Making Paper Reviewing Robust to Bid Manipulation Attacks

Ruihan Wu [*1] **Chuan Guo** [*2] **Felix Wu** [3†] **Rahul Kidambi** [4†] **Laurens van der Maaten** [2] **Kilian Q. Weinberger** [1]

Abstract

Most computer science conferences rely on paper bidding to assign reviewers to papers. Although paper bidding enables high-quality assignments in days of unprecedented submission numbers, it also opens the door for dishonest reviewers to adversarially influence paper reviewing assignments. Anecdotal evidence suggests that some reviewers bid on papers by "friends" or colluding authors, even though these papers are outside their area of expertise, and recommend them for acceptance without considering the merit of the work. In this paper, we study the efficacy of such *bid manipulation attacks* and find that, indeed, they can jeopardize the integrity of the review process. We develop a novel approach for paper bidding and assignment that is much more robust against such attacks. We show empirically that our approach provides robustness even when dishonest reviewers collude, have full knowledge of the assignment system's internal workings, and have access to the system's inputs. In addition to being more robust, the quality of our paper review assignments is comparable to that of current, non-robust assignment approaches.

1. Introduction

Peer review is a cornerstone of scientific publishing. It also functions as a gatekeeper for publication in top-tier computer-science conferences. To facilitate high-quality peer reviews, it is imperative that paper submissions are reviewed by qualified reviewers. In addition to assessing a reviewer's qualifications based on their prior publications (Charlin & Zemel, 2013), many conferences implement a *paper bidding* phase in which reviewers express their interest in reviewing particular papers. Facilitating bids is important because the review quality is higher when reviewers are interested in a paper (Stent & Ji, 2018).

Unfortunately, paper bidding also creates the potential for difficult-to-detect adversarial behavior by reviewers. In particular, a reviewer may place high bids on papers by "friends" or colluding authors, even when those papers are outside of the reviewer's area of expertise, with the purpose of accepting the papers without merit. Anecdotal evidence suggests that such *bid manipulation attacks* may have, indeed, influenced paper acceptance decisions in recent top-tier computer science conferences (Vijaykumar, 2020; Littman, 2021).

This paper investigates the efficacy of bid manipulation attacks in a realistic paper-assignment system. We find that such systems are, indeed, very vulnerable to adversarial bid, which is corroborated by prior work (Jecmen et al., 2020). Furthermore, we design a paper-assignment system that is robust against bid manipulation attacks. Specifically, our system treats paper bids as supervision for a model of reviewer preferences, rather than directly using bids to assign papers. We then detect atypical patterns in the paper bids by measuring their influence on the model, and remove such high-influence bids as they are potentially malicious.

We evaluate the efficacy of our system on a novel, synthetic dataset of paper bids and assignments that we developed to facilitate the study of robustness of paper-assignment systems. We carefully designed this dataset to match the statistics of real bidding data from recent computer-science conferences. We find that our system produces high-quality paper assignments on the synthetic dataset, while also providing robustness against groups of colluding, adversarial reviewers in a *white-box setting* in which the adversaries have full knowledge of the system's inner workings and its inputs. We hope our findings will help computer-science conferences in performing high-quality paper assignments at scale, while also minimizing the surface for adversarial behavior by a few bad actors in their community.

2. Bid Manipulation Attacks

We start by investigating the effectiveness of bid manipulation attacks on a typical paper assignment system.

Paper assignment system. Most paper assignment systems utilize a computed score $s_{r,p}$ for each reviewer-paper pair

---

[*] Equal contribution. [†] Work done while at Cornell University. [1] Department of Computer Science, Cornell University [2] Facebook AI Research [3] ASAPP [4] Amazon Search & AI. Correspondence to: Ruihan Wu <rw565@cornell.edu>, Chuan Guo <chuanguo@fb.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

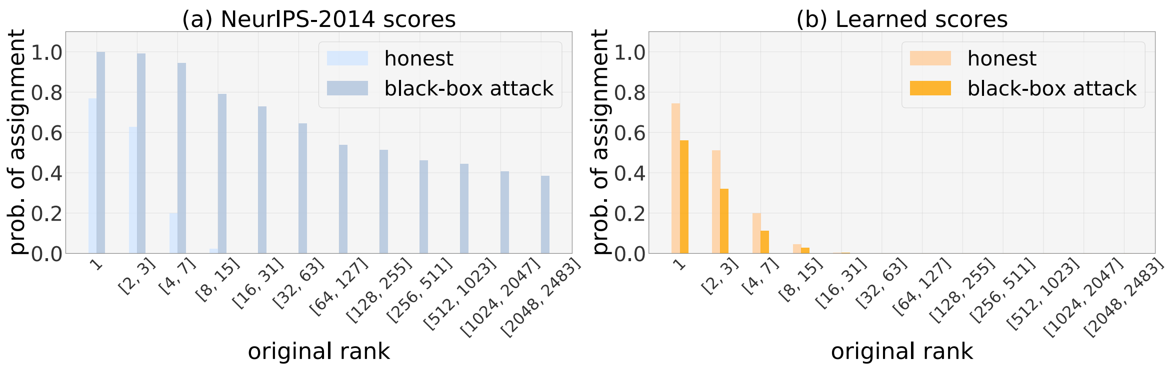

Figure 1. Probability of assigning an adversarial reviewer to the target paper before and after the reviewer executes their black-box bid manipulation attack. See text for details.

(r, p) that reflects the degree of relevance between the reviewer and the paper (Hartvigsen et al., 1999; Goldsmith & Sloan, 2007; Tang et al., 2012; Charlin & Zemel, 2013). The conference organizer can then maximize utility metrics such as the total relevance score whilst maintaining appropriate balance constraints: *i.e.*, there are an adequate number of, say, R reviewers per paper and every reviewer receives a manageable load of at most P papers. This approach gives rise to the following optimization problem:

$$\max_{a\in\{0,1\}^{m\times n}} \quad \sum_{r=1}^{m}\sum_{p=1}^{n} a_{r,p}s_{r,p} \tag{1}$$
$$\text{subject to} \quad \sum_{r=1}^{m} a_{r,p} = R \quad \forall p, \qquad \sum_{p=1}^{n} a_{r,p} \leq P \quad \forall r,$$

where m and n refer to the total number of reviewers and papers, respectively. Eq. (1) is an assignment problem that can be solved using standard techniques such as the Hungarian algorithm (Kuhn, 1955).

The reviewer-paper relevance score, $s_{r,p}$, is critical in obtaining high-quality assignments. Arguably, an ideal relevance score incorporates both the reviewer's *expertise* and *interest* towards the paper (Stent & Ji, 2018). Approaches for measuring expertise include computing the similarity of textural features between reviewers and papers (Dumais & Nielsen, 1992; Mimno & McCallum, 2007; Charlin & Zemel, 2013) as well as using authorship graphs (Rodriguez & Bollen, 2008; Liu et al., 2014). In addition to these features, paper assignment systems generally consider reviewer interest obtained via self-reported paper bids. For example, the NeurIPS-2014 assignment system (Lawrence, 2014) uses a formula for $s_{r,p}$ that incorporates the reviewer's and paper's subject area, TPMS score (Charlin & Zemel, 2013), and the reviewer's bid. Each reviewer may bid on a paper as *none*, *in a pinch*, *willing*, or *eager*[1] to express their preference. The *none* option is the default bid when a reviewer did not enter a bid.

Bid manipulation attacks. Although incorporating reviewer interest via self-reported bids is beneficial to the overall assignment quality, it also allows a malicious reviewer to bid *eager* on a paper that is outside their area of expertise, with the sole purpose of influencing the acceptance decision of a paper that was authored by a "friend" or a "rival". If a single bid has too much influence on the overall assignment, such bid manipulation attacks may be effective and jeopardize the integrity of the review process.

We demonstrate the feasibility of a simple *black-box* bid manipulation attack against the assignment system in Eq. (1). For a target paper p, the malicious reviewer attacks the assignment system by bidding *eager* for p and *none* for all other papers. We evaluate the effectiveness of the attack by randomly picking 400 papers from our synthetic conference dataset (see Section 5), and determine paper assignments using Eq. (1) (with $R = 3$ and $P = 6$) using relevance scores from the NeurIPS-2014 system (Lawrence, 2014). Fig. 1 (*left*) shows the fraction of adversarial reviewers ($m = 2,483$) that can secure their target paper in the final assignment via the bid manipulation attack. As an attack is easier if a reviewer is already ranked high for a particular paper (*e.g.*, because nobody else bids on this paper, or the subject areas match), we visualize the success rate as a function of rank of the "true" paper-reviewer relevance score. More precisely, we rank all reviewers by their original (pre-manipulation) relevance score $s_{r,p}$ and group them into bins of increasing size.

The light gray bar in each bin reports the assignment success rate if all reviewers bid honestly. In the absence of malicious reviewers, the majority of assignments go to reviewers ranked 1 to 7. However, with malicious bids, *any* reviewer stands a good chance of being assigned the target paper. For instance, the chance of getting a target paper for a reviewer ranked between 16 and 31 increases from 0% to

[1]For simplicity, we exclude the option *not willing* that expresses negative interest.

over 70% when bidding maliciously. Even reviewers with the lowest ranks (2048 and lower) have a 40% chance of being assigned the target paper by just changing their bids. This possibility is especially concerning because it may be much easier for an author to corrupt a non-expert reviewer (*i.e.*, a reviewer with a relatively low rank), simply because there are many more such reviewer candidates.

3. Predicting Relevance Scores

The success of the bid manipulation attack exposes an inherent tension in the assignment process. Assigning papers to a reviewer who has expressed explicit interest helps in eliciting high-quality feedback. However, relying too heavily on individual bids paves the way for misuse by malicious reviewers. To achieve a better trade-off, we propose to use the bids from all reviewers (of which the vast majority are honest) as labels to train a supervised model that *predicts* bids as the similarity score $s_{r,p}$, and all other indicators (*e.g.*, subject area matches, TPMS score (Charlin & Zemel, 2013), and paper title) as features. This indirect use of bids allows the scoring function to capture reviewer preferences but reduces the potential for abuse. Later, we will show that this approach also allows for the development of active defenses against bid manipulation attacks.

Scoring model. Let $X \in \mathbb{R}^{(mn)\times d}$ be a feature matrix consisting of d-dimensional feature vectors for every pair of m reviewers and n papers. Let $\mathcal{Y}$ denote the set of possible bids in numerical form, *e.g.* $\mathcal{Y} = \{0, 1, 2, 3\}$. We define $\mathbf{y} \in \mathcal{Y}^{mn}$ as the label vector containing the numerical bids for all reviewer-paper pairs. We define a ridge regressor that maps reviewer-paper features to corresponding bids, similar to the linear regression model from Charlin & Zemel (2013):

$$\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w}} \|X\mathbf{w} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_2^2. \tag{2}$$

To ensure that no single reviewer has disproportionate influence on the model, we restrict the maximum number of positive bids from a reviewer to be at most $U = 60$ and subsample bids of a reviewer whenever the number of bids exceeds U. In a typical CS conference, most reviewers bid on no more than 60 papers (out of thousands of submissions) (Shah et al., 2018).

The trained model $\mathbf{w}^*$ can predict reviewer interest by computing a score $s_{r,p}$ for a reviewer-paper pair (r, p) as follows:

$$s_{r,p} = X_{r,p}\mathbf{w}^* = X_{r,p}H^{-1}X^\top\mathbf{y}, \tag{3}$$

where $H = X^\top X + \lambda I$ is the ridge Hessian (size $d \times d$) and $X_{r,p}$ is the feature vector for the pair (r, p). These predicted scores can then be used in the assignment algorithm in place of bids. In Appendix B, we validate the prediction accuracy of our model using the average precision-at-k (AP@k) metric.

There is an important advantage to our method: bidding is a laborious and monotonous task, and as mentioned above most reviewers only bid on very limited papers. It is likely that only a partial set of bids is observed among all papers that the reviewer is interested in. The scoring model could fill in missing scores by learning the latent interest from the features of papers and reviewers. Completing the full bidding matrix improves the assignment quality, particularly for papers that received few bids originally.

The choice of regression loss serves an important purpose. Since the bid value (between 0 and 3) reflects the *degree of interest* from a reviewer, the loss should reflect the severity of error when making a wrong prediction. For example, if a reviewer expresses *eager* interest (bid score 3), predicting *no bid* (bid score 0) would incur a much greater loss than predicting *willing* (bid score 2).

Effect against simple black-box attack. Fig. 1 (*right*) shows the effect of the proposed scoring model against the bid manipulation attack from Section 2. The assignment probability for honest bidders (light orange) is similar to that of the NeurIPS-2014 system across different bins of reviewer rank. However, deviations from benign bidding behavior are clearly corrected by the model: in fact, the assignment probability *decreases* after the attack (dark orange). This can be explained by the fact that our approach does not use bids to assign reviewers to papers directly, but instead to learn for what type of papers a reviewer may be suitable. The reviewer is actually well-suited for high ranking submissions, but by only bidding on the target paper (instead of honest bids on similar submissions) the model receives less signal that suggests the reviewer is a match for the target paper.

4. Defending Against Colluding Bid Manipulation Attackers

Although the learning-based approach appears robust against manipulation of bids by one reviewer, attackers may have stronger capabilities. Specifically, an adversary can modify their bids based on knowledge of a friend/rival's submissions or another reviewer's bids. Moreover, adversarial reviewers may *collude* to secure the assignment of a specific paper. We capture such capabilities in a *threat model* that describes our assumptions about the adversary. We design an optimal *white-box attack* in this threat model that drastically improves the adversary's success rate. Both the threat model and the white-box attack are intentionally designed to provide very broad capabilities to the adversary. Next, we design a defense that detects and removes white-box adversaries from the reviewer pool to provide security even under the new threat model.

Threat Model. We make the following assumptions about adversarial reviewers: **1.** The adversary may collude with one or more reviewers to secure a target paper's assignment.[2] If any of the colluding reviewers are assigned the paper in question, the attack is considered successful. Collusion with *any* reviewer is allowed except the top-ranked candidates (based on honest bidding), as this would not be an abuse of the *bidding* process[3]. **2.** The adversary cannot manipulate any training features. We are interested in preventing against the *additional* security risk enabled by the bidding mechanism. An attack that succeeds by manipulating features can also be used against an automated assignment system that does not allow bidding. **3.** The adversary may have full knowledge of the assignment system. **4.** The adversary may have direct access to the features and bids of all other reviewers. **5.** The adversary may be able to arbitrarily manipulate his/her bids and those of anyone in the colluding group.

4.1. White-box Attack

To successfully attack the assignment system under these assumptions, the adversary needs to maximize the predicted relevance score of the target paper for him/herself and/or the other colluding reviewers. This amounts to executing a data poisoning attack (Biggio et al., 2012; Xiao et al., 2015; Mei & Zhu, 2015; Jagielski et al., 2018; Koh et al., 2018) against the regression model that is used to predict scores, aiming to alter the score prediction for a specific paper-reviewer pair.

Non-colluding attack. We first devise an attack that maximizes the malicious reviewer's score $s_{r,p}$ for target paper p in the non-colluding setting. We represent reviewers as $[m] = \{1, \ldots, m\}$ and let

$$\mathcal{Y}_{\text{feas}} = \{\mathbf{y}' \in \mathcal{Y}^n : |\{q : \mathbf{y}'_q > 0\}| \leq U\}$$

denote the feasible set of bidding vectors for a particular reviewer for which the number of positive bids is at most U. Adversary r can change $\mathbf{y}_r$ to the $\mathbf{y}'_r \in \mathcal{Y}_{\text{feas}}$ that maximizes the relevance score:

$$\begin{aligned} s^*_{r,p} &:= \max_{\mathbf{y}'_r \in \mathcal{Y}_{\text{feas}}} X_{r,p} H^{-1} (X_r^\top \mathbf{y}'_r + X_{[m]\setminus\{r\}}^\top \mathbf{y}_{[m]\setminus\{r\}}) \\ &= \max_{\mathbf{y}'_r \in \mathcal{Y}_{\text{feas}}} X_{r,p} H^{-1} (X_r^\top \mathbf{y}'_r - X_r^\top \mathbf{y}_r + X^\top \mathbf{y}). \end{aligned}$$

It is straightforward to see that $s^*_{r,p}$ maximally increases the score prediction for reviewer r:

$$\Delta s^*_{r,p} := s^*_{r,p} - s_{r,p} = \max_{\mathbf{y}'_r \in \mathcal{Y}_{\text{feas}}} X_{r,p} H^{-1} X_r^\top (\mathbf{y}'_r - \mathbf{y}_r). \tag{4}$$

[2]*e.g.* by posting the paper ID in a private chat channel of college alumni or like minded members of the community.

[3]For this reason, our framework is not suitable for preventing the attack in (Vijaykumar, 2020) since collusion likely occurred in the author stage.

Note that Eq. (4) maximizes the inner product between $\mathbf{z} := X_{r,p} H^{-1} X_r^\top$ and $\mathbf{y}'_r - \mathbf{y}_r$. To achieve the maximum, papers q corresponding to the top-U positive values in $\mathbf{z}$ should be assigned $\mathbf{y}_{r,q} = \max \mathcal{Y}$, and the remaining bids are set to 0. This requires the adversary to solve a top-U selection problems, which can be done in $O(d^2 + n(d + \log U))$ (Cormen et al., 2009).

Colluding attack. Adversarial reviewers can collude to more effectively maximize the predicted score for reviewer r. An attack in this setting maximizes over the colluding group, $\mathcal{M}$, and over the bids of every reviewer in $\mathcal{M}$. We note that Eq. (4) is not specific to reviewer r, but that the influence of any reviewer t's bids on score prediction $s_{r,p}$ has the form:

$$\Delta_t s_{r,p} := \max_{\mathbf{y}'_t \in \mathcal{Y}_{\text{feas}}} X_{r,p} H^{-1} X_t^\top (\mathbf{y}'_t - \mathbf{y}_t).$$

Hence, the influence from the members of $\mathcal{M}$ on $s_{r,p}$ are *independent*, which implies the adversaries can adopt a greedy approach. Specifically, M_a colluding adversaries can alter the $(M_a n)$-dimensional training label vector $\mathbf{y}_\mathcal{M}$ to $\mathbf{y}'_\mathcal{M} \in \mathcal{Y}_{\text{feas}}^{M_a}$ to maximize the score prediction for reviewer r via:

$$\begin{aligned} \Delta s^*_{r,p} &= \max_{(\mathcal{M}, \mathbf{y}'_\mathcal{M}) \in \mathcal{P}(r, M_a)} X_{r,p} H^{-1} X_\mathcal{M}^\top (\mathbf{y}'_\mathcal{M} - \mathbf{y}_\mathcal{M}), \\ &= \max_{\mathcal{M} \subseteq [m] : r \in \mathcal{M}, |\mathcal{M}| = M_a} \sum_{t \in \mathcal{M}} \max_{\mathbf{y}'_t \in \mathcal{Y}_{\text{feas}}} X_{r,p} H^{-1} X_t^\top (\mathbf{y}'_t - \mathbf{y}_t) \\ &= \max_{\mathcal{M} \subseteq [m] : r \in \mathcal{M}, |\mathcal{M}| = M_a} \sum_{t \in \mathcal{M}} \Delta_t s_{r,p}, \end{aligned} \tag{5}$$

where $\mathcal{P}(r, M_a)$ denotes the set of possible colluding parties of size M_a and their bids:

$$\mathcal{P}(r, M_a) := \{(\mathcal{M}, \mathbf{y}'_\mathcal{M}) : \mathcal{M} \subseteq [m], r \in \mathcal{M}, |\mathcal{M}| = M_a \text{ and } \mathbf{y}'_\mathcal{M} \in \mathcal{Y}_{\text{feas}}^{M_a}\}.$$

The last line in Eq. (5) can be computed by first evaluating $\Delta_t s_{r,p}$ for every $t \in [m] \setminus \{r\}$, and then greedily selecting the top-$(M_a - 1)$ reviewers to form the colluding party with r. The computational complexity of the resulting attack is $O(d^2 + mn(d + \log U) + m \log M_a))$.

4.2. Active Defense Against Colluding Bid Manipulation Attacks

Both the black-box attack from Section 2 and the white-box attack described above adversarially manipulate paper bids. In contrast to honest reviewers whose bids are strongly correlated with their expertise and subject of interest, attackers provide "surprising" bids that have a large influence on the predictions of the scoring model. This allows us to detect potentially malicious bids using an outlier detection algorithm. Specifically, we make our paper assignment system robust against the colluding bid manipulation attacks by

Algorithm 1 Paper assignment system that is robust against colluding bid manipulation attacks.

1: Predict relevance scores $s_{r,p}$ for all reviewer-paper pairs;
2: Initialize candidate set $C = \{(r,p) : \text{rank}(s_{r,p}) \text{ is at least } K \text{ for paper } p\}$;
3: **for** reviewer-paper pair $(r,p) \in C$ **do**
4: Compute relevance score $s^\dagger_{r,p}$ using Eq. (7)
5: Remove (r,p) from C if $\text{rank}(s^\dagger_{r,p})$ is below K for paper p;
6: **end for**
7: Solve the assignment problem in Eq. (1) using $s_{r,p}$ for pairs in C.

detecting and removing training examples that have a disproportional influence on model predictions. We make the same assumptions about the attacker as in Section 4.1, and, in addition, that they are unaware of our active defense.

To implement this system, we note that given a set of malicious reviewers $\mathcal{M}$, we can re-compute the relevance scores for a reviewer-paper pair (r,p) by removing these reviewers from the training set:

$$\tilde{s}_{r,p} = X_{r,p} H^{-1}_{\mathcal{M}^c} X^\top_{\mathcal{M}^c} \mathbf{y}_{\mathcal{M}^c},$$

where $H_{\mathcal{M}^c} = X^\top_{\mathcal{M}^c} X_{\mathcal{M}^c} + \lambda I$ is the Hessian matrix for data points in the complement of the malicious reviewer set $\mathcal{M}$. We assume that at most M_d reviewers collude to form set $\mathcal{M}$. Intuitively, $\tilde{s}_{r,p}$ reflects the relevance score for the pair (r,p) *as predicted by other reviewers*. Relying on the assumption that the vast majority of reviewers are benign, $\tilde{s}_{r,p}$ is likely close to the unobserved true preferences had r been benign.

Following work on robust regression (Jagielski et al., 2018; Chen et al., 2013; Bhatia et al., 2015), this allows us to compute relevance scores that ignore the most likely malicious reviewers in $\mathcal{M}$ by evaluating:

$$s^\dagger_{r,p} = \min_{\mathcal{M} \subseteq [m] : r \in \mathcal{M}, |\mathcal{M}| = M_d} X_{r,p} H^{-1}_{\mathcal{M}^c} X^\top_{\mathcal{M}^c} \mathbf{y}_{\mathcal{M}^c} \leq \tilde{s}_{r,p}. \tag{6}$$

That is, $s^\dagger_{r,p}$ overestimates the decrease in the predicted relevance score for (r,p) had r been benign. The optimization problem in Eq. (6) is intractable because it searches over $\binom{m-1}{M_d-1} = \Theta(m^{M_d})$ subsets of reviewers, $\mathcal{M}$, and because it inverts a $d \times d$ Hessian for every $\mathcal{M}$. To make optimization tractable, we approximate the Hessian $H^{-1}_{\mathcal{M}^c}$ by H^{-1}, which is accurate for small M_d. This approximation facilitates a greedy search for $\mathcal{M}$ because it allows Eq. (6) to be decomposed:

$$\begin{aligned} s^\dagger_{r,p} &\approx \min_{\mathcal{M} \subseteq [m] : r \in \mathcal{M}, |\mathcal{M}| = M_d} X_{r,p} H^{-1} X^\top_{\mathcal{M}^c} \mathbf{y}_{\mathcal{M}^c} \\ &= X_{r,p} H^{-1} X^\top \mathbf{y} - \\ &\quad \max_{\mathcal{M} \subseteq [m] : t \in \mathcal{M}, |\mathcal{M}| = M_d} \sum_{t \in \mathcal{M}} X_{r,p} H^{-1} X_t \mathbf{y}_t. \end{aligned} \tag{7}$$

Eq. (7) can be computed efficiently by sorting the values of $S = \{X_{r,p} H^{-1} X_t \mathbf{y}_t : t \neq r\}$ and selecting r as well as the top $M_d - 1$ corresponding reviewers in S. The computational complexity of the resulting algorithm is $O(d^2 + mnd + m \log M_d))$ for each pair (r,p).

Assignment algorithm. Efficient approximation for the robust relevance score $s^\dagger_{r,p}$ enables our robust assignment algorithm, which proceeds as follows. We first form the *candidate set* C of reviewer-paper pairs by selecting the top-K reviewers for each paper according to the predicted relevance score $s_{r,p}$. For each pair $(r,p) \in C$, the algorithm marks r as potentially malicious and removes the pair (r,p) from C if r *would not have belonged to the candidate set* using the robust relevance score $s^\dagger_{r,p}$. Since $s^\dagger_{r,p} \leq \tilde{s}_{r,p}$, an M_a-colluding attack is always marked as malicious if $M_a \leq M_d$. After removing every potentially malicious pair from C, the assignment problem in Eq. (1) is solved over the remaining reviewer-paper pairs in the candidate set to produce the final assignment[4]. The resulting assignment algorithm is summarized in Algorithm 1. The algorithm trades off two main goals:

1. Every paper needs to be assigned to a sufficient number of reviewers that have the expertise and willingness to review. Therefore, the approach that removes potentially malicious reviewer candidates needs to have a low false positive rate (FPR).
2. The final assignment should be robust against collusion attacks. Therefore, the approach that filters out potentially malicious reviewers needs to have a high true positive rate (TPR).

This trade-off between FPR and TPR is governed by the hyperparameter M_d. Using a higher value of M_d can provide robustness against larger collusions, but it may also remove many benign reviewers from the candidate set even when insufficient alternative reviewers are available. We perform a detailed study of this trade-off in Section 5.

5. Experiments

We empirically study the efficacy of our robust paper bidding and assignment algorithm. Our experiments show that our assignment algorithm removes a large fraction of malicious reviewers, while still preserving the utility of bids for honest reviewers.

[4]This can be achieved by setting $s_{r,p} = -\infty$ for all $(r,p) \notin C$.

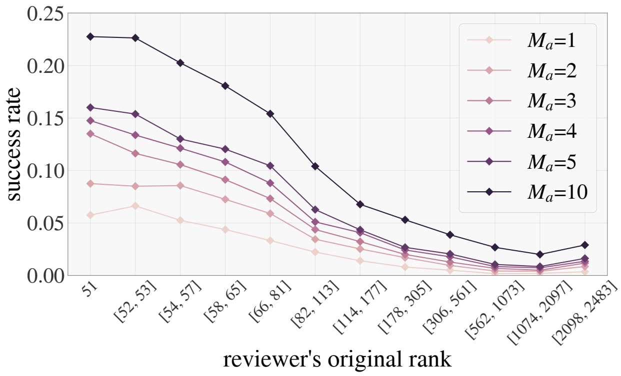

Figure 2. Success rate after the white-box bid manipulation attack against an undefended linear regression scoring model.

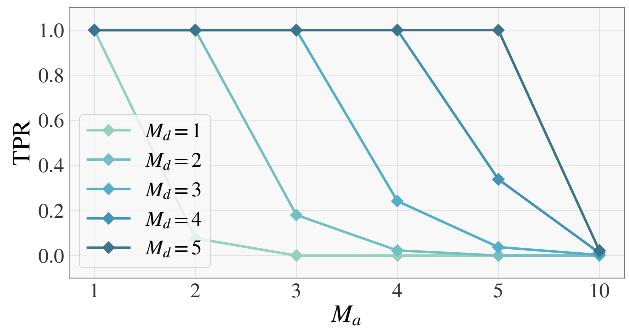

Figure 3. TPR for detecting successful *white-box* attacks using Algorithm 1. For colluding parties of size $M_a \leq M_d$, the detection algorithm has a *near-perfect* TPR. Detection remains viable even when $M_a > M_d$ for moderately high values of M_d.

Dataset. Because real bidding data is not publicly available, we construct a synthetic conference dataset from the Semantic Scholar Open Research Corpus (Ammar et al., 2018). This corpus contains publicly available academic papers annotated with attributes such as citation, venue, and field of study. To simulate a NeurIPS-like conference environment, we collect $n = 2446$ papers published in AI conferences between 2014 and 2015 to serve as submitted papers. We also select $m = 2483$ authors to serve as reviewers, and generate bids based on paper citations. Generated bids are selected from the set $\mathcal{Y} = \{0, 1, 2, 3\}$, corresponding to the bids *none*, *in a pinch*, *willing*, and *eager*.

We generated bids in such a way as to mimic bidding statistics from a recent, major AI conference. Our paper and reviewer features include paper/reviewer subject area, paper title, and a TPMS-like similarity score. We refer to the appendix for more details on our synthetic dataset. For full reproducibility we release our code[5] and synthetic data[6] publicly and invite program chairs across disciplines to use our approach on their real bidding data.

5.1. Effectiveness of White-Box Attacks

We first show that the white-box attack from Section 4.1 can succeed against our relevance scoring model if detection of malicious reviewers is not used. We perform the white-box attacks as follows:

1. The relevance scoring model is trained to predict scores $s_{r,p}$ for every reviewer-paper pair.
2. We randomly select 400 papers and rank all $m = 2483$ reviewers for these papers based on $s_{r,p}$.
3. We discard the $K = 50$ highest-ranked reviewers as attacker candidates for paper p because high-ranked reviewers need not act maliciously to be assigned.
4. We group the remaining reviewers into bins of exponentially growing size (powers of two), and sample 10 malicious reviewers from each bin without replacement.
5. Each selected reviewer chooses its most suitable M_a colluders and modifies their bids using the attack from Section 4.1, targeting paper p.

Result. We run our assignment algorithm on the maliciously modified bids and evaluate the chance of assignment for reviewer r before and after the attack. Fig. 2 shows the fraction of malicious reviewers that successfully alter the paper assignments and is assigned their target paper. Each line shows the attack success rate with a certain colluding party size of M_a. When bidding honestly, all reviewers are below rank $K = 50$ and have no chance of being assigned. With a colluding party size of $M_a = 10$, a reviewer has a 22% chance of being assigned the target paper at an original rank of 51. At the same rank, the success rate is up to 5% even when no collusion occurs. Increasing the collusion size M_a strictly increases the assignment probability, while attackers starting from a lower original rank have a lower success rate. The latter trend shows that the model provides a limited degree of robustness even without the detection mechanism.

5.2. Effectiveness of the Robust Assignment Algorithm

We evaluate the robust assignment algorithm against successful attacks from Section 4.1.

What percentage of attacks is accurately detected? Fig. 3 shows the true positive rate (TPR) of detecting malicious reviewers as a function of collusion size, M_a (on the x-axis), for different values of the hyperparameter M_d. First, we measure the algorithm against all attacks that succeeded against the undefended scoring model (*cf.* Fig. 2). Fig. 3 shows that when $M_a \leq M_d$, the detection TPR is very close to 100%, which implies *almost all* malicious reviewers are removed in this case. The TPR decreases as

[5] https://github.com/facebookresearch/secure-paper-bidding
[6] https://drive.google.com/drive/folders/1khI9kaPy_8F0GtAzwR-48Jc3rsQmBhfe

Setting	FPR		Assignment Quality				# of under-reviewed
	Top-5	Top-50	Frac. of pos.	Avg. bid score	Avg. TPMS	Avg. max. TPMS	
NeurIPS-2014	–	–	0.990	2.732	0.732	0.737	–
TPMS only	–	–	0.323	0.872	0.949	0.997	–
$M_d = 0$	–	–	0.442	1.200	0.848	0.943	–
$M_d = 1$	0.022	0.259	0.443	1.201	0.849	0.943	0
$M_d = 2$	0.046	0.428	0.442	1.199	0.850	0.944	0
$M_d = 3$	0.069	0.528	0.439	1.191	0.852	0.945	4
$M_d = 4$	0.100	0.600	0.435	1.181	0.855	0.947	7
$M_d = 5$	0.139	0.657	0.433	1.172	0.859	0.950	24

Table 1. FPR and assignment quality after detection using different settings of M_d. A higher value of M_d offers a better protection against large colluding parties (see Fig. 3), but also increases the detection FPR. Nevertheless, assignment quality is minimally impacted even with a high FPR since the majority of false positives have low rank and are unlikely to be assigned to begin with.

the size of the collusion, M_a increases but still provides some protection even when $M_a > M_d$. For instance, when $M_a = 5$ and $M_d = 4$ (darkest blue line), approximately 40% of the successful attacks are detected. Increasing M_d will protect against larger colluding parties at the cost of increasing the false positive rate (FPR), that is, the number of times in which an honest reviewer is mistaken for an adversary. A high FPR can negatively impact the quality of the assignments.

What is the quality of the final assignments? To study the effect of false positives from detection on the final paper assignments, we also evaluate assignment quality in terms of *fraction of positive bids*, *average bid score*, *average TPMS*, and *average maximum TPMS* (*i.e.*, maximum TPMS score among assigned reviewers for each paper averaged over all papers). Higher values of these metrics indicate a higher assignment quality. The first row in Table 1 shows the assignment quality when using the NeurIPS-2014 (Lawrence, 2014) relevance scores. As expected, it over-emphasizes positive bids, which constitutes its inherent vulnerability. The second line shows the assignment quality when using only the TPMS score, which serves as a baseline for evaluating how much utility from bids is our robust assignment framework preserving. In contrast, using TPMS scores over-emphasizes average TPMS and average maximum TPMS.

The third line shows our assignment algorithm using the linear regression model without malicious reviewer detection ($M_d = 0$). As it fills in the initially sparse bidding matrix, it has significantly more papers to choose from and yields assignments with fewer positive bids — however the assignment quality is *substantially higher* in terms of TPMS metrics compared to when using NeurIPS-2014 scores. The regression model offers a practical trade-off between relying on bids that reflect reviewer preference and relying on factors related to expertise (such as TPMS).

The remaining rows report results for the robust assignment algorithm with increasing values of M_d. As expected, detection FPR increases as M_d increases, but only has a limited effect on the assignment quality metrics. The main reason for this is that most false positives are low-ranked reviewers, who are unlikely to be assigned the paper even if they were not excluded from the candidate set. Indeed, detection FPR is significantly lower for top-5 reviewers (second column) compared to that of top-50 reviewers (third column). Overall, our results show that the assignment quality is hardly impacted by the detection mechanism.

We observed that a small number of papers were not assigned sufficient reviewers because the detection removed too many reviewers from the set of candidate reviewers for those papers. We report this number in the last column (# of under-reviewed) of Table 1. Although this is certainly a shortcoming of the robust assignment algorithm, the number of papers with insufficient candidates is small enough that it is still practical for conference organizers to assign them manually.

Comparison with robust regression. One effective defense against label-poisoning attacks for linear regression is the TRIM algorithm (Jagielski et al., 2018), which fits the model on a subset of the points that incur the least loss. The algorithm assumes that L out of the mn training points are poisoned and optimize:

$$\begin{aligned} \min_{\mathbf{w},\mathcal{I}} \quad & \|X^{\mathcal{I}}\mathbf{w} - \mathbf{y}^{\mathcal{I}}\|_2^2 + \lambda\|\mathbf{w}\|_2^2 \\ \text{s.t.} \quad & \mathcal{I} \subseteq \{1, \ldots, mn\}, |\mathcal{I}| = mn - L, \end{aligned}$$

where $X^{\mathcal{I}}, \mathbf{y}^{\mathcal{I}}$ denote the subset of $mn - L$ training data points selected by the index set $\mathcal{I}$. We apply TRIM to identify the L poisoned pairs (r, p) and remove them from the assignment candidate set. We then proceed to assign the remaining $mn - L$ pairs using Eq. (3).

Table 2 shows the comparison between TRIM and our robust assignment algorithm in terms of assignment quality and

Defense	Assignment Quality				Detection TPR				
	Frac. of pos.	Avg. bid score	Avg. TPMS	Avg. max. TPMS	$\mathbf{M_a = 1}$	$\mathbf{M_a = 2}$	$\mathbf{M_a = 3}$	$\mathbf{M_a = 4}$	$\mathbf{M_a = 5}$
TRIM ($L = 10000$)	0.439	1.19	0.848	0.943	0.201	0.081	0.037	0.035	0.054
TRIM ($L = 30000$)	0.219	0.439	0.816	0.917	0.986	0.966	0.942	0.924	0.919
Algorithm 1 ($M_d = 1$)	0.443	1.201	0.849	0.943	1.000	0.077	0.000	0.000	0.000
Algorithm 1 ($M_d = 5$)	0.433	1.172	0.859	0.950	1.000	1.000	1.000	1.000	1.000

Table 2. Comparison of assignment quality and detection TPR against white-box attack between the TRIM robust regression algorithm and our robust assignment algorithm. See text for details.

detection TPR. The first and third rows correspond to the TRIM algorithm and Algorithm 1 that achieve a comparable assignment quality. Both methods fail to detect colluding attacks with $M_a > 1$, but Algorithm 1 is drastically more effective when $M_a = 1$. The second and fourth rows compare settings of TRIM and Algorithm 1 that achieve a similar detection TPR. Indeed, both have close to 100% detection rate for $M_a = 1, \ldots, 5$. However, the assignment quality for TRIM is much worse, with all quality metrics being lower than using TPMS score alone (*cf.* row 2 in Table 1). Note that TRIM requires a drastic *overestimate* of the number of poisoned data ($L = 30,000$) in order to detect most attack instances, which means that many benign training samples are being misidentified as malicious.

Running time. As described in Section 4.2, our detection algorithm has a computational complexity of $O(d^2 + mnd + m \log M_d)$ for each reviewer-paper pair. In practice, pairs belonging to the same paper can be processed in a batch to re-use intermediate computation, which amounts to an average of 26 seconds per paper. This process can be easily parallelized across papers for efficiency.

5.3. Improved Black-box Attack

The white-box attack from Section 4.1 assumed that the adversary has extensive knowledge about the assignment system and all reviewers' features and bids. In this section, we propose a more realistic *colluding black-box attack*, where the adversary only has access to the features and bids of reviewers in the colluding party. This attack represents a reasonable approximation of what a real world adversary could achieve, and we show that it is potent against the scoring model in Section 3 absent of any detection mechanism. We further show the effectiveness of our detection algorithm against this *colluding black-box attack*.

Colluding black-box attack. The failure of the simple black-box attack from Section 2 is due to the malicious reviewer r bidding positively only on a single paper, instead of also on a group of papers that are similar to p. We alter the attack strategy by giving the largest bid score to $U = 60$ papers p' that are most similar to p (including p itself). In practice, this can be done by comparing the titles and abstracts of p' to the target paper p. We simulate this

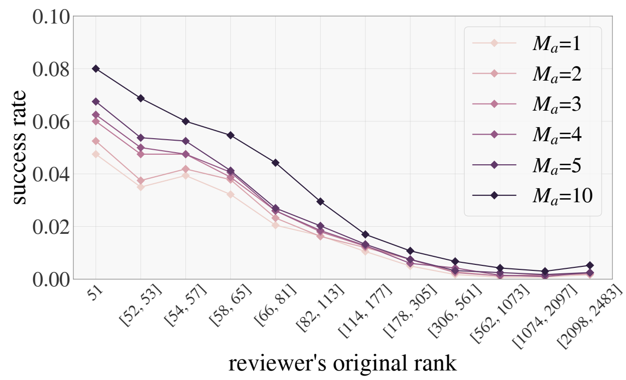

Figure 4. Success rate after the *colluding black-box attack* against an undefended linear regression scoring model.

attack in our experiment by select papers p' whose feature vector $X_{r,p'}$ have a high inner product with $X_{r,p}$.

We can extend this strategy to allow for colluding attacks. The malicious reviewer first selects $M_a - 1$ reviewers with the most similar background to form the colluding group. In simulation, we measure reviewer similarity by the inner product between their respective reviewer-related features. Mimicking r's paper selection strategy, every reviewer r' in the colluding group now gives the largest bid score to the $U = 60$ papers p' with the highest inner product between $X_{r',p'}$ and $X_{r,p}$.

Attack performance. Fig. 4 shows the success rate of the colluding black-box attack against the linear regression model. Note that this attack is much more successful than the *simple black-box attack* from Section 2, which had a success rate of 0% for all reviewers below rank 16. Here, the success rate before attack is initially 0%, which increased to close to 5% after attack even without collusion ($M_a = 1$). Increasing the colluding party size strictly improves attack performance, while attackers with lower initial rank are less successful. Compared to the white-box attack from Section 4.1 (see Fig. 2), the colluding black-box attack is substantially less potent as expected.

Detection performance. For completeness, we evaluate the detection algorithm from Section 4.2 against successful colluding black-box attacks. In Fig. 5, we plot detection TPR as a function of the size of the colluding party (M_a)

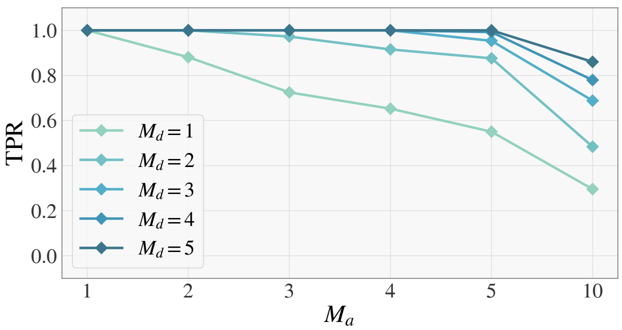

Figure 5. TPR for detecting successful *colluding black-box attacks* using Algorithm 1.

for various choices of the detection parameter M_d. Results show that detection TPR is close to 1 when $M_a \leq M_d$, and remains very high for $M_a > M_d$. For instance, at $M_a = 10$ and $M_d = 5$, detection TPR is above 80% for successful attacks, which is in sharp contrast with the same setting in Fig. 3 for the white-box attack, where TPR is reduced to 0%. The detection performance against this more realistic colluding black-box attack further validates our robust assignment algorithm as a practical countermeasure against bid manipulation.

6. Related Work

Our work fits in a larger body of work on automatic paper assignment systems, which includes studies on the design of relevance scoring functions (Dumais & Nielsen, 1992; Mimno & McCallum, 2007; Rodriguez & Bollen, 2008; Liu et al., 2014) and appropriate quality metrics (Goldsmith & Sloan, 2007; Tang et al., 2012). These studies have contributed to the development of conference management platforms such as EasyChair, HotCRP, and CMT that support most major computer science conferences.

Despite advances in automatic paper assignment, (Rennie, 2016) highlights shortcomings of peer-review systems owing to issues such as prejudices, misunderstandings, and corruption, all of which serve to make the system inefficient. For instance, the standard objective for assignment (say, Eq. (1)) seeks to maximize the total relevance of assigned reviewers for the entire conference, which may be unfair to papers from under-represented areas. This has led to efforts that design objective functions and constraints to promote fairness in the assignment process for all submitted papers (Garg et al., 2010; Long et al., 2013; Stelmakh et al., 2018; Kobren et al., 2019).

Furthermore, the assignment problem faces the additional challenge of coping with the implicit bias of reviewers (Stelmakh et al., 2019). This issue is particularly prevalent when authors of competing submissions participate in the review process, as they have an incentive to provide negative reviews in order to increase the chance of their own paper being accepted (Anderson et al., 2007; Thurner & Hanel, 2011). In order to alleviate this problem, recent studies have devised assignment algorithms that promote impartiality in reviewers (Aziz et al., 2016; Xu et al., 2018). We contribute to this line of work by identifying and removing reviewers who adversarially alter their bids to be assigned papers for which they have adverse incentives.

More recently, Jecmen et al. (2020) studied the bid manipulation problem and considered an orthogonal approach to defending against it. Their method focuses on probabilistic assignment and upper limits the assignment probability for any paper-reviewer pair. As a result, the success rate of a bid manipulation attack is reduced. In contrast, our work seeks to limit the *disproportional influence* of malicious bids rather than uniformly across all paper-reviewer pairs, and further considers the influence of colluding attackers on the assignment system.

7. Conclusion

This study demonstrates some of the risks of paper bidding mechanisms that are commonly utilized in computer-science conferences to assign reviewers to paper submissions. Specifically, we show that bid manipulation attacks may allow adversarial reviewers to review papers written by friends or rivals, even when these papers are outside of their area of expertise. We developed a novel paper assignment system that is robust against such bid manipulation attacks, even in settings when multiple adversaries collude and have in-depth knowledge about the assignment system. Our experiments on a synthetic but realistic dataset of conference papers demonstrate that our assignment system is, indeed, robust against such powerful attacks. At the same time, our system still produces high-quality paper assignments for honest reviewers. Our assignment algorithm is computationally efficient, easy to implement, and should be straightforward to incorporate into modern conference management systems. We hope that our study contributes to a growing body of work aimed at developing techniques that can help improve the fairness, objectivity, and quality of the scientific peer-review process at scale.

Acknowledgements

This research is supported by grants from the National Science Foundation NSF (III-1618134, III- 1526012, IIS-1149882, IIS-1724282, and TRIPODS-1740822, OAC-1934714), the Bill and Melinda Gates Foundation, and the Cornell Center for Materials Research with funding from the NSF MRSEC program (DMR-1719875), and SAP America.

References

Ammar, W., Groeneveld, D., Bhagavatula, C., Beltagy, I., Crawford, M., Downey, D., Dunkelberger, J., Elgohary, A., Feldman, S., Ha, V., et al. Construction of the literature graph in semantic scholar. *arXiv preprint arXiv:1805.02262*, 2018.

Anderson, M. S., Ronning, E. A., De Vries, R., and Martinson, B. C. The perverse effects of competition on scientists' work and relationships. *Science and engineering ethics*, 13(4):437–461, 2007.

Aziz, H., Lev, O., Mattei, N., Rosenschein, J. S., and Walsh, T. Strategyproof peer selection: Mechanisms, analyses, and experiments. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Bhatia, K., Jain, P., and Kar, P. Robust regression via hard thresholding. In *Advances in Neural Information Processing Systems*, pp. 721–729, 2015.

Biggio, B., Nelson, B., and Laskov, P. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

Charlin, L. and Zemel, R. The toronto paper matching system: an automated paper-reviewer assignment system. In *ICML*, 2013.

Chen, Y., Caramanis, C., and Mannor, S. Robust sparse regression under adversarial corruption. In *International Conference on Machine Learning*, pp. 774–782, 2013.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009. ISBN 0262033844.

Dean, J. and Henzinger, M. R. Finding related pages in the world wide web. *Computer networks*, 31(11-16): 1467–1479, 1999.

Dumais, S. T. and Nielsen, J. Automating the assignment of submitted manuscripts to reviewers. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 233–244, 1992.

Garg, N., Kavitha, T., Kumar, A., Mehlhorn, K., and Mestre, J. Assigning papers to referees. *Algorithmica*, 58(1): 119–136, 2010.

Goldsmith, J. and Sloan, R. H. The ai conference paper assignment problem. In *Proc. AAAI Workshop on Preference Handling for Artificial Intelligence, Vancouver*, pp. 53–57, 2007.

Hartvigsen, D., Wei, J. C., and Czuchlewski, R. The conference paper-reviewer assignment problem. *Decision Sciences*, 30(3):865–876, 1999.

Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., and Li, B. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 19–35. IEEE, 2018.

Jecmen, S., Zhang, H., Liu, R., Shah, N. B., Conitzer, V., and Fang, F. Mitigating manipulation in peer review via randomized reviewer assignments. *arXiv preprint arXiv:2006.16437*, 2020.

Kobren, A., Saha, B., and McCallum, A. Paper matching with local fairness constraints. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1247–1257, 2019.

Koh, P. W., Steinhardt, J., and Liang, P. Stronger data poisoning attacks break data sanitization defenses. *arXiv preprint arXiv:1811.00741*, 2018.

Kuhn, H. W. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Lawrence, N. Paper allocation for nips, 2014. https://inverseprobability.com/2014/06/28/paper-allocation-for-nips. [Online; accessed on 2020-10-02].

Littman, M. L. Collusion rings threaten the integrity of computer science research. *Communications of the ACM*, 64(6):43–44, 2021.

Liu, X., Suel, T., and Memon, N. A robust model for paper reviewer assignment. In *Proceedings of the 8th ACM Conference on Recommender systems*, pp. 25–32, 2014.

Long, C., Wong, R. C.-W., Peng, Y., and Ye, L. On good and fair paper-reviewer assignment. In *2013 IEEE 13th International Conference on Data Mining*, pp. 1145–1150. IEEE, 2013.

Mei, S. and Zhu, X. Using machine teaching to identify optimal training-set attacks on machine learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Mimno, D. and McCallum, A. Expertise modeling for matching papers with reviewers. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 500–509, 2007.

Rennie, D. Let's make peer review scientific. *Nature*, 2016.

Rodriguez, M. A. and Bollen, J. An algorithm to determine peer-reviewers. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 319–328, 2008.

Shah, N. B., Tabibian, B., Muandet, K., Guyon, I., and Von Luxburg, U. Design and analysis of the nips 2016 review process. *The Journal of Machine Learning Research*, 19(1):1913–1946, 2018.

Stelmakh, I., Shah, N. B., and Singh, A. Peerreview4all: Fair and accurate reviewer assignment in peer review. *arXiv preprint arXiv:1806.06237*, 2018.

Stelmakh, I., Shah, N., and Singh, A. On testing for biases in peer review. In *Advances in Neural Information Processing Systems*, pp. 5287–5297, 2019.

Stent, A. and Ji, H. A review of reviewer assignment methods, 2018. https://naacl2018.wordpress.com/2018/01/28/a-review-of-reviewer-assignment-methods. [Online; accessed on 2020-10-02].

Tang, W., Tang, J., Lei, T., Tan, C., Gao, B., and Li, T. On optimization of expertise matching with various constraints. *Neurocomputing*, 76(1):71–83, 2012.

Thurner, S. and Hanel, R. Peer-review in a world with rational scientists: Toward selection of the average. *The European Physical Journal B*, 84(4):707–711, 2011.

Vijaykumar, T. N. Potential organized fraud in acm/ieee computer architecture conferences, 2020. https://medium.com/@tnvijayk/potential-organized-fraud-in-acm-ieee-computer-architecture-conferences-ccd61169370d. [Online; accessed on 2020-10-13].

Weinberger, K., Dasgupta, A., Langford, J., Smola, A., and Attenberg, J. Feature hashing for large scale multitask learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 1113–1120, 2009.

Xiao, H., Biggio, B., Nelson, B., Xiao, H., Eckert, C., and Roli, F. Support vector machines under adversarial label contamination. *Neurocomputing*, 160:53–62, 2015.

Xu, Y., Zhao, H., Shi, X., and Shah, N. B. On strategyproof conference peer review. *arXiv preprint arXiv:1806.06266*, 2018.