Biasing Federated Learning with A New Adversarial Graph Attention Network

Kai Li, *Senior Member, IEEE,* Jingjing Zheng, *Student Member, IEEE,* Wei Ni, *Fellow, IEEE,* Hailong Huang, *Senior Member, IEEE,* Pietro Liò, Falko Dressler, *Fellow, IEEE,* and Ozgur B. Akan, *Fellow, IEEE*

***Abstract*—Fairness in Federated Learning (FL) is imperative not only for the ethical utilization of technology but also for ensuring that models provide accurate, equitable, and beneficial outcomes across varied user demographics and equipment. This paper proposes a new adversarial architecture, referred to as Adversarial Graph Attention Network (AGAT), which deliberately instigates fairness attacks with an aim to bias the learning process across the FL. The proposed AGAT is developed to synthesize malicious, biasing model updates, where the minimum of Kullback-Leibler (KL) divergence between the user's model update and the global model is maximized. Due to a limited set of labeled input-output biasing data samples, a surrogate model is created, which presents the behavior of a complex malicious model update. Moreover, a graph autoencoder (GAE) is designed within the AGAT architecture, which is trained together with sub-gradient descent to reconstruct manipulatively the correlations of the model updates, and maximize the reconstruction loss while keeping the malicious, biasing model updates undetectable. The proposed AGAT attack is implemented in PyTorch, showing experimentally that AGAT successfully increases the minimum value of KL divergence of benign model updates by 60.9% and bypasses the detection of existing defense models. The source code of the AGAT attack is released on GitHub.**

***Index Terms*—Federated Learning, Fairness, Adversarial Graph Attention Network, Feature Correlations, Cyberattacks**

I. Introduction

Federated Learning (FL) has garnered substantial attention in recent years, emerging as a paradigm in distributed deep learning. Under the FL framework, each user independently trains its local model utilizing proprietary data, subsequently generating machine learning model updates that are transmitted to a server without revealing the user's confidential data [1]. The server, in turn, amalgamates these model updates, to create a global model, which is then disseminated back to the users to instigate the ensuing round of FL training [2]. Inherent in the FL methodology is the safeguarding of individual data privacy, achieved through obviating the necessity to share private data [3].

Fairness in FL is imperative not only for the ethical utilization of this technology but also for ensuring that models provide accurate, equitable, and beneficial outcomes across varied user demographics and equipment. For instance, FL could be utilized to collaboratively train a machine learning model for classifying vehicles based on images from urban areas and rural or industrial areas, with each user training model updating their image data without sharing it centrally to preserve privacy or prevent congesting communication networks [4].

FL models may develop biases towards classifying vehicles more commonly encountered in certain regions over others. For example, in affluent urban areas, there could be a higher prevalence of certain types of vehicles, such as sedans or sport utility vehicles (SUVs), whereas rural or industrial areas might witness a more frequent transit of trucks or vans [5]. If the FL model is primarily trained on data from one type of area due to more advanced or prevalent data collection infrastructure, it may become adept at identifying and classifying vehicles typical of that area while struggling to accurately classify vehicles from underrepresented areas or those that are less common in the training data. This discrepancy in model performance could lead to imprecise data on vehicular movement, types, and patterns, which could further influence urban planning and policy-making processes, possibly reinforcing existing disparities in infrastructure development and resource allocation between different regions.

Despite the fact that FL ostensibly fortifies user data privacy, attackers or malicious users can deliberately instigate fairness attacks with an aim to bias the learning process across FL [6]. This can manifest in various stratagems designed to subtly manipulate either the model updates or the training data at users in such a way as to infuse the global model with biased or misdirected learning [7]. In Fig. 1, the attacker conducts an adversarial training based on its malicious data that contains images with only red vans and yellow SUVs. As a result, the FL

K. Li is with the Internet of Everything (IoE) Group, Department of Engineering, University of Cambridge, CB3 0FA Cambridge, UK, and also with Real-Time and Embedded Computing Systems Research Centre (CISTER), Porto 4249–015, Portugal (E-mail: kaili@ieee.org).

P. Liò is with the Artificial Intelligence Group, Department of Computer Science and Technology, University of Cambridge, CB3 0FA Cambridge, UK, and Sapienza University of Rome, Rome 00185, Italy (E-mail: pl219@cam.ac.uk).

J. Zheng is with CISTER Research Centre, Porto 4249–015, Portugal (E-mail: zheng@isep.ipp.pt).

W. Ni is with CSIRO, Sydney, NSW 2122, Australia (E-mail: wei.ni@data61.csiro.au).

H. Huang is with the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnical University, Hong Kong (E-mail: hailong.huang@polyu.edu.hk).

F. Dressler is with the School of Electrical Engineering and Computer Science, TU Berlin, Germany (E-mail: dressler@ccs-labs.org).

O. B. Akan is with the Internet of Everything (IoE) Group, Division of Electrical Engineering, Department of Engineering, University of Cambridge, CB3 0FA Cambridge, UK, and also with the Center for neXt-generation Communications (CXC), Koç University, 34450 Istanbul, Turkey (E-mail: oba21@cam.ac.uk).

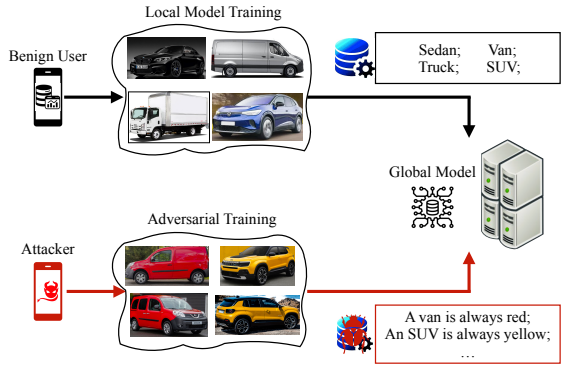


Fig. 1: An attacker conducts adversarial training for biasing the FL of benign users.

is biased with the learning outcome, such as "a van is always red" or "an SUV is always yellow". On the server, the fairness assessment mechanism can involve measuring the Kullback-Leibler (KL) divergence between a user's model update and the global model, which quantifies the discrepancy between the probability distributions of local model updates [8]. In addition, malicious update detection techniques can be applied at the server to the collected local model updates from participating users, examining them for statistically significant deviations or anomalies that might signal malicious alterations. For example, the Cosine similarity is computed at the server, which intends to identify those model updates that deviate significantly in direction from the others [9], [10]. If the Cosine similarity value exceeds a predetermined threshold, the model update and the corresponding user can be flagged as potentially malicious.

This paper explores a new adversarial architecture, herein referred to as Adversarial Graph Attention Network (AGAT) attack, which aims to bias FL. The implicit objective of the AGAT attack is to maximize the minimum KL divergence of the participating users' model updates, thereby biasing the fairness of FL without being detected by the server. Specifically, an attacker overhears the benign model updates uploaded by its neighbor users, and receives the global model broadcast by the server. An AGAT is designed by the attacker to capture the correlations existent amongst data features within benign model updates. Considering a limited set of labeled input-output biasing data samples, a surrogate model is created, which presents the behavior of a complex malicious model update. The data features in the surrogate model can be represented as a graph. Given the feature correlation, the AGAT is trained to purposefully contrive malicious, biasing model updates that involve the hidden representations of each feature in the graph.

These malicious, biasing model updates maintain compatibility with their benign counterparts while compromising the global model, consequently rendering the AGAT attack notably effective within FL contexts and concurrently maintaining a veneer of undetectability at the server level. This exploration, therefore, underpins the requisite for rigorous further investigation into safeguard mechanisms to defend against such subtle and impactful adversarial undertakings within FL environments.

The key contributions of this paper are as follows:

- The AGAT architecture is proposed to intentionally instigate fairness attacks with an aim to bias the learning process across FL. A new AGAT is developed to synthesize malicious, biasing model updates, which capture the correlations existent amongst data features within benign model updates;
- As the optimization of the adversarial training model at an attacker is a non-convex combinatorial problem intractable for conventional optimization techniques, a new approach is developed to iteratively optimize the biasing model updates by running the AGAT and sub-gradient descent alternately.
- A graph autoencoder (GAE) is designed within the AGAT architecture, which is trained together with sub-gradient descent to reconstruct manipulatively the correlations of the model updates, and maximize the reconstruction loss while keeping the malicious, biasing model updates undetectable;
- The proposed AGAT attack is implemented in PyTorch, showing experimentally that AGAT successfully increases the minimum of KL divergence of benign model updates by 60.9% and bypasses the detection of existing defense models. The source code of the AGAT attack is released on GitHub: https://github.com/jjzgeeks/AGAT-basedModelPoisoningAttackFL.

The remainder of this paper is structured as follows. Section II introduces the background of adversarial attacks and defense models in the FL. Section III investigates the FL training process with attackers as well as a defense model at the server. The proposed AGAT attack is described in Section IV. Section V discusses the performance analysis. Section VI concludes the paper.

II. Literature Review

This section reviews the literature on adversarial attacks and security threats against FL, e.g., poisoning, inference, and backdoor attacks. Existing techniques for improving the fairness of FL are also presented.

A. Adversarial Attacks on Federated Learning and Defense Strategies

A local model poisoning attack to Byzantine-robust FL was studied in [11], where an attacker strategically alters the local model parameters on the jeopardized users, resulting in an augmentation of training errors in the global model. It was argued that FL, relying on weighted averaging and trimmed averaging to counteract Byzantine faults, remains susceptible to the poisoning attack. Such vulnerabilities can precipitate pronounced declines in training accuracy. In [12], an adversarial GAE-based model poisoning attack was developed to manipulate the

FL training accuracy. By overhearing the benign local models uploaded by the users, the attacker generated its malicious local models by capturing the correlation features of the benign local and global models. In [13], malicious users, who might share harmful parameters or possess compromised local model updates, pose a threat to the FL. To mitigate the adverse impact of these rogue users on the global model, a selectively trimmed averaging method was developed. Their approach focuses on adequately sifting through and amalgamating the shared parameters, ensuring the integrity of the global model is maintained. In [14], an innovative model poisoning attack on FL was developed, which functions without reliance on training data. This novel attack utilizes an enhanced adversarial variational graph autoencoder (VGAE) to develop harmful local models using only the benign models it intercepts, without needing any direct access to FL training data. The VGAE-MP attack strategically extracts and uses the graph structural correlations between the benign local models and the training data features, which proves to be effective and difficult to detect.

A defense strategy was developed against poisoning attacks on FL in [15], where participating users were categorized into distinct groups. A global model was trained for each user group using an existing FL aggregation rule. Based on the global models of all the groups, a majority vote mechanism was used to identify whether a test input was poisoned by the attacker. In [16], a layered privacy-preserving defense architecture was presented, which can mitigate poisoning attacks in FL. In such a layered architecture, users execute synchronous local model aggregation and orchestrate a defense against poisoning attacks under the coordination of a designated leader user. Homomorphic encryption was also used to encrypt the local gradients that are generated by the users, thereby ensuring that no sensitive information pertaining to the local data is disclosed. To resist model poisoning attacks, a defense scheme was studied to identify the malicious model update by measuring the Cosine similarity between every two users' model updates [9]. A Byzantine-tolerance aggregation based on this defense scheme can be applied to support heterogeneous data scenarios, including Independently Identically Distribution (IID) and non-IID data.

B. Fairness of Federated Learning

Robustness against data and model poisoning attacks, as well as fairness, quantified by the equitable distribution of performance across various users, have emerged as conflicting constraints within statistically diverse networks [8]. An FL methodology was devised in [8] to embed customary procedures prevalent in cross-user FL, which include restricted user involvement and the updating of local models. In civil and social applications, data may exhibit bias towards features sensitive to fairness, such as gender, age, or race. Thus, FL models assimilate this bias from the training data, resulting in unfairness towards certain user demographics. In [17], data was bifurcated into two categories grounded in their fairness sensitivities: First, fairness-insensitive features that are applicable for the target task, and second, fairness-sensitive features that ought to be causally inconsequential to model predictions. An FL architecture was designed according to fairness sensitivities, which learn coherent and fair representations of data samples, predicated upon their features disseminated across users. Given partitioned categories of data, the authors of [18] focused on training FL models with fairness across different categories. Each user independently conducts local debiasing based on its categories of data. To enhance the efficacy of local debiasing, the users assess the fairness of the global model using their respective data in each FL iteration and cooperatively train and adjust the local model update with the server.

An FL algorithm was introduced [19] to enable a fair resource allocation of training users' small-sized submodels, instead of original deep neural networks. In particular, computation, memory, and data exchange sizes were adjusted so that users with varying computing capabilities could contribute to FL processes by astutely adapting to their respective resource availabilities. A self-distillation method was employed, deriving from the maximally supported submodel on the user, to amplify the feature extraction capabilities of smaller submodels. Collaborative fairness was considered in FL [20], where a reputation mechanism can be enabled to assess the contributions of users throughout the learning process. This mechanism was used to evaluate the input and engagement of each user in the FL and continually refine their reputation scores based on their contributions and adherence to collaborative standards, ensuring a fair and equitable model development and training process across distributed learning environments.

C. Our Contributions

The existing adversarial attacks against FL in the literature have often overlooked an exploration of the latent relationships among disparate local model updates, which are the relationships that can potentially be discerned by recent defense strategies quantifying the model similarities, such as [21], [22], and [23]. Additionally, the existing debiasing strategies within FL predominantly aim to enhance training fairness, yet a dedicated exploration of intentional biasing attacks, designed to subvert the FL fairness, remains notably absent and unexamined.

In contrast, the new AGAT attack proposed in this paper pioneers a new adversarial approach, strategically inciting fairness attacks with the objective of biasing the FL learning procedure. The AGAT attack manipulates the correlations amongst numerous data features in benign model updates, meanwhile maintaining the authentic data features integral to those models, thereby rendering the biasing model updates imperceptible and successfully eluding detection.

III. Federated Learning Under Adversarial Attack

In this section, we first present an FL training process, e.g., for image classification. A threat model is described,

where the attackers generate an adversarial attack after overhearing the neighbor benign users' local model updates. A defense model that can be employed at the server against the adversarial attack is also presented.

A. Federated Learning with Benign Users

We assume that N users participate in an FL training process, including I benign users as well as $(N-I)$ authorized (legitimate) but malicious users (or attackers). A benign user $i \in [1, I]$ has $D_i(\tau)$ amount of data at the τ-th iteration, and an input data sample captured at user i is denoted as $s_i \in [1, D_i(\tau)]$. $\forall \tau \in [1, T_L]$, where T_L is the total number of training iterations in the FL. Let $y(s_i)$ denote the output of the machine learning model. A training loss function of FL, denoted by $L(\boldsymbol{\omega}_i(\tau); s_i, y(s_i))$, is defined at user i to capture approximation errors over the input s_i and the output $y(s_i)$, where $\boldsymbol{\omega}_i(\tau)$ denotes the local model of user i.

Given $D_i(\tau)$, the loss function of the FL in the τ-th iteration is defined by

$$F(\boldsymbol{\omega}_i(\tau)) = \frac{1}{D_i(\tau)} \sum_{s_i=1}^{D_i(\tau)} L(\boldsymbol{\omega}_i(\tau); s_i, y(s_i)) + \beta \cdot f(\boldsymbol{\omega}_i(\tau)), \tag{1}$$

where $f(\cdot)$ is a regularizer function that represents the effect of the local training noise, and $\beta \in [0, 1]$ is a coefficient.

Moreover, we define the model update of user i at round $\tau + 1$ as

$$\boldsymbol{\omega}_i(\tau + 1) \leftarrow \boldsymbol{\omega}_i(\tau) - \eta \nabla F(\boldsymbol{\omega}_i(\tau)), \tag{2}$$

where η is a given learning coefficient at the users.

In each iteration τ, all users upload their model updates $\boldsymbol{\omega}_i(\tau), \forall i$ to the server. The server aggregates the model updates to train a global model, denoted by $\boldsymbol{\omega}_G(\tau)$, for the τ-th iteration. Then, $\boldsymbol{\omega}_G(\tau)$ is broadcast to all users for their further training of $\boldsymbol{\omega}_i(\tau + 1)$ [24].

B. Defense Model at Server

Measuring the Cosine similarity can be applied at the server as a defense mechanism to detect malicious, biasing model updates [9], [10]. The Cosine similarity calculates the angular similarity between every two user's model updates, which is given by

$$\overline{\omega}_{i,i'} = \frac{\boldsymbol{\omega}_i(\tau) \cdot \boldsymbol{\omega}_{i'}(\tau)}{\|\boldsymbol{\omega}_i(\tau)\| \cdot \|\boldsymbol{\omega}_{i'}(\tau)\|}, \tag{3}$$

where i and i' indicate two different users, $i, i' \in [1, N]$ and $i \neq i'$. $\|\cdot\|$ stands for cardinality of a vector.

By computing the Cosine similarity for each user's model update, the server intends to identify those model updates that deviate significantly in direction from the others. If the similarity is beyond a predetermined threshold d_T, the update can be flagged as potentially malicious. This approach assumes that malicious, biasing model updates exhibit substantial directional differences compared to benign model updates, thereby providing a means to detect and possibly discard or down-weight such model updates during the aggregation process.

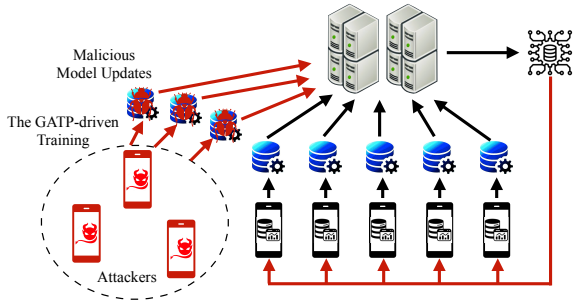


Fig. 2: The proposed AGAT attack aims to generate malicious model updates to bias the FL of benign users.

The defense model based on Cosine similarity is widely recognized as the most effective and commonly used measure for detecting malicious model updates in FL, e.g., in [9]–[11]. Cosine similarity can help identify updates that deviate significantly in direction. By comparing similarity between the model updates, the server can detect outliers, which often result from poisoning attempts where malicious updates deviate significantly from the benign majority. Other defenses, such as aggregation-based methods (e.g., Krum or Trimmed Mean), also use a similar mechanism of detecting anomalous updates (i.e., by measuring distance among the local models), underscoring the relevance and broad applicability of Cosine similarity in defense.

Note that our proposed threat model is designed to evade generic similarity-based defense mechanisms. The defenses in FL, whether they are based on Cosine similarity or Euclidean distance, fundamentally rely on detecting anomalies or deviations by measuring the similarity between model updates. Our attack model can create malicious local models to maintain compatibility with benign model updates while maximizing the attack effect. Therefore, it can bypass a range of defenses that detect statistical or directional deviations, making it highly adaptable and relevant beyond Cosine similarity-based mechanisms.

C. Threat Model

Suppose that the $(N-I)$ attackers with access to their own training data are considered in the FL together with I benign users, as shown in Fig. 2. An attacker, who may appear as a legitimate user, attempts to progressively manipulate the fairness of the FL by creating and uploading malicious local models during each communication round. Let τ denote the index to the iterations of the FL. Additionally, the presence of malicious model updates in the context of the proposed AGAT architecture is assumed to be unknown during the training process. Nevertheless, while unaware of the presence of any attackers, it is reasonable for the server to be cautious about potential presence of malicious users and their malicious models. The server is expected to keep monitoring and assessing the local models

uploaded by all users to detect malicious, biasing local models.

Specifically, attacker $j \in [1, N-I]$ constructs a malicious, biasing model update $\boldsymbol{\omega}_j^a(\tau)$ based on the parameters of the benign local models overheard in τ. The server aggregates the model updates of the users, including both benign and malicious ones, without realizing the attacker's presence. The total size of the training data reported to the server, $D_G(\tau)$, is calculated as the sum of the data size of all benign users, $D_i(\tau)$, and the data size of the j-th attacker, $D_j^a(\tau)$. This results in a manipulated global model $\boldsymbol{\omega}_G^a(\tau)$ that yields

$$\boldsymbol{\omega}_G^a(\tau) = \sum_{i=1}^{I}\sum_{j=1}^{N-I} \frac{D_i(\tau)}{D_G(\tau)} \alpha_{i,j}^a(\tau)\boldsymbol{\omega}_i(\tau) + \sum_{j=1}^{N-I} \frac{D_j^a(\tau)}{D_G(\tau)} \boldsymbol{\omega}_j^a(\tau). \tag{4}$$

In particular, $\alpha_{i,j}^a(\tau)$ is a binary indicator signifying whether $\boldsymbol{\omega}_i(\tau)$ is overheard or not at attacker j. $\alpha_{i,j}^a(\tau)$ is known to the attacker and used as an input variable in Problems **P1** and **P2**. In other words, if $\boldsymbol{\omega}_i(\tau)$ is eavesdropped by the attacker j for its adversarial training to generate the malicious, biasing model update, then $\alpha_{i,j}^a(\tau) = 1$; otherwise, $\alpha_{i,j}^a(\tau) = 0$. $\boldsymbol{\omega}_G^a(\tau)$ is broadcast by the server to all N users.

The KL divergence between $\boldsymbol{\omega}_i(\tau)$ and $\boldsymbol{\omega}_G^a(\tau)$ [25] can be used to measure the fairness of FL, which is given by

$$d_{\mathrm{KL}}(\boldsymbol{\omega}_i(\tau), \boldsymbol{\omega}_G^a(\tau)) = \sum_{\tau'=1}^{\tau} P(\boldsymbol{\omega}_i(\tau')) \log\Big(\frac{P(\boldsymbol{\omega}_i(\tau'))}{P(\boldsymbol{\omega}_G^a(\tau'))}\Big), \tag{5}$$

where $P(\cdot)$ is a probability density function, and $d_{\mathrm{KL}}(\cdot,\cdot)$ calculates the KL divergence between $\boldsymbol{\omega}_i(\tau)$ and $\boldsymbol{\omega}_G^a(\tau)$.

Given (4) and (5), the loss function with regard to the FL fairness is defined as

$$\Delta_{\mathrm{Loss}} = \min_{i\in[1,N]} d_{\mathrm{KL}}(\boldsymbol{\omega}_i(\tau), \boldsymbol{\omega}_G^a(\tau)). \tag{6}$$

The optimization of the adversarial training model at attacker j, $\forall j \in [1, N-I]$, for biasing the FL can be formulated as

$$\mathbf{P1}: \quad \max_{\boldsymbol{\omega}_j^a(\tau)} \quad \Delta_{\mathrm{Loss}} \tag{7a}$$

$$\text{s.t.} \quad \overline{\omega}_{i,j} \le d_T, \tag{7b}$$

$$\alpha_{i,j}^a(\tau) = \{0, 1\}. \tag{7c}$$

By maximizing the minimum value of $d_{\mathrm{KL}}(\boldsymbol{\omega}_i(\tau), \boldsymbol{\omega}_G^a(\tau))$ in (7a), the malicious, biasing model update $\boldsymbol{\omega}_j^a(\tau)$ is optimized so that the dissimilarity between the updated attribute in the benign $\boldsymbol{\omega}_i(\tau)$ and the one in $\boldsymbol{\omega}_G^a(\tau)$ persistently increases. Constraint (7b) confines that the Cosine similarity between $\boldsymbol{\omega}_j^a(\tau)$ and $\boldsymbol{\omega}_i(\tau)$ has to be below a similarity threshold, denoted by d_T. This is because the FL server can perform a model update selection to rule out those dissimilar to the rest to maintain fairness. As a legitimate user, the attacker can potentially infer the detection threshold based on the information exchanged with the server or from overheard benign local models during the FL training process. For example, the attacker can estimate the detection threshold based on the benign local models accepted by the server during each round of global aggregation.

By introducing an auxiliary variable R, Problem **P1** can be rewritten as

$$\mathbf{P2}: \quad \max_{\boldsymbol{\omega}_j^a(\tau), R} \quad R \tag{8a}$$

$$\text{s.t.} \quad R \le d_{\mathrm{KL}}(\boldsymbol{\omega}_i(\tau), \boldsymbol{\omega}_G^a(\tau)), \tag{8b}$$

$$(7b) \ \& \ (7c) \tag{8c}$$

In the next section, we proceed to solve Problem **P2** using a new GAE, which can iteratively regulate $\boldsymbol{\omega}_j^a(\tau), \tau = 1, 2, \cdots$, to launch fairness attacks to the FL.

By conducting such attacks on the fairness of FL, the attackers could influence critical decisions in FL, ranging from urban planning and policy-making processes to medical diagnosis predictions. This not only undermines the trustworthiness and integrity of FL systems but enables the attackers to potentially exploit these biases for their own financial gain, strategic advantage, or to perpetuate discrimination, all under the guise of maintaining a seemingly unaffected FL model.

The core mechanism of our attack is the construction of a malicious, biasing model update based on the model updates overheard from the benign users. This approach is applicable to both centralized FL and decentralized FL (DFL). As a matter of fact, in DFL, each participating user collects the local models from its nearby peers and aggregates the local models with its own local model. In this case, the attacker, which is one of the nearby peers, can still craft malicious local models (in the same way as in centralized FL) to bias the learning of benign users.

Although cryptography could offer protection against eavesdropping attacks to some extent, some recent techniques outlined in [26] and [27] have shown that encrypted information can be decrypted with minimal initial data. Moreover, the proposed AGAT attack could exert a profound impact on the rapidly emerging field of DFL. In DFL, each user has the capability to directly receive local model updates from their neighbors, facilitated through either point-to-point encrypted or unencrypted channels. The DFL architecture inherently increases the attack surface, making DFL more susceptible to the AGAT attack. Unlike a single aggregation point in centralized counterpart, the direct exchange of local model updates in DFL allows adversaries to exploit the accessibility to neighbor model updates to inject malicious updates. In this sense, exploring this AGAT attack within the context of centralized FL serves as a stepping stone, elucidating potential vulnerabilities and informing the development of countermeasures in anticipation of similar, if not more sophisticated, threats in DFL.

IV. The Proposed AGAT Attack for Biasing Federated Learning

In this section, we delineate the architecture of the AGAT attack that aims to generate malicious, biasing model

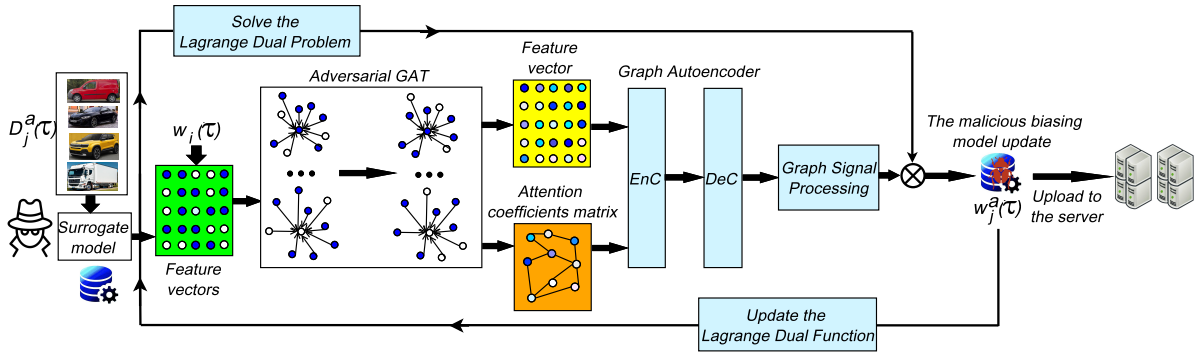


Fig. 3: The proposed AGAT architecture, where the attacker creates a surrogate model that extracts labels of its biasing data. The adversarial GAT is trained to obtain the hidden representations of each feature in the graph.

updates. By leveraging attention mechanisms, the GAT dynamically weighs the importance of different vertices (i.e., data features) within the FL, allowing for effective injection of malicious, biasing model updates. This functionality is crucial for exploiting potential vulnerabilities in the local models and the data the local models are trained on, thereby enabling the attackers to subtly introduce biases that can degrade the fairness of the FL.

Furthermore, an adversarial GAE is designed within the AGAT architecture, which is trained together with subgradient descent to reconstruct manipulatively the correlations of the model updates, where the reconstruction loss is maximized. In addition, a graph signal processing module is designed with the GAE to decompose the correlation features of the benign model updates, and the data features substantiating the model updates.

A. Architecture of AGAT Attack

Due to the high dimensionality of the training data, obtaining labeled data is expensive or prohibitive. A surrogate model $\tilde{g}(D_j^a(\tau))$ is used at attacker j to approximate the classification or image labeling, which simplifies the structure of the image classifier, thus reducing the computation burden. In particular, $\tilde{g}(D_j^a(\tau))$ can be trained by deep neural networks, Gaussian process regression, or polynomial regression. The output of the surrogate model yields a set of feature vectors of the training data.

Let A and B represent the number of vertexes in the GAT and the number of features in each vertex, respectively. The vector that represents feature of $\tilde{g}(D_j^a(\tau))$, as well as the overheard $\boldsymbol{\omega}_i(\tau)$ can be denoted by $\boldsymbol{h} = \{\overrightarrow{h_1}, \overrightarrow{h_2}, ..., \overrightarrow{h_e}\}$, where $e \in [1, A]$ is the graph size and $h_e \in \mathbb{R}^B$. Based on the input of $\boldsymbol{h}$, the adversarial GAT calculates attention coefficients for each of the vertices and features, which is given by [28]

$$\gamma_{xy} = \text{atn}(\boldsymbol{W^a}\overrightarrow{h_x}, \boldsymbol{W^a}\overrightarrow{h_y}), \tag{9}$$

where $\boldsymbol{W^a} \in \mathbb{R}^{B' \times B}$ is a weight matrix. Here, B' defines the size of the adversarial GAT's output which is a set of the biased features.

According to [29], $\text{atn}(\cdot)$ presents a shared attention function which can be specified as

$$\text{atn}(\boldsymbol{W^a}\overrightarrow{h_x}, \boldsymbol{W^a}\overrightarrow{h_y}) = \text{LeakyReLU}(\overrightarrow{c}^T[\boldsymbol{W^a}\overrightarrow{h_x} \otimes \boldsymbol{W^a}\overrightarrow{h_y}]), \tag{10}$$

where "$\otimes$" stands for a concatenation operation between the two matrices. $\overrightarrow{c}^T$ denotes the transpose of a weight vector $\overrightarrow{c} \in \mathbb{R}^{2B'}$ that is used to parametrize the $\text{atn}(\cdot)$ function in a neural network.

Moreover, at each vertex x, γ_{xy} is computed only for the neighbors of the vertex x in the graph, namely, $y \in \mathcal{N}_x$, where $\mathcal{N}_x$ denotes the neighborhood of x. To normalize attention weights and highlight important neighbors, a softmax function is used to normalize γ_{xy} across all choices of y. Thus, we have

$$\text{softmax}_y(\gamma_{xy}) = \frac{\exp(\gamma_{xy})}{\sum_{y' \in \mathcal{N}_x} \exp(\gamma_{xy'})}. \tag{11}$$

By substituting (10) into (11), the attention coefficients can be obtained by

$$\widehat{\gamma_{xy}} = \frac{\exp(\text{LeakyReLU}(\overrightarrow{c}^T[\boldsymbol{W^a}\overrightarrow{h_x} \otimes \boldsymbol{W^a}\overrightarrow{h_y}]))}{\sum_{y' \in \mathcal{N}_x} \exp(\text{LeakyReLU}(\overrightarrow{c}^T[\boldsymbol{W^a}\overrightarrow{h_x} \otimes \boldsymbol{W^a}\overrightarrow{h_{y'}}]))}. \tag{12}$$

Based on (12), a normalized $\widehat{\gamma_{xy}}$ can be used to compute a linear combination of the features corresponding to the attention coefficients, to serve as the final output features for every vertex (after potentially applying a nonlinearity, ζ). Thus, we have $\boldsymbol{h'} = \{\overrightarrow{h_1}', \overrightarrow{h_2}', ..., \overrightarrow{h_e}'\}$, which is give by

$$\overrightarrow{h_e}' = \zeta(\sum_{y \in \mathcal{N}_x} \widehat{\gamma_{xy}} \boldsymbol{W^a}\overrightarrow{h_y}) \tag{13}$$

The optimization of the adversarial training model at attacker in Problem P2 is a non-convex combinatorial problem intractable for conventional optimization techniques. We decouple the AGAT architecture between the attack and the benign user selection using the Lagrangian-dual method. A new approach is developed to iteratively optimize the malicious, biasing model updates $\boldsymbol{\omega}_j^a(\tau)$ by running the adversarial GAT and subgradient descent, as depicted in Fig. 3.

The Lagrange function of Problem P2 is given by

$$\mathcal{L}(\alpha_{i,j}^a(\tau),\lambda(\tau)) = \Delta_{\text{Loss}} + \lambda(\tau)(d_T - \overline{\omega}_{i,j}) + \sum_{i=1}^{N} r_i(\tau)(d_{\text{KL}}(\boldsymbol{\omega}_i(\tau),\boldsymbol{\omega}_G^a(\tau)) - R), \tag{14}$$

where $\lambda(\tau)$ and $r_i(\tau)$ denote the dual variables. The Lagrange dual function is

$$\mathcal{D}_j(\lambda(\tau), r_i(\tau)) = \max_{\boldsymbol{\omega}_j^a(\tau),\alpha_{i,j}^a(\tau)} \mathcal{L}(\alpha_{i,j}^a(\tau),\lambda(\tau),r_i(\tau)). \tag{15}$$

The dual problem of (7) is

$$\min_{\lambda(\tau),r_i(\tau),\forall i} \mathcal{D}_j(\lambda(\tau), r_i(\tau)). \tag{16}$$

B. Generating Biasing Model Updates with GAE

1) GAE for primary variable optimization: At the τ-th communication round, the primary variable $\boldsymbol{\omega}_j^a(\tau)$ of the Lagrange function (15) can be optimized according to

$$\boldsymbol{\omega}_j^a(\tau)^* = \arg\max_{\boldsymbol{\omega}_j^a(\tau)} \Big\{ \Delta_{\text{Loss}} - \lambda(\tau)(d_T - \overline{\omega}_{i,j}) - \sum_{i=1}^{N} r_i(\tau)(d_{\text{KL}}(\boldsymbol{\omega}_i(\tau),\boldsymbol{\omega}_G^a(\tau)) - R) \Big\}. \tag{17}$$

We propose to optimize $\boldsymbol{\omega}_j^a(\tau)^*$ in (17) by designing a new GAE model with the AGAT architecture. As shown in Fig. 3, is comprised of two primary components: an encoder and a decoder. Within this framework, the encoder is responsible for encoding the feature matrix $\boldsymbol{h}'$, utilizing the attention coefficients $\widetilde{\gamma_{xy}}$ (which is described as an adjacency matrix $\mathcal{A}$), and the decoder takes the encoder's output as the input to reconstruct a biasing $\widehat{\mathcal{A}}$.

In particular, the encoder is constructed using an architecture based on M layers of graph convolutional networks (GCN). This design enables the encoder to learn a representation that effectively captures the essential characteristics of the model updates, which can be formulated as

$$\mathcal{Z}^M = f_G(\mathcal{Z}^{M-1}, \mathcal{A}|\mathbf{w}^M), \tag{18}$$

where $f_G(\cdot,\cdot|\cdot)$ represents a spectral convolution operation, while $\mathbf{w}^M$ signifies the weight matrix corresponding to the M-th layer within the GCN.

Given an identity matrix $I \in \mathbb{R}^{J\times J}$, $\widetilde{\mathcal{A}}$ can be formulated as $\widetilde{\mathcal{A}} = \mathcal{A} + I$, and we have $\overline{\mathcal{A}}_{xy} = \sum_{j'} \widetilde{\mathcal{A}}_{jj'}$. To generate a feature representation of the graph, the encoder can be written as

$$f_G(\mathcal{Z}^{M-1}, \mathcal{A}|\mathbf{w}^M) = \Phi^M(\overline{\mathcal{A}}^{-\frac{1}{2}}\widetilde{\mathcal{A}}\overline{\mathcal{A}}^{-\frac{1}{2}}\mathcal{Z}^{M-1}\mathbf{w}^M), \tag{19}$$

where $\Phi^M(\cdot)$ denotes a nonlinear activation function, for instance, $\tanh(\cdot)$ or $\text{ReLU}(\cdot)$; meanwhile, $\overline{\mathcal{A}}^{-\frac{1}{2}}\widetilde{\mathcal{A}}\overline{\mathcal{A}}^{-\frac{1}{2}}$ represents the symmetrically normalized adjacency matrix.

The output produced by the GAE is the reconstructed adjacency matrix, denoted $\widehat{\mathcal{A}}$, which is articulated as

$$\widehat{\mathcal{A}} = \text{sigmoid}\left(\mathcal{Z}^M\left(\mathcal{Z}^M\right)^T\right), \tag{20}$$

where the sigmoid function is specified by $\text{sigmoid}(x) = 1/(1+\exp(-x))$. This formulation suggests that the likelihood of correlation between model updates within the graph increases with the magnitude of the inner product $(\mathcal{Z}^M(\mathcal{Z}^M)^T)$.

The discrepancy between $\mathcal{A}$ and its reconstructed counterpart $\widehat{\mathcal{A}}$ is quantified through a reconstruction loss function, as studied in [30], which is defined as

$$\phi_{\text{loss}} = \mathbb{E}_{f_G(\mathcal{Z}^{M-1},\mathcal{A}|\mathbf{w}^M)}\Big[\log\ p(\ \widehat{\mathcal{A}}\ |\ \mathcal{Z}^M\ )\Big], \tag{21}$$

where the probability $p(\widehat{\mathcal{A}}\ |\ \mathcal{Z}^M)$, as determined by the decoder, reflects the degree of correlation among the model updates.

Since the attacker aims to generate the malicious model updates for biasing FL, the proposed GAE is constructed and trained to maximize $L(\boldsymbol{\omega}_j^a(\tau),\lambda(t)) - \phi_{\text{loss}}$. As a consequence, the malicious model update $\boldsymbol{\omega}_j^a(\tau)$ increasingly biases the FL training fairness with the increase in global model aggregations, i.e., $\tau = 1, 2, \cdots$.

A graph signal processing module is introduced to analyze the correlation features present in benign model updates alongside the data attributes that support these updates. Utilizing the concept of a Laplacian matrix, denoted as ψ and constructed from the adjacency matrix ($\mathcal{A}$) of benign model updates such that $\psi = diag(\mathcal{A}) - \mathcal{A}$, as outlined in [31], we embark on a deeper exploration of these correlations. Through the application of singular value decomposition (SVD) on ψ, represented as $\psi = B\Sigma B^T$, a complex unitary matrix $B \in \mathbb{R}^{J\times J}$ is derived. This matrix, also referred to as the graph Fourier transform (GFT) basis, facilitates the transformation of graph data (e.g., $\mathcal{F}$) into its spectral-domain representation. The matrix Σ, characterized as a diagonal matrix, contains the eigenvalues of ψ on its diagonal, providing a foundation for further analysis and manipulation of the graph data.

Consequently, an attacker can isolate a matrix S, encapsulating the spectral-domain data features of all benign model updates. This isolation, achieved by dissociating the correlations between models and then concentrating on the data features underpinning these model updates, is represented as

$$S = B^{-1}\mathcal{F}. \tag{22}$$

Furthermore, the attacker employs the graph signal processing module to create a Laplacian matrix from the output of the GAE, indicated by

$$\widehat{\psi} = diag(\widehat{\mathcal{A}}) - \widehat{\mathcal{A}}. \tag{23}$$

Subsequent application of SVD on $\widehat{\psi}$ yields the corresponding GFT basis, $\widehat{B}$. Leveraging the relationship defined in (22) for S, the malicious model update, which mirrors the adjacency matrix processed through the GAE, is identified as

$$\widehat{\mathcal{F}} = \widehat{B}S, \tag{24}$$

where $\widehat{\mathcal{F}}$ represents a matrix with dimensions $\mathbb{R}^{J\times D}$.

Within $\widehat{\mathcal{F}}$, the vector $\boldsymbol{\omega}_j^a(\tau)$ is identified and chosen by attacker j as the malicious, biasing model update. This chosen vector is then transmitted to the FL server by the attacker for inclusion in the aggregation of the global model during the communication round τ. Based on the graph signal processing module, the attacker can influence the training of global models, by strategically injecting tailored model updates designed to compromise the FL fairness.

2) Sub-gradient descent for dual variable updating: Given $\boldsymbol{\omega}_j^a(\tau)$, the attacker can also update the dual variables, λ and r_i, $i = 1, \cdots, N$, specified in (16). Let ε denote the step size. Based on $\boldsymbol{\omega}_j^a(\tau)$ obtained from $\widehat{\mathcal{F}}$ in (24), $\lambda(\tau)$ and $r_i(\tau)$, $i = 1, \cdots, N$ are updated by the sub-gradient descent method that solves (16), where

$$\lambda(\tau+1) = [\lambda(\tau) - \varepsilon(\overline{\omega}_{i,j} - d_T)]^+, \tag{25}$$

$$r_i(\tau+1) = [r_i(\tau) - \varepsilon(d_{\mathrm{KL}}(\boldsymbol{\omega}_i(\tau), \boldsymbol{\omega}_G^a(\tau)) - R)]^+, \quad i = 1, \cdots, N, \tag{26}$$

and $[x]^+ = \max(0, x)$. At initialization, $\lambda(\tau)$ is non-negative, i.e., $\lambda(1) \geq 0$, to ensure that (25) converges. Moreover, $\boldsymbol{\omega}_j^a(\tau)$ is optimized with the updates of $\overline{\omega}_{i,j}$ and $\boldsymbol{\omega}_G^a(\tau)$ according to (3) and (4), respectively, after the global model aggregation.

C. Training Algorithm of AGAT Attack

According to the design of the new AGAT attack in Fig. 3, Algorithm 1 is developed along with the FL training of the benign users and the FL server. Specifically, the environment and parameters, such as the graph structure $\mathcal{G} = (\mathcal{V}, E, \mathcal{F})$, the total number of learning iterations T_L, and the datasets $D_i(\tau)$ and $D_j^a(\tau)$ are initialized. During each training iteration of the FL, benign users train and upload $\boldsymbol{\omega}_i(\tau)$ to the server. The attacker conducts **AGAT**$(\tilde{g}(D_j^a(\tau)), \boldsymbol{\omega}_i(\tau), \lambda(\tau))$ to generate $\boldsymbol{\omega}_j^a(\tau)$, which trains $\tilde{g}(D_j^a(\tau))$ and extracts features using attention coefficients $\widehat{\gamma_{xy}}$ to emphasize correlations. These features are encoded using GAE to produce the adjacency matrix $\mathcal{A}$ that represents connections between model features. The attacker then creates a biased model update in an attempt to manipulate the global model and uploads it to the server. The server aggregates these updates, including the malicious ones, to update $\boldsymbol{\omega}_G^a(\tau)$, which is then distributed across the users, including the benign ones, for the next training iteration. This cyclic process progressively biases the global model, undermining the integrity and effectiveness of the FL. As $\boldsymbol{\omega}_j^a(\tau)$ is highly correlated with $\boldsymbol{\omega}_i(\tau)$ from the benign users, the FL server is unlikely to detect and identify the attackers.

Algorithm 1 The training algorithm of the proposed AGAT attack

1: **1. Initialize**: $\mathcal{G} = (\mathcal{V}, E, \mathcal{F})$, T_L, N, I, d_T, $D_i(\tau)$, and $D_j^a(\tau)$.
Proposed AGAT attack:
2: **for** Training iteration $\tau = 1, 2, 3, \cdots, T_L$. **do**
3: Benign users train the local model updates $\boldsymbol{\omega}_i(\tau)$, $i \in [1, I]$ according to (2).
4: Benign users upload $\boldsymbol{\omega}_i(\tau)$, $i \in [1, I]$ to the server, and the attacker $j \in [1, N-I]$ overhears the benign model updates of its neighbours.
5: The proposed AGAT in Fig. 3 is conducted at the attacker to generate malicious, biasing model updates $\boldsymbol{\omega}_j^a(\tau)$, i.e., **AGAT**$(\tilde{g}(D_j^a(\tau)), \boldsymbol{\omega}_i(\tau), \lambda(\tau))$:
6: Training the surrogate mode $\tilde{g}(D_j^a(\tau)) \rightarrow \boldsymbol{h}$.
7: The attention coefficients $\widehat{\gamma_{xy}} \leftarrow$ (12).
8: Based on (13), $\boldsymbol{h}'$ that contains correlated features is obtained.
9: The GAE encoder encodes $\boldsymbol{h}'$ with $\widehat{\gamma_{xy}}$ (which represents an adjacency matrix $\mathcal{A}$).
10: $\widehat{\mathcal{A}}$ is reconstructed by training the GAE, which maximizes $L(\boldsymbol{\omega}_j^a(\tau), \lambda(t)) - \phi_{\text{loss}}$.
11: According to the proposed graph signal processing module (22) $\sim$ (24), the malicious, biasing model update $\boldsymbol{\omega}_j^a(\tau)$ is obtained and uploaded to the server.
12: At the server, (7b) is checked to detect the biasing model update.
13: According to (4), the server aggregates the selected model updates to generate the global model $\boldsymbol{\omega}_G^a(\tau)$ that is broadcasted to all the users.
14: The benign users conduct training of their next model updates $\boldsymbol{\omega}_i(\tau+1)$ with the received global model, i.e., $\boldsymbol{\omega}_i(\tau) \leftarrow \boldsymbol{\omega}_G^a(\tau)$, $\forall i \in [1, I]$.
15: **end for**

V. Performance Evaluation

This section presents the implementation of the AGAT attack using PyTorch. When subjected to this attack, we assess the training accuracy of both local and global models. Moreover, the detection efficacy of the AGAT attack is examined through the metric of Cosine similarity between the local models and the global one. In addition, the source code for the AGAT attack has been released on GitHub: https://github.com/jjzgeeks/AGAT-basedModelPoisoningAttackFL.

A. Experimental Settings

The benign FL is designed to improve image classification accuracy, while the proposed AGAT attack aims to maximize the KL divergence between the user's model update and the global model, which leads to a biased FL training of the label classification. The total number of users N increases from 6 to 35, while the number of benign users I increases from 5 to 30. The global model $\boldsymbol{\omega}_G^a(\tau)$ in FL is trained with 100 communication rounds, and training of the local model $\boldsymbol{\omega}_i(\tau)$ is carried out in 10 iterations.

For building the architecture of the AGAT, the number of attention heads, the hidden layer size, dropout rate, weight decay, and the number of layers are set to 4, 80, 0.4, 2×10^3, and 2, respectively. The activation function is rectified linear unit (ReLU), as given in (10), for the intermediate layers due to its simplicity and effectiveness

in alleviating the vanishing gradient problem, and softmax is used for the output layer when dealing with classification tasks. For building the adjacency matrix $\boldsymbol{\mathcal{A}}$ in GAE at each attacker, the number of selected model parameters in $\boldsymbol{\omega}_i(\tau)$, i.e., M, is set to 100, 200, or 300. The GAE encoder is a two-layer GCN network with a dropout layer to prevent overfitting. The GAE decoder is an inner product. The Adam optimizer with a learning rate 0.01 is adopted to optimize the network. For all datasets, we use the same encoder, decoder and SVM models.

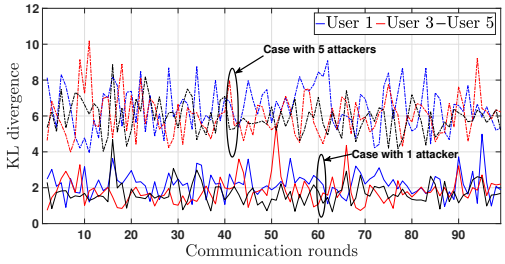
The implementation of the proposed AGAT attack was conducted on a SVM model, utilizing PyTorch version 1.12.1 and Python version 3.9.12. This setup was deployed on a Linux-based workstation, equipped with an Intel(R) Core(TM) i7-9700K CPU at 3.60GHz, featuring 8 cores, and supported by 16 GB of DDR4 memory operating at 2400 MHz. The experimentation involved the application of the AGAT attack across two distinct datasets, demonstrating the attack's efficacy and potential impacts on SVM models under specified computational environments and data conditions:

- The CIFAR-10 dataset [32], consists of 60,000 images in color, each with a dimension of 32×32 pixels, and distributed across ten distinct classes. Each class is represented by 6,000 images. This dataset is organized into two subsets: 50,000 images designated for training purposes and 10,000 images allocated for testing. This structure supports a wide range of image recognition tasks by providing a diverse set of visual inputs for model training and evaluation.
- The Street View House Numbers (SVHN) dataset [33], includes over 600,000 real-world digit images, featuring house numbers in their natural, unsegmented form, captured in a wide range of lighting conditions, angles, and backgrounds.

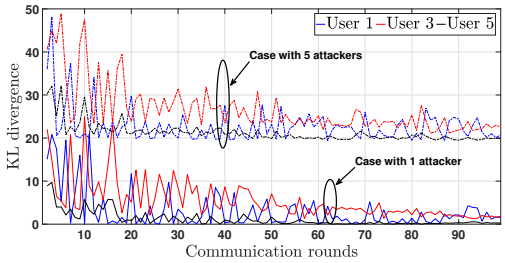
For our experiments, the CIFAR-10 and SVHN datasets are balanced in terms of their class distributions [34], [35]. This characteristic of the datasets is crucial, as our primary interest lies in a new fairness attack on the FL. By training the AGAT based on balanced datasets, we ensure that the baseline conditions of our experiments do not inherently contain biases or imbalances that could confound the effects of the proposed attack. This allows accurate assessment and demonstration of the impact of the AGAT in biasing the FL.

Three key performance metrics are investigated:

- KL divergence measures the difference between the probability distributions of a user's model update and the global model, providing insight into how much a local model deviates from the expected global distribution.
- The local model's testing accuracy assesses to what extent fairness is compromised without reducing FL accuracy under the proposed AGAT attack, making the attack difficult to detect at the server.
- Cosine similarity measures the angular similarity between the local models and the corresponding global model, which is used to evaluate the invisibility of the proposed AGAT attack.

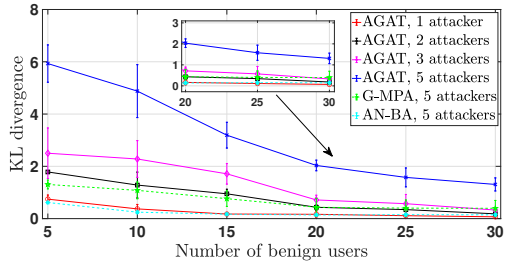


(a) the CIFAR-10 dataset.

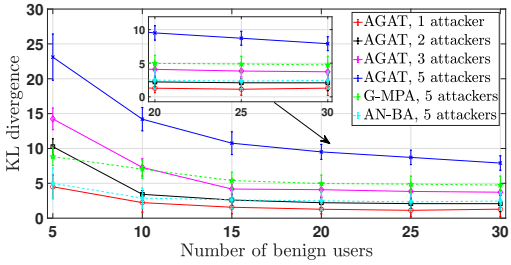


(b) the SVHN dataset.

Fig. 4: Given I = 5, the KL divergence $d_{\mathrm{KL}}(\boldsymbol{\omega}_i(\tau), \boldsymbol{\omega}_G^a(\tau))$ under attacks with one or five attackers.



(a) the CIFAR-10 dataset.

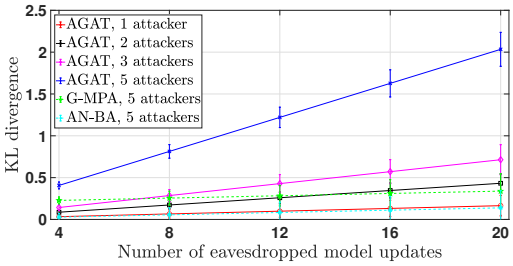


(b) the SVHN dataset.

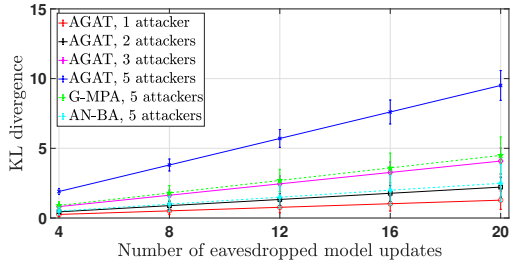
Fig. 5: When I increases from 5 to 30, the KL divergence $d_{\mathrm{KL}}(\boldsymbol{\omega}_i(\tau), \boldsymbol{\omega}_G^a(\tau))$ in the presence of one, two, three, or five attackers.

In addition, the proposed AGAT attack is compared with an existent adversarial GAE-based model poisoning attack (G-MPA), as well as an existing fairness attack on FL, i.e., additive noise-based biasing attack (AN-BA):

- G-MPA focuses on compromising the integrity of



(a) the CIFAR-10 dataset.



(b) the SVHN dataset.

Fig. 6: Given 20 users, the KL divergence $d_{\mathrm{KL}}(\boldsymbol{\omega}_i(\tau), \boldsymbol{\omega}_G^a(\tau))$, where the number of $\boldsymbol{\omega}_i(\tau)$ overheard increases from 4 to 20.

benign local models by fabricating malicious training samples, thereby reducing the test accuracy of these models. This technique has been used in existing works, such as [36] and [37]. Specifically, the G-MPA involves the attacker disrupting the training process through the introduction of a counterfeit user. This fake user transmits malicious local models to the server, effectively manipulating the collective learning outcome.

- AN-BA, considered in [8] and [38], generates malicious local models by injecting a Gaussian random noise into the received global model, which can enlarge the magnitudes of the random local model updates using a scaling factor.

B. Attacking Performance

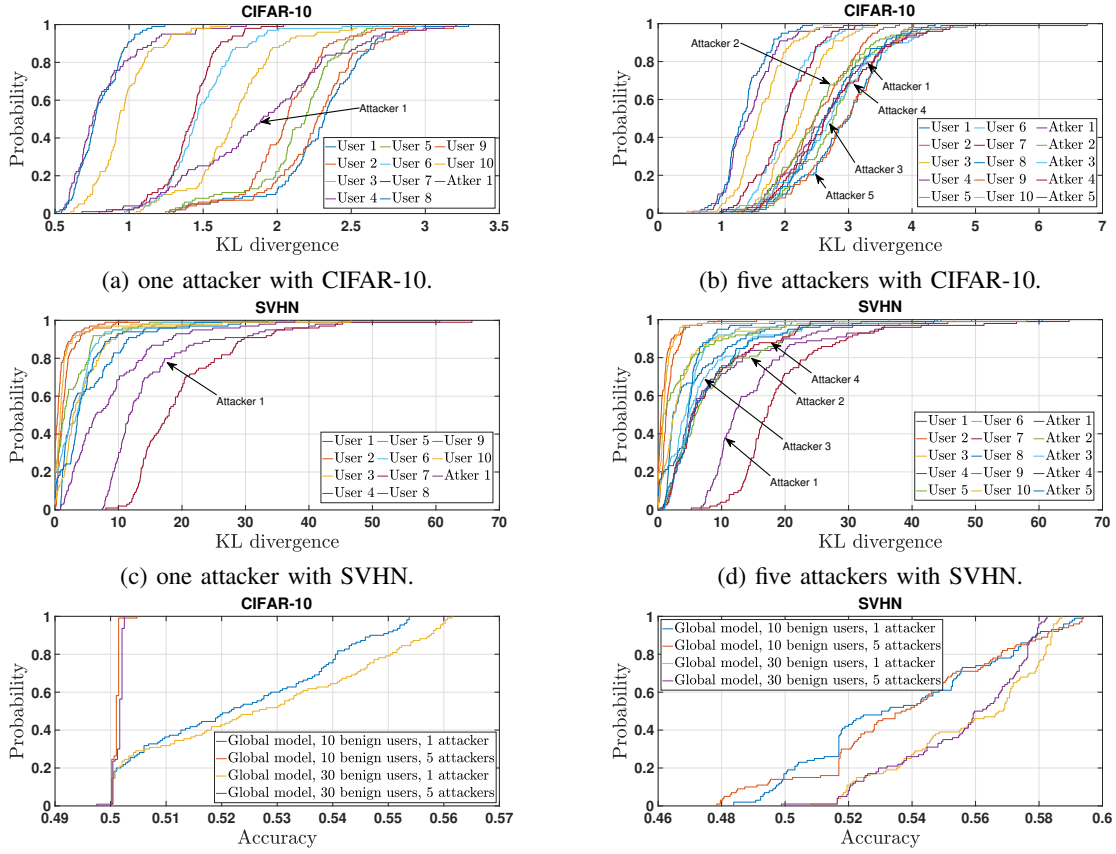
1) KL divergence: Given $I = 5$, Fig. 4 shows the KL divergence of each user i's $\boldsymbol{\omega}_i(\tau)$ under the proposed AGAT attack, i.e., $d_{\mathrm{KL}}(\boldsymbol{\omega}_i(\tau), \boldsymbol{\omega}_G^a(\tau))$ in (5). The performance is tested with the CIFAR-10 dataset in Fig. 4(a) or the SVHN dataset in Fig. 4(b), given one or five attackers in the FL. Generally, the KL divergence given five attackers is about three times higher than the case with a solo attacker. This is reasonable since the increasing number of attackers leads to more malicious, biasing model updates, i.e., $\boldsymbol{\omega}_j^a(\tau)^*$, being aggregated in the FL. Consequently, the maximum loss function with regard to the FL fairness in (6) increases.

In Fig. 5, we conduct a comparative analysis of the average KL divergence pertaining to local models subjected to the proposed AGAT attack versus those affected by the existing G-MPA and AN-BA. This comparison spans an increase in the number of benign users I from 5 to 30, alongside varying numbers of attackers from 1 to 5. Specifically, within the context of the CIFAR-10 dataset and with the presence of five attackers, Fig. 5(a) elucidates that the KL divergence under the AGAT attack exhibits a substantial elevation, 70.2% and 85.4% higher compared to the divergences under the G-MPA and AN-BA, respectively. Similarly, Fig. 5(b) illustrates that, when considering the SVHN dataset, the AGAT attack results in a KL divergence that surpasses that of the G-MPA and AN-BA attacks by 60.9% and 78.6%, respectively. Such findings underscore a significant bias in FL fairness induced by the AGAT attack. This bias stems from our innovative architecture based on GAT and GAE, which tailors the adversarial adjacency matrix in alignment with the unique features of the users' local model updates. As a result, this leads to the generation of maliciously biased model updates aimed at maximizing the loss differential, Δ_{Loss}, as studied in (6).

As shown in Fig. 5, the KL divergence associated with local models increases concomitantly with the augmentation in the number of attackers, given a fixed I. This substantiates the detrimental impact of the proposed AGAT attack on the fairness of FL. Furthermore, the KL divergence decreases with the increment of I, since increasing the number of benign users can fortify the resilience of FL against fairness attacks. This enhancement in resistance is attributable to the aggregation of an increased number of benign local models, which inherently dilutes the adversarial influence exerted by the attackers, thereby preserving the integrity and fairness of the FL process.

Fig. 6 illustrates the KL divergence $d_{\mathrm{KL}}(\boldsymbol{\omega}_i(\tau), \boldsymbol{\omega}_G^a(\tau))$ as the number of overheard updates $\boldsymbol{\omega}_i(\tau)$ increases from 4 to 20, where $I = 20$. It is observed that the KL divergence grows in proportion to the number of model updates overheard. For instance, with five attackers, the KL divergence rises by 75.8%. By contrast, with a single attacker, it increases by 84.6%. This demonstrates that the more benign model updates the attacker can eavesdrop on, the more correlated features the proposed AGAT can exploit to generate malicious, biasing model updates. As a result, the fairness of the FL process is further compromised, with model updates becoming increasingly skewed.

To further study the probability distribution of malicious, biasing model updates generated by the proposed AGAT attack, Fig. 7 plots the cumulative distribution function (CDF) of the KL divergence and global model accuracy. Based on the CIFAR-10 dataset in Figs. 7(a) and 7(b) as well as the SVHN dataset in Fig. 7(c) and 7(d), we observe that the CDF with more attackers results in a higher probability of the KL divergence. More importantly, the CDF of the malicious, biasing model updates exactly follows the same probability distribution pattern as the benign ones. Therefore, it is impossible for the server to identify the attacker. This is achieved by our innovative design of the AGAT architecture, namely, the adversarial GAT captures the correlations existent amongst data features within benign model updates, while the GAE is trained together with sub-gradient descent to reconstruct



(a) one attacker with CIFAR-10.

(b) five attackers with CIFAR-10.

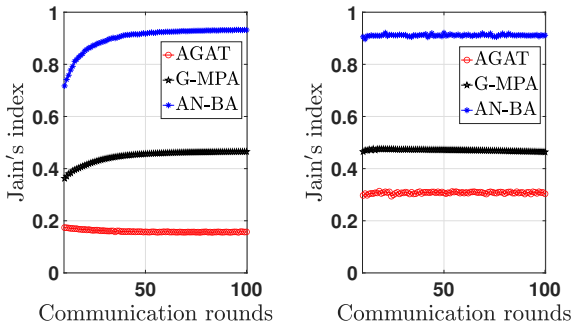
(c) one attacker with SVHN.

(d) five attackers with SVHN.

(e) Global model accuracy with CIFAR-10.

(f) Global model accuracy with SVHN.

Fig. 7: Given I = 10, CDF of the KL divergence and global model accuracy under the proposed AGAT attack.



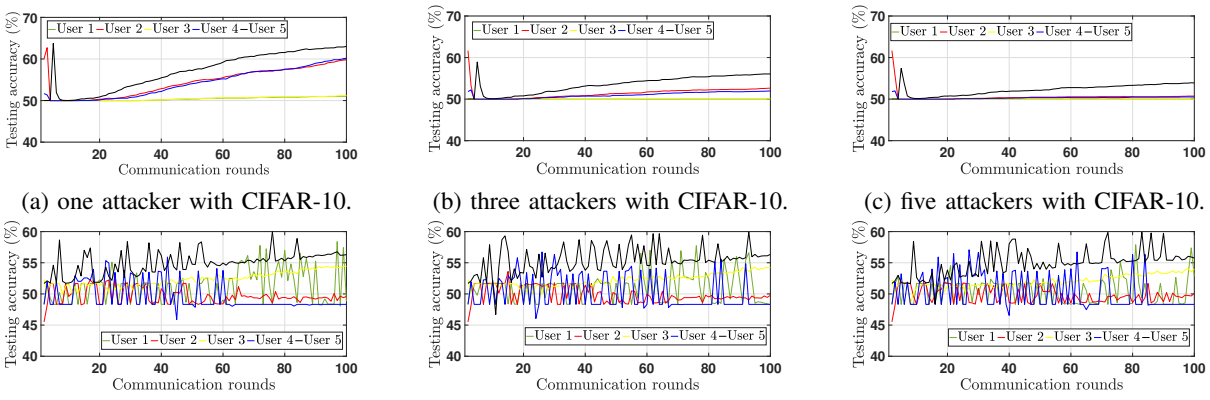
(a) the CIFAR-10 dataset. (b) the SVHN dataset.

Fig. 8: The Jain's fairness index of the model updates, when $I = 20$ and five attackers.

manipulatively the correlations of the model updates, and maximize the reconstruction loss.

Fig. 8 presents the Jain's fairness index for the model updates when $I = 20$ and five attackers are present. Using the CIFAR-10 dataset, the performance of the proposed AGAT method is 25.6% and 74.2% lower than that of G-MPA and AN-BA, respectively. When evaluating the SVHN dataset, the Jain's fairness index for AGAT is 15.7% and 61.4% lower compared to G-MPA and AN-BA, respectively. These results confirm that the AGAT attack significantly compromises fairness in FL. This bias is driven by our novel architecture, which leverages GAT and GAE to adaptively construct an adversarial adjacency matrix aligned with the distinct features of the users' local model updates.

2) FL accuracy: Given I = 5, Fig. 9 shows the local model's testing accuracy under the proposed AGAT attack, based on the CIFAR-10 and SVHN datasets. As observed from Figs. 9(a) to 9(f), despite the adversarial interventions, the FL accuracy not only remains unaffected but also continues to converge. This is attributed to the objective of the AGAT attack, which diverges from traditional adversarial tactics by specifically aiming to bias the FL, rather than undermining their testing accuracy. Furthermore, the new adversarial GAT design captures the correlations among data features within benign model updates, thereby skewing the model's decision boundaries without detrimentally affecting the FL accuracy. This nuanced strategy highlights a sophisticated attack vector that compromises the fairness and integrity of the FL without the conventional hallmark



(a) one attacker with CIFAR-10. (b) three attackers with CIFAR-10. (c) five attackers with CIFAR-10.

(d) one attacker with with SVHN. (e) three attackers with SVHN. (f) five attackers with SVHN.

Fig. 9: Given 100 FL communication rounds and $I = 5$, the local model's testing accuracy under the proposed AGAT attack on the CIFAR-10 and SVHN datasets.

of reduced accuracy, thus posing a more insidious threat that can elude standard detection mechanisms.

3) Cosine similarity: To evaluate the invisibility of the proposed AGAT attack, we further investigate the Cosine similarity between the local and the global models [39], i.e., Constraint (7b), based on the CIFAR-10 and SVHN datasets in Fig. 10. As shown in Figs. 10(a), 10(b), 10(c) and 10(d), the Cosine similarities between the malicious, biasing model updates generated by the new AGAT attack and the corresponding global models are always below that of the benign local model updates. This complicates detecting and defending against fairness biases at the server as malicious updates can blend with legitimate data. In contrast, as depicted in Figures 10(e) and 10(f), both the G-MPA method and the AN-BA approach lead to a markedly higher Cosine similarity between the malicious, biasing model updates and the aggregate global models. This increased similarity offers a clearer signal for detection mechanisms. This contrast underlines the superior tactical advantage of the proposed AGAT attack: AGAT crafts malicious model updates by exploiting the feature correlations between benign local updates and the global model. This strategy effectively obfuscates the distinctions between benign and malicious updates, rendering the latter virtually undetectable and showcasing the sophistication and potential efficacy of AGAT in compromising model integrity.

VI. Conclusions and Future Work

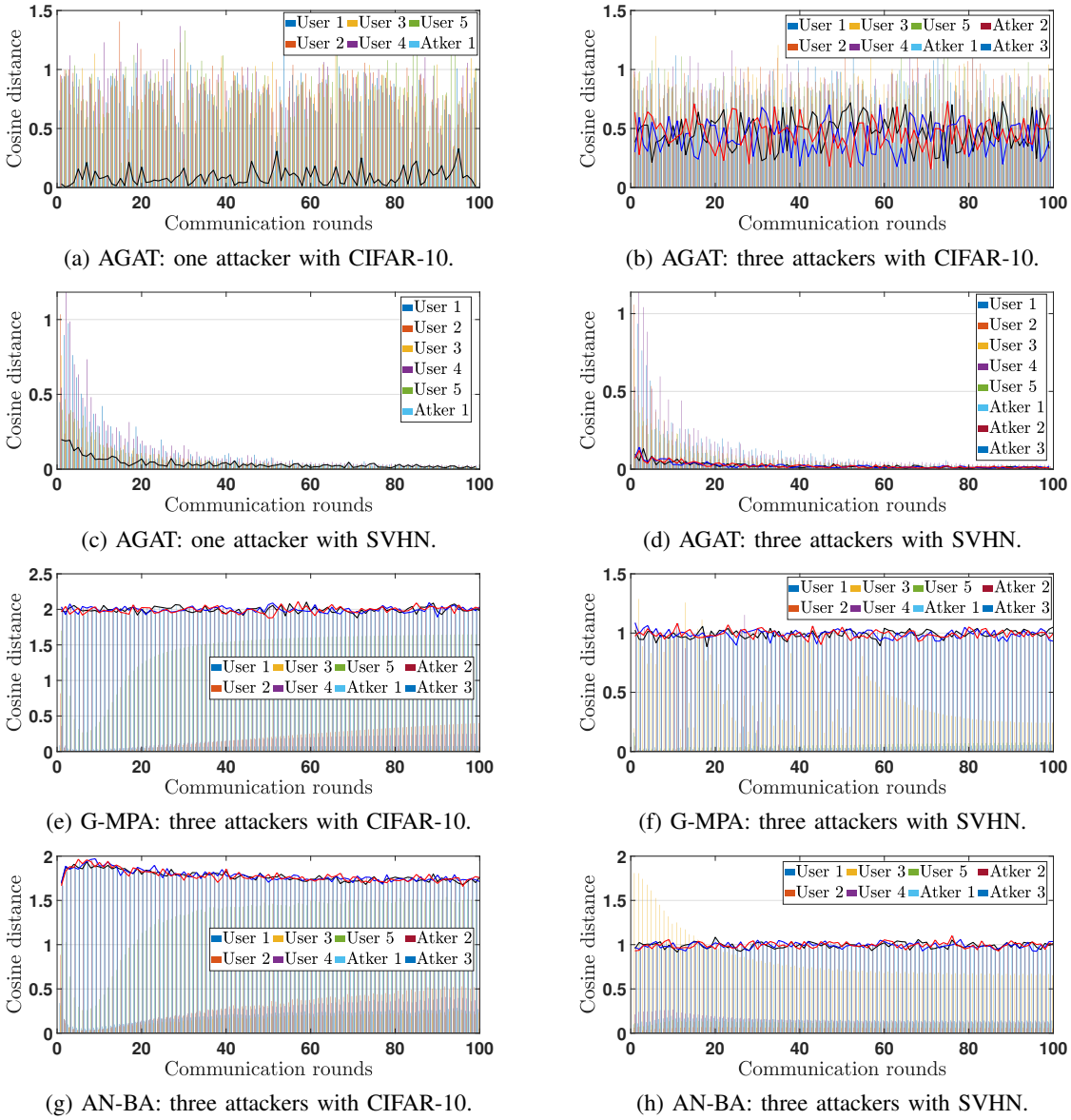
In this paper, a new AGAT architecture was proposed to intentionally instigate fairness attacks with an aim to bias the learning process across the FL. The proposed AGAT was developed to synthesize malicious, biasing model updates, which capture the correlations among data features within benign model updates. Moreover, an adversarial GAE was designed within the AGAT architecture, which can be trained together with sub-gradient descent to manipulatively reconstruct the correlations of the model updates and maximize the reconstruction loss while keeping the malicious, biasing model updates undetectable. The proposed AGAT attack was implemented in PyTorch, showing experimentally that AGAT successfully increases the minimum value of KL divergence of benign model updates by 60.9% and bypasses detection of the existing defense model. The source code of the AGAT attack is released on GitHub.

For future work, we aim to extend the proposed AGAT attack to DFL, where the users collaborate by sharing updates directly with their peers. We will explore how the AGAT attack adapts to decentralized communication patterns, and assess its effectiveness in this context. Moreover, future work could explore the development of a defense model against the AGAT attack, where the model updates can be dynamically selected and weighted. By using a reward mechanism based on metrics, such as the consistency and reliability of updates over time, the server can iteratively adjust its strategy to minimize the influence of potentially malicious updates. The defense model can be developed to allow the server to learn from historical data, gradually identifying and reducing the weight of updates that exhibit unusual or biased behavior, as might be induced by the AGAT attack.

Acknowledgements

This work was supported by the CISTER Research Unit (UIDP/UIDB/04234/2020) and project ADANET (PTDC/EEICOM/3362/2021), financed by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology); and also supported in part by the AXA Research Fund (AXA Chair for Internet of Everything at Koç University).

The authors would like to thank Dr. Petar Veličković (a Staff Research Scientist at Google DeepMind and Affiliated Lecturer at the University of Cambridge, https://petar-v.com/) for his assistance with the formulation of the AGAT architecture, and constructive comments on the article.



(a) AGAT: one attacker with CIFAR-10.

(b) AGAT: three attackers with CIFAR-10.

(c) AGAT: one attacker with SVHN.

(d) AGAT: three attackers with SVHN.

(e) G-MPA: three attackers with CIFAR-10.

(f) G-MPA: three attackers with SVHN.

(g) AN-BA: three attackers with CIFAR-10.

(h) AN-BA: three attackers with SVHN.

Fig. 10: Given 100 FL communication rounds and $I = 5$, the Cosine similarities of the local models are measured at the server in order to detect an adversarial attack, based on the CIFAR-10 and SVHN datasets.

REFERENCES

[1] Y. Zheng, S. Lai, Y. Liu, X. Yuan, X. Yi, and C. Wang, "Aggregation service for federated learning: An efficient, secure, and more resilient realization," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 988–1001, 2022.

[2] K. Li, Y. Cui, W. Li, T. Lv, X. Yuan, S. Li, W. Ni, M. Simsek, and F. Dressler, "When internet of things meets metaverse: Convergence of physical and cyber worlds," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 4148–4173, 2022.

[3] C. Dong, J. Weng, M. Li, J.-N. Liu, Z. Liu, Y. Cheng, and S. Yu, "Privacy-preserving and byzantine-robust federated learning," *IEEE Transactions on Dependable and Secure Computing*, 2023.

[4] J. Le, X. Lei, N. Mu, H. Zhang, K. Zeng, and X. Liao, "Federated continuous learning with broad network architecture," *IEEE Transactions on Cybernetics*, vol. 51, no. 8, pp. 3874–3888, 2021.

[5] N. Chen, G. Akar, S. I. Gordon, and S. Chen, "Where do you live and what do you drive: Built-environmental and spatial effects on vehicle type choice and vehicle use," *International Journal of Sustainable Transportation*, vol. 15, no. 6, pp. 444–455, 2021.

[6] W. Hao, M. El-Khamy, J. Lee, J. Zhang, K. J. Liang, C. Chen, and L. C. Duke, "Towards fair federated learning with zero-shot data augmentation," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 3310–3319.

[7] D. Y. Zhang, Z. Kou, and D. Wang, "Fairfl: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models," in *IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 1051–1060.

[8] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *Proceedings of International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 6357–6368.

[9] Z. Ma, J. Ma, Y. Miao, Y. Li, and R. H. Deng, "ShieldFL: Mitigating model poisoning attacks in privacy-preserving federated learning,"

IEEE Transactions on Information Forensics and Security, vol. 17, pp. 1639–1654, 2022.

[10] Z. Li, J. Liu, and K. Wu, "A multiobjective evolutionary algorithm based on structural and attribute similarities for community detection in attributed networks," *IEEE Transactions on Cybernetics*, vol. 48, no. 7, pp. 1963–1976, 2017.

[11] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to {Byzantine-Robust} federated learning," in *USENIX security symposium*, 2020, pp. 1605–1622.

[12] K. Li, J. Zheng, X. Yuan, W. Ni, O. B. Akan, and H. V. Poor, "Data-agnostic model poisoning against federated learning: A graph autoencoder approach," *IEEE Transactions on Information Forensics and Security*, 2024.

[13] M. Kaheni, M. Lippi, A. Gasparri, and M. Franceschelli, "Selective trimmed average: A resilient federated learning algorithm with deterministic guarantees on the optimality approximation," *IEEE Transactions on Cybernetics*, 2024.

[14] K. Li, X. Yuan, J. Zheng, W. Ni, F. Dressler, and A. Jamalipour, "Leverage variational graph representation for model poisoning on federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[15] X. Cao, Z. Zhang, J. Jia, and N. Z. Gong, "Flcert: Provably secure federated learning against poisoning attacks," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 3691–3705, 2022.

[16] X. Chen, H. Yu, X. Jia, and X. Yu, "Apfed: Anti-poisoning attacks in privacy-preserving heterogeneous federated learning," *IEEE Transactions on Information Forensics and Security*, 2023.

[17] T. Qi, F. Wu, C. Wu, L. Lyu, T. Xu, H. Liao, Z. Yang, Y. Huang, and X. Xie, "Fairvfl: A fair vertical federated learning framework with contrastive adversarial learning," *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 7852–7865, 2022.

[18] Y. H. Ezzeldin, S. Yan, C. He, E. Ferrara, and A. S. Avestimehr, "Fairfed: Enabling group fairness in federated learning," in *Proceedings of AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, 2023, pp. 7494–7502.

[19] S. Horvath, S. Laskaridis, M. Almeida, I. Leontiadis, S. Venieris, and N. Lane, "Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout," *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 12 876–12 889, 2021.

[20] L. Lyu, X. Xu, Q. Wang, and H. Yu, "Collaborative fairness in federated learning," *Federated Learning: Privacy and Incentive*, pp. 189–204, 2020.

[21] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the tails: Yes, you really can backdoor federated learning," *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 16 070–16 084, 2020.

[22] S. Awan, B. Luo, and F. Li, "Contra: Defending against poisoning attacks in federated learning," in *Proceedings of European Symposium on Research in Computer Security (ESORICS)*. Springer, 2021, pp. 455–475.

[23] J. Shi, W. Wan, S. Hu, J. Lu, and L. Y. Zhang, "Challenges and approaches for mitigating byzantine attacks in federated learning," in *Proceedings of IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2022, pp. 139–146.

[24] J. Zheng, K. Li, N. Mhaisen, W. Ni, E. Tovar, and M. Guizani, "Exploring deep-reinforcement-learning-assisted federated learning for online resource allocation in privacy-preserving edgeiot," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21 099–21 110, 2022.

[25] S. Park, S. Han, F. Wu, S. Kim, B. Zhu, X. Xie, and M. Cha, "Feddefender: Client-side attack-tolerant federated learning," in *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 1850–1861.

[26] G. Beck, M. Zinkus, and M. Green, "Automating the development of chosen ciphertext attacks," in *Proceedings of USENIX Security Symposium*, 2020, pp. 1821–1837.

[27] Y. Ding, G. Zhu, D. Chen, X. Qin, M. Cao, and Z. Qin, "Adversarial sample attack and defense method for encrypted traffic data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18 024–18 039, 2022.

[28] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.

[29] N. Jiang, W. Jie, J. Li, X. Liu, and D. Jin, "Gatrust: A multi-aspect graph attention network model for trust assessment in osns," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[30] N. Mrabah, M. Bouguessa, M. F. Touati, and R. Ksantini, "Rethinking graph auto-encoder models for attributed graph clustering," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[31] P. Van Mieghem, *Graph spectra for complex networks*. Cambridge university press, 2023.

[32] R. C. Çalik and M. F. Demirci, "Cifar-10 image classification with convolutional neural networks for embedded systems," in *IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 2018, pp. 1–2.

[33] N. Yuval, "Reading digits in natural images with unsupervised feature learning," in *Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[34] S. Park, J. Lim, Y. Jeon, and J. Y. Choi, "Influence-balanced loss for imbalanced visual classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 735–744.

[35] G. K. Nayak, K. R. Mopuri, and A. Chakraborty, "Effectiveness of arbitrary transfer sets for data-free knowledge distillation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1430–1438.

[36] K. Li, X. Yuan, J. Zheng, W. Ni, and M. Guizani, "Exploring adversarial graph autoencoders to manipulate federated learning in the internet of things," in *Proceedings of IEEE International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2023, pp. 898–903.

[37] L. Gao, L. Li, Y. Chen, W. Zheng, C. Xu, and M. Xu, "Fifl: A fair incentive mechanism for federated learning," in *Proceedings of the International Conference on Parallel Processing*, 2021, pp. 1–10.

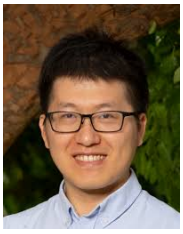
[38] J. Zheng, K. Li, X. Yuan, W. Ni, and E. Tovar, "Detecting poisoning attacks on federated learning using gradient-weighted class activation mapping," in *Companion Proceedings of the ACM on Web Conference (WWW)*, 2024, pp. 714–717.

[39] G. Chen, K. Li, A. M. Abdelmoniem, and L. You, "Exploring representational similarity analysis to protect federated learning from data poisoning," in *Companion Proceedings of the ACM on Web Conference (WWW)*, 2024, pp. 525–528.

Kai Li (S'09–M'14–SM'20) received the B.E. degree from Shandong University, China, in 2009, the M.S. degree from The Hong Kong University of Science and Technology, Hong Kong, in 2010, and the Ph.D. degree in computer science from The University of New South Wales, Sydney, NSW, Australia, in 2014. Currently, he is a Visiting Research Scholar with the School of Electrical Engineering and Computer Science, TU Berlin, Germany, and a Senior Research Scientist with the CISTER Research Centre, Porto, Portugal. He is also a CMU-Portugal Research Fellow, jointly supported by Carnegie Mellon University (CMU), Pittsburgh, PA, USA, and the Foundation for Science and Technology (FCT), Lisbon, Portugal. From 2023 to 2024, he was a Visiting Research Scientist with the Division of Electrical Engineering, Department of Engineering, University of Cambridge, UK. In 2022, he was a Visiting Research Scholar with the CyLab Security and Privacy Institute, CMU. Prior to this, he was a Post-Doctoral Research Fellow with the SUTD-MIT International Design Centre, Singapore University of Technology and Design, Singapore, from 2014 to 2016. He has also held positions as a Visiting Research Assistant with the ICT Centre, CSIRO, Brisbane, QLD, Australia, from 2012 to 2013, and a full-time Research Assistant with the Mobile Technologies Centre, The Chinese University of Hong Kong, Hong Kong, from 2010 to 2011. He has been an Associate Editor of journals, such as *Internet of Things* (Elsevier) since 2024, *Nature Computer Science* (Springer) since 2023, *Computer Communications* (Elsevier) and *Ad Hoc Networks* (Elsevier) since 2021, and IEEE ACCESS from 2018 to 2024.

Jingjing Zheng (S'22) is currently near to completion of pursuing the Ph.D. degree in electrical and computer engineering with the University of Porto, Porto, Portugal. He is a Student Researcher with CISTER Research Center, Porto, Portugal. In 2022, he was a Visiting Research Scholar with the CyLab Security and Privacy Institute, CMU. His main research interests include federated learning, machine learning security, and edge computing.

Wei Ni (M'09–SM'15–F'24) received the B.E. and Ph.D. degrees in communication science and engineering from Fudan University, Shanghai, China, in 2000 and 2005, respectively. He is a Senior Principal Research Scientist with CSIRO and a Conjoint Professor with the University of New South Wales. He has (co)authored one book, ten book chapters, more than 300 journal articles, more than 100 conference papers, 26 patents, and ten standard proposals accepted by IEEE. His research interests include machine learning, online learning, stochastic optimization, and their applications to the security, integrity, and efficiency of network systems. He has been an Editor of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, since 2022; IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, since 2018; IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, since 2024; IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, since 2024; and *Cambridge Press New Research Directions: Cyber-Physical Systems*, since 2022.



Hailong Huang received his Ph.D degree in Systems and Control from the University of New South Wales, Sydney, Australia, in 2018. He was a post-doctoral research fellow at the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia. He is now an Assistant Professor at the Department of Aeronautical and Aviation Engineering, the Hong Kong Polytechnic University, Hong Kong. His current research interests include guidance, navigation, and control of UAVs and mobile robots. He is an Associate Editor of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INTELLIGENT VEHICLES, and INTELLIGENT SERVICE ROBOTICS and *International Journal of Advanced Robotic Systems*, an editorial board member of the *International Journal of Dynamics and Control*.

Pietro Liò received the M.A. degree from the University of Cambridge, the Ph.D. degree in complex systems and non-linear dynamics from the Department of Engineering, School of Informatics, University of Firenze, Italy, and the Ph.D. degree in (theoretical) genetics from the University of Pavia, Italy. He is currently a Full Professor of computational biology with the Computer Laboratory, University of Cambridge, and a member of the Artificial Intelligence Group. His research interests include bioinformatics, computational biology modeling, and machine learning to integrate various types of data (molecular and clinical, drugs, social, and lifestyle) across different spatial and temporal scales of biological complexity to address personalized and precision medicine.

Falko Dressler (F'17) received the M.Sc. and Ph.D. degrees from the Department of Computer Science, University of Erlangen, in 1998 and 2003, respectively. He is currently a Full Professor and the Chair of Telecommunication Networks with the School of Electrical Engineering and Computer Science, TU Berlin. He has been the Associate Editor-in-Chief of IEEE TRANSACTIONS ON MOBILE COMPUTING and *Computer Communications* (Elsevier) and an Editor of journals, such as IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, *Ad Hoc Networks* (Elsevier), and *Nano Communication Networks* (Elsevier). He has been chairing conferences, such as IEEE INFOCOM, ACM MobiSys, ACM MobiHoc, IEEE VNC, and IEEE GLOBECOM. He has authored the textbooks *Self-Organization in Sensor and Actor Networks* (Wiley & Sons) and *Vehicular Networking* (Cambridge University Press). He has been an IEEE Distinguished Lecturer and an ACM Distinguished Speaker. He is an ACM Distinguished Member. He is a member of the German National Academy of Science and Engineering (acatech). He has been serving on the IEEE COMSOC Conference Council and the ACM SIGMOBILE Executive Committee. His research objectives include adaptive wireless networking (sub-6GHz, mmWave, visible light, and molecular communication) and wireless-based sensing with applications in ad-hoc and sensor networks, the Internet of Things, and cyber-physical systems.

Ozgur B. Akan (F'16) received the PhD from the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, in 2004. He is currently the Head of Internet of Everything (IoE) Group, with the Department of Engineering, University of Cambridge, UK and the Director of Centre for neXt-generation Communications (CXC), Koç University, Turkey. His research interests include wireless, nano, and molecular communications and Internet of Everything.