Securing Distributed Network Digital Twin Systems Against Model Poisoning Attacks

Zifan Zhang, Minghong Fang, Mingzhe Chen, *Member, IEEE*, Gaolei Li, *Member, IEEE*, Xi Lin, *Member, IEEE*, and Yuchen Liu, *Member, IEEE*

***Abstract*—In the era of 5G and beyond, the increasing complexity of wireless networks necessitates innovative frameworks for efficient management and deployment. Digital twins (DTs), embodying real-time monitoring, predictive configurations, and enhanced decision-making capabilities, stand out as a promising solution in this context. Within a time-series data-driven framework that effectively maps wireless networks into digital counterparts, encapsulated by integrated vertical and horizontal twinning phases, this study investigates the security challenges in distributed network DT (NDT) systems, which potentially undermine the reliability of subsequent network applications, such as wireless traffic forecasting. Specifically, we consider a minimal-knowledge scenario for all attackers, in that they do not have access to network data and other specialized knowledge, yet can interact with previous iterations of server-level models. In this context, we spotlight a novel fake traffic injection attack designed to compromise a distributed NDT system for wireless traffic prediction. In response, we then propose a defense mechanism, termed global-local inconsistency detection (GLID), to counteract various model poisoning threats. GLID strategically removes abnormal model parameters that deviate beyond a particular percentile range, thereby fortifying the security of network twinning process. Through extensive experiments on real-world wireless traffic data sets, our experimental evaluations show that both our attack and defense strategies significantly outperform existing baselines, highlighting the importance of security measures in the design and implementation of DTs for 5G and beyond network systems.**

***Index Terms*—Digital twin (DT), distributed learning, poisoning attack, security, traffic prediction, wireless networks.**

I. Introduction

IN THE realm of telecommunications, wireless networks are experiencing a paradigm shift, primarily driven by the advent of edge computing, spectrum sharing, and millimeter-wave communication technologies in the 5G era. These technological advancements are foundational to a multitude of novel applications and services, notably enhancing mobile broadband and facilitating the seamless integration of the Internet of Things (IoT) [1], [2], autonomous transportation [3], smart urban infrastructure [4], and remote healthcare delivery [5]. Further, the nascent stages of 6G research are indicative of potential revolutionary leaps in hybrid physical-virtual network technologies, paving the way for ubiquitous and intelligent connectivity worldwide. Parallel to these advancements, the concept of digital twin (DT) has surfaced as a significant technological breakthrough in the mixed reality era [6], [7], [8]. The DTs embody intricate virtual representations of physical entities or systems and gain traction in the context of the Fourth Industrial Revolution. This concept synergistically harnesses the capabilities of IoT, machine learning, and big data analytics, meticulously constructing a comprehensive digital model that mirrors the physical attributes, processes, interconnection, and dynamics of its real-world counterpart. Such models play a pivotal role in facilitating predictive simulations, what-if analysis, and system optimizations within a virtual environment, thereby offering tangible insights into operational challenges and maintenance requirements [9], [10], [11].

While DTs offer a wide range of benefits and applications, ensuring their security remains a critical concern that necessitates a comprehensive understanding and robust countermeasures. Common security threats to DTs include data breaches, unauthorized access, and cyber-attacks, which can disrupt the seamless interaction between the physical and virtual systems [12]. In the realm of wireless networks, these DTs face additional challenges, such as Byzantine attacks, man-in-the-middle attacks, and signal interference, which can severely impact their availability and reliability. Furthermore, robust countermeasures, including encryption techniques and secure communication protocols, are needed to protect DTs from potential adversarial attacks in open wireless environments [13]. As DTs are increasingly integrated into the metaverse applications [14], addressing the security and privacy challenges becomes paramount to ensure a trustworthy user interface [15]. In particular, trust evaluation schemes, such as in [16], have been proposed for using federated learning (FL) in DT systems, aiming to enhance data usage security by evaluating the trustworthiness of participating network entities. In the realm of wireless networks, FL leverages its decentralized nature to facilitate multiple network services. With

Manuscript received 16 May 2024; accepted 14 June 2024. Date of publication 24 July 2024; date of current version 24 October 2024. This work was supported by the National Science Foundation under Award SaTC–2350075, Award CNS–2312138, Award CNS–2312139, and Award SaTC–2350076. *(Corresponding author: Yuchen Liu.)*

Zifan Zhang and Yuchen Liu are with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695 USA (e-mail: zzhang66@ncsu.edu; yuchen.liu@ncsu.edu).

Minghong Fang is with the Department of Computer Science and Engineering, University of Louisville, Louisville, KY 40292 USA (e-mail: minghong.fang@duke.edu).

Mingzhe Chen is with the Department of Electrical and Computer Engineering and Frost Institute for Data Science and Computing, University of Miami, Coral Gables, FL 33146 USA (e-mail: mingzhe.chen@miami.edu).

Gaolei Li and Xi Lin are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: gaolei_li@sjtu.edu.cn; linxi234@sjtu.edu.cn).

Digital Object Identifier 10.1109/JIOT.2024.3421895

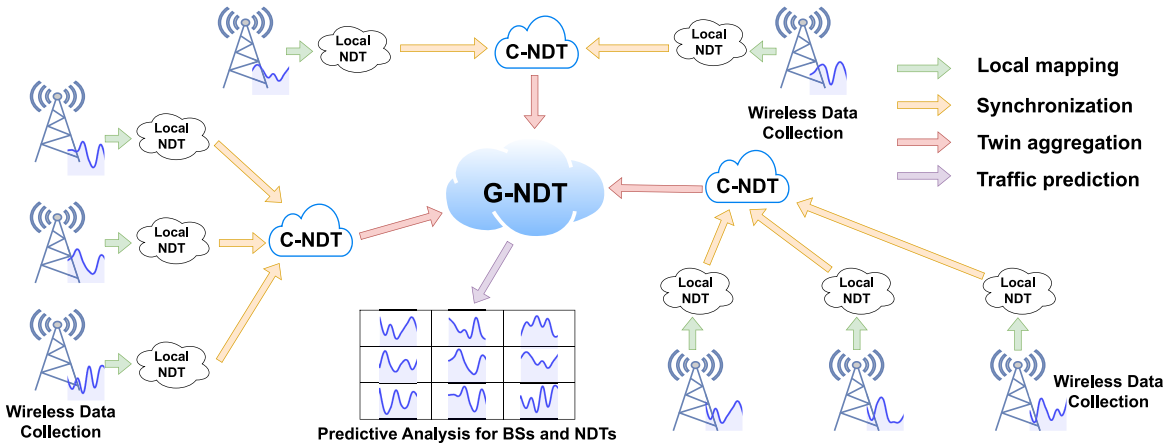


Fig. 1. Distributed NDT framework for WTP.

the exponential growth in the number of connected devices and the ever-increasing demand for data-intensive applications like streaming and IoT services, constructing precise network DTs (NDTs) accurately becomes vital for ensuring various downstream forecasting tasks, such as wireless traffic prediction (WTP) [17], [18], [19]. Despite distributed learning's potential in accuracy, efficiency, and privacy preservation, its integration into NDT creation and operation is not devoid of challenges. Notably, Byzantine attacks, particularly model poisoning attacks, pose significant threats to the effectiveness and trustworthiness of NDT systems.

In a model poisoning attack, malicious network entities introduce adversarial modifications to the model parameters during the mapping process of NDTs. This tampering results in a compromised server-level twin, i.e., global twin model, when aggregated at the central network controller, subsequently producing incorrect operations on the physical infrastructure. Such inaccuracies lead to the risk of network inefficiencies and even severe service disruptions, especially in real-time applications like autonomous driving systems. In more extreme scenarios, these attacks may serve as gateways to further malicious network intrusions, instigating broader security and privacy concerns as illustrated in [20] and [21]. The grave implications of model poisoning attacks underscore the pressing need for robust security measures to ensure the integrity, reliability, and resilience of distributed NDT systems against Byzantine failures, thereby safeguarding the overarching network infrastructure and the services reliant on it. While most existing DT mapping algorithms and their associated security strategies are typically assessed within the context of *classification* problems [22], [23], scant attention has been paid to the *regression* problems, as observed in examined WTP scenarios within NDTs, introducing distinct challenges related to data distribution, model complexity, and evaluation metrics. The distinction between data manipulation strategies in regression and classification problems, as well as their detection methodologies, underscores the nuanced challenges in safeguarding twin models against emerging adversarial attacks. For instance, in a regression-based DT-assisted WTP problem, attackers typically target the model's continuous output by altering the distribution or magnitude of input time-series data, intending to steer predictions in a specific direction. This differs from classification tasks, where the manipulation revolves around modifying input features to induce misclassification without noticeably changing the input's appearance to human observers.

To bridge this gap, we make the first attempt to introduce a novel attack centered on injecting disruptive traffic data from malicious NDTs into wireless networks. Existing model poisoning attacks have predominantly depended on additional access knowledge and direct intrusions on physical base stations (BSs) [22], [24], [25]. However, in a practical cellular network system, BSs have exhibited a commendable level of resilience against attacks, making the extraction of training data from them a challenging endeavor. In contrast, the cost of deploying fake NDTs that mimic their behaviors is comparatively lower than the resources required for compromising authentic BSs [26]. This assumption asserts that these compromised NDTs lack insight into the training data and only have access to the initial and current global twin models, aligning with the practical settings studied in [26]. Importantly, other information, such as data aggregation rules and model parameters from benign NDTs or BSs, remains inaccessible to these compromised NDTs. In this work, we consider a distributed DT-assisted network architecture as depicted in Fig. 1, where wireless traffic data collected from BSs is mapped into local NDTs to establish an initial and private NDT for each BS. Within each cluster, a cluster-level NDT (C-NDT) is constructed by aggregating these local twin models. Subsequently, at the backend, a global twin model (G-NDT) is established by merging the C-NDT model parameters during each iterative phase. This global twin model is then synchronized with each local NDT, serving as a foundation for predictive analysis and enabling specific applications for each BS and its associated NDTs. In this situation, our threat model envisions a minimum-knowledge scenario for an adversary. First, we propose fake traffic injection (FTI), a methodology designed to create undetectable fake NDTs with minimal prior knowledge. Each fake NDT employs both its initial model and current global information to determine the optimizing

trajectory of the twinning process, as shown in Fig. 4. These malicious participants aim to subtly align the global model toward an outcome that undermines the integrity and reliability of the NDT system. Numerous numerical experiments are conducted to validate that our FTI demonstrates efficacy across various state-of-the-art model aggregation rules, outperforming other poisoning attacks in terms of vulnerability impacts.

On the contrary, we propose an innovative defensive strategy known as global-local inconsistency detection (GLID), aimed at neutralizing the effects of model poisoning attacks on NDT systems. This defense scheme involves strategically removing abnormal model parameters that deviate beyond a specific percentile range estimated through statistical methods in each dimension. Such an adaptive approach allows us to trim varying numbers of malicious model parameters instead of a fixed quantity [27]. Next, a weighted mean mechanism is employed to update the global twin model parameter, subsequently disseminated back to each NDT. Our extensive evaluations, conducted on real-world data sets, demonstrate that the proposed defensive mechanism substantially mitigates the impact of model poisoning attacks on NDT systems, thereby showcasing a promising avenue for securing distributed NDT systems with trustworthiness. This article is an extended version of our previous work in [28], where we expand upon it by adapting the proposed attack and defense strategies from traditional FL settings to the practical NDT system.

The contributions are briefly summarized into three folds: 1) we present a novel model poisoning attack, employing fake NDTs for traffic injection into distributed NDT systems under a minimum-knowledge scenario; 2) conversely, we propose an effective defense strategy tailored to counteract various model poisoning attacks, which proactively trims an adaptive number of twin model parameters by leveraging the percentile estimation technique; and 3) finally, we evaluate both the proposed poisoning attack and the defensive mechanism using real-world traffic data sets from Milan City, where the results demonstrate that the FTI attack indeed compromises distributed NDT systems, and the proposed defensive strategy proves notably more effective than other baseline approaches in mitigating various attacks.

II. Related Works and Preliminaries

A. Distributed Network Digital Twin Systems

The integration of DTs into the realm of wireless networking represents a significant leap forward in this rapidly evolving field. As outlined in [29], the use of DTs involves the creation of detailed virtual replicas of network components and infrastructure. This approach enables real-time analytics and optimization, providing deep insights into network behavior under various scenarios. Such strategies are crucial for predictive maintenance and performance monitoring, greatly enhancing network reliability and efficiency. Furthermore, Wang et al. [30] introduced the use of Graph Neural Networks to enhance DTs in network slicing, aimed at predicting network performance and optimizing resources in high-bandwidth and low-latency scenarios. Additionally, the application of DTs in vehicular networks is detailed in [31], showcasing DTs' ability to model and control software-defined vehicular networks, thereby improving the effectiveness and reliability of vehicular communications. Moreover, Yin et al. [32] proposed a DT-assisted security scheme for multiresource heterogeneous RANs in space-air-ground integrated networks. Despite various explorations into DT applications within wireless networks, there is a gap in the literature regarding the development and mapping of NDTs, which our research seeks to address.

At the forefront of DTs, distributed learning emerges as a revolutionary approach, especially for large-scale networks and the Industrial IoT. The combination of these technologies not only enhances system efficiency but also transforms data handling capabilities. Zhang et al. [33] proposed a Joint Vertical and Horizontal Learning-based digital twinning strategy to perform a precise mapping from physical networks to DTs. Zhang et al. [34] exemplified their potential in reliable edge caching and real-time data-driven optimization. Additionally, addressing the challenge of efficient data communication in distributed learning systems, Lu et al. [35] and [36] proposed strategies to enhance data exchange and processing—crucial for scaling up applications with massive access.

B. Poisoning Attacks on Distributed Systems

The decentralized architecture of distributed DT systems renders them vulnerable to Byzantine attacks, as explored in previous study [22], [23], [24], [26], [37], [38]. In these scenarios, adversaries can compromise BSs and their corresponding NDT models to undermine the entire distributed DT system. These malicious BSs may tamper with their local training data or directly modify their local twin models to negatively impact the global twin model. For example, the Trim attack [22] involves malicious BSs deliberately distorting their local twin models to create a significant discrepancy in the aggregated model post-attack compared to its preattack state. The MPAF attack [26] sees each compromised BS applying a negative scalar to the global twin model update before forwarding this tampered update to the server-level twin. In the Random attack [22], malicious DTs generate and send a random vector, drawn from a Gaussian distribution, to the server as their update. Furthermore, a recent study by [24] introduced specific poisoning attacks targeting distributed DT systems. Here, an attacker manipulates some BSs under their control, each with its local training data set. These DTs adjust their local models using this data and then scale their model updates by a factor before dispatching these altered updates to the server. These strategies highlight the critical challenge of securing the entire system against various forms of data and model tampering attacks, underscoring the need for robust defense mechanisms.

Existing attacks in our considered setting suffer from the following limitations. In the MPAF attack, model updates from fake NDTs are exaggerated by a factor, such as 1×10^6. However, this approach is impractical because the server can easily identify these excessive updates as anomalies and discard them. Furthermore, such blatant manipulation lacks

subtlety, making it easy to detect and counter. On the other hand, our method involves carefully crafting model updates on fake clients by solving an optimization problem. This ensures that the server is unable to differentiate these fake updates from benign ones, allowing the attacker to simultaneously breach the integrity of the system without detection. Our approach maintains the updates within a plausible range, avoiding the pitfalls of easily detectable anomalies. The attack described in [24] is not feasible because it is based on the unrealistic assumption that an attacker can easily take control of authentic BSs or DTs. In reality, it is highly challenging for an attacker to gain such influence over existing, authentic facilities. Moreover, this attack does not consider the sophisticated security measures typically in place to protect these systems. Our research, however, focuses on developing more realistic attack scenarios that account for the complexities and security protocols of modern distributed DT systems, ensuring a more accurate assessment of their vulnerabilities.

C. Byzantine-Robust Aggregation Rules

In environments free from adversarial intentions, server-level twins typically aggregate incoming local twin model updates through a simple averaging process [39]. However, recent studies [28], [40] have revealed vulnerabilities in this averaging method of aggregation, particularly its susceptibility to poisoning attacks. In such attacks, a single malicious local twin model can significantly alter the aggregated result. To counter these vulnerabilities, the literature offers a range of Byzantine-resistant aggregation algorithms [27], [40], [41], [42], [43], [44], [45]. For instance, the Krum method [40] assesses each local twin's update by calculating the sum of Euclidean distances to updates from other twin models, selecting the update with the smallest sum for global aggregation. The Median aggregation strategy [27] involves the server-level twin computing median values across each parameter from all local updates, improving resistance to outlier manipulations. These strategies introduce robustness against adversarial actions, ensuring the integrity of the aggregated twin model in distributed systems.

III. Creation and Synchronization of Network Digital Twins for Wireless Traffic Prediction

This section introduces a novel framework for creating and synchronizing NDTs specifically designed for WTP. The framework is structured around three main stages: 1) dynamic connectivity segmentation (DCS); 2) vertical twinning (V-twinning); and 3) horizontal twinning (H-twinning). The overall framework is shown in Fig. 1. The primary objective of the NDTs is to minimize prediction errors across all BSs for a better understanding of the physical network. This can be formulated as an optimization problem

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \frac{1}{Mz} \sum_{m=1}^{M} \sum_{n=1}^{z} F\big(f\big(r_m^n, \boldsymbol{\alpha}\big), s_m^n\big) \tag{1}$$

where F is the quadratic loss function, M is the number of NDTs, z is the number of data points, r_m^n is the input traffic sequence, and s_m^n is the corresponding output traffic prediction. The optimization problem is resolved through FL with distributed NDTs, following the synchronization, local updating, and model aggregation process. This single-level mapping approach utilizes a classical FL strategy to aggregate multiple local twin models, serving as the baseline for comparing with our proposed joint vertical-horizontal mapping scheme.

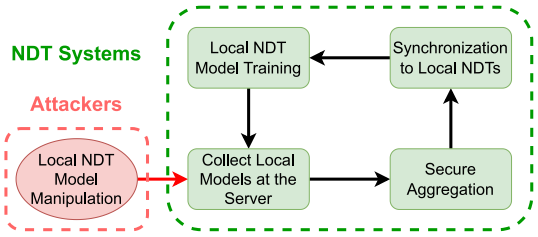


Fig. 2. Process of model poisoning attack and secure aggregation defense.

Specifically, (1) can be resolved in a distributed fashion in traditional FL settings with the following three steps in each global training round t, as shown in Fig. 2.

1) *Step I (Local Twin Update):* Each NDT $i \in [n]$ utilizes its private time-series training data along with the current global model to refine its own local model, then transmits the updated local model $\boldsymbol{\theta}_i^t$ back to a central server.
2) *Step II (Local Twin Manipulation/Model Poisoning Attack):* Each malicious NDT utilizes its knowledge to modify or create local twin models, and then send these malicious twin models to the server.
3) *Step III (Aggregation of Local Twin Models):* The central server leverages the aggregation rule (AR) to merge the n received local models and subsequently updates the global model as follows:

$$\boldsymbol{\theta}^{t+1} = \mathrm{AR}\big\{\boldsymbol{\theta}_1^t, \boldsymbol{\theta}_2^t, \ldots, \boldsymbol{\theta}_n^t\big\}. \tag{2}$$

The commonly used aggregation rule is the FedAvg [39], where the server simply averages the received n local models from distributed NDTs, i.e., $\mathrm{AR}\{\boldsymbol{\theta}_1^t, \boldsymbol{\theta}_2^t, \ldots, \boldsymbol{\theta}_n^t\} = (1/n)\sum_{i=1}^{n} \boldsymbol{\theta}_i^t$.
4) *Step IV (Synchronization):* The central server sends the current global model $\boldsymbol{\theta}^t$ to all NDTs.

Specifically, our multilevel mapping framework encompasses a central DT, named global NDT (G-NDT), coordinating with a network of M NDTs, and multiple cluster NDT (C-NDT). Each NDT, denoted as m in the set $[M]$, independently holds a proprietary data set $d_m = \{d_m^1, d_m^2, \ldots, d_m^L\}$. In this data set, L indicates the total number of time intervals, and d_m^l represents the traffic load at NDT m during the lth interval, where l ranges over $[L]$. The NDT involves constructing input–output predictive traffic sequences locally, denoted as $\{r_m^n, s_m^n\}_{n=1}^{z}$, for each NDT to generate future traffic predictions. Here, r_m^n is a suNDTet of historical traffic data corresponding to the output $s_m^n = \{d_m^{l-1}, \ldots, d_m^{l-a}, d_m^{l-\rho 1}, \ldots, d_m^{l-\rho b}\}$. The parameters a and b represent sliding windows that capture immediate and cyclical temporal dependencies, respectively, while ρ reflects inherent

periodicities in the network, which might be influenced by user activity patterns or application service demands.

The first stage, DCS, is employed periodically to ensure effective clustering of NDTs with similar communication characteristics and networking configurations. This clustering step is integral to the efficient creation and updates of multiple distributed NDTs, i.e., C-NDTs, which demonstrate distinct behaviors and perform parallel synchronization with the G-NDT. The DCS algorithm clusters the NDTs based on attributes, such as geological distances, capacity of backhaul links, coverage area overlaps, and similarity of frequency of occurrence distribution. The relationship between two NDTs, n_1 and n_2, is quantified by a metric Φ_{n_1,n_2}

$$\Phi_{n_1,n_2} = \frac{\omega_g}{g_{n_1,n_2}} + \omega_k \cdot k_{n_1,n_2} + \omega_\beta \cdot \beta_{n_1,n_2} + \omega_\tau \cdot \tau_{n_1,n_2} \quad (3)$$

where ω represents the weights for each attribute. This dynamic clustering enhances the twinning performance in real time and forms the basis for accurate WTP by grouping NDTs with similar traffic patterns.

In the V-Twinning stage, initial NDTs are created with historical data on caching requests and their frequency. It employs an FL strategy, where model parameters are shared among NDTs instead of raw data, enabling collaborative training of a global model. This approach efficiently distributes twinning tasks across NDTs while ensuring content data privacy. Specifically, the V-Twinning stage initializes a concrete G-NDT and synchronizes C-NDTs with the G-NDT after the twinning aggregation process. The aggregation of C-NDTs to form the G-NDT is given by

$$\boldsymbol{\alpha}^{t+1} = \frac{1}{C}\sum_{c=1}^{C} \boldsymbol{\alpha}_c^t \quad (4)$$

where $\boldsymbol{\alpha}_c^t$ represents the model parameters of the C-NDT for cluster c at time t and C is the number of clusters. This stage is crucial for initializing the NDTs with historical traffic data, which serves as a foundation for future traffic prediction.

The H-Twinning stage is designed to periodically synchronize the physical network and NDTs with real-time data. It adopts an asynchronous FL approach to update with dynamics from the physical network, providing a scalable and flexible solution for wireless networks composed of multiple clusters. This stage updates the twins regularly, ensuring that all NDTs remain relevant and accurately simulate and predict wireless traffic patterns. The update rule for the G-NDT based on the deviation ϵ between a C-NDT and the current G-NDT is as follows:

$$\boldsymbol{\alpha}^{t+1} = \begin{cases} \frac{1}{C}\sum_{c=1}^{C} \boldsymbol{\alpha}_c^t, & \text{if } \epsilon > \psi \\ \boldsymbol{\alpha}^t, & \text{otherwise} \end{cases} \quad (5)$$

where ψ is a predefined threshold, and $\epsilon = (\boldsymbol{\alpha}_c^t - \boldsymbol{\alpha}^t)^2$ measures the deviation between the C-NDT and the G-NDT. This stage is critical for incorporating real-time traffic data into the NDTs, enabling them to adapt to changing network conditions and improve traffic prediction accuracy.

IV. Threat Model for Distributed Network Digital Twins

Built upon the constructed distributed NDT system, this section discusses the threat model and explores a novel attack that poses a security breach to system functionality and network operations.

A. Objective of the Attacker

The fundamental aim of an attacker targeting a distributed NDT system is to impair the performance of the composite global twin model significantly. Such impairment directly undermines the precision of real-time traffic forecasts, which is crucial for effective network management and resource distribution. The ramifications of compromised traffic predictions include network congestion, diminished service quality, and suboptimal resource utilization, presenting considerable operational hurdles for network operators. This disturbance extends beyond the service providers, affecting end-users dependent on stable and efficient network services.

B. Capabilities of the Attacker

To achieve their goal, attackers introduce counterfeit NDT models into the system, as illustrated in Fig. 3. These fabricated NDTs can replicate the functionality of legitimate NDTs with minimal investment and effort. This tactic, which entails deploying fake BSs and NDTs using readily available open-source tools or emulators [26], [46], [47], [48], presents a low-barrier, high-feasibility threat vector distinct from the strategies like those in [24] that require compromising actual NDTs. Given the stringent security measures of contemporary networks, which complicate the direct manipulation of authentic twin models, this approach of deploying spurious BSs and NDTs emerges as a notably viable method for attack.

C. Knowledge of the Attacker

The attacker's limited understanding of the intricacies of the targeted distributed NDT system adds to the challenge of mounting a successful attack. In many practical scenarios, acquiring comprehensive knowledge about the aggregation algorithms or details of legitimate NDTs proves exceedingly difficult due to robust security measures and encryption. Consequently, an attack necessitating minimal specialized knowledge and training data not only appears more feasible but also carries a lower risk of detection. The operation of the counterfeit NDTs—receiving the global model and dispatching malicious updates—demands only basic intelligence, effectively lowering the threshold for entry for would-be attackers. This characteristic of the threat model heightens its potential danger, broadening the pool of possible adversaries to include those with scant technical skills or resources.

D. Fake Traffic Injection Attack

The proposed Algorithm 1, named the FTI Algorithm, presents a strategy for a Byzantine model poisoning attack aimed at compromising the prediction accuracy of an NDT system under specific assumptions.

Algorithm 1 FTI

Require: Current global twin model $\boldsymbol{\theta}^t$, base model $\hat{\boldsymbol{\theta}}$, n benign NDTs, m fake NDTs, η
Ensure: Fake models $\boldsymbol{\theta}_i^t, i \in [n+1, n+m]$
1: step $\leftarrow \eta$
2: PreDist $\leftarrow -1$
3: **for** $r = 1, 2, \ldots, R$ **do**
4: **for** each fake NDT i **do**
5: $\boldsymbol{\theta}_i^t \leftarrow \eta\hat{\boldsymbol{\theta}} - (\eta - 1)\boldsymbol{\theta}^t$
6: **end for**
7: Dist $\leftarrow \left\|\boldsymbol{\theta}_i^t - \boldsymbol{\theta}^t\right\|_2$
8: **if** PreDist $<$ Dist **then**
9: $\eta \leftarrow \eta + \frac{\text{step}}{2}$
10: **else**
11: $\eta \leftarrow \eta - \frac{\text{step}}{2}$
12: **end if**
13: step $\leftarrow \frac{\text{step}}{2}$
14: PreDist $\leftarrow$ Dist
15: **end for**
16: **return** $\boldsymbol{\theta}_i^t, i \in [n+1, n+m]$

At the core of the FTI attack is an iterative procedure. In each iteration, the current global twin model $\boldsymbol{\theta}^t$ and the base model $\hat{\boldsymbol{\theta}}$ undergo a detailed examination. For each fake BS i, a malicious local model $\boldsymbol{\theta}_i^t$ is constructed by blending the global model $\boldsymbol{\theta}^t$ with the base model $\hat{\boldsymbol{\theta}}$ in a weighted manner, as delineated in line 5 of Algorithm 1. Subsequent to the formation of $\boldsymbol{\theta}_i^t$, its deviation from the global model is assessed using the Euclidean norm, as depicted in line 7. The algorithm then evaluates whether this distance has increased compared to the previous measurement, denoted as *PreDist*. If an increase is observed, indicating that the malicious local model $\boldsymbol{\theta}_i^t$ is diverging further from the global model $\boldsymbol{\theta}^t$, the value of η is incremented. Conversely, if no increase in distance is detected, η is decremented. The adjustment of η is executed in half-steps of its initial value, as outlined in lines 8 to 12.

The algorithm aims to steer the global model toward greater alignment with a predefined base model in each round. Specifically, during the tth round, fake NDTs compute the direction of local model updates, determined by the difference between the current global twin model and the base model, denoted as $\boldsymbol{H} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^t$. Progressing in this direction signifies that the global model is becoming more akin to the base model. A straightforward method to obtain the local model of a fake BS involves scaling $\boldsymbol{H}$ by a factor η. However, this direct approach yields suboptimal attack performance.

Assuming n represents the number of benign NDTs, and the attacker intends to inject m fake NDTs into the system, we propose a method for calculating $\boldsymbol{\theta}_i^t$ for each fake NDT $i \in [n+1, n+m]$

$$\boldsymbol{\theta}_i^t = \eta\hat{\boldsymbol{\theta}} - (\eta - 1)\boldsymbol{\theta}^t. \tag{6}$$

In such scenarios, an attacker tends to opt for a higher η to ensure the sustained effectiveness of the attack, as illustrated in Fig. 4 with an initial η of 10. This remains valid even after the server amalgamates the manipulated local updates from fake NDTs with legitimate updates from benign NDTs.

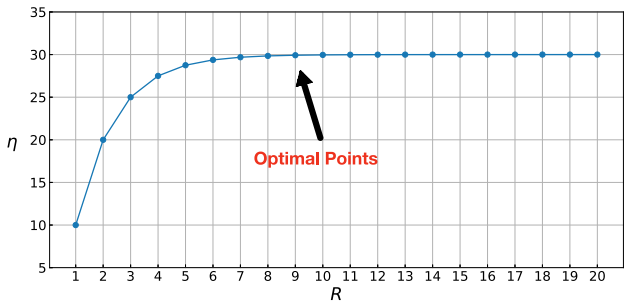


Fig. 3. Optimal value of η over communication round of R.

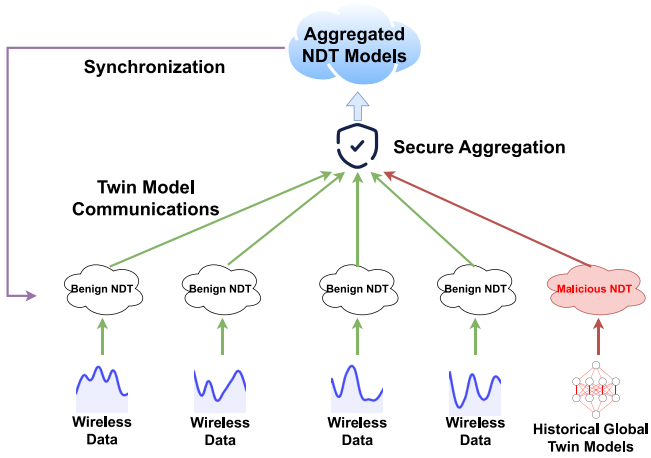


Fig. 4. Framework of NDT system protections.

V. Global-Local Inconsistency Detection

The defense against model poisoning attacks is founded on an aggregation protocol designed to identify malicious NDTs, termed the GLID method, as elaborated in Algorithm 2. In each global round t, GLID primarily examines anomalies present in each dimension of the model parameters $\boldsymbol{\theta}_i^t$, aiding in the identification of potentially malicious entities, where $i \in [1, n+m]$ and $n+m$ denotes the total number of NDTs in the system. This robust and versatile approach enables the system to adapt to various operational contexts without necessitating intricate similarity assessments. Then, choosing the parameter for the percentile range when trimming outliers for secure aggregation becomes crucial, as it directly influences the model's balance between robustness and accuracy. Typically, a narrow percentile range might exclude legitimate variations in data, reducing the model's accuracy and potentially leading to biased or incomplete representations. Conversely, a broad percentile range may fail to eliminate malicious or anomalous data contributions, compromising the model's security by allowing adversarial inputs to skew the aggregation process. Therefore, selecting an appropriate percentile range ensures that most benign data points are retained while effectively filtering out outliers or adversarial inputs. This balance is essential for maintaining both the performance and security of

Algorithm 2 GLID

Require: Local models $\boldsymbol{\theta}_1^t, \boldsymbol{\theta}_2^t, \ldots, \boldsymbol{\theta}_{n+m}^t$, current global model $\boldsymbol{\theta}^t$, k
Ensure: Aggregated global model $\boldsymbol{\theta}^{t+1}$

```
for d = 1, 2, ..., D do
    θ̄_d^t ← (1/(n+m)) Σ_{i=1}^{n+m} θ_{d,i}^t
    σ_d^t ← sqrt((1/(n+m)) Σ_{i=1}^{n+m} (θ_{d,i}^t − θ̄_d^t)^2)
    percentile_d^t ← (g(θ̄_d^t − k·σ_d^t), g(θ̄_d^t + k·σ_d^t))
    Identify malicious NDTs based on percentile pairs
    for each NDT i do
        if θ_{d,i}^t is benign then
            α_{d,i}^t ← σ_d^t / |θ_{d,i}^t − θ̄_d^t|
        else
            α_{d,i}^t ← 0
        end if
    end for
    θ_d^{t+1} ← (Σ_{i=1}^{n+m} α_{d,i}^t · θ_{d,i}^t) / (Σ_{i=1}^{n+m} α_{d,i}^t)
end for
θ^{t+1} ← [θ_1^{t+1}, θ_2^{t+1}, ..., θ_D^{t+1}]
return θ^{t+1}
```

DT models, protecting against data poisoning attacks without sacrificing the overall quality and representativeness of the aggregated twinning data.

Specifically, the GLID approach enhances the detection of potential malicious activities within the network by employing *percentile-based trimming* on each dimension of the model parameters. To establish an effective percentile pair for identifying abnormalities, four statistical methods can be adopted: 1) standard deviation (SD); 2) interquartile range (IQR); 3) z-scores; or 4) one-class support vector machine (One-class SVM).

Suppose the total count of dimensions of the model parameter is D. For the default SD method, the percentile pair for each dimension d can be calculated as follows:

$$\text{percentile pair}_d^t = \left(g\left(\bar{\boldsymbol{\theta}}_d^t - k \cdot \sigma_d^t\right),\ g\left(\bar{\boldsymbol{\theta}}_d^t + k \cdot \sigma_d^t\right)\right) \quad (7)$$

where $\bar{\boldsymbol{\theta}}_d^t$ is the mean of the dth dimension across all models in the tth global training round, σ_d^t is the SD of the dth dimension, and k is a predefined constant dictating the sensitivity of outlier detection. $g(\cdot)$ is the interpolation function based on the SD bound to estimate percentile pairs, defined as

$$g(x) = \left(\frac{P(x) - 0.5}{n+m}\right) \times 100 \quad (8)$$

where $P(x)$ is the position of x in the sorted data set. We use $k = 3$ for general purposes. Given that different tasks may require varied percentile bounds, a precise estimation method is crucial for generalizing our defense strategy. The detailed percentile estimation methods are discussed later in this section. In the FL-based WTP system, model parameters in the dth dimension exceeding these percentile limits are flagged as malicious, and their weights α_i^t are assigned as 0. The other benign values in this dimension are aggregated using a weighted average rule, where the weights $\alpha_{d,i}^t$ are inversely proportional to the absolute deviation of each value $\boldsymbol{\theta}_{d,i}^t$ from the mean $\bar{\boldsymbol{\theta}}_d^t$, and normalized by the SD σ_d^t. It can be represented as follows:

$$\alpha_{d,i}^t = \frac{\sigma_d^t}{\left|\boldsymbol{\theta}_{d,i}^t - \bar{\boldsymbol{\theta}}_d^t\right|}. \quad (9)$$

These weights of the dth dimension are then normalized and applied to aggregate each BS's local model $\boldsymbol{\theta}_i^t$ into a global model $\boldsymbol{\theta}^{t+1}$, which can be represented as follows in the view of each dimension:

$$\boldsymbol{\theta}_d^{t+1} = \frac{\sum_{i=1}^{n+m} \alpha_{d,i}^t \cdot \boldsymbol{\theta}_{d,i}^t}{\sum_{i=1}^{n+m} \alpha_{d,i}^t}. \quad (10)$$

Subsequently, the server broadcasts this aggregated global model parameter $\boldsymbol{\theta}^{t+1}$ back to all NDTs for synchronization.

There are three additional percentile estimation strategies listed below. Based on the upper and lower bound computed below, we can get a final percentile estimation decision to detect abnormal values in each dimension.

1) *IQR:* The IQR method calculates the range between the first and third quartiles (25th and 75th percentiles) of the data, identifying outliers based on this range. For each dimension d, the outlier bounds are

$$\text{lower bound}_{d,\text{IQR}}^t = Q1_d^t - k_{\text{IQR}} \cdot \text{IQR}_d^t \quad (11)$$

$$\text{upper bound}_{d,\text{IQR}}^t = Q3_d^t + k_{\text{IQR}} \cdot \text{IQR}_d^t \quad (12)$$

where $Q1_d^t$ and $Q3_d^t$ are the first and third quartiles, and k_{IQR} adjusts sensitivity.

2) *Z-Scores:* The Z-score method measures how many SDs a point is from the mean. For each dimension d, the normal range bounds are

$$\text{lower bound}_{d,\text{Z-score}}^t = g\left(\bar{\boldsymbol{\theta}}_d^t - k_{\text{Z}} \cdot \sigma_d^t\right) \quad (13)$$

$$\text{upper bound}_{d,\text{Z-score}}^t = g\left(\bar{\boldsymbol{\theta}}_d^t + k_{\text{Z}} \cdot \sigma_d^t\right) \quad (14)$$

where k_{Z} is the number of SDs for the normal range.

3) *One-Class SVM:* One-class SVM constructs a decision boundary for anomaly detection. The decision function for each dimension d is

$$f_d^t(\boldsymbol{\theta}) = \text{sign}\left(\sum_{i=1}^{n_{\text{SV}}} \gamma_i \cdot K\left(\boldsymbol{\theta}_{\text{SV}_i,d}^t, \boldsymbol{\theta}\right) - \rho\right) \quad (15)$$

where $\boldsymbol{\theta}_{\text{SV}_i,d}^t$ are the support vectors
γ_i are the Lagrange multipliers
$K(\cdot,\cdot)$ is the kernel function, and
ρ is the offset.

A point $\boldsymbol{\theta}$ is an outlier if $f_d^t(\boldsymbol{\theta}) < 0$.

In essence, this defense mechanism is a strategic amalgamation of direct statistical trimming and aggregation, targeting the preservation of the global model's integrity against poisoning attacks. By accurately isolating and excluding malicious NDTs prior to aggregation, it significantly diminishes the likelihood of adversarial disruption in the FL framework. Additionally, its capacity to accommodate various dimensions and adapt

to different inconsistency metrics and aggregation protocols considerably extends its applicability across a broad spectrum of distributed wireless network scenarios.

VI. Experimental Evaluation

In this section, we present an extensive evaluation of our proposed FTI poisoning attack and the GLID defense mechanism. We provide extensive results across various performance metrics to demonstrate their effectiveness in multiple dimensions.

A. Experimental Setup

1) Data Sets: To assess our methods, we employ real-world data sets from Telecom Italia [49]. The Milan wireless traffic data set is partitioned into 10 000 grid cells, each served by an NDT covering an area of approximately 235 m squared. The data set comprises three subsets: 1) "Milan-Internet;" 2) "Milan-SMS;" and 3) "Milan-Calls," which capture diverse wireless usage patterns. Our primary focus is on the "Milan-Internet" subset, which facilitates a detailed analysis of urban telecommunications behavior.

2) Baseline Schemes: We benchmark our FTI attack against several state-of-the-art model poisoning attacks to underscore its effectiveness. Additionally, we employ these baseline attacks to demonstrate the efficacy of our GLID defense strategy.

1) *Trim Attack [22]:* Processes each key in a model dictionary, using extremes in a specific dimension to determine a *directed* dimension. Model parameters are then selectively zeroed or retained to influence the model's behavior.
2) *History Attack [26]:* Iterates over model parameters, replacing current values with historically scaled ones to warp the model parameters using past data and misguide the aggregation process.
3) *Random Attack [26]:* Disrupts the model by replacing parameters with random values drawn from a normal distribution, scaled to maintain a semblance of legitimacy and inject controlled chaos into the aggregation process.
4) *MPAF [26]:* Calculates a directional vector from the difference between initial and current parameters, adjusting model values to intentionally diverge from the original trajectory and introduce adversarial bias. Fake NDTs are then injected into the system.
5) *Zheng Attack [24]:* Inverts the direction of model updates by incorporating the negative of previous global updates, refined through error maximization to generate a poison that is challenging to detect due to its alignment with the twin model's error landscape.

Furthermore, we consider several baseline defensive mechanisms to evaluate the robustness of our proposed attack and defense.

1) *Mean [39]:* Calculates the arithmetic mean of updates in each dimension, assuming equal trustworthiness among all NDTs. This method is susceptible to the influence of extreme values.
2) *Median [27]:* Identifies the median value in each dimension for each parameter across updates, discarding extreme contributions to enhance robustness against outliers.
3) *Trim [27]:* Discards a specified percentage of the highest and lowest updates before computing the mean in each dimension, reducing the influence of anomalous or malicious updates on the aggregate model.
4) *Krum [40]:* Scores each NDT's update based on the sum of Euclidean distances to other NDTs' updates, selecting the update from the NDT with the minimum score for the global update.
5) *FoolsGold [42]:* Calculates a cosine similarity matrix among all NDTs and adjusts the weights for each NDT based on these similarities, aggregating the weighted gradients to form a global model.
6) *FABA [50]:* Computes the Euclidean distance for each NDT's model from the mean of all received models, excluding a specific percentage of the most distant models to filter out potential outliers or malicious updates.
7) *FLTrust [41]:* Calculates cosine similarity between the server's current model and each NDT's model to generate trust scores, which are then used to weigh the NDT's contribution to the final aggregated model.
8) *FLAIR [43]:* Each NDT calculates "flip-scores" from the changes in gradient directions and "suspicion-scores" based on historical behavior, using these scores to adjust the weights assigned to each NDT's contributions to the global twin model.

3) Experimental Settings and Performance Metrics: For our experiments, we randomly select 100 BSs and their corresponding NDTs to evaluate the impact of poisoning attacks and the effectiveness of defense mechanisms. We primarily report results on the Milan-Internet data set. Model training is configured with a learning rate of 0.001 and a batch size of 64. We inject a 20% percentage of fake NDTs to simulate benign ones in the system for the FTI attack and assume a scenario where 20% of the NDTs are compromised for other baseline attacks. Our proposed FTI attack employs a parameter $\eta = 10$, while other attacks utilize a scaling factor of 1000. For the Trim aggregation rule, we discard 20% of the twin model parameters from all NDTs. In our GLID defense, we use the SD method as the default percentile estimation method. We adopt mean absolute error (MAE) and mean squared error (MSE) as the primary metrics for performance evaluation, with larger MAE and MSE values indicating better attack effectiveness.

B. Numerical Results

1) Performance of Proposed Methods: Tables I and II demonstrate the significant vulnerabilities introduced by the proposed FTI Attack across various aggregation methods within our NDT construction. It is observed that under our FTI Attack, the Mean Rule is completely compromised over both the V-twinning and H-twinning stages, as reflected by their MAE and MSE values reaching over 100.0 (values exceeding 100 are capped at 100). This result denotes a

TABLE I
PERFORMANCE EVALUATION WITH MILAN-INTERNET DATA SET DURING V-TWINNING STAGE

Aggregation Rule	Metric	Attack						
		NO	Trim	History	Random	MPAF	Zheng	***FTI***
Mean	MAE	0.281	100.0	100.0	100.0	100.0	0.768	**100.0**
	MSE	0.106	100.0	100.0	100.0	100.0	0.314	**100.0**
Median	MAE	0.281	0.283	0.281	0.282	0.281	0.287	**100.0**
	MSE	0.106	0.106	0.107	0.106	0.106	0.115	**100.0**
Trim	MAE	0.281	0.282	0.282	0.281	0.282	0.309	**100.0**
	MSE	0.106	0.107	0.109	0.106	0.108	0.126	**100.0**
Krum	MAE	0.291	0.295	100.0	0.295	100.0	0.295	**100.0**
	MSE	0.111	0.113	100.0	0.114	100.0	0.114	**100.0**
FoolsGold	MAE	0.283	100.0	100.0	100.0	100.0	1.004	**100.0**
	MSE	0.115	100.0	100.0	100.0	100.0	0.627	**100.0**
FABA	MAE	0.289	100.0	0.297	0.289	100.0	0.693	**100.0**
	MSE	0.109	100.0	0.105	0.101	100.0	0.269	**100.0**
FLTrust	MAE	0.312	0.304	100.0	0.310	100.0	3.252	**100.0**
	MSE	0.114	0.112	100.0	0.114	100.0	1.278	**100.0**
FLAIR	MAE	0.286	0.298	100.0	100.0	100.0	0.320	**100.0**
	MSE	0.114	0.108	100.0	100.0	100.0	0.116	**100.0**
GLID	MAE	**0.281**	**0.281**	**0.282**	**0.281**	**0.281**	**0.282**	**72.453**
	MSE	**0.106**	**0.107**	**0.106**	**0.106**	**0.107**	**0.106**	**27.548**

TABLE II
PERFORMANCE EVALUATION WITH MILAN-INTERNET DATA SET DURING H-TWINNING STAGE

Aggregation Rule	Metric	Attack						
		NO	Trim	History	Random	MPAF	Zheng	***FTI***
Mean	MAE	0.266	100.0	100.0	100.0	100.0	0.753	**100.0**
	MSE	0.101	100.0	100.0	100.0	100.0	0.309	**100.0**
Median	MAE	0.296	0.298	0.296	0.297	0.296	0.302	**100.0**
	MSE	0.101	0.102	0.102	0.105	0.101	0.110	**100.0**
Trim	MAE	0.296	0.297	0.297	0.296	0.297	0.294	**100.0**
	MSE	0.101	0.102	0.104	0.101	0.108	0.121	**100.0**
Krum	MAE	0.276	0.280	100.0	0.280	100.0	0.280	**100.0**
	MSE	0.106	0.108	100.0	0.109	100.0	0.109	**100.0**
FoolsGold	MAE	0.268	100.0	100.0	100.0	100.0	0.989	**100.0**
	MSE	0.110	100.0	100.0	100.0	100.0	0.622	**100.0**
FABA	MAE	0.274	100.0	100.0	100.0	100.0	0.678	**100.0**
	MSE	0.104	100.0	100.0	100.0	100.0	0.264	**100.0**
FLTrust	MAE	0.297	0.289	100.0	0.295	100.0	3.237	**100.0**
	MSE	0.109	0.107	100.0	0.109	100.0	1.223	**100.0**
FLAIR	MAE	0.271	0.283	100.0	100.0	100.0	0.305	**100.0**
	MSE	0.109	0.103	100.0	100.0	100.0	0.111	**100.0**
GLID	MAE	**0.266**	**0.266**	**0.267**	**0.266**	**0.266**	**0.267**	**72.458**
	MSE	**0.101**	**0.102**	**0.101**	**0.101**	**0.102**	**0.101**	**27.543**

total breakdown in their WTP functionality. The Median Rule further emphasizes the severity of the FTI Attack, with both its MAE and MSE escalating from modest baseline figures to 100. This sharp contrast highlights the FTI attack's reliable performance against other defenses, such as the trim attack against the median rule, where the increase in MAE and MSE is relatively minor at 0.283 and 0.106 for V-twinning, respectively. Additionally, the Trim Rule, typically considered robust, exhibits a drastic increase in MAE to over 100.0, a significant rise from its baseline without any attack (denoted as NO in Tables I and II) of 0.281. This surge underscores the Trim Rule's vulnerability to the FTI Attack, marking a notable departure from its typical resilience. Similar results can also be found in other aggregation rules under FTI attacks, such as Krum, FoolsGold, FABA, FLTrust, and FLAIR, where the FTI attack demonstrates the best overall performance against the given defenses. The Zheng Attack, however, presents a distinct pattern of disruption. When subjected to this attack, FLTrust, which typically exhibits lower error metrics, shows a significant compromise, evidenced by the dramatic increase in its MAE to 3.252 and MSE to 1.278. Such a tailored nature of the Zheng Attack appears to target specific vulnerabilities within FLTrust, which are not as apparent in other scenarios, such as the trim attack, where the rise in MAE and MSE for FLTrust is relatively modest. Regarding the MPAF Attack, most aggregation rules in the table do not show a convincing defense, except for a few like Median, Trim, and GLID.

During the H-twinning stage, most baseline schemes demonstrate a similar performance under attacks. However, although the Median and Trim Rules could protect the NDT systems from being attacked, they do not perform well in maintaining a precise NDT after a valid initial construction, i.e., the V-twinning stage. For instance, the MAE and MSE values of Median Rule under NO attacks are 0.281 and 0.106, respectively. These performance metrics increase to 0.296 and 0.101 after a period of maintenance, which leads to inaccurate predictions compared to the initial twin models. This is due to the heterogeneous nature of data distribution, with the Median and Trim Rules trimming out too many participants during the twin model aggregation process.

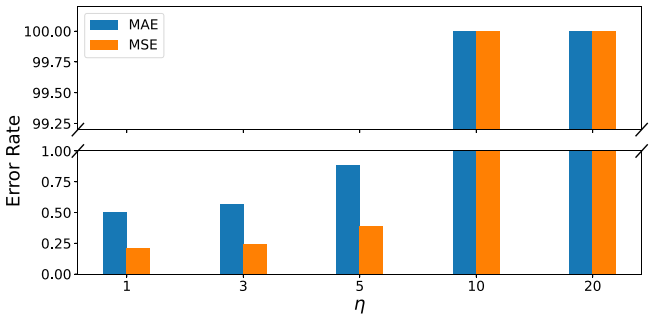


Fig. 5. Impact of values of η.

From the defender's standpoint, the proposed GLID aggregation method demonstrates consistent performance stability across various attacks. Both its MAE and MSE values remain close to their baseline levels. Even in the case of our FTI attack, GLID manages to keep errors below 100, with MAE and MSE values of 72.453 and 27.548, respectively. This stability is particularly noteworthy, especially when compared to other rules, such as FLAIR, which exhibit a significant deviation from their nonattacked baselines under the same adversarial conditions. GLID's ability to sustain its performance in the face of diverse and severe attacks underscores its potential as a resilient aggregation methodology. Other rules, such as FABA, also experience inconsistent defense performance during the V-twinning and H-twinning stages. FABA maintains good performance under History and Random attacks during the V-twinning process but is degraded to over 100.0 during the H-twinning process. Such performance is led by the various data distribution and sample sizes from the real-time data stream. In later evaluations, we focus on the entire twinning process, combining V-twinning and H-twinning, to evaluate the effects of other parameters on our proposed poisoning attacks and defense mechanisms for NDT systems.

2) Evaluation on the Impact of η: The step size η in our proposed FTI attack (see Algorithm 1) serves as a dynamic scaling factor, and its initial value significantly influences the NDT's performance metrics. This impact is illustrated in Fig. 5, where the Median aggregation rule is employed as the baseline defense strategy. A notable observation is the correlation between increasing values of η and the corresponding rise in MAE and MSE of twin models. For example, at $\eta = 1$, the MAE and MSE are relatively low, recorded at 0.517 and 0.215, respectively. However, increasing η to higher values, such as 10 or 20, results in a dramatic surge that reaches the maximum error rate. This increase suggests a significant compromise in the twin models, surpassing the predefined threshold for effective detection of the attack. The rationale behind this analysis emphasizes the pivotal role of η in determining the *strength* of a poisoning attack. An increased initial η tends to degrade model performance, deviating significantly from its expected operational state. Simultaneously, a higher η also raises the risk of the attack's perturbations being detected and eliminated during the defense process.

TABLE III
IMPACT OF PERCENTAGES OF FAKE NDTS

Pct.	Metric	Attack					
		Trim	Hist	Rand	MPAF	Zhe.	**FTI**
5%	MAE	0.276	0.270	0.274	0.270	0.268	**0.284**
	MSE	0.093	0.094	0.093	0.093	0.093	**0.094**
10%	MAE	0.275	0.268	0.273	0.268	0.269	**0.313**
	MSE	0.092	0.095	0.093	0.095	0.101	**0.109**
20%	MAE	0.278	0.273	0.273	0.271	0.324	**100.0**
	MSE	0.092	0.101	0.092	0.097	0.141	**100.0**
30%	MAE	100.0	100.0	100.0	100.0	6.045	**100.0**
	MSE	100.0	100.0	6.146	100.0	1.159	**100.0**
40%	MAE	100.0	100.0	100.0	100.0	100.0	**100.0**
	MSE	100.0	100.0	100.0	100.0	100.0	**100.0**

TABLE IV
IMPACT OF PERCENTILE ESTIMATION METHODS

Method	Metric	Attack						
		NO	Trim	Hist	Rand	MPAF	Zhe.	**FTI**
SD	MAE	0.274	0.274	0.274	0.273	0.274	0.274	**72.43**
	MSE	0.092	0.092	0.092	0.092	0.092	0.092	**27.53**
IQR	MAE	0.274	0.275	0.275	0.274	0.265	0.273	**100.0**
	MSE	0.092	0.092	0.092	0.092	0.092	0.093	**100.0**
Z-scores	MAE	0.274	0.274	0.274	0.274	0.275	1.102	**100.0**
	MSE	0.092	0.092	0.093	0.092	0.092	0.416	**100.0**
SVM	MAE	0.274	100.0	100.0	0.275	100.0	0.768	**100.0**
	MSE	0.092	100.0	100.0	0.092	100.0	0.290	**100.0**

3) Evaluation on Percentage of Fake NDTs: The degree of compromise in NDTs significantly influences the model's performance, as evidenced in Table III. By adopting the Median aggregation as the defensive approach, the model first exhibits resilience at lower compromise levels, such as with only 5%–10% fake NDTs in the scenario. However, a noticeable decline in performance is observed as the percentage of fake NDTs increases to 20% or higher. This deterioration is evident as the MAE and MSE values reach 100.0 in all categories, signaling a complete model failure. The underlying principle behind this trend suggests the model's limited tolerance to malicious interference. More precisely, the network system can withstand below 20% compromise without significant performance degradation. However, beyond this threshold, the model's integrity is severely undermined, resulting in a complete system breakdown. This observation highlights the critical importance of implementing robust security measures to prevent excessive compromise of NDTs, ensuring the model's reliability and effectiveness.

4) Evaluations on Percentile Estimation Methods: The dynamic trimming of an adaptive number of model parameters through percentile estimation, which is adapted in GLID, proves to be an effective defense strategy against various model poisoning attacks. In the comparative analysis of various estimation methods, as shown in Table IV, SD estimation emerges as the best technique, exhibiting marked consistency and robustness across a spectrum of estimation approaches. This is evidenced by the consistently low MAE and MSE values for SD across these approaches, at 0.219 and 0.087, respectively. In contrast, other methods have varying degrees of inconsistency and vulnerability. For instance, One-class SVM exhibits pronounced variability, with MAE and MSE values reaching the maximal error level of over 100.0 under Trim, History, and MPAF attacks. Such a disparity in performance, particularly the stably lower error rates of SD compared to the

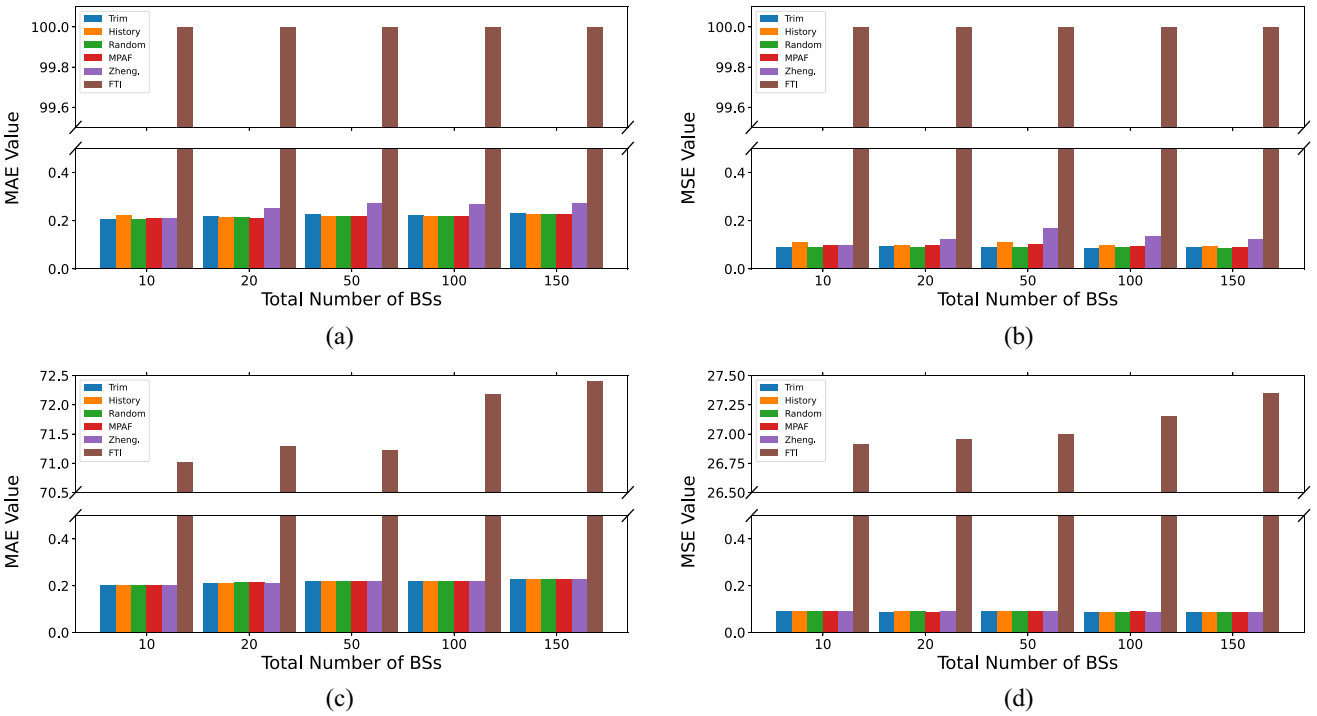


Fig. 6. Impact of NDT density on the performance of median and GLID methods with respect to MAE and MSEs. (a) Median AR w.r.t. MAE. (b) Median AR w.r.t. MSE. (c) GLID AR w.r.t. MAE. (d) GLID AR w.r.t. MSE.

significant fluctuations in other estimation methods, positions SD as a reliable and effective percentile estimation technique in GLID.

5) Evaluations on the Impact of NDT Density: Given a 20% proportion of fake NDTs, Fig. 6(a)–(d) compare the Median and GLID rules with varying densities of NDTs in the network scenario. The total number of NDTs does not significantly impact the performance of any attack and defense mechanisms, especially for our FTI and GLID strategies, which is consistent with traditional FL settings [28]. Under Median aggregation, FTI consistently shows maximal errors, with MAE and MSE exceeding 100 across different NDT densities, indicating the failure of the defense. This consistency of performance across varying participants in the distributed NDT system suggests that the total number of NDTs does not substantially influence the effectiveness of the attack and defense strategies.

6) Evaluations on the Percentile Range of GLID: Table V presents an evaluation of performance across a variety of percentile pairs used in the proposed GLID method on different attack methods. The configuration of the percentile pair guides the GLID method in identifying and eliminating outliers. For example, specifying a percentile pair of [10, 70] means that values below the tenth percentile and above the 70th percentile are trimmed away, focusing the analysis on the data within these bounds. It is observed that, when the percentile pair is set at [10, 70], most methods, except for the Zheng Attack, register a metric over 100.0, suggesting the models are fully attacked. Similarly, the percentile pair of [10, 90] yields a value over 100 for all methods except the Zheng Attack. The Zheng attack consistently records low metrics across all settings, such as 0.880 and 0.346 for the pair [10, 70], raising questions about its attack efficacy. On the other hand, the FTI attack shows varied performance; it achieves over 100.0 for most percentile pairs like [10, 70] and [20, 90] but drops to 79.634 and 29.849 for the pair [30, 70]. These results underscore the importance of fine-tuning the percentile pair parameters in the GLID method. Proper parameter selection can effectively trim outliers without significantly impacting overall network performance.

TABLE V
IMPACT OF DIFFERENT PERCENTILE PAIRS

Pair	Metric	Method					
		Trim	Hist	Rand	MPAF	Zhe.	**FTI**
[10, 70]	MAE	100.0	100.0	100.0	100.0	0.880	**100.0**
	MSE	100.0	100.0	100.0	100.0	0.346	**100.0**
[20, 70]	MAE	0.266	0.265	0.269	0.268	0.267	**100.0**
	MSE	0.102	0.106	0.104	0.101	0.106	**100.0**
[30, 70]	MAE	0.269	0.271	0.272	0.266	0.268	**79.634**
	MSE	0.112	0.109	0.110	0.106	0.109	**29.849**
[10, 80]	MAE	100.0	100.0	100.0	100.0	0.883	**100.0**
	MSE	100.0	100.0	100.0	100.0	0.340	**100.0**
[20, 80]	MAE	0.268	0.266	0.269	0.265	0.267	**76.468**
	MSE	0.106	0.102	0.104	0.101	0.106	**28.776**
[30, 80]	MAE	0.271	0.269	0.271	0.267	0.268	**75.411**
	MSE	0.109	0.110	0.106	0.109	0.112	**28.619**
[10, 90]	MAE	100.0	100.0	100.0	100.0	0.884	**100.0**
	MSE	100.0	100.0	100.0	100.0	0.339	**100.0**
[20, 90]	MAE	0.266	0.268	0.269	0.267	0.265	**100.0**
	MSE	0.109	0.106	0.105	0.110	0.106	**100.0**
[30, 90]	MAE	0.268	0.269	0.271	0.267	0.266	**100.0**
	MSE	0.106	0.109	0.110	0.105	0.109	**100.0**

VII. CONCLUSION AND FUTURE WORK

In this study, we introduced a novel approach to perform model poisoning attacks on NDTs through FTI. Operating under the assumption that real-world BSs are challenging to attack, we inject fake traffic distribution within NDTs

with minimum knowledge that disseminates malicious model parameters into distributed network systems. Furthermore, we presented an innovative GLID mechanism, designed to safeguard the NDT systems. It employs an adaptive trimming strategy, relying on percentile estimations that preserve accurate model parameters while effectively removing outliers. Extensive evaluations demonstrate the effectiveness of our attack and defense, outperforming existing baselines.

With the advent of the digitalization era, the development of an effective security framework for NDT systems presents numerous opportunities for future research and development. Future work could focus on enhancing the capabilities of secure NDT to incorporate real-time data streams and predictive analytics, enabling proactive security management and optimization. Additionally, exploring the integration of explainable artificial intelligence to elucidate model aggregation decisions, detect biases, and ensure the reliability and trustworthiness of the models is a crucial area of research. Further, investigating the application of secure DTs in emerging technologies, such as the IoT and 6G cellular systems, offers promising avenues for integrated intelligence and autonomy.

REFERENCES

[1] D. C. Nguyen et al., "6G Internet of Things: A comprehensive survey," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 359–383, Jan. 2022.

[2] Z. Qadir, K. N. Le, N. Saeed, and H. S. Munawar, "Towards 6G Internet of Things: Recent advances, use cases, and open challenges," *ICT Exp.*, vol. 9, no. 3, pp. 296–312, 2023.

[3] A. Miglani and N. Kumar, "Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges," *Veh. Commun.*, vol. 20, Dec. 2019, Art. no. 100184.

[4] F. Cugurullo, "Urban artificial intelligence: From automation to autonomy in the smart city," *Front. Sustain. Cities*, vol. 2, p. 38, Jul. 2020.

[5] A. Qayyum, J. Qadir, M. Bilal, and A. Al-Fuqaha, "Secure and robust machine learning for healthcare: A survey," *IEEE Rev. Biomed. Eng.*, vol. 14, pp. 156–180, Jul. 2020.

[6] E. VanDerHorn and S. Mahadevan, "Digital twin: Generalization, characterization and implementation," *Decis. Support Syst.*, vol. 145, Jun. 2021, Art. no. 113524.

[7] C. Semeraro, M. Lezoche, H. Panetto, and M. Dassisti, "Digital twin paradigm: A systematic literature review," *Comput. Ind.*, vol. 130, Sep. 2021, Art. no. 103469.

[8] D. M. Botín-Sanabria, A.-S. Mihaita, R. E. Peimbert-García, M. A. Ramírez-Moreno, R. A. Ramírez-Mendoza, and J. d. J. Lozoya-Santos, "Digital twin technology challenges and applications: A comprehensive review," *Remote Sens.*, vol. 14, no. 6, p. 1335, 2022.

[9] K. Feng, J. Ji, Y. Zhang, Q. Ni, Z. Liu, and M. Beer, "Digital twin-driven intelligent assessment of gear surface degradation," *Mech. Syst. Signal Process.*, vol. 186, Mar. 2023, Art. no. 109896.

[10] W. Yu, P. Patros, B. Young, E. Klinac, and T. G. Walmsley, "Energy digital twin technology for industrial energy management: Classification, challenges and future," *Renew. Sustain. Energy Rev.*, vol. 161, Jun. 2022, Art. no. 112407.

[11] L. Li, B. Lei, and C. Mao, "Digital twin in smart manufacturing," *J. Ind. Inf. Integr.*, vol. 26, Mar. 2022, Art. no. 100289.

[12] C. Alcaraz and J. Lopez, "Digital twin: A comprehensive survey of security threats," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 3, pp. 1475–1503, 3rd Quart., 2022.

[13] E. Karaarslan and M. Babiker, "Digital twin security threats and countermeasures: An introduction," in *Proc. Int. Conf. Inf. Secur. Cryptol. (ISCTURKEY)*, 2021, pp. 7–11.

[14] H. Ning et al., "A survey on the metaverse: The state-of-the-art, technologies, applications, and challenges," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14671–14688, Aug. 2023.

[15] S. B. Far and A. I. Rad, "Applying digital twins in metaverse: User interface, security and privacy challenges," *J. Metaverse*, vol. 2, no. 1, pp. 8–15, 2022.

[16] J. Guo et al., "TFL-DT: A trust evaluation scheme for federated learning in digital twin for mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 11, pp. 3548–3560, Nov. 2023.

[17] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, "Spatio-temporal wireless traffic prediction with recurrent neural network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, pp. 554–557, Aug. 2018.

[18] Y. Xu, W. Xu, F. Yin, J. Lin, and S. Cui, "High-accuracy wireless traffic prediction: A GP-based machine learning approach," in *Proc. IEEE Global Commun. Conf.*, 2017, pp. 1–6.

[19] C. Zhang, S. Dang, B. Shihada, and M.-S. Alouini, "Dual attention-based federated learning for wireless traffic prediction," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[20] M. Joshi and T. H. Hadi, "A review of network traffic analysis and prediction techniques," 2015, *arXiv:1507.05722*.

[21] J. Fan, D. Mu, and Y. Liu, "Research on network traffic prediction model based on neural network," in *Proc. 2nd Int. Conf. Inf. Syst. Comput. Aided Educ. (ICISCAE)*, 2019, pp. 554–557.

[22] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to Byzantine-robust federated learning," in *Proc. 29th USENIX Security Symp. (USENIX Security)*, 2020, pp. 1605–1622.

[23] V. Shejwalkar and A. Houmansadr, "Manipulating the Byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *Proc. NDSS*, 2021, pp. 1–18.

[24] T. Zheng and B. Li, "Poisoning attacks on deep learning based wireless traffic prediction," in *Proc. IEEE Conf. Comput. Commun.*, 2022, pp. 660–669.

[25] C. Xie, O. Koyejo, and I. Gupta, "Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation," in *Proc. 35th Uncertain. Artif. Intell.*, 2020, pp. 261–270.

[26] X. Cao and N. Z. Gong, "MPAF: Model poisoning attacks to federated learning based on fake clients," in *Proc. CVPR Workshops*, 2022, pp. 1–9.

[27] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5650–5659.

[28] Z. Zhang, M. Fang, J. Huang, and Y. Liu, "Poisoning attacks on federated learning-based wireless traffic prediction," in *Proc. IFIP/IEEE Netw. Conf.*, 2024, pp. 1–9.

[29] P. Almasan et al., "Network digital twin: Context, enabling technologies, and opportunities," *IEEE Commun. Mag.*, vol. 60, no. 11, pp. 22–27, Nov. 2022.

[30] H. Wang, Y. Wu, G. Min, and W. Miao, "A graph neural network-based digital twin for network slicing management," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1367–1376, Feb. 2022.

[31] L. Zhao, G. Han, Z. Li, and L. Shu, "Intelligent digital twin-based software-defined vehicular networks," *IEEE Netw.*, vol. 34, no. 5, pp. 178–184, Sep./Oct. 2020.

[32] Z. Yin, N. Cheng, T. H. Luan, Y. Song, and W. Wang, "DT-assisted multi-point symbiotic security in space-air-ground integrated networks," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 5721–5734, 2023.

[33] Z. Zhang, M. Chen, Z. Yang, and Y. Liu, "Mapping wireless networks into digital reality through joint vertical and horizontal learning," in *Proc. IFIP/IEEE Netw. Conf.*, 2024, pp. 1–9.

[34] Z. Zhang, Y. Liu, Z. Peng, M. Chen, D. Xu, and S. Cui, "Digital twin-assisted data-driven optimization for reliable edge caching in wireless networks," *IEEE J. Sel. Areas Commun.*, to be published.

[35] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-efficient federated learning and permissioned blockchain for digital twin edge networks," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2276–2288, Feb. 2021.

[36] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-efficient federated learning for digital twin edge networks in Industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5709–5718, Aug. 2021.

[37] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *Proc. 25th Eur. Symp. Res. Comput. Security*, 2020, pp. 480–501.

[38] L. Liang, X. Li, H. Huang, Z. Yin, N. Zhang, and D. Zhang, "Securing multidestination transmissions with relay and friendly interference collaboration," *IEEE Internet Things J.*, vol. 11, no. 10, pp. 18782–18795, May 2024.

[39] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1–10.
[40] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. 31st Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
[41] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "FLTrust: Byzantine-robust federated learning via trust bootstrapping," 2020, *arXiv:2012.13995*.
[42] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating Sybils in federated learning poisoning," 2018, *arXiv:1808.04866*.
[43] A. Sharma, W. Chen, J. Zhao, Q. Qiu, S. Bagchi, and S. Chaterji, "FLAIR: Defense against model poisoning attack in federated learning," in *Proc. ACM CCS*, 2023, pp. 553–566.
[44] Q. Xia, Z. Tao, and Q. Li, "Defending against Byzantine attacks in quantum federated learning," in *Proc. 17th Int. Conf. Mobil., Sens. Netw. (MSN)*, 2021, pp. 145–152.
[45] M. Fang et al., "Byzantine-robust decentralized federated learning," in *Proc. CCS*, 2024, pp. 1–18.
[46] "Android-x86 run android on your PC." Accessed: Jul. 13, 2024. [Online]. Available: https://www.android-x86.org/
[47] "NoxPlayer, the perfect android emulator to play mobile games on PC." Accessed: Jul. 13, 2024. [Online]. Available: https://www.bignox.com/
[48] "The world's first cloud-based android gaming platform." Accessed: Jul. 13, 2024. [Online]. Available: https://www.bluestacks.com/
[49] G. Barlacchi et al., "A multi-source dataset of urban life in the city of Milan and the province of Trentino," *Sci. Data*, vol. 2, Oct. 2015, Art. no. 150055.
[50] Q. Xia, Z. Tao, Z. Hao, and Q. Li, "FABA: An algorithm for fast aggregation against Byzantine attacks in distributed neural networks," in *Proc. IJCAI*, 2019, pp. 4824–4830.

Zifan Zhang received the bachelor's and master's degrees in electrical and computer engineering from The Ohio State University, Columbus, OH, USA, in 2021 and 2023, respectively. He is currently pursuing the Ph.D. degree with the Department of Computer Science, North Carolina State University, Raleigh, NC, USA.

His research interests include wireless networking, digital twins, distributed learning, and model security.

Mr. Zhang received the Best Paper Award Runner-Up at the IFIP/IEEE Networking Conference 2024.

Minghong Fang received the Ph.D. degree from the Department of Electrical and Computer Engineering from The Ohio State University, Columbus, OH, USA, in 2022.

He is an Incoming Tenure-Track Assistant Professor with the Department of Computer Science and Engineering, University of Louisville, Louisville, KY, USA. He was a Postdoctoral Associate with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA, from 2022 to 2024. His research interests include security, privacy, and machine learning.

Mingzhe Chen (Member, IEEE) is currently an Assistant Professor with the Department of Electrical and Computer Engineering and the Institute of Data Science and Computing, University of Miami, Coral Gables, FL, USA. His research interests include federated learning, reinforcement learning, virtual reality, unmanned aerial vehicles, and Internet of Things.

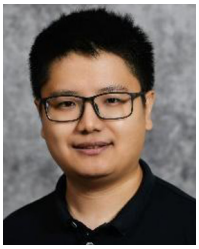
Dr. Chen has received four IEEE Communication Society Journal Paper Awards and four Conference Best Paper Awards at IEEE conferences. He currently serves as an Associate Editor for IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE WIRELESS COMMUNICATIONS LETTERS, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, and IEEE TRANSACTIONS ON MACHINE LEARNING IN COMMUNICATIONS AND NETWORKING.

Gaolei Li (Member, IEEE) received the B.S. degree in electronic information engineering from Sichuan University, Chengdu, China, in 2015, and the Ph.D. degree in cyber security from Shanghai Jiao Tong University, Shanghai, China, in 2020.

He is currently an Assistant Professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. His research interests include adversarial machine learning and network security.

Xi Lin (Member, IEEE) received the B.S. degree from the School of Precision Instrument and Optoelectronics Engineering, Tianjin University, Tianjin, China, in 2016, and the Ph.D. degree in cyber security from Shanghai Jiao Tong University, Shanghai, China, in 2021.

He is an Assistant Professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. His research interests include blockchain, privacy computing, edge computing, and the Internet of Things.



Yuchen Liu (Member, IEEE) received the Ph.D. degree from the Georgia Institute of Technology, Atlanta, GA, USA, in 2022.

He is currently an Assistant Professor with the Department of Computer Science, North Carolina State University, Raleigh, NC, USA. His research interests include wireless networking, digital twins, generative AI, distributed learning, mobile computing, and software simulation.

Dr. Liu has received several Best Paper Awards at IEEE and ACM conferences. He currently serves as Associate Editor for IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON MACHINE LEARNING IN COMMUNICATIONS AND NETWORKING, and *Computer Networks* (Elsevier).