

An Analysis of Untargeted Poisoning Attack and Defense Methods for Federated Online Learning to Rank Systems

Shuyi Wang

The University of Queensland
Brisbane, QLD, Australia
shuyi.wang@uq.edu.au

Guido Zuccon

The University of Queensland
Brisbane, QLD, Australia
g.zuccon@uq.edu.au

ABSTRACT

Federated online learning to rank (FOLTR) aims to preserve user privacy by not sharing their searchable data and search interactions, while guaranteeing high search effectiveness, especially in contexts where individual users have scarce training data and interactions. For this, FOLTR trains learning to rank models in an online manner – i.e. by exploiting users’ interactions with the search systems (queries, clicks), rather than labels – and federatively – i.e. by not aggregating interaction data in a central server for training purposes, but by training instances of a model on each user device on their own private data, and then sharing the model updates, not the data, across a set of users that have formed the federation. Existing FOLTR methods build upon advances in federated learning.

While federated learning methods have been shown effective at training machine learning models in a distributed way without the need of data sharing, they can be susceptible to attacks that target either the system’s security or its overall effectiveness.

In this paper, we consider attacks on FOLTR systems that aim to compromise their search effectiveness. Within this scope, we experiment with and analyse data and model poisoning attack methods to showcase their impact on FOLTR search effectiveness. We also explore the effectiveness of defense methods designed to counteract attacks on FOLTR systems. We contribute an understanding of the effect of attack and defense methods for FOLTR systems, as well as identifying the key factors influencing their effectiveness.

CCS CONCEPTS

• Information systems → Learning to rank.

KEYWORDS

Online Learning to Rank, Federated Learning, Federated Online Learning to Rank, Model Poisoning, Data Poisoning

ACM Reference Format:

Shuyi Wang and Guido Zuccon. 2023. An Analysis of Untargeted Poisoning Attack and Defense Methods for Federated Online Learning to Rank Systems. In *Proceedings of the 2023 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR ’23)*, July 23, 2023, Taipei, Taiwan. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3578337.3605117>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICTIR ’23, July 23, 2023, Taipei, Taiwan

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0073-6/23/07...\$15.00
<https://doi.org/10.1145/3578337.3605117>

1 INTRODUCTION

In Online Learning to Rank (OLTR), all documents are stored in a server, and users’ queries and interaction data (e.g., clicks) are also collected in the server. The ranker is then trained in a centralised and online manner. However, this setting could potentially infringe on users’ privacy as users may not want to share their queries and interactions. In addition, documents containing personal information, like in email search [14] or desktop search [9], may not be appropriate to surrender to a third party search service. To address this issue, a new paradigm – Federated Online Learning to Rank (FOLTR) – has been explored [13, 29–31]. In FOLTR (as in Figure 1), clients retain their data locally, train a local ranker, and then share the local model weights (or gradients) with the server instead of the raw data. The server plays a very different role – aggregating the received weights in an effective manner (e.g., via federated averaging [18]) and then broadcasting the obtained global ranker to the clients, which in turns use the global ranker to replace their local ranker. The whole process is carried out iteratively. Compared with conventional OLTR, FOLTR provides a mechanism to safeguard users’ privacy. Also, the collaborative training makes the local rankers more effective than if they were trained separately with only the data of each single user.

Existing FOLTR systems however are not necessarily secure: the federation mechanism provides malicious clients with opportunities for attacking the effectiveness of the global ranker. For example, malicious clients can send arbitrary weights to the server so that the convergence of the global ranker can be perturbed after aggregation. This kind of attack is termed as *untargeted poisoning attack* and aim to compromise the integrity of the global model trained federatively [1]. This issue is critical for federated learning systems, but it has not yet been studied for FOLTR. In this work, we initiate the investigation of poisoning attacks and corresponding defense methods in the context of FOLTR systems.

Outside of FOLTR systems, poisoning attacks on federated learning systems has been shown successful in compromising model integrity across several federated machine learning tasks [3, 4, 17, 24, 35], including in natural language processing and recommender systems. To mitigate or remove the threat posed by poisoning attacks, defense strategies have been designed and optimised [5, 11, 34]. Defense strategies typically act upon the aggregation rules used in the global model updating phase. The vulnerability of existing FOLTR methods to these attacks and the effectiveness of the related defense mechanism is unknown. Previous work in FOLTR has shown that findings obtained with respect to federated learning in other areas of Machine Learning or Deep Learning do not directly translate to the online learning to rank context, and therefore the study of

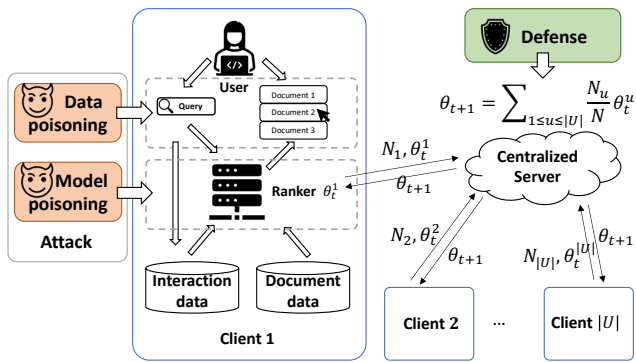


Figure 1: Overview of a FOLTR system with attack and defense modules (the arrows point to where these modules will be applied).

these techniques in the context of FOLTR is important. For example, Wang and Zuccon [31] have found that methods for dealing with non identical and independently distributed data in federated learning systems do not generalise to the context of FOLTR. Therefore, the performance of poisoning attacks and defense methods proposed in general domain cannot be guaranteed when applied to FOLTR: we address this limitation by adapting and investigating these methods to the setting of FOLTR and establish baselines for future studies.

In this paper, we complement the state-of-the-art FOLTR system with one untargeted attack module and one defense module (shown in Figure 1). For the untargeted attack module, we implement a *data poisoning* method that compromises the local training data to affect the trained model, and two *model poisoning* methods that directly corrupt the local model updates. As for the defense module, we implement four Byzantine-robust aggregation rules to safeguard against such attacks. These defense mechanisms rely on statistical techniques to identify outliers among the received weights and subsequently exclude them during the aggregation process. Through extensive empirical experiments, we (1) investigate the vulnerability of FOLTR systems to untargeted poisoning attacks, and show under which conditions poisoning attacks can represent a real threat to FOLTR systems; and (2) demonstrate the effectiveness of defense strategies, and importantly reveal the presence of issues with defense strategies if applied to FOLTR systems for which an attack is not in place.

2 RELATED WORK

2.1 Federated OLTR

Unlike traditional Learning to Rank (LTR), Online Learning to Rank (OLTR) optimizes rankers through implicit user feedback (e.g., clicks) to directly influence search engine result pages in real-time production. The earliest method, Dueling Bandit Gradient Descent (DBGD) [36], uniformly samples variations of the ranking model and updates the ranker based on online interleaving evaluation. To mitigate the high variance and regret inherent in DBGD, subsequent methods have improved it through techniques like multiple interleaving [19, 23], projected gradient [28], and counterfactual evaluation [37]. In contrast to DBGD-based approaches, Pairwise Differentiable Gradient Descent (PDGD) [20] utilizes a

Plackett-Luce model to sample the ranking list and estimates gradients from inferred pairwise preferences. This method has been found to exhibit greater resilience to noise and higher effectiveness in optimizing neural models.

OLTR methods have been thoroughly investigated in a centralized setting, where a central server possesses the data to be searched and gathers users' search interactions, such as queries and clicks. The training of the ranker also takes place on this server. However, this centralized paradigm is not well-suited for privacy-preserving requirement where each client may not wish to, or cannot, share the searchable data, queries and other interactions. This is the case, for example, of hospitals wanting to collaborate together to create powerful rankers to identify the cohort of patients for specific rare conditions (and as such, each hospital only holds limited data that would not be sufficient to train an effective ranker individually), but that by legislation they are forbidden to share the actual data.

To handle this issue, Federated Online Learning to Rank (FOLTR) methods have been proposed. These methods consider a decentralized machine learning scenario where data owners (clients) collaboratively train the model without sharing their data under the coordination of a central server. One such method is the Federated OLTR with Evolutionary Strategies (FOLTR-ES) [13], which extends the OLTR optimization scenario to the Federated SGD [18] and utilizes Evolution Strategies as optimization method [22]. While FOLTR-ES performs well on small-scale datasets under certain evaluation metrics, its effectiveness does not generalise to large-scale datasets and standard OLTR metrics [30]. Because of this, we do not consider FOLTR-ES in our study. An alternative method is the FPDGD [29], which builds upon the state-of-the-art OLTR method, the Pairwise Differentiable Gradient Descent (PDGD) [20], and integrates it into the Federated Averaging (FedAvg) framework [18]. FPDGD exhibits effectiveness comparable to centralized OLTR methods, representing the current state-of-the-art FOLTR method. Thus, our empirical investigation of attack and defense methods on FOLTR systems relies on the FPDGD method, which is further described in Section 3.

2.2 Poisoning Attacks on Federated Learning

Poisoning attacks on federated learning systems aim to compromise the integrity of the system's global model. Poisoning attacks can be grouped according to the goals of the attack into two categories: untargeted poisoning attacks, and targeted poisoning attacks (also known as backdoor attacks).

Targeted poisoning attacks aim to manipulate a global model according to the attacker's objectives, such as misclassifying a group of data with certain features to a label chosen by the attacker, while maintaining normal model effectiveness under other conditions. This is accomplished through backdoor attacks [1, 3], which are designed to allow the targeted manipulations to transpire stealthily and without detection.

In contrast, untargeted poisoning attacks (also known as Byzantine failures [5, 10, 15, 25, 34]) aim to decrease the overall effectiveness of the global model indiscriminately for all users and data groups. Current untargeted poisoning methods can be divided into two categories: data poisoning and model poisoning. Label flipping [4] is a representative data poisoning method: the labels of honest training data are changed without altering their features.

Model poisoning, on the other hand, directly affects the local model updates before they are sent to the centralized server. For example, Baruch et al. [2] poisons the local model updates through the addition of noise computed from the variance between the before-attack model updates, while Fang et al. [10]'s attacks are optimized to undermine specific robust aggregation rules.

Among untargeted poisoning attacks on federated learning systems, model poisoning methods have been found to be the most successful [1]. In particular, data poisoning attacks have limited success when Byzantine-robust defense aggregation rules are in use [10]; we introduce these defense methods in Section 5. Furthermore, most data poisoning attacks assume that the attacker has prior knowledge about the entire training dataset, which is often unrealistic in practice.

In this paper, we focus on untargeted poisoning attacks, delving into the effectiveness of both data poisoning and model poisoning methods. These attack methods are studied within the framework of a FOLTR system based on FPDGD, with and without the integration of defense countermeasures.

3 PRELIMINARIES

3.1 Online Learning to Rank (OLTR)

In OLTR, the ranker is learned directly from user interactions (clicks in our study), rather than editorial labels. In this context, each client performs searches on several queries during each local training phase. For each query q , the candidate documents set is D_q and the local training data held by each client is $\{(x_i, c_i), i = 1 \dots |D_q|\}$ q with feature representation (x_i) and user's click signal (c_i) for each (q, d_i)-pair (where $d_i \in D_q$). The value of the click feedback c_i is either 0 (unclicked) or 1 (clicked). In practice, the *click* is dependent on the relevance degree of the candidate document d_i to the query q , the rank position of d_i , and other noise or randomness factors.

3.2 Federated Pairwise Differentiable Gradient Descent (FPDGD)

We add our attacking and defense modules to the current state-of-the-art FOLTR system, the Federated Pairwise Differentiable Gradient Descent (FPDGD) [29], which is outlined in Algorithm 1. Within each iteration t , each client u considers N_u interactions and updates the local ranker using Pairwise Differentiable Gradient Descent (PDGD) [20]. After the local update is finished, each client sends the trained weights θ_t^u to the server. The server then leverages the widely-used Federated Averaging [18] to aggregate the local model updates. Afterwards, the new global weights θ_{t+1} are sent back to the clients as their new local rankers. We refer the reader to the original FPDGD paper for more details [29].

4 ATTACKS TO FOLTR SYSTEMS

4.1 Problem Definition and Threat Model

Attacker's capability: Poisoning attacks can come from both members (insiders) and non-members (outsiders) of the FOLTR system. Insiders include both the central server and the clients, while outsiders include eavesdroppers on communication channels and users of the final ranker (this is similar to adversarial attacks during inference). In this study, we focus on insider attacks by malicious

Algorithm 1 FederatedAveraging PDGD.

- set of clients participating training: U , each client is indexed by u ;
- local interaction set: B , number of local interactions: N_u .

Server executes:

- 1: initialize θ_0 ; scoring function: f ; learning rate: η
- 2: **for** each round $t = 1, 2, \dots$ **do**
- 3: **for** each client $u \in U$ **in parallel do**
- 4: $\theta_t^u, N_u \leftarrow \text{ClientUpdate}(u, \theta_t)$
- 5: $\theta_{t+1} \leftarrow \sum_{u=1}^{|U|} \frac{N_u}{\sum_{u=1}^{|U|} N_u} \theta_t^u$

ClientUpdate(u, θ): // Run on client u

- 1: **for** each local update i from 1 to N_u **do**
 - 2: $\theta \leftarrow \theta + \eta \nabla f_\theta$ //PDGD update with data from B
 - 3: **return** (θ, N_u) to server
-

participants in the FOLTR system since insider attacks are generally more effective than outsider attacks [17].

We assume the attacker has control over m collusive clients, which means that the training data and local model updates can be exchanged among the malicious clients. We restrict the percentage of collusive clients to less than 50%: higher amounts would make it trivial to manipulate the global model.

Attacker's background knowledge: We assume that the attacker has only access to the compromised clients: the training data and local rankers of all remaining clients remain not accessible to the attacker. Thus, the attacker has limited prior knowledge: the training data and the locally updated models from the poisoned clients, and the shared global model. The exception of having full prior knowledge¹ will only be for the purpose of analysis and will be clarified in place.

Problem Formulation: Assume n clients are involved in the FOLTR system. Among them, m clients are malicious. Without loss of generality, we assume the first m participants are compromised. Be \mathbf{w}_i the local model that the i -th client sends to the central server. The global ranking model is updated through aggregating all \mathbf{w}_i :

$$\mathbf{w}_g = \text{agg}(\mathbf{w}_1, \dots, \mathbf{w}_m, \mathbf{w}_{m+1}, \dots, \mathbf{w}_n) \quad (1)$$

4.2 Data Poisoning

Data poisoning methods aim to corrupt the training data in order to degrade the model's effectiveness. This can be done by adding malicious instances or altering existing instances in an adversarial manner.

Our data poisoning attack to FOLTR is inspired by the label flipping strategy [4, 27], in which the labels of honest training samples from one class are flipped to another class, while the features of the flipped samples are kept unchanged. In our case, we want to change the label of irrelevant documents into "high-relevant" and vice versa, without any changes to the feature representation of the corresponding query-document pairs. To achieve so, the attacker needs to intentionally flip the feedback by clicking on irrelevant documents to bring arbitrary noise thus poison the training.

In our experiments, as no click data is available with the considered datasets, we follow the common practice from previous

¹i.e. the attacker can also access information (training data, model updates) of non-poisoned clients.

Table 1: Instantiations of SDBN click model for simulating user behaviour in experiments. $rel(d)$ denotes the relevance label for document d . Note that in the MQ2007 dataset, only three-levels of relevance are used. We demonstrate the values for MQ2007 in bracket.

$rel(d)$	$P(\text{click} = 1 \mid rel(d))$				
	0	1	2	3	4
<i>perfect</i>	0.0 (0.0)	0.2 (0.5)	0.4 (1.0)	0.8 (-)	1.0 (-)
<i>navigational</i>	0.05 (0.05)	0.3 (0.5)	0.5 (0.95)	0.7 (-)	0.95 (-)
<i>informational</i>	0.4 (0.4)	0.6 (0.7)	0.7 (0.9)	0.8 (-)	0.9 (-)
<i>poison</i>	1.0 (1.0)	0.8 (0.5)	0.4 (0.0)	0.2 (-)	0.0 (-)

$rel(d)$	$P(\text{stop} = 1 \mid \text{click} = 1, rel(d))$				
	0	1	2	3	4
<i>perfect</i>	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (-)	0.0 (-)
<i>navigational</i>	0.2 (0.2)	0.3 (0.5)	0.5 (0.9)	0.7 (-)	0.9 (-)
<i>informational</i>	0.1 (0.1)	0.2 (0.3)	0.3 (0.5)	0.4 (-)	0.5 (-)
<i>poison</i>	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (-)	0.0 (-)

literature in OLTR and FOLTR [20, 29, 31] of simulating click behaviour based on the extensively-used *Simplified Dynamic Bayesian Network* (SDBN) click model [8]. This click model has been shown to produce reasonable predictions of real-world user click behaviour. Under SDBN, users examine a search engine result page (SERP) from top to bottom. Each document is inspected and clicked with click probability $P(\text{click} = 1 \mid rel(d))$, conditioned on the actual relevance label $rel(d)$ of the document. After a document is clicked, the user decides to stop the search session with stopping probability $P(\text{stop} = 1 \mid \text{click} = 1, rel(d))$, or continue otherwise. Commonly, three instantiations of SDBN are considered: (1) a *perfect* user examines every document and clicks on all relevant documents thus provides very reliable feedback, (2) a *navigational* user searches for reasonably relevant documents with a higher probability to stop searching after one click, (3) an *informational* user typically clicks on many documents without a specific information preference thus provides the noisiest click feedback.

Inspired by the three instantiations, we manipulate one *poison* instantiation to simulate malicious clicking behaviour. The click probability of *poison* instantiation is the reverse version of the *perfect* click behaviour: the highest probability of clicking is associated with the least relevance label. All stop probabilities in *poison* instantiation are set to zero as we assume the attacker wants to poison as many clicks as possible. The values we adopt for the four instantiations of SDBN are reported in Table 1.

4.3 Model Poisoning

Unlike data poisoning, model poisoning directly modifies the local model updates (through poisoning gradients or model parameter updates) before sending them to the server. Some literature shows that model poisoning is more effective than data poisoning [1, 3] while it also requires sophisticated technical capabilities and high computational resources than solely poisoning data. In this section, we investigate two existing model poisoning methods.

4.3.1 Little Is Enough (LIE). Baruch et al. [2] find that if the variance between local updates is sufficiently high, the attacks can make use of this by adding small amounts of noise to the compromised local models and bypass the detection of defense methods. They provide a perturbation range in which the attackers

can successfully poison the learning process. To conduct the attack, the adversaries first compute the average μ and standard deviation σ of the before-attack benign local model updates of all collusive attackers ($\mathbf{w}_1, \dots, \mathbf{w}_m$). A coefficient z is used and computed based on the number of benign and malicious clients. Finally, the local model of attackers is manipulated as $\mathbf{w}_i^m = \mu - z\sigma$ for $i \in \{1, \dots, m\}$ and sent to the central server who aggregates updates from all participants under certain rules in Equation 1. Baruch et al. [2] observe that, for image classification tasks, the small noises sufficiently compromise the global model while being sufficiently small to evade detection from defense strategies.

4.3.2 Fang’s Attack. Fang et al. [10] proposed an optimization-based model poisoning attack tailored to specific robust aggregation rules (Krum, Multi-Krum, Trimmed Mean and Median), as will be explained in Section 5.

Fang’s attack is conducted separately under two assumptions: (1) full knowledge, and (2) partial knowledge. Under full knowledge, the attacker has full access to local model updates of all benign clients. This is a strong and impractical assumption, and it is often not the case in real attacks on federated learning systems. In the partial knowledge scenario, the attacker only knows the local training data and models of the compromised clients.

In their attack to the robust aggregation rules Krum and Multi-Krum, the attacker computes the average μ of the benign updates in their possession, computes a perturbation $\mathbf{s} = -\text{sign}(\mu - \mathbf{w}_g)$, and finally computes a malicious update as $\mathbf{w}_i^m = (\mathbf{w}_g + \lambda \cdot \mathbf{s})$ by solving for the coefficient λ , where \mathbf{w}_g is the before-attack global model during each federated training step. Thus, under the full knowledge assumption, the average μ and perturbation signal \mathbf{s} are computed based on all benign updates ($\mathbf{w}_1, \dots, \mathbf{w}_m, \mathbf{w}_{m+1}, \dots, \mathbf{w}_n$). For updates of the malicious clients ($\mathbf{w}_1, \dots, \mathbf{w}_m$), the before-attack benign updates are leveraged. Under the partial knowledge scenario, only the before-attack benign updates ($\mathbf{w}_1, \dots, \mathbf{w}_m$) are used to estimate the real values for average μ and the reversed deviation vector \mathbf{s} .

When attacking Trimmed Mean and Median, the goal is to craft the compromised local models based on the maximum $w_{max,j}$ or minimum $w_{min,j}$ benign parameters for each dimension j of the local model (this is one of the key features used by Trimmed Mean and Median for defending). The choice of $w_{max,j}$ or $w_{min,j}$ depends on which one deviates the global model towards the inverse of its update direction without attacks. Similar to when attacking Krum, the reversed deviation vector \mathbf{s} is computed with full knowledge or estimated under partial knowledge with only before-attack updates from all attackers, so as the estimation of $w_{max,j}$ and $w_{min,j}$. After getting the j -th value of vector \mathbf{s} , the j -th dimension of the compromised local model is randomly sampled from the range built based on $w_{max,j}$ (if $s_j = 1$) or $w_{min,j}$ (if $s_j = -1$).

5 DEFENSE FOR FOLTR SYSTEMS

The current state-of-the-art defense methods against untargeted poisoning attacks focus on enhancing the robustness of the aggregation rules (Equation 1) used during the global update phase, to counteract attempts by malicious clients to corrupt the training.

Next, we describe four robust aggregation rules that have been shown effective in general federated learning, but have not been evaluated for FOLTR.

5.1 Krum and Multi-Krum

The intuition behind the Krum method for robust aggregation [5] is that the malicious local model updates need to be far from the benign ones in order for the success of poisoning the global model. To evaluate how far a model update \mathbf{w}_i is from the others, Krum computes the Euclidean distances between \mathbf{w}_i and \mathbf{w}_j for $i \neq j$. We denote $i \rightarrow j$ if \mathbf{w}_j belongs to the set of $n - m - 2$ closest local models of \mathbf{w}_i . Then the sum of $n - m - 2$ shortest distances to \mathbf{w}_i is computed and denoted as $s(i) = \sum_{i \rightarrow j} \text{Euc_dist}(\mathbf{w}_i, \mathbf{w}_j)$. After computing the distance score $s(i)$ for all local updates, Krum selects the local model with the smallest $s(i)$ as the global model \mathbf{w}_g :

$$\mathbf{w}_g = \text{Krum}(\mathbf{w}_1, \dots, \mathbf{w}_m, \mathbf{w}_{m+1}, \dots, \mathbf{w}_n) = \arg \min_{\mathbf{w}_i} s(i) \quad (2)$$

Multi-Krum is a variation of the Krum method. Multi-Krum, like Krum, calculates the distance score $s(i)$ for each \mathbf{w}_i . However, instead of choosing the local model with the lowest distance score as the global model (as Krum does), Multi-Krum selects the top f local models with the lowest scores and computes the average of these f models (\mathbf{w}'_i , where $i \in \{1, \dots, f\}$) to be the global model.

$$\mathbf{w}_g = \text{Multi-Krum}(\mathbf{w}_1, \dots, \mathbf{w}_m, \mathbf{w}_{m+1}, \dots, \mathbf{w}_n) = \frac{1}{f} \sum_{i=1}^f \mathbf{w}'_i \quad (3)$$

In our empirical investigation, we set the Multi-Krum parameter $f = n - m$, as in previous work [5].

5.2 Trimmed Mean and Median

Assume that w_{ij} is the j -th parameter of the i -th local model. For each j -th model parameter, the Trimmed Mean method [34] aggregates them separately across all local models. After removing the β largest and smallest among w_{1j}, \dots, w_{nj} , the Trimmed Mean method computes the mean of the remaining $n - 2\beta$ parameters as the j -th parameter of the global model. We denote $U_j = \{w_{1j}, \dots, w_{(n-2\beta)j}\}$ as the subset of $\{w_{1j}, \dots, w_{nj}\}$ obtained by removing the largest and smallest β fraction of its elements. That is, the j -th parameter of the global model updated by Trimmed Mean is:

$$w_j = \text{Trimmed Mean}(w_{1j}, \dots, w_{nj}) = \frac{1}{n - 2\beta} \sum_{w_{ij} \in U_j} w_{ij} \quad (4)$$

In our implementation, as in previous work on general federated learning [10, 25, 26], we set β to be the number of compromised clients m .

The Median method, like the Trimmed Mean method, sorts the j -th parameter of n local models. Instead of discarding the β largest and smallest values (as in Trimmed Mean), the Median uses the median of w_{1j}, \dots, w_{nj} as the j -th parameter of the global model:

$$w_j = \text{Median}(w_{1j}, \dots, w_{nj}) \quad (5)$$

In case n is an even number, the median is calculated as the average of the middle two values.

6 EXPERIMENTAL SETUP

We next describe our experimental setup to evaluate the considered attack and defense mechanism in the context of a FOLTR system.

Datasets. Our experiments are performed on four commonly-used LTR datasets: MQ2007 [21], MSLR-WEB10k [21], Yahoo [7], and Istella-S [16]. Each dataset consists of a set of queries and the corresponding pre-selected candidate documents for each query.

Each query-document pair is represented by a multi-dimensional feature vector, and have a corresponding annotated relevance label. Among the selected four datasets, MQ2007 [21] is the smallest with 1,700 queries, 46-dimensional feature vectors, and 3-level relevance assessments (from *not relevant* (0) to *very relevant* (2)). The other three datasets are larger, more recent, and provided by commercial search engines. MSLR-WEB10k has 10,000 queries and each query is associated with 125 documents on average, each query-represented with 136 features. Yahoo has 29,900 queries and each query-document pair has 700 features. Istella-S is the largest, with 33,018 queries, 220 features, and an average of 103 documents per query. These three commercial datasets are all annotated for relevance on a five-grade-scale: from *not relevant* (0) to *perfectly relevant* (4).

Federated setup. We consider 10 participants ($n = 10$) in our experiments, among which m clients are attackers. This setup is representative of a cross-silo FOLTR system, typical of a federation of a few institutions or organisations, e.g. hospitals creating a ranker for cohort identification from electronic health records [12]. In this paper we will not consider the setup of a cross-device FOLTR system, where many clients are involved in the federation: this is representative of a web-scale federation.

We assume that the malicious clients can collude with each other to exchange their local data and model updates to enhance the impact of attacks. In the federated setting, each client holds a copy of the current ranker and updates the local ranker through issuing $N_u = 5$ queries along with the respective interactions. The attackers can only compromise the local updating phase through poisoning the training data or model updates of the controlled malicious clients. After the local updating finishes, the central server will receive the updated ranker from each client and aggregate all local messages to update the global ranker. In our experiments, we consider the following aggregation rules: (1) FedAvg, (2) other robust aggregation rules introduced in Section 5. Unless otherwise specified, we train the global ranker through $T = 10,000$ global updating times.

User simulations. We follow the standard setup for user simulations in OLTR [20, 29, 31, 37]. We randomly sample from the set of queries in the static dataset to determine the query that the user issues each time. After that, the pre-selected documents for the query are ranked by the current local ranking model to generate a ranking result. For every query, we limit the SERP to 10 documents. User interactions (clicks) on the displayed ranking list are simulated through the SDBN click models introduced in Sec 4.2. For the user simulation in model poisoning, we simulate three types of users using the three click instantiations: perfect, navigational, and informational. We experiment on the three types of users separately in order to show the impact of attacking on different types of users. For data poisoning, we simulate the poisoned click based on the *poison* click combining with benign users on the aforementioned three types of click models separately to show the impact of our data poisoning strategies on different types of benign clicks.

Ranking models. We experiment on a linear and neural model as the ranking model when training with FPDGD. For the linear model, we set the learning rate $\eta = 0.1$ and zero initialization was used. As in the original PDGD and FPDGD studies [20, 29], the neural ranker is optimized using a single hidden-layer neural network with 64 hidden nodes, along with $\eta = 0.1$.

Evaluation. We evaluate the attack methods by comparing the gap in offline performance obtained when a specific attack is performed and when no attack is performed. The higher the performance degradation, the more effective the attack.

As we limit each SERP to 10 documents, we use $nDCG@10$ for offline evaluation. The offline performance is measured through averaging the $nDCG$ scores of the global ranker over the queries in the held-out test dataset with the actual relevance label. We record the offline $nDCG@10$ score of the global ranker during each federated training update.

7 RESULTS FOR DATA POISONING

We perform data poisoning attack and four defense methods across different settings of user behaviours (i.e. click models) and number of attackers ($\{10\%, 20\%, 30\%, 40\%\}$). Results on MSLR10k with a linear ranker are shown as solid lines in Figure 2 – results for other datasets are similar and omitted for space constraints.

7.1 Attacks

In the plots of Figure 2, the solid lines represent the results of data poisoning when no defense method is deployed. Among them, the black line represents no attacking situation (“honest” baseline). We can observe that the effect of data poisoning depends on the settings of user behaviors (i.e. click models) and the number of attackers.

Effect of number of attackers. By comparing the solid curves in each plot of Figure 2, we can observe that the overall performance of the FOLTR system decreases as the number of attackers increases, compared to the “honest” baseline. Thus, the higher the number of attackers, the more degradation on the FOLTR system is experienced.

Ease of attack under different user behaviours. By comparing the plots within each row, we see the effect of data poisoning is different under different user behaviours. In the navigational and informational settings, attacks carried by as little as 20% of clients can significantly affect the system. However, to successfully attack the perfect click, a higher number of malicious clients is needed. Across all datasets, the informational click model is the most affected by attacks, while the perfect click model only experiences considerable losses when a large number of clients has been compromised.

Neural ranker vs. linear ranker. The findings from results for the neural ranker under data poisoning attack are similar to those for the linear ranker – and this pattern is valid across all remaining experiments we report. Therefore, we only report experiments using the linear ranker due to limited space.

7.2 Defense

Next, we demonstrate the effectiveness of our four defense mechanisms against data poisoning attack. The results on MSLR10k are shown by the dashed curves in Figure 2. Each row corresponds to one defense method.

Krum. Overall, Krum performs well across all datasets and for all three types of click models once the percentage of malicious clients reaches 20% or higher, with the exception of MQ2007. However, Krum does not work when defending against 10% of clients, except for Istella-S. The accuracy drop from deploying Krum (as shown in Section 9) outweighs its effectiveness in defense, especially when

there is a relatively small impact on the effectiveness of the model, as is in the case when 10% of the clients are malicious. Additionally, Krum does not show any improvement in defending certain scenarios under the informational click model, such as for MQ2007 under all percentages of malicious clients, and for MSLR10k when 40% of clients are malicious.

Multi-Krum. The results obtained for Multi-Krum show similar effectiveness on the perfect click model as Krum. It is important to note that the perfect click model is the hardest to attack among the three types of click models considered. Multi-Krum provides slightly better defense performance on navigational clicks compared to Krum, especially when there are fewer attackers (30% or less). However, for the informational click model, Multi-Krum does not perform as well as Krum. This is because the variance of the local model updates is relatively higher in the noisier informational click model. After averaging the selected local models, the advantage of Multi-Krum is reduced, especially when there are more than 30% malicious clients.

Trimmed Mean. Across all experiments, Trimmed Mean does not perform well on the noisiest click model (informational) when there are more than 30% malicious clients involved. When the malicious clients are 20% or 30%, Trimmed Mean provides lower performance gains compared to Krum, but it performs similarly to Krum when only 10% of the clients are malicious.

Median. Like Trimmed Mean, Median does not provide improved performance on the noisy informational click model when 30% or 40% of clients are malicious. Similarly, and like other robust aggregation rules, Median does not show significant improvements when only 10% of clients are malicious. In fact, the Median’s performance even decreases on the navigational click model for MSLR10k with 10% of malicious clients. When the malicious clients are 20% and 30% of all clients in the federation, the performance gain provided by Median is similar to that of Trimmed Mean.

Summary. Overall, Krum and Multi-Krum work better than Trimmed Mean or Median when defending against data poisoning attacks, with the exception that Trimmed Mean and Median perform better on the smaller MQ2007 dataset.

8 RESULTS FOR MODEL POISONING

We implement the model poisoning strategies specified in Section 4.3 and report their results, specifically comparing their poisoning effectiveness with that of data poisoning methods.

8.1 Little Is Enough (LIE)

The experimental results obtained for LIE are partially shown in Figure 3(a), along with a comparison with data poisoning.

Ineffectiveness of LIE. The results indicate that LIE is less effective in attacking the performance of the global model compared to data poisoning, with one exception for the perfect click model on the Yahoo dataset when 40% of the clients are malicious. This shows that adding random noise to compromise the local models is less effective for attacking the global ranker performance than compromising the click signals directly. Because of the poor attacking effectiveness of LIE, we do not investigate how it performs when defense strategies are put in place.

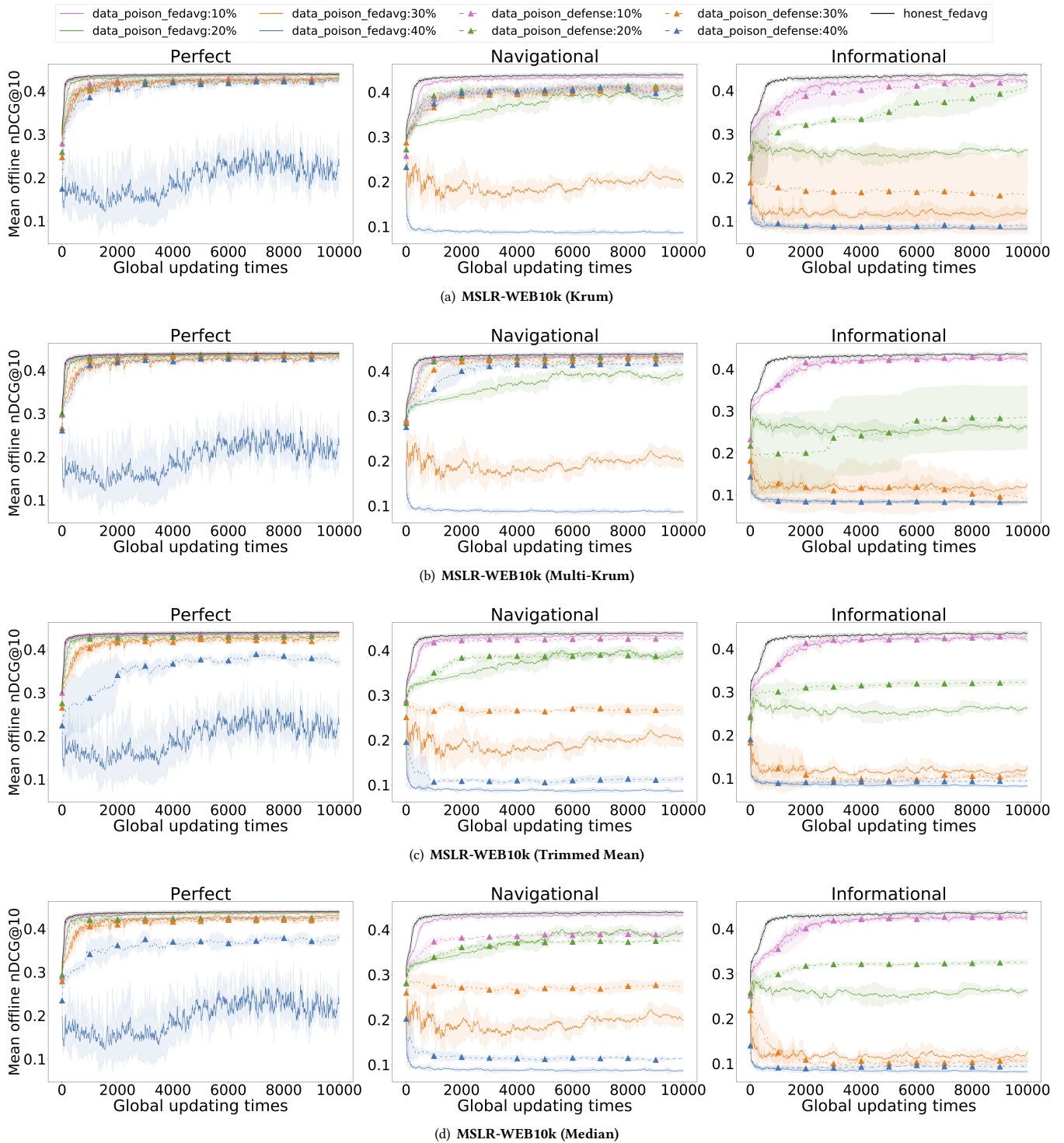


Figure 2: Offline performance (nDCG@10) for MSLR-WEB10k under data poisoning attack and defense strategies, simulated with three benign instantiations of SDBN click model and different percentage of attackers equaling to {10%, 20%, 30%, 40%}; results averaged across all dataset splits and experimental runs.

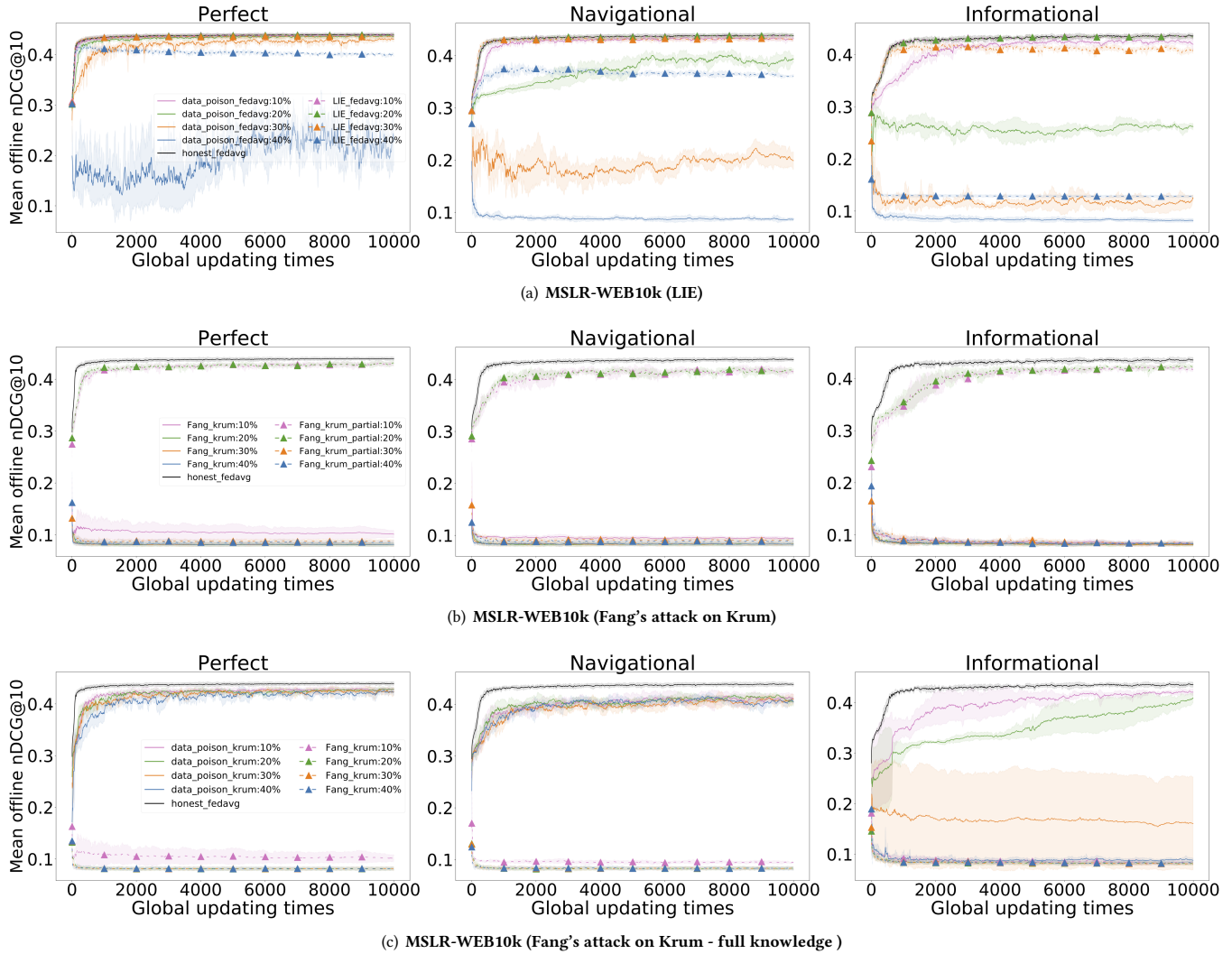


Figure 3: Offline performance (nDCG@10) under model poisoning attacks, simulated with three benign instantiations of SDBN click model under different percentage of attackers equaling to {10%, 20%, 30%, 40%}; results averaged across all dataset splits and experimental runs.

8.2 Fang's Attack

In our experiments, we implement Fang's attacks on four robust aggregation rules, with each attacking strategy tailoring specific defense strategies except that the same attack method is shared for Trimmed Mean and Median.

Full knowledge vs. partial knowledge. First, we compare the attacking performance under both full knowledge and partial knowledge assumptions. According to previous findings in general federated learning [10], attacking with full knowledge performs consistently better than with partial knowledge as the tailored attack can be optimised with auxiliary information about benign clients. From our results (results on MSLR10k under Krum are shown in Figure 3(b)), we observe that full knowledge performs better with fewer malicious clients (10% and 20%), but the gap in effectiveness obtained between full and partial knowledge decreases as the number of malicious clients increases (30% and 40%), thus

leading to differences compared to the general results in federated learning. This is because with more malicious clients, partial knowledge (knowledge of before-attack local model updates for compromised clients) provides enough information to effectively poison the global model while avoiding detection by robust defense strategies.

Fang's Attack vs. data poisoning. Next, we compare Fang's attack under the full knowledge assumption against the data poisoning method under the same robust-aggregation rule (results on MSLR10k under Krum are shown in Figure 3(c)). We find that Fang's attack can successfully poison FOLTR and mitigate the impact of defense methods compared to data poisoning. This finding aligns with the original results from Fang et al. [10].

9 IMPACT OF DEFENSE UNDER NO-ATTACK

Robust aggregation rules exhibit improvements in defending against poisoning attacks under some circumstances. But in real-world settings, the administrator of the FOLTR system has no knowledge

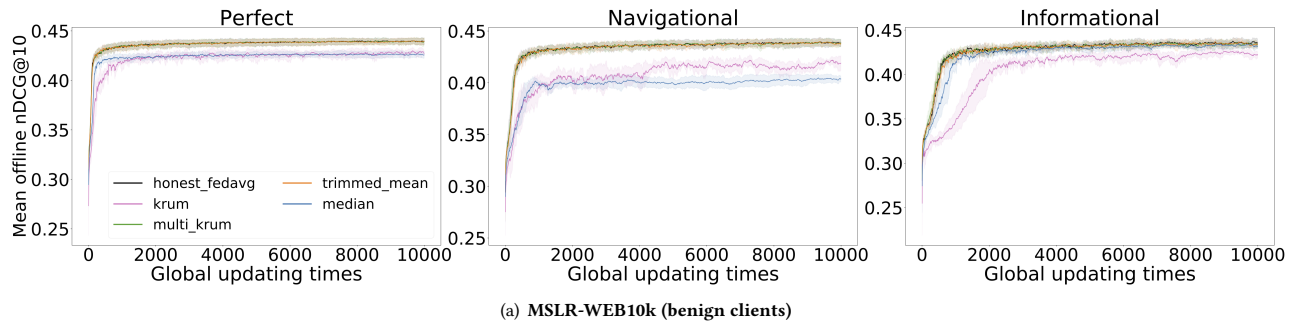


Figure 4: Offline performance (nDCG@10) of FOLTR system when no attack is present but defense strategies are deployed; results averaged across all dataset splits and experimental runs.

of whether an attack is taking place. Thus, if the system administrator wishes to ensure protection against attacks, they may be required to deploy defense strategies irrespective of an attack ever taking place, or not. However, is there a price to pay, in terms of search effectiveness, if a defense strategy is deployed on a FOLTR system that is not exposed to an attack? We investigate this next, by comparing the effectiveness of a FOLTR system with no malicious clients and with different defense strategies implemented against the effectiveness of the same system with no defense.

The experimental results on MSLR10k reported in Figure 4 show that using Krum and Median leads to a decrease in performance compared to the FedAvg baseline when no attacks are present. Results for other datasets are similar and are omitted for space reasons. This finding has also been reported before in general federated learning literature [6, 32, 33], especially when each client’s local training data is non independent and identically distributed (non-IID). This is because those Byzantine-robust FL methods exclude some local model updates when aggregating them as the global model update [6, 32]. This decrease raises questions about the use of these methods in FOLTR systems when no malicious client is present – and it suggests that if reliable methods for attack detection were available, then defense mechanisms may better be deployed only once the attack takes place.

10 SUMMARY OF KEY FINDINGS

Based on the presented empirical results above, we identify the following key findings:

- In general, the perfect click type is more difficult to attack compared to the other two click models, whether it be data or model poisoning methods, except in specific instances when employing Fang’s attack under the full knowledge assumption. To successfully attack a FOLTR system when perfect click feedback is present, a larger number of attackers is required due to the relatively low variance between local updates. As a result, more clients must be compromised to inject noise, otherwise the attack is more likely to be detected by robust aggregation rules.
- Among all attacking strategies studied in this paper, Fang’s attack with full knowledge emerged as the most successful in diminishing the performance of the global model, though some exceptions were observed in the noisy informational click scenario. When there were more malicious clients (i.e. 30% or 40% of the total clients), Fang’s attack with partial knowledge is just as effective as with full knowledge. This indicates that model poisoning

is more effective than data poisoning. Furthermore, when defense measures were implemented, Fang’s attack demonstrated greater success against Krum and Multi-Krum aggregation rules in comparison to Trimmed Mean and Median.

- It is essential to highlight that although Krum has proven effective in countering data poisoning and Trimmed Mean in defending against Fang’s attack, deploying these two aggregation rules should be exercised with caution as they result in an overall decrease in search performance if the system is not exposed to attacks. Thus, the selection of these defense mechanisms should be carefully considered, taking into account the specific context and risk of potential attacks to strike the right balance between security and search effectiveness.

11 CONCLUSION

In this paper we explore attacks and defense mechanisms for federated online learning to rank (FOLTR) systems, focusing on the potential degradation of ranking performance caused by untargeted poisoning attacks. We investigate both data and model poisoning strategies and evaluate the effectiveness of various state-of-the-art robust aggregation rules for federated learning in countering these attacks. Our findings indicate that sophisticated model poisoning strategies outperform data poisoning methods, even when defense mechanisms are in place. We also reveal that deploying defense mechanisms without an ongoing attack can lead to ranker performance degradation. This finding recommends care in the deployment of such mechanisms and suggests that future research should explore defense strategies that do not deteriorate FOLTR ranker performance if no attack is underway.

This is the first study that systematically analyses the threats brought by untargeted poisoning attacks and demonstrates the effectiveness (and associated drawbacks) of existing defense methods on mitigating the impact of malicious adversaries under federated online learning to rank system.

Due to space limitations, we could not include all experiment results in the paper. The complete results, along with code and settings are available at <https://github.com/ielab/foltr-attacks>.

ACKNOWLEDGMENTS

Shuyi Wang is the recipient of a Google PhD Fellowship. This research is partially funded by Beijing Baidu Netcom Technology Co, Ltd, for the project "Federated Online Learning of Neural Rankers", under funding schema 2022 CCF-Baidu Pinecone.

REFERENCES

- [1] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2938–2948.
- [2] Gilad Baruch, Moran Baruch, and Yoav Goldberg. 2019. A little is enough: Circumventing defenses for distributed learning. *Advances in Neural Information Processing Systems* 32 (2019).
- [3] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. 2019. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*. PMLR, 634–643.
- [4] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on International Conference on Machine Learning*. 1467–1474.
- [5] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems* 30 (2017).
- [6] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. 2021. FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping. In *28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021*.
- [7] Olivier Chapelle and Yi Chang. 2011. Yahoo! learning to rank challenge overview. In *Proceedings of the learning to rank challenge*. PMLR, 1–24.
- [8] Olivier Chapelle and Ya Zhang. 2009. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th international conference on World wide web*. 1–10.
- [9] Sara Cohen, Carmel Domshlak, and Naama Zwerdling. 2008. On ranking techniques for desktop search. *ACM Transactions on Information Systems (TOIS)* 26, 2 (2008), 1–24.
- [10] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2020. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. In *Proceedings of the 29th USENIX Conference on Security Symposium*. 1623–1640.
- [11] Rachid Guerraoui, Sébastien Rouault, et al. 2018. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning*. PMLR, 3521–3530.
- [12] William Hersh. 2020. *Information Retrieval: A Biomedical and Health Perspective*. Springer Nature.
- [13] Eugene Kharitonov. 2019. Federated online learning to rank with evolution strategies. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 249–257.
- [14] Jin Young Kim, Nick Craswell, Susan Dumais, Filip Radlinski, and Fang Liu. 2017. Understanding and modeling success in email search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 265–274.
- [15] Leslie Lamport, Robert Shostak, and Marshall Pease. 1982. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems* 4, 3 (1982), 382–401.
- [16] Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri, and Salvatore Trani. 2016. Post-learning optimization of tree ensembles for efficient ranking. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 949–952.
- [17] Lingjuan Lyu, Han Yu, and Qiang Yang. 2020. Threats to Federated Learning: A Survey. *arXiv preprint arXiv:2003.02133* (2020).
- [18] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.
- [19] Harrie Oosterhuis and Maarten de Rijke. 2017. Balancing speed and quality in online learning to rank for information retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 277–286.
- [20] Harrie Oosterhuis and Maarten de Rijke. 2018. Differentiable unbiased online learning to rank. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 1293–1302.
- [21] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 datasets. *arXiv preprint arXiv:1306.2597* (2013).
- [22] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. 2017. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864* (2017).
- [23] Anne Schuth, Harrie Oosterhuis, Shimon Whiteson, and Maarten de Rijke. 2016. Multileave gradient descent for fast online learning to rank. In *proceedings of the ninth ACM international conference on web search and data mining*. 457–466.
- [24] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suci, Christoph Studer, Tudor Dumitras, and Tom Goldstein. 2018. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems* 31 (2018).
- [25] Virat Shejwalkar and Amir Houmansadr. 2021. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*.
- [26] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. 2022. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1354–1371.
- [27] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursay, and Ling Liu. 2020. Data poisoning attacks against federated learning systems. In *European Symposium on Research in Computer Security*. Springer, 480–501.
- [28] Huazheng Wang, Sonwoo Kim, Eric McCord-Snook, Qingyun Wu, and Hongning Wang. 2019. Variance reduction in gradient exploration for online learning to rank. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 835–844.
- [29] Shuyi Wang, Bing Liu, Shengyao Zhuang, and Guido Zuccon. 2021. Effective and Privacy-preserving Federated Online Learning to Rank. In *Proceedings of the 2021 ACM SIGIR international conference on theory of information retrieval*. 3–12.
- [30] Shuyi Wang, Shengyao Zhuang, and Guido Zuccon. 2021. Federated Online Learning to Rank with Evolution Strategies: A Reproducibility Study. In *European Conference on Information Retrieval*.
- [31] Shuyi Wang and Guido Zuccon. 2022. Is Non-IID Data a Threat in Federated Online Learning to Rank?. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2801–2813.
- [32] Qi Xia, Zeyi Tao, Zijiang Hao, and Qun Li. 2019. FABA: an algorithm for fast aggregation against byzantine attacks in distributed neural networks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 4824–4830.
- [33] Jian Xu, Shao-Lun Huang, Linqi Song, and Tian Lan. 2022. Byzantine-robust federated learning through collaborative malicious gradient filtering. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 1223–1235.
- [34] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*. PMLR, 5650–5659.
- [35] Yang Yu, Qi Liu, Likang Wu, Runlong Yu, Sanshi Lei Yu, and Zaixi Zhang. 2023. Untargeted attack against federated recommendation systems via poisonous item embeddings and the defense. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 4854–4863.
- [36] Yisong Yue and Thorsten Joachims. 2009. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*. 1201–1208.
- [37] Shengyao Zhuang and Guido Zuccon. 2020. Counterfactual online learning to rank. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I 42*. Springer, 415–430.