using a beta distribution on the basis of positive $(+1)$ and negative (-1) observations reported by our witnesses. The detail being:

1. The attacker injects one positive rating from witness c_1 regarding p_1, increasing c_1's overall view of p_1 to $+2$;
2. It injects one negative rating from witness c_3 to p_3, reducing c_3's overall view of p_3 down by -1; and
3. It injects three negative ratings from witness c_4 regarding p_5, dropping this from $+1$ to -2.

One of the interesting characteristics of this attack (identified by our model) is that it distributes ρ across a number of p_1's competitors as well as investing a small amount in promoting p_1. In this case the trust model is quite simple, but it illustrates the kinds of orchestrated attack strategy that can be identified. The questions remaining are: what is an optimal attack, and how do we discover them efficiently?

3.3 Optimal Attacks

The optimal attack can be characterised in a number of ways. We may consider setting the objective to maximise the absolute trust rating of the attacker, p_a:

$$\begin{aligned} \mathcal{E}^* = &\operatorname*{arg\,max}_{\mathcal{E}'} \ \tau(\delta, p_a, \tilde{\mathcal{E}}^{\delta}) - \tau(\delta, p_a, \mathcal{E}^{\delta}) \\ &\text{subject to } \tilde{\mathcal{E}}^{\delta} = \mathcal{E}' + \mathcal{E}^{\delta} \end{aligned} \quad (3)$$

where $\tau(\delta, p_a, \mathcal{E})$ is the decision maker's trust in the attacker, p_a. This may, however, have no impact on the *rank* of the attacker, even in the cases where the optimal attack is found. This depends on the underlying formulation of the TRS. To this end, we change the attacker's objective to focus on its rank, in this case, it will be improving its rank:

$$\begin{aligned} \mathcal{E}^* = &\operatorname*{arg\,max}_{\mathcal{E}'} \ r(\delta, p_a, \mathcal{E}^{\delta}) - r(\delta, p_a, \tilde{\mathcal{E}}^{\delta}) \\ &\text{subject to } \tilde{\mathcal{E}}^{\delta} = \mathcal{E}' + \mathcal{E}^{\delta} \end{aligned} \quad (4)$$

In this case, this attacker's objective is a hard (in this case, discrete) optimisation problem, which is strongly non-convex. The difficulty stems from the fact that the objective function in Equation (4) depends on the implemented TRS, a "black box" whose formulation may be impossible to access or intrinsically complex. Gradient-based methods are

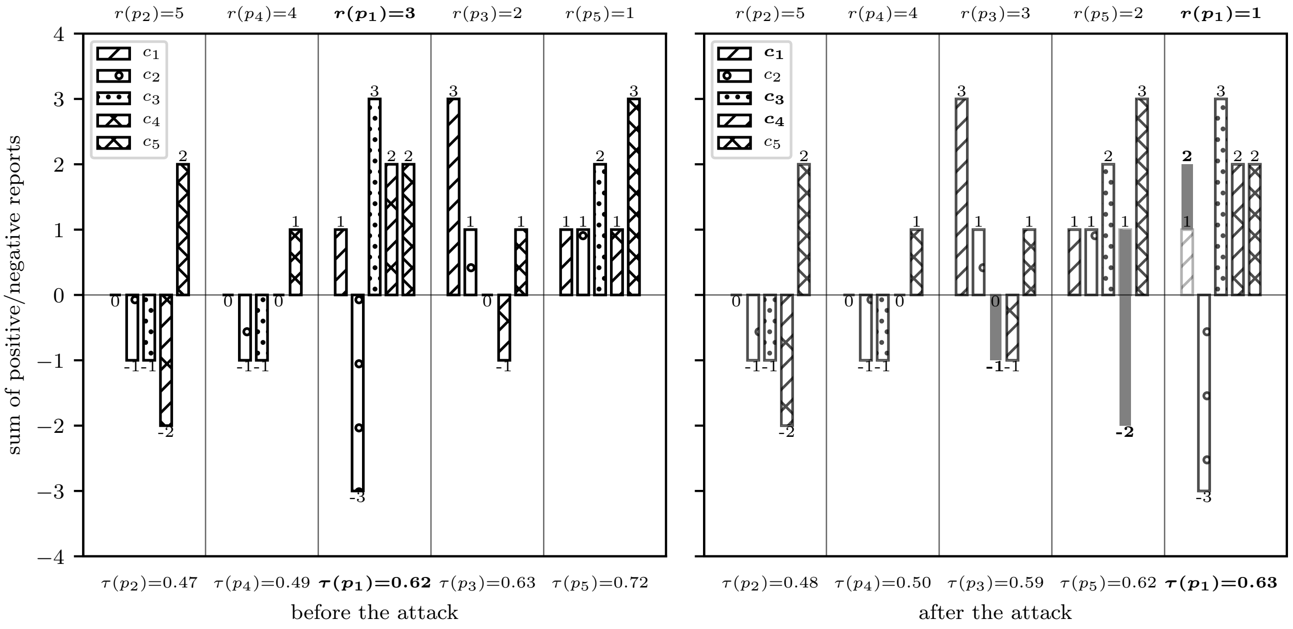

Figure 1: Agent δ's relative rankings of service providers before and after a strategic attack, where the ρ=5 and s=3. The malicious attacker, p_1, has control over witnesses $c_1, c_3, c_4 \in \mathcal{W}'$.

unlikely to be suitable in this case, failing to escape from local minima. For this reason, we apply two sampling-based optimisation strategies to search for attacks.

3.4 Attack Search Strategies

Given the size of the space of possible attacks in realistic scenarios, an attacker will be able to sample only a small proportion, the extent of which will depend on computational resources available. To solve the attacker's optimisation problem, we apply Monte Carlo and hierarchical sampling-based optimisation techniques.

Monte Carlo Sampling, MCS, uses Monte Carlo simulation to randomly sample the objective function, approximating the expected reward via the empirical mean [Kleywegt *et al.*, 2002].

Hierarchical Sampling, HS, is an optimisation technique that is designed to exploit smoothness properties of an objective function (local Lipschitz) [Bubeck *et al.*, 2011]. The smoothness property, in our case, would be manifest if similarly rewarding attacks are closely ordered in the space.

These sampling methods can, of course, be halted at any time, returning the best attack identified.

4 Evaluation

In order to evaluate our model for identifying trust and reputation system vulnerabilities we provide a simulation environment, through which controlled experiments can be conducted for attacks focused on a target decision maker, δ. This simulation and analysis environment, along with implementations of the trust models used in this section, is freely available [Güneş *et al.*, 2019].

Parameter	Value	Description
$\lvert\mathcal{P}\rvert$	20	The number of provider agents
$\lvert\mathcal{W}\rvert$	20	The number of witness agents
s	2	The number of witnesses under the attacker's control
t	10	The number of provider observations made by each witness

Table 1: Experimental constants.

The simulation environment is designed to assess the effect of a series of attack trials (determined by the sampling method employed). Before each attack, a set of witnesses, $\mathcal{W}$, interact with a set of providers, $\mathcal{P}$, over a number of rounds. Observations made by witnesses are drawn from Bernoulli distributions characterising the behaviour of each provider. The parameters of these Bernoulli distributions are drawn from either a Uniform distribution or a Dirichlet with all its parameters set to 20 to produce providers that behave in a similar manner.

To capture variety in connectivity between witnesses and providers, we introduce an *indirect knowledge degree*, d, that denotes the chance of each witness interacting with a provider, and t is the number of times they interact (see Table 1). Witnesses transform observations received from each provider via a k-by-k behaviour matrix θ_{c_i}, which allows us to control how each witness reports observed behaviour from each provider. Their reports are categorically distributed by this matrix, with values for each row in θ_{c_i} drawn from a distribution such that the sum of each row is 1.0. If, for example, we have binary observations, $k = 2$, and if $\theta_{c_i} = \begin{bmatrix} 0 & 1.0 \\ 1.0 & 0 \end{bmatrix}$ the witness reports the reverse of their true observation without noise. Note, therefore, that all providers behave in a similar manner, but witnesses vary in reliability.

We consider four experimental variables: the strategy used to search for attacks (MCS or HS); the connectivity between witnesses and providers (d); the power of the attacker (ρ); and the behaviour of witnesses (θ_{c_i}). Other parameters are fixed as specified in Table 1. We further restrict the attacker to explore only 1% of the search space for all strategies across all experiments.

Four widely studied TRSs along with a simple baseline (average) function are selected for our investigation. These models (summarised below) represent a variety of commonly employed techniques for handling malicious witnesses. We implemented them based on information from respective papers, choosing reasonable values for parameters after a set of runs to ensure that performance is not hindered, and all implementations are freely available [Güneş *et al.*, 2019].

BRS [Jøsang and Ismail, 2002] uses Bayesian update to fuse observations from different providers and witnesses. The work by Whitby et. al. [2004] extends the model by adding a filtering mechanism where evidence that deviates from the majority up to a degree is discarded.

TRAVOS [Teacy *et al.*, 2006] discounts the influence of witnesses by heuristically calculating the similarity between distributions of witness observations; in contrast, BRS discards divergent reports. In TRAVOS, similarity is calculated by tabulating the outcomes by using a particular selection of bins that denote regions of the outcome distribution.

HABIT [Teacy *et al.*, 2012] is a hierarchical Bayesian model to estimate trustworthiness by similarities between providers. The decision maker calculates the similarity between the opinions of witnesses about a provider in comparison to other providers and the weighted average is calculated.

EIGEN [Kamvar *et al.*, 2003] uses power iteration to capture transitivity of trust between parties. The outcomes of observations are normalised and stored in a global matrix. A global trust value is then calculated using the left-principal eigenvector of this matrix.

The objective of the attacker, p_a, is to maximise rank gain $r_{TRS}(\delta, p_a, \mathcal{E}) - r_{TRS}(\delta, p_a, \mathcal{E}^a)$ from the perspective of δ. The attacker has no knowledge about any theoretically identified vulnerabilities of TRSs, and has no predetermined strategy. The attacker observes a snapshot of the system and creates a strategy, given ρ and s, in order to increase its rank.

To gain further insights into the TRSs considered, we categorise the most effective attack identified in each sampling run to determine the type of attack it represents and its effect. Our categories include, but extend those outlined by Hoffman *et al.* [2009]. The definitions we use are: *self-promoting* SP, positive reports to the attacker, *self-slandering* SS, negative reports to the attacker, *self-orchestrated* SO, both negative and positive reports to the attacker, *slandering* S negative reports to other providers, *promoting* P, positive reports to other provider, *orchestrated* O positive and negative reports to other providers and *complete-orchestrated* CO negative or positive reports to both the attacker and providers. The frequency of each type of attack identified and the degree of rank gain that is achieved are measured.

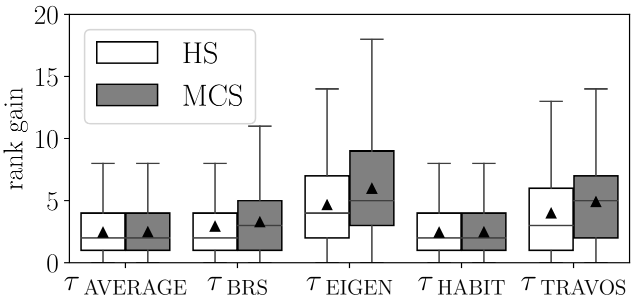

Figure 2: Comparing MCS and HS in varying TRSs. Triangles denote the mean of the corresponding distribution.

4.1 Results

Here, we present the results of our experiments. Each experimental condition is repeated 3000 different instances, so that we minimise the effect of the starting point of the attacker before the attack. In our results, we plot the distribution of rank gain and *mean rank gain* over these scenarios to illustrate the performance of our attacker model. To validate the statistical significance, we performed pairwise Mann–Whitney U tests with Bonferroni correction[1].

Identifying an Effective Search Strategy

Figure 2 shows the performance of the attacker across different TRSs given the selected search strategy. The attacker achieves a minimum of 2 rank gains on average for all TRSs, but EIGEN and TRAVOS are significantly more vulnerable. With respect to our search strategies, MCS performed at least as well as HS for all TRSs, and showed a significantly higher performance against BRS, EIGEN and TRAVOS. All three of these cases were statistically significant with $p < 0.001$. The cause of the differences between MCS and HS is, we believe, because there is little structure in the space of attacks (smoothness property) that may be exploited by HS. In subsequent experiments we use MCS as our search strategy.

Optimising Attack Strategies

Figure 3 shows the performance (rank gain) as power ρ (Figure 3a), population behaviour (Figure 3b) and connectivity between witnesses and providers (Figure 3c) is varied. As is expected, increasing the power of the attacker enables it to achieve a greater rank gain against all TRSs (Figure 3a-b). The rate of increase does, however, vary across the two population profiles. When the results from Figure 3b is compared with Figure 3a, the rank gain achieved against EIGEN on average decreased when providers have similar behaviours. Figure 3c shows Average and HABIT models had smaller mean rank gain comparing to BRS, EIGEN and TRAVOS, starting from $t = 6$.

[1] Our reason for choosing this test is that the resulting rank gain distributions were not normally distributed according to the Shapiro-Wilk test.

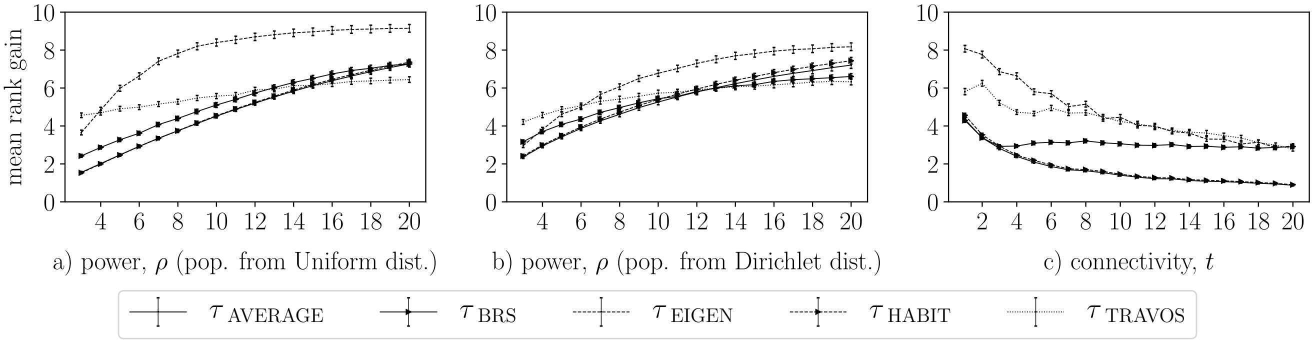

Figure 3: Comparing TRSs where power of the attacker, the evidence available and the population behaviour is varied. Error bars denote the standard error of the mean rank gain.

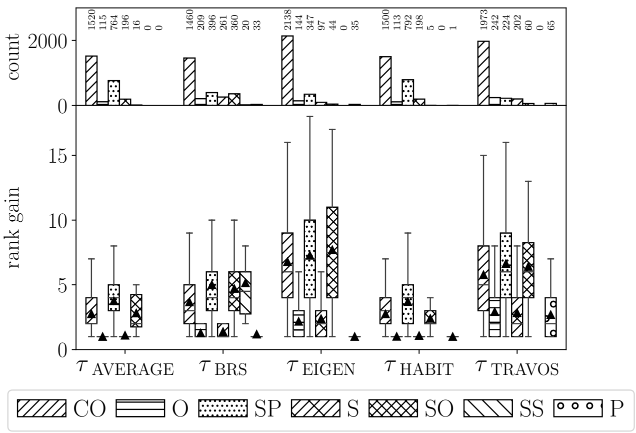

Figure 4: Distributions of rank gain achieved when an attack type is selected in varying TRSs.

Identifying Vulnerabilities

Figure 4 shows the types of attacks that are found by our attacker model. In the top section, we show the count of each type of attack and below the distribution of rank gain achieved. *Complete-orchestrated* (CO) attacks were most effective for all TRSs, and utilising CO attacks returned higher rank gain than other strategies. In the remaining cases, SP was the second mostly selected attack strategy with a lower expectation.

5 Discussion

The results show that, under challenging scenarios where the attacker power and the number of controlled witnesses are limited, our model was able to affect all TRSs considered such that the attacker attained a rank position increase of at least 2. Selected TRSs vary in their resilience to attacks, and many perform poorly in in comparison to a simple averaging mechanism. Further insight into the structure of effective attacks can be acquired by clustering attack patterns around specific categories. This reveals that the best attack strategies (at least for a rank gain objective) consist of a combination of actions against a range of providers. A robustness analysis of this kind provides a TRS designer with a useful tool to understand the types of strategies that might be employed by a sophisticated attacker, and hence focus development of mitigation methods.

Assumptions made in this research include that the attacker can observe all available evidence, and knows the TRS being employed. The attacker can, therefore, calculate the ranks of each provider whenever the evidence changes. In practice, the attacker will have some uncertainty of the TRS being used in the target system. From the perspective of the designer, however, it is reasonable to analyse resilience of a TRS from this worst-case perspective.

It is worth mentioning that our attacker model selects witnesses according to the objective function without considering the cost of using a particular witness. Costs associated with witness selection may vary; e.g. employing a witness considered trustworthy may incur higher cost. This could, however, be captured by adapting the objective function.

We view the TRS analysis method proposed as a basis for reducing vulnerabilities in future trust models. Coordinated attack patterns identified for a specific TRS may be used as a basis for automated attack recognition mechanisms to supplement the system. Suspicious patterns identified can be passed on for further investigation.

6 Conclusion

We have introduced and demonstrated the practical value of a new and generic method for identifying vulnerabilities in TRSs. Given a characterisation of the space of possible attacks, we define an attacker model. Our model may then be employed to search for effective strategies through derivative-free optimisation methods. The outcome is a set of attack profiles and an estimate of the vulnerability of the TRS to an attack of this kind. In this way, we contribute to the development of future trust and reputation systems that are less vulnerable to sophisticated external threats.

Acknowledgments

The authors gratefully acknowledge financial support from the EPSRC Doctoral Training Partnership, and the use of IRIDIS HPC facility at the University of Southampton.

References

[Bidgoly and Ladani, 2016] A. J. Bidgoly and B. T. Ladani. Modeling and quantitative verification of trust systems against malicious attackers. *The Computer Journal*, 59(7):1005–1027, 2016.

[Brundage *et al.*, 2018] M. Brundage, S. Avin, J. Clark, H. Toner, P. Eckersley, B. Garfinkel, A. Dafoe, P. Scharre, T. Zeitzoff, B. Filar, et al. The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. *arXiv preprint arXiv:1802.07228*, 2018.

[Bubeck *et al.*, 2011] S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári. X-armed bandits. *J. Mach. Learn. Res.*, 12:1655–1695, 2011.

[Burnett *et al.*, 2010] C. Burnett, T. J. Norman, and K. Sycara. Bootstrapping trust evaluations through stereotypes. In *Proc. AAMAS*, pages 241–248, 2010.

[Emont and Bürge, 2018] J. Emont and C. Bürge. How Scammers in China Manipulate Amazon. *Wall Street Journal*, 2018. https://www.wsj.com/articles/how-scammers-in-china-manipulate-amazon-11545044402 Accessed on 02/01/2019.

[Godefroid *et al.*, 2008] P. Godefroid, M. Y. Levin, and D. Molnar. Automated Whitebox Fuzz Testing. In *NDSS*, volume 8, pages 151–166, 2008.

[Güneş *et al.*, 2019] T. D. Güneş, L. Tran-Thanh, and T. J. Norman. Attack strategies and analysis for trust and reputation systems. (dataset). https://doi.org/10.5258/SOTON/D0937, 2019. Accessed on 03/06/2019.

[Hoffman *et al.*, 2009] K. Hoffman, D. Zage, and C. Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Comput. Surv.*, 42(1):1–31, 2009.

[Jøsang and Golbeck, 2009] A. Jøsang and J. Golbeck. Challenges for robust trust and reputation systems. In *Proc. SMT*, pages 52–64, 2009.

[Jøsang and Ismail, 2002] A. Jøsang and R. Ismail. The beta reputation system. In *Proc. 15th Bled Electronic Commerce Conference*, volume 5, pages 2502–2511, 2002.

[Kamvar *et al.*, 2003] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust algorithm for reputation management in P2P networks. In *Proc. WWW*, pages 640–651, 2003.

[Kerr and Cohen, 2009] R. Kerr and R. Cohen. Smart cheaters do prosper: Defeating trust and reputation systems. In *Proc. AAMAS*, pages 993–1000, 2009.

[Kleywegt *et al.*, 2002] A. J. Kleywegt, A. Shapiro, and T. Homem-de-Mello. The sample average approximation method for stochastic discrete optimization. *SIAM J. Optimization*, 12(2):479–502, 2002.

[Liu *et al.*, 2009] X. Liu, A. Datta, K. Razdca, and E. P. Lim. Stereotrust: A group-based personalized trust model. In *Proc. 18th ACM CIKM*, pages 7–16, 2009.

[Muller *et al.*, 2015] T. Muller, Y. Liu, and J. Zhang. The fallacy of Endogenous Discounting of Trust Recommendations. In *Proc. AAMAS*, pages 563–572, 2015.

[Muller *et al.*, 2016] T. Muller, D. Wang, Y. Liu, and J. Zhang. How to use information theory to mitigate unfair rating attacks. In *Trust Management X*, pages 17–32. Springer, 2016.

[Ruan and Durresi, 2016] Y. Ruan and A. Durresi. A survey of trust management systems for online social communities – Trust modeling, trust inference and attacks. *Knowledge-Based Systems*, 106:150 – 163, 2016.

[Teacy *et al.*, 2006] W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck. TRAVOS: Trust and reputation in the context of inaccurate information sources. *Auton. Agent. Multi-Agent Syst.*, 12(2):183–198, 2006.

[Teacy *et al.*, 2012] W. T. L. Teacy, M. Luck, A. Rogers, and N. R. Jennings. An efficient and versatile approach to trust and reputation using hierarchical Bayesian modelling. *Artif. Intell.*, 193:149–185, 2012.

[Wang *et al.*, 2014] D. Wang, T. Muller, Y. Liu, and J. Zhang. Towards robust and effective trust management for security: A survey. In *Proc. PST*, pages 511–518, 2014.

[Wang *et al.*, 2015] D. Wang, T. Muller, A. A. Irissappane, J. Zhang, and Y. Liu. Using information theory to improve the robustness of trust systems. In *Proc. AAMAS*, pages 791–799, 2015.

[Whitby *et al.*, 2004] A. Whitby, A. Jøsang, and J. Indulska. Filtering out unfair ratings in Bayesian reputation systems. In *Proc. 7th Int. Workshop on Trust in Agent Societies*, pages 106–117, 2004.