Received 3 June 2025; revised 3 July 2025; accepted 17 July 2025. Date of publication 22 July 2025; date of current version 5 August 2025.

*Digital Object Identifier 10.1109/OJCOMS.2025.3591535*

# A Trust-By-Learning Framework for Secure 6G Wireless Networks Under Native Generative AI Attacks

**MD SHIRAJUM MUNIR[1] (Member, IEEE), SRAVANTHI PRODDATOORI[2], MANJUSHREE MURALIDHARA[2], TRINIDAD MARIO DENA[1], WALID SAAD[3] (Fellow, IEEE), ZHU HAN[4] (Fellow, IEEE), AND SACHIN SHETTY[2] (Senior Member, IEEE)**

[1]School of Computing, Analytics, and Modeling, University of West Georgia, Carrollton, GA 30118, USA
[2]Center for Secure and Intelligent Critical Systems, Old Dominion University, Norfolk, VA 23529, USA
[3]Electrical and Computer Engineering, Virginia Tech, Arlington, VA 22203, USA
[4]Electrical and Computer Engineering, University of Houston, Houston, TX 77004, USA

CORRESPONDING AUTHOR: M. S. MUNIR (e-mail: mmunir@westga.edu)

This work was supported in part by the National Science Foundation (NSF) under Grant 2451402, and in part by the OARU 2025 Ralph E. Powe Junior Faculty Enhancement Award.

**ABSTRACT** Sixth-generation (6*G*) wireless networks will become vulnerable due to native generative AI (GenAI)-driven intelligent poisoning attacks in both the radio unit and the core network. In particular, network parameters and metrics in cross-layer design pose fundamentally uncertain conditions and can be compromised through the native GenAI mechanism, which leverages data augmentation and reconstruction capabilities. This work investigates the capabilities of native GenAI to create novel poisoning attacks in wireless networks, while investigating their impact through uncertainty-informed root analysis. Then, detected attacks are mitigated by developing a trustworthy service aggregation in the wireless network. First, a joint decision problem is formulated to generate intelligent poisoning attacks, understand their root cause by defining a new measure of uncertainty as plausibility, and mitigate them through trustworthy service aggregation in wireless networks. Second, to address the challenges of the formulated problem, a novel Trust-By-Learning (TBL) framework is developed. The proposed TBL framework primarily consists of three components: 1) a native GenAI mechanism that can penetrate intelligent poisoning attacks in wireless networks' metrics and parameters; 2) a Dempster-Shafer-based evidence theoretic mechanism that is developed to understand the root cause of inherently uncertainty of those attacks to quantify the trust for further mitigation; and 3) a meta-reinforcement-based Markov Decision Process learning framework that can mitigate the intelligent attacks by enforcing trustworthy service aggregation. Extensive experimental analysis demonstrates that native GenAI methods, such as generative adversarial network (GAN), variational autoencoder (VAE), and autoencoder have significant capability to enforce poisoning attacks. Results show that the autoencoder performs significantly better in generating poisoning attacks capabilities of 98.2%, 97.4%, and 95% for Amazon, Netflix, and Download services, respectively. The proposed TBL framework effectively replicates intelligent attack dependencies by achieving a trust score of 0.972, 0.922, and 0.892 for Amazon, Download, and Netflix services, respectively. Finally, the proposed TBL framework shows efficacy in understanding the trust in GenAI-driven intelligent poisoning attacks on network parameters and metrics by quantifying root causes and mitigating rates.

**INDEX TERMS** Generative AI, 6G, intelligent attacks, evidence theory, trustworthy AI, meta-reinforcement learning.

## I. INTRODUCTION

THE EMERGENCE of the sixth-generation (6*G*) wireless networks promises significant advancements in speed, capacity, and reliability, enabling a future of interconnected devices and services with tight integration of AI Technologies [1], [2], [3]. However, these advancements also introduce new security challenges, particularly in the context of generative AI attacks. The need for a new

generation of wireless networks, such as 6G, stems from the limitations of current 5G networks in handling the increasing demand for data, connectivity, and intelligent services. 6G networks aim to provide seamless intelligent connectivity, ultra-low latency, and enhanced mobile broadband, extending these capabilities to include sensing and AI-driven services [2], [3]. These next-generation 6G networks will face a growing threat from intelligent attacks that leverage generative AI (GenAI) to compromise their trust. GenAI, with its ability to generate realistic data and mimic system behavior, can be weaponized to create sophisticated attacks that bypass traditional security measures [4]. This can erode trust in the network and its services, posing a significant challenge to the successful deployment of 6G. In particular, the attacks include artificial intelligence and machine learning based intelligent attacks, zero-day attacks, quantum attacks, and physical layer attacks [4].

For instance, AI-native 6G networks [3], [5], [6], [7], [8], [9], [10] in will face major security challenges in poisoning attacks in network metrics and parameters. Further, the emerging 6G applications such as Extended Reality (XR), Connected and Autonomous Vehicles (CAV), Holographic Telepresence, the Internet of Everything (IoE), Smart Grid 2.0, UAV-based mobility, Hyper-intelligent IoT, Digital Twin, and so on significantly relies on AI/ML methods [4], [5], [6], [11], [12], [13], [14]. Therefore, the chance of intelligent poisoning attacks in 6G wireless networks significantly increases due to the rigorous deployment of AI methods in both the application level and wireless infrastructure.

Protecting and securing 6G wireless networks from such intelligent attacks, it is essential to understand the new attack surface of network parameters and metrics while establishing trust on services. The challenges incorporate due to the highly uncertain behavior of intelligent attacks in wireless parameters and metrics such as received signal strength indicator (RSSI), reference signal received quality (RSRQ), reference signal received power (RSRP), channel quality indicator (CQI), user mobility, and so on. Further, establishing long-term temporal dependencies among the network parameters and metrics and coping with the high-dimensional attack space of heterogeneous wireless services become major challenges. The studies [4], [5], [6], [11], [12], [13], [14], [15], [16], [17] have not investigated intelligent attacks on wireless networks' parameters and metrics that are created by native GenAI methods such as Autoencoder and Variational Autoencoder (VAE), Generative Adversarial Networks (GANs), and so on. In particular, the main motivation behind this research is to investigate the capabilities of native such GenAI to create novel poisoning attacks in wireless networks, while investigating their impact through uncertainty-informed root analysis. Subsequently, the detected attacks are mitigated by developing a trustworthy service aggregation in the wireless network. In essence, prior works have not adequately studied the complex and adaptive nature of intelligent attacks in high-dimensional spaces [1], [4], [5], [6], [11], [12], [13], [14], [15], [16], [17], highlighting the need for a comprehensive framework like TBL to address these challenges.

Our initial study in [1] shows the capability of intelligent poisoning attack generation by native GenAI models and understands their attack vectors in a quantifiable trust metric. *The main contribution of the paper is a novel Trust-By-Learning framework that can help understand the uncertain behavior of GenAI-driven intelligent attacks on network change, parameters, and metrics while protecting the wireless network through uncertainty-informed trustworthy service aggregation.* Our contributions are summarized as follows:

- We propose a novel Trust-By-Learning framework for understanding and mitigating intelligent cyber attacks in next-generation wireless systems. In particular, the proposed framework can create intelligent poisoning attacks on communication parameters and metrics while understanding the root cause and protecting the 6G services by providing trustworthy aggregation.
- We develop a new narrow GenAI framework capable of creating new intelligent adversarial attack surfaces in wireless systems for further understanding the attack characteristics, severity, and the possible ways for mitigating next-generation cyber-attacks.
- We develop a trust quantification mechanism based on evidence theory that effectively captures the uncertainty of intelligent poisoning attacks on wireless communication, aiming to safeguard next-generation wireless systems from such sophisticated threats.
- We develop a meta-reinforcement-based Markov decision process learning framework to understand the intelligent attacks and trustworthy service aggregation in wireless networks by taking into account long-term temporal dependencies among the intelligent attack vectors.
- The experimental results show mutual information values for dependency replication in the TBL framework across various generative models show interesting results. Autoencoder and Variational Autoencoder effectively model attack patterns, achieving high correlation coefficients of 0.972 for Amazon, 0.922 for Download, and 0.892 for Netflix. In contrast, Generative Adversarial Networks perform less effectively in this context, with lower correlation coefficients of 0.212 for Amazon, 0.445 for Download, and 0.454 for Netflix. This demonstrates that Autoencoders and VAEs are more adept at replicating and understanding attack behaviours, which is crucial for developing robust defenses.

The rest of the paper is organized as follows. In Section II, we present important related works based on the existing

literature. In Section III, we describe the proposed trustworthy wireless network system model. Then, we formulate the TBL problem in Section IV. The proposed TBL framework is designed in Section V. In Section VI, we present and analyze our experimental results. Finally, we conclude our discussion in Section VII. Abbreviations are summarized in the APPENDIX.

## II. RELATED WORKS

The design of security solutions for AI-native wireless networks attracted significant attention recently [2], [3]. For instance, the work in [18], [19] identifies security technologies and research challenges specific to 6G wireless networks with a focus on the role of AI and blockchains. The authors in [11] studied the problem of identifying and understanding the emerging trends, applications, requirements, technologies, and future research directions in the context of 6G networks. However, in [11], the authors do not provide any technical direction of how to address the intelligent attacks in wireless network infrastructure. The work in [4] identifies future challenges by focusing on security and privacy concerns that might arise with the development of 6G. In particular, the authors survey potential challenges associated with different 6G technologies and applications. The work in [19] identifies the cyber attack prediction mechanisms that leverage traditional machine learning to GenAI models. However, our work focuses on investigating intelligent poisoning attacks in wireless networks that are posed by a native GenAI model. The works [4], [11], [18], [19] primarily focus on various aspects of 6G wireless security, such as general trends and existing privacy measures. However, these studies have not extensively explored the emerging challenge of GenAI attacks in wireless networks or detailed specific protection mechanisms against them. This paper aims to fill that gap by investigating the potential for GenAI-driven attacks and proposing effective defense strategies.

A few of the works [15], [16], [17], [20], [21] studied the problem of securing wireless networks against intelligent attacks. The authors in [15] address the problem of securing 6G network-assisted Internet of Things(IoT) systems against adversarial attacks. It explores various defense strategies and evaluates their effectiveness through theoretical analysis, up-to-date research, and Monte Carlo simulations. While their work significantly advances the understanding of such defense mechanisms, there is an opportunity to further explore the potential roles of Generative AI in this context. The authors in [20] investigate a key foundational aspect of GenAI-driven risk in cybersecurity. In particular, the authors raised concern that data poisoning attacks can be a crucial threat from the GenAI. Further, the authors in [21] study a theoretical and empirical study on response to GenAI-driven in smart grid communication. Particularly, the authors developed a Bayesian belief network framework to understand the GenAI attack surface in smart grid

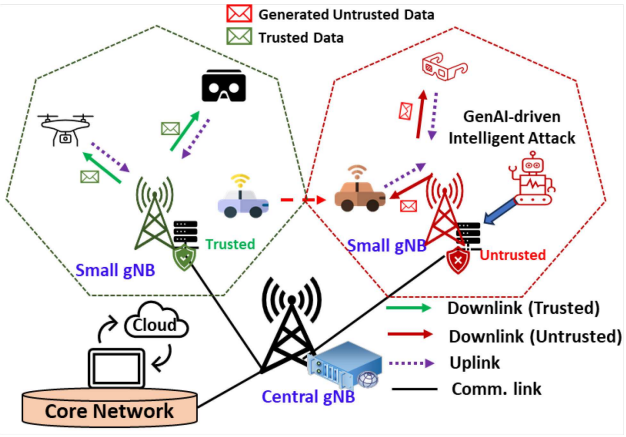



**FIGURE 1.** A system model of securing 6G service aggregation under generative AI-driven intelligent attacks in a wireless network.

communication. Our research builds on these findings by leveraging Generative AI to develop robust and adaptive strategies for enhancing the security of 6G wireless networks.

The research [17] presented a robust-by-design framework for anti-jamming in MIMO-OFDM wireless communications. Robust anti-jamming methods are created without requiring presumptions about the adversary's configuration, thanks to the use of sensing-assisted information. The work [16] solved network transparency and improving user interactions in Zero Touch Networks (ZTNs) by integrating Large Language Models (LLMs) that can distill complex deep reinforcement learning (DRL)-based anti-jamming techniques. The research [17] presented a robust-by-design framework for anti-jamming in MIMO-OFDM wireless communications. Robust anti-jamming methods are created without requiring presumptions about the adversary's configuration, thanks to the use of sensing-assisted information. However, these works [15], [16], [17] do not consider the intelligent attacks that can be imposed by the GenAI models while not explore how to integrate GenAI approaches in a realistic way for understanding the new intelligent data poisoning attacks in wireless networks.

## III. SYSTEM MODEL OF TRUSTWORTHY WIRELESS COMMUNICATION

Considering a wireless network consisting of a set $\mathcal{I} = \{0, 1, 2, \ldots, I\}$ of $I+1$ 6G NodeBs (gNBs) that encompass $I$ small cell base stations (SBSs) overlaid over an AI-native central gNB $i = 0$ as shown in Figure 1. Figure 1 illustrates a typical system model of securing service aggregation under generative AI-driven intelligent attacks in a wireless network. To make the system model more understandable to a broader audience, we have omitted the detailed blocks related to the core network in Figure 1. Each gNB $i \in \mathcal{I}$ can serve a set $\mathcal{J}_i$ of $J$ heterogeneous 6G services such as UAV-based mobility, connected and autonomous vehicles (CAV), intelligent health care, and so on. We consider a time-slotted system with each

**TABLE 1.** Summary of notations.

| Notation | Description |
|---|---|
| $\mathcal{I}$ | Set of next-generation NodeBs (gNBs) |
| $\mathcal{J}$ | Set of heterogeneous 6G services |
| $\alpha_{it}^{\text{up}}$ | Uplink data rate at gNB $i$ |
| $\beta_{it}^{\text{down}}$ | Downlink data rate at gNB $i$ |
| $\Upsilon(t)$ | Trust score of $j$ service aggregation decision |
| $\boldsymbol{a}_{it}$ | Aggregation decisions |
| $\Gamma_{tij}$ [dBm] | Reference signal received power (RSRP) of service $j \in \mathcal{J}_i$ |
| $\Lambda_{ij}(t)$ | Channel quality indicator (CQI) |
| $m_{tij}$ | User mobility |
| $\gamma_{tij}$ | Received signal strength indicator (RSSI) |
| $S_{tij}$ | signal-to-interference-plus-noise ratio (SINR) |
| $\psi_{k\neq i}$ | Near cell RSRP |
| $\phi$ | Attack vector generation parameters |
| $\theta$ | Intelligent poisoning attacks parameters |
| $\zeta$ | Latent space (i.e., attack vector) |
| $\delta^{\max}$ | Maximum downlink data rate |
| $\omega^{\max}$ | Maximum uplink data rate |
| $\Phi_j$ | Recurrent neural network (RNN) learning parameters |
| $\Theta_\pi$ | meta learned weight parameters |
| $h_j$ | RNN hidden state |

time slot $t$ belonging to a finite time horizon $\mathcal{T}$. In each slot $t$, a gNB $i \in \mathcal{I}$ establishes service aggregation decisions $\boldsymbol{a}_{it} \in \forall a_{tij}$. Each gNB $i \in \mathcal{I}$ will assign data rates $\alpha_{it}^{\text{up}}$ and $\beta_{it}^{\text{down}}$ for uplink and downlink at gNB $i$. The considered system model meets the 6G wireless network requirements [4], [5], [6], [11], [12], [13], [14] by enabling native AI-agent at each gNB for allocating uplink $\alpha_{tij}^{\text{up}}$ and downlink $\beta_{tij}^{\text{down}}$ data rates to a particular 6G service $j \in \mathcal{J}$. Therefore, the AI-native agent of each gNB $i \in \mathcal{I}$ can be affected by AI/ML-based intelligent attacks such as poisoning, AI compromises, and eavesdropping, during network operation etc [4], [13]. To characterize such intelligent attacks and protect the 6G wireless network, we need to understand three models: 1) the communication model, 2) the intelligent attack generation model, and 3) the trust model in the following subsections, respectively.

### A. COMMUNICATION MODEL

In our system, we consider both downlink and uplink communication models to serve each 6G service $j \in \mathcal{J}_i$ demands at gNB $i \in \mathcal{I}$. We define $G_{tij}^{\text{down}}$ as the downlink channel gain and $\sigma^2$ as the additive white gaussian noise (AWGN) at gNB $i$. Therefore, for reference signal received power (RSRP) $\Gamma_{tij}$ and co-channel interference $\Pi_{tij}^{\text{down}}$, the signal-to-interference-plus-noise ratio (SINR) $S_{tij}$ of gNB $i \in \mathcal{I}$ can be calculated as follow [22], [23], [24], [25], [26],

$$S_{tij} = \sum_{\forall j \in \mathcal{J}_i} \frac{\Gamma_{tij} G_{tij}^{\text{down}}}{\sigma^2 + \Pi_{tij}^{\text{down}}}. \tag{1}$$

The Channel Quality Indicator (CQI) $\Lambda_{ij}(t)$ [26] is a crucial metric used to capture and assess the performance of wireless communication, including both downlink and uplink behaviors. Therefore, we can estimate a CQI of gNB $i \in \mathcal{I}$ as follow [23], [24], [26], [27]:

$$\Lambda_{ij}(t) = 0.5223 \times S_{tij} + 4.6176, \tag{2}$$

where 0.5223 and 4.6176 are constant [23], [24], [27] on the linear model (2).

We consider a 20 MHz wide bandwidth based on orthogonal frequency-division multiplexing (OFDM). Each resource block (RB) consists of 12 sub-carriers, totaling 180 KHz per RB, and there are 100 RBs available for each channel bandwidth [25], [26]. Then, for a fixed bandwidth $B_{tij}^{\text{down}}$, the downlink data rate $\beta_{tij}^{\text{down}}$ of the considered communication model can be calculated as follows:

$$\beta_{tij}^{\text{down}} = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_{it}} B_{tij}^{\text{down}} \log_2 \left(1 + \frac{\Gamma_{tij} G_{tij}^{\text{down}}}{\sigma^2 + \Pi_{tij}^{\text{down}}}\right). \tag{3}$$

Similarly, the uplink data rate $\alpha_{tij}^{\text{up}}$ can be calculated as follows:

$$\alpha_{tij}^{\text{up}} = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_{it}} B_{tij}^{\text{up}} \log_2 \left(1 + \frac{\Gamma_{tij} G_{tij}^{\text{up}}}{\sigma^2 + \Pi_{tij}^{\text{up}}}\right), \tag{4}$$

where $B_{tij}^{\text{up}}$ is the uplink bandwidth. In this work, we investigate the effect of a new intelligent attack vector on communication parameters and metrics generated by AI. Therefore, we describe an intelligent attack generation model in the following subsection.

### B. INTELLIGENT ATTACK GENERATION

Considering a vector $\boldsymbol{x}_{tij} = (x_1, x_2, \ldots, x_X)$ of $X$ elements wireless communication metrics and parameters, where $\forall \boldsymbol{x}_{tij} \in \mathcal{X}$. Therefore, the elements of the vector $\boldsymbol{x}_{tij}$ include user mobility $m_{tij}$, RSSI $\gamma_{tij}$, RSRP $\Gamma_{tij}$, SINR $S_{tij}$, uplink data rate $\alpha_{tij}^{\text{up}}$, downlink data rate $\beta_{tij}^{\text{down}}$, near-cell RSRP $\psi_{k\neq i}$, CQI $\Lambda_{tij}$. Attackers target such parameters and metrics for generating AI/ML-based intelligent data poisoning attacks in the wireless network so as to compromise the AI services and gain control over the service execution.

We consider an intelligent poisoning attack vector parameters is $\phi$. $\phi$ that maps with $\mathcal{X}$ in a latent space (i.e., attack vector) $\zeta$, where $\phi : \mathcal{X} \rightarrow \zeta$. In other words, $\zeta$ becomes an intelligent attack's latent space for the given network parameters and metrics $\mathcal{X}$ on attack generation parameters $\phi$. Now, consider $\theta$ is an actual attack reconstruction parameter that decodes the intelligent attack latent space $\zeta$ into poisonous network metrics and parameters $\boldsymbol{x'}_{tij}$. The network data poisoning parameter $\theta$ generates the actual attacks $\boldsymbol{x'}_{tij}$ through intelligent attack latent space $\zeta$, $\theta : \zeta \rightarrow \mathcal{X}$, $\forall \boldsymbol{x}_{tij} \in \mathcal{X}$. Therefore, the intelligent attack generation model can be defined as follows:

$$F(\phi, \theta) = \mathbb{E}_{\boldsymbol{x}_{tij} \sim \zeta}\left[\Omega\left(\boldsymbol{x_{tij}}, g_\theta(f_\phi(\boldsymbol{x_{tij}}))\right)\right], \tag{5}$$

where $\Omega(.)$ is function of intelligent attack vector latent space $\zeta$ generation parameters $\phi$ and actual data poisoning parameters $\theta$.

Therefore, we formulate the intelligent poisoning attack generation model as follows:

$$\min_{\phi,\theta} F(\phi,\theta) = F(\phi,\theta) = \frac{1}{N}\sum_{n=1}^{N} ||\boldsymbol{x}_{tij} - g_\phi\big(f_\theta(\boldsymbol{x}_{tij})\big)||_2^2, \quad (6)$$

where $\boldsymbol{x'}_{tij} = g_\theta(f_\phi(\boldsymbol{x}_{tij}))$ is the generated network parameters and metrics and $\zeta_{tij}$:$\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_N\} \subset \mathcal{X}$ for $N$ sample. As a result, $F(\phi,\theta)$ can generate intelligent poisoning attacks in wireless communication systems by generating metrics and parameters like RSRP, RSRQ, SINR, CQI, mobility, as well as uplink/downlink data rates, among others. Thus, it is imperative to protect the communication systems from such attacks. Our goal is, therefore, to design a trust model inspired by the concept of the Dempster–Shafer [28] evidence theory. The following subsection describes the proposed trust model between 6G service and gNB.

### C. TRUST MODEL BETWEEN SERVICES AND GNB

The proposed trust model aims to protect the wireless system from intelligent poisoning attacks by establishing trustworthy communication between the service and the gNB. First, we define the belief of network parameters and metrics between service $j \in \mathcal{J}_i$ and gNB $i \in \mathcal{I}$. We consider a power set $2^{\boldsymbol{x}_{tij}}$ of wireless communication metrics and parameters $\boldsymbol{x}$ of service $j \in \mathcal{J}_i$. For example, the power set $2^{\boldsymbol{x}_{tij}}$ contains all combinations of belief of user mobility, RSSI, RSRP, SINR, uplink data rate, downlink data rate, near-cell RSRP, and CQI, including the empty set. We define a belief assignment mass function as $M\colon 2^{\boldsymbol{x}_{tij}} \rightarrow [0,1]$, where $\sum_{x\in 2^{\boldsymbol{x}_{tij}}} M(\boldsymbol{x})$. Thus, for a non-empty set $2^{\boldsymbol{x}_{tij}}$, total sum of beliefs will be 1 such that $\sum_{2^{x\in\boldsymbol{x}_{tij}}} = M(\boldsymbol{x}) = 1$. We estimate the belief through a Bayesian approximation,

$$\hat{M}(\boldsymbol{x}) = \begin{cases} \frac{\sum_{\boldsymbol{x'}|\boldsymbol{x}\subset\boldsymbol{x'}} M(\boldsymbol{x'})}{\sum_{\mathcal{X}} M(\mathcal{X})}, & \text{if } \sum_{2^{x\in\boldsymbol{x}_{tij}}} M(\boldsymbol{x}) = 1, \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

In (7), the belief in a service request lies in quantifying the uncertainty of a trusted request. In other words, (7) is a degree of belief or mass (belief function) for a particular service's network parameters and metrics. We consider plausibility [28] as a measure of uncertainty of network parameters or metrics and define it as follows:

$$Pl(x) = 1 - \hat{M}\big(\forall x' \in \boldsymbol{x}_{tij}, x \neq \boldsymbol{x}_{tij}\big), \quad (8)$$

where $\forall x' \in \boldsymbol{x}_{tij}$ is an intelligent attacks parameter of service $j \in \mathcal{J}_i$. We define a binary decision variable $a_{tij}$ to indicate whether the network parameters of service $j$ have been trusted or not. Therefore, $a_{tij}$ becomes a binary decision variable that decides the service aggregation decision of service $j \in \mathcal{J}_j$,

$$a_{tij} = \begin{cases} 1, & \text{if } \hat{M}(\boldsymbol{x}) \leq 1, Pl(x) < 1, \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

Where $Pl(x)$ represents the plausibility of network parameter $x \in \boldsymbol{X}$. We quantify the trust score of network parameters as follows:

$$\Upsilon(t) = \begin{cases} \frac{\hat{M}(\boldsymbol{x'})}{\hat{M}(\boldsymbol{x})} \times \frac{1}{Pl(x)} \times \Lambda_{ij}(t), \text{ if } \hat{M}(\boldsymbol{x}) > 0, \\ 0, \qquad \text{otherwise.} \end{cases} \quad (10)$$

The trust score in (10) depends on belief $\hat{M}(\boldsymbol{x'})$ and plausibility $Pl(x)$ of $\boldsymbol{x'}$ while we capture CQI $\Lambda_{ij}(t)$ in (2). That means the trust score (10) relies on both prior evidence of network parameters and metrics, and the current channel quality index of service $j \in \mathcal{J}_i$ at gNB $i \in \mathcal{I}$. The physical meaning of the trust score is that lower plausibility indicates higher trust, enabling us to better understand the intelligent poisoning parameters and metrics $\boldsymbol{x'}$, and vice-versa. In this work, we formulate a joint optimization problem that can generate an intelligent attack, understand the attack vectors such as generated communication parameters and metrics, and protect the communication system by providing service aggregation decisions based on trust. A detailed description of the Trust-By-Learning problem for a 6G wireless network is discussed next.

## IV. TRUST-BY-LEARNING DECISION PROBLEM FORMULATION

This section proposes a joint decision problem to secure wireless networks from AI/ML-driven intelligent attacks. The objective of the proposed decision problem is to maximize the trust score while finding the trustworthy service aggregating decision $a_{tij}$ of a service $j \in \mathcal{J}$ to gNB $i \in \mathcal{I}$ by determining belief $\hat{M}(\boldsymbol{x'})$ of the generative intelligent attack $\boldsymbol{x'}$. The proposed Trust-By-Learning is as follows:

$$\max_{\boldsymbol{x'},\hat{M}(\boldsymbol{x'}),\boldsymbol{a},} \sum_{t=1}^{T}\sum_{i=1}^{|\mathcal{I}|}\sum_{j=1}^{|\mathcal{J}|} \mathbf{E}_{\pi_{\Phi_j}}[\Upsilon(t)], \quad (11)$$

$$\text{s.t.} \quad \phi \leq \mathcal{X} \rightarrow \zeta, \zeta_{tij}\text{:}\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_N\} \subset \mathcal{X}, \quad (11a)$$

$$\theta \leq \zeta \rightarrow \boldsymbol{x'}, \forall \boldsymbol{x'}_{tij} \in \boldsymbol{x'}, \quad (11b)$$

$$Pl\big(x'\big) \leq 1, \forall x' \in \boldsymbol{x'}, \quad (11c)$$

$$a_{tij}\hat{M}\big(\boldsymbol{x'}\big) \leq 1, \forall x' \in \boldsymbol{x'}, \quad (11d)$$

$$a_{tij}\Lambda_{ij}(t) \leq \eta, \quad (11e)$$

$$\beta_{tij}^{\text{down}} \leq \delta^{\max}, \quad (11f)$$

$$a_{tij}\alpha_{tij}^{\text{up}} \leq \omega^{\max}, \quad (11g)$$

$$\Phi_j \leq H_\Theta\big(\mathcal{Y}_j\big), \mathcal{Y}_j = \Big\{\mathcal{X}, \mathcal{A}, \hat{M}(\boldsymbol{x}), \Upsilon(t)\Big\}, \quad (11h)$$

$$a_{tij} \in \{0,1\}, \forall i \in \mathcal{I}, \forall j \in \mathcal{J}_i. \quad (11i)$$

The formulated decision problem (11) consists of three decision variables, intelligent attack generated network parameters and metrics $\boldsymbol{x'}$, belief $\hat{M}(\boldsymbol{x'})$ on generated attacks, and trustworthy service aggregation decision $\forall a_{tij} \in \boldsymbol{a}$. The generated network parameters and metrics $\boldsymbol{x'}$ rely on parameters $\phi$, $\theta$, and the attack vector latent space $\zeta$. Constraint (11a) ensures the mapping from attack generation vector parameters $\phi$ to latent space $\zeta$. Constraint (11b)

ensures the mapping from attack latent space $\zeta$ to communication metrics $\boldsymbol{x}'$ with generated parameters $\theta$. The parameters $\phi$ and $\theta$ are determined by the intelligent attacks model in (6). The plausibility is one of the key performance metrics for determining the trust score of generated communication metrics and parameters $\boldsymbol{x}'$, constraint (11c) assures that this value must be positive and between 0 to 1. Similarly, the marginal probability of belief is constrained between 0 and 1 in constraint (11d). Constraint (11e) ensures that the CQI of a particular service fulfillment session between the service $j \in \mathcal{J}_i$ and gNB $i \in \mathcal{I}$ does not exceed the maximum $\eta = 15$ CQI [25], [26]. Constraints (11f) and (11g) ensure that the allocated downlink and uplink data rates of a service $j \in \mathcal{J}_i$ always bounded by maximum downlink $\delta^{\max}$ and uplink capacity $\omega^{\max}$, respectively. Constraint (11h) transforms the formulated problem into a Markov Decision Process $\mathcal{Y}_j = \{\mathcal{X}, \mathcal{A}, \hat{M}(\boldsymbol{x}'), \Upsilon(t)\}$, where $\boldsymbol{x} \in \mathcal{X}$ is the observations, $\boldsymbol{a} \in \mathcal{A}$ represents trusted service aggregation decisions, $\hat{M}(\boldsymbol{x}')$ presents evidence-based belief of a generated intelligent attacks $\boldsymbol{x}'$, and $\Upsilon(t)$ is the trust score of the AI-generated network parameters and metrics $\boldsymbol{x}'$. In particular, the constraint (11h) ensures that the formulated problem (11) establishes a Trust-By-Learning problem by accumulating a Markov Decision Process, where parameters $\Phi_j$ relies on the distribution of evidence $\Theta$ that belongs to the belief distribution $\mathcal{Y}_j \sim \hat{M}(\boldsymbol{x}')$ of the AI-generated network and parameters $\boldsymbol{x}'$. Finally, constraint (11i) ensures that at time slot $t \in T$, each service $j \in \mathcal{J}_i$ cannot be aggregated with more than one gNB.

The formulated problem (11) is hard to solve in polynomial time complexity in an optimization problem solver due to its non-linear constraints and uncertain dependencies on intelligent attack parameters. Therefore, the formulated problem (11) does not guarantee an optimal solution due to the uncertainty constraint (11c). In consequence, we will divide the solution of the formulated decision problem into three sub-problems. First, we solve the intelligent attack generation of network parameters and metrics $\boldsymbol{x}'$. Second, we determine the uncertainty-informed evidence creation $\hat{M}(\boldsymbol{x}')$. Finally, we develop a TBL framework to protect from intelligent poisoning attacks by trustworthy service aggregation $\boldsymbol{a}$ while fulfilling the service demands. A detailed description of the proposed solution approaches is given in the following section.

## V. PROPOSED TRUST-BY-LEARNING FRAMEWORK

In this paper, we propose a novel TBL framework to understand and mitigate intelligent cyber attacks in next-generation wireless communication systems. We illustrate the proposed three steps Trust-By-Learning framework in Figure 2. In step 1 (yellow block in Figure 2), we generate intelligent poisoning attacks on network parameters and metrics by deploying the native GenAI model. Step 2 (green block in Figure 2) represents the procedure of root-cause analysis of the intelligent attacks. Finally, step 3 (light brown block in Figure 2) presents the trustworthy learning

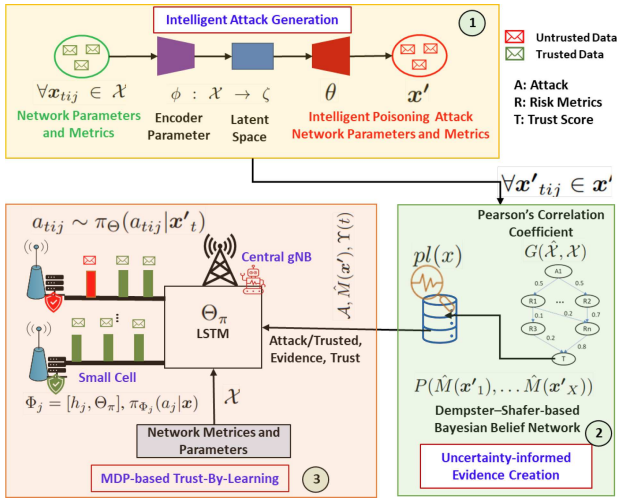



**FIGURE 2.** The proposed Trust-By-Learning framework for mitigating intelligent attack in wireless networks.

procedure of the proposed framework. Our approach involves generating intelligent poisoning attacks on communication parameters using a narrow GenAI framework to explore new adversarial attack surfaces. We introduce an evidence-theory-based trust quantification mechanism to capture the uncertainty of these attacks and ensure the protection of 6G services through trustworthy service aggregation. We outline the algorithmic procedure for Dempster–Shafer-based uncertainty-informed evidence creation, including generating sample space from poisoning attacks, estimating evidence with a Bayesian Belief Network, and calculating the plausibility of affected network parameters and metrics. Additionally, we develop a meta-reinforcement learning framework utilizing Markov decision processes to understand long-term temporal dependencies among intelligent attack vectors. Experimental results demonstrate that autoencoders and VAEs effectively replicate dependencies in various datasets, outperforming GANs in this task.

### A. INTELLIGENT ATTACK GENERATION FRAMEWORK

We will adopt the concept of unsupervised machine learning to generate intelligent poisoning attacks related to wireless network metrics and parameters. In particular, we create intelligent attacks $\boldsymbol{x}'$ by developing an autoencoder-based deep learning network. Algorithm 1 illustrates the overall procedure of the intelligent attack generation framework. In particular, Algorithm 1 solves the decision variable $\boldsymbol{x}'$ of the formulated problem in (11) in a data-driven manner.

The input of Algorithm 1 is network metrics and parameters of all $\boldsymbol{x}_{tij} \in \mathcal{X}$ services $\mathcal{J}_i$ in a finite time domain $T$. Each service parameters and metrics vector $\boldsymbol{x}_{tij}$ contains mobility $m_{tij}$, RSSI $\gamma_{tij}$, RSRP $\Gamma_{tij}$, SINR $S_{tij}$, uplink data rate $\alpha_{tij}^{\text{up}}$, downlink data rate $\beta_{tij}^{\text{down}}$, near-cell RSRP $\psi_{k\neq i}$, and CQI $\Lambda_{tij}$ information. Therefore, the role of Algorithm 1 is to generate the poisoning attack such that given network parameters and metrics $\boldsymbol{x}_{tij}$.

**Algorithm 1** Autoencoder-Based Intelligent Attack Generation

**Input:** $\boldsymbol{x}_{tij} = (x_1, x_2, \ldots, x_X) \approx (m_{tij},\ \gamma_{tij},\ \Gamma_{tij},\ S_{tij},\ \alpha_{tij}^{\text{up}},\ \beta_{tij}^{\text{down}},\ \psi_{k\neq i},\ \Lambda_{tij}),\ \forall \boldsymbol{x}_{tij} \in \mathcal{X}$.
**Output:** $\forall \boldsymbol{x'}_{tij} \in \boldsymbol{x'}$.
**Initialization**: $\phi$, $\theta$, $\zeta$, $\boldsymbol{W}$, $w_t$, $split(\mathcal{X})$.
1: **while** $\forall \boldsymbol{x}_{tij} \in \mathcal{X}$ **do**
2: **for** *epochs* $\leq$ *max epochs* **do**
3: **NN backpropagation:** $||\boldsymbol{x} - \sigma'(\boldsymbol{W}(\sigma(\boldsymbol{W}\boldsymbol{x} + b)) + b')||^2$ using (16)
4: **Execute ADAM optimizer:** using (17), (18), (19), (20)
5: **Estimate:** $\phi \rightarrow \zeta$ using (12)
6: **Execute ReLU function:** $g(x)$ using (13)
7: **Estimate weight change:** $\Delta w_t$ using (21)
8: **Estimate:** $\theta$ using (14)
9: **Execute Sigmoid function:** $s(x)$ using (15)
10: **end for**
11: **NN weight update:** $w_{t+1} = w_t + \Delta w_t$ using (22)
12: **end while**
13: **return** $\zeta$, $\boldsymbol{x'}$

Algorithm 1 initializes attack vector generation parameters $\phi$, intelligent poisoning attack parameters $\theta$, latent space (i.e., attack vector) $\zeta$, and neural network (NN) initial weight parameters $\boldsymbol{W}$. Thus, then, generative parameters $\phi$ maps with input network parameters and metrics $\mathcal{X}$ in a latent space $\zeta$, $\phi$:$\mathcal{X} \rightarrow \zeta$.

After estimating latent space $\zeta$, we estimate parameters of attack vectors $\theta$:$\zeta \rightarrow \mathcal{X}, \forall \boldsymbol{x}_{tij} \in \mathcal{X}$.

In line 3 of Algorithm 1, we execute neural network standard back-propagation to estimate latent space $\zeta$. Therefore, we define an encoder function as follows:

$$\zeta = \sigma(\boldsymbol{W}\boldsymbol{x} + b), \tag{12}$$

where $\boldsymbol{W}$ is neural network weights and $b$ is the bias. We employ a rectified linear unit (ReLU) activation function to capture the positive part of input $\boldsymbol{x}$ from neural network output. The ReLU activation function is defined as follows:

$$g(x) = \max(0, x) \ = \frac{x + |x|}{2}, \tag{13}$$

where $x > 0$. We consider the outcome of (13) as the latent space of a potential intelligent attack vector $\zeta$. Therefore, we can generate the poisoning attack as follows:

$$\boldsymbol{x'} = \sigma'\big(\boldsymbol{W}\zeta + b'\big), \tag{14}$$

where $b'$ becomes the bias regularization term of attacked parameters and metrics. We consider the output layer of the attack generation neural network as the sigmoid activation function [29] to capture the distribution from negative to positive value. The Sigmoid activation function is defined as follows:

$$s(x) = \frac{e^x}{1 + e^x}. \tag{15}$$

Finally, we can model a neural network for standard backpropagation loss function as follows:

$$\boldsymbol{L}\big(\boldsymbol{x}, \boldsymbol{x'}\big) = \min ||\boldsymbol{x} - \boldsymbol{x'}||^2 = ||\boldsymbol{x} - \sigma'\big(\boldsymbol{W}(\sigma(\boldsymbol{W}\boldsymbol{x} + b)) + b'\big)||^2, \tag{16}$$

where $||\boldsymbol{x} - \boldsymbol{x'}||^2$ is equivalent to the L2 norm or least squares function [30] during neural network training. We used the Adaptive Moment Estimation (Adam) optimization [31] to train the neural network due to the capability of capturing the first and second moments of the loss function the function (16). The first and second moments of the loss function (16) are defined by,

$$\phi_{t+1}^w = \vartheta_1 \phi_t^w + (1 - \vartheta_1)\nabla^w \boldsymbol{L}\big(\boldsymbol{x}, \boldsymbol{x'}\big), \tag{17}$$

$$\nu_{t+1}^w = \vartheta_2 \nu_t^w + (1 - \vartheta_2)\big(\nabla^w \boldsymbol{L}(\boldsymbol{x}, \boldsymbol{x'})\big)^2, \tag{18}$$

where $\vartheta_1$ and $\vartheta_2$ are the decay rates. As this approximation algorithm employs adaptive learning rates, the step size is essential in the initial iterations of the training process. Consequently, the algorithm conducts a bias correction before estimating the weight updates. The bias correction functions for the first and second moments are:

$$\hat{\phi^w} = \frac{\phi_{t+1}^w}{1 - (\vartheta_1)_{t+1}}, \tag{19}$$

$$\hat{\nu^w} = \frac{\nu_{t+1}^w}{1 - (\vartheta_2)_{t+1}}. \tag{20}$$

Hence, the function of weight updates $\Delta w_t$ with corrected bias is defined by,

$$\Delta w_t = -r\frac{\hat{\phi^w}}{\sqrt{\hat{\nu^w}} + \varepsilon}, \tag{21}$$

where $r$ represents the learning rate and $\varepsilon$ is a small value that prevents division by zero. Thus, the updated weight for the next time slot $t + 1$ is given by:

$$w_{t+1} = w_t + \Delta w_t. \tag{22}$$

In Algorithm 1, lines 3 and 4 execute the backpropagation and ADAM optimization for neural network training, respectively. Lines from 5 to 8 execute the procedure for intelligent poisoning attack generation training in Algorithm 1. Line 11 of Algorithm 1 updated the neural network trained weight for testing. Algorithm 1 generates a poisoning attack $\boldsymbol{x'}$ in network parameters and metrics. We have used the output of Algorithm 1 to create uncertainty-informed evidence to establish the trustworthy protection of next-generation wireless systems.

The computational complexity primarily comes from neural network backpropagation and the Adam optimizer used for weight updates. Each epoch involves recalculating weights based on the backpropagation of errors, and the use of activation functions like ReLU and Sigmoid. For a network with n neurons and $\boldsymbol{l}$ layers, the complexity per epoch is $O(n{*}l)$. The number of epochs further multiplies this complexity, resulting in an overall complexity of $O(epochs * n * l)$.

### B. UNCERTAINTY-INFORMED EVIDENCE CREATION

Building evidence $\hat{M}(\boldsymbol{x}')$ from previous knowledge is necessary to establish trust in wireless network parameters and metrics to defend against new sophisticated attacks. Therefore, the proposed intelligent attack generation Algorithm 1 is utilized for generating such attacks $\forall \boldsymbol{x}'$. We create a sample space using Pearson's correlation coefficient [32] from the generated attack as follows:

$$G\left(\hat{\mathcal{X}}, \mathcal{X}\right) = \frac{\sum_{j\in\mathcal{J}}\left(\boldsymbol{x'}_{tij} - \bar{\boldsymbol{x}'}\right)\left(\boldsymbol{x}_{tij} - \bar{\boldsymbol{x}}\right)}{\sqrt{\sum_{j\in\mathcal{J}}\left(\boldsymbol{x'}_{tij} - \bar{\boldsymbol{x}'}\right)^2 \sum_{j\in\mathcal{J}}\left(\boldsymbol{x}_{tij} - \bar{\boldsymbol{x}}\right)^2}}. \quad (23)$$

We will now leverage the concept of Dempster–Shafer [28] evidence theory to capture the uncertainty of evidence. In particular, we capture plausibility [28] as a measure of uncertainty of network parameters or metrics in (8). Thus, we build a Bayesian network [27] that consists of prior probability distribution $P(G(\hat{\mathcal{X}}, \mathcal{X}))$, likelihood function $P(G(\hat{\mathcal{X}}, \mathcal{X})|\hat{M}(\boldsymbol{x}'))$, and posterior probability $P(\hat{M}(\boldsymbol{x}')|G(\hat{\mathcal{X}}$. Therefore, the posterior probability can be presented as follows:

$$P(\hat{M}(\boldsymbol{x}')|G(\hat{\mathcal{X}}, \mathcal{X})) \propto P(G(\hat{\mathcal{X}}, \mathcal{X})|\hat{M}(\boldsymbol{x}')) \times P(G(\hat{\mathcal{X}}, \mathcal{X})). \quad (24)$$

For each network parameter or metric $x' \in \boldsymbol{x}'$, the product of the individual density functions becomes the conditional on their parent variables $v \in \mathcal{X}$. Therefore, the conditional parent will be given by:

$$P(\hat{M}(\boldsymbol{x}')) = \prod_{v\in\mathcal{X}} P(\hat{M}(\boldsymbol{x}')_v|\hat{M}(\boldsymbol{x}')_{\boldsymbol{Pa}(v)}). \quad (25)$$

Thus, the joint distribution of intelligent attacks $\boldsymbol{x}'$ can be calculated from conditional probabilities using the chain rule,

$$P(\hat{M}(\boldsymbol{x'}_1), \ldots \hat{M}(\boldsymbol{x'}_X)) = \prod_{v\in\mathcal{X}} P(\hat{M}(\boldsymbol{x'}_v))|\hat{M}(\boldsymbol{x}'_k), \hat{M}(\boldsymbol{x}'_k) \in \boldsymbol{Pa}(\hat{M}(\boldsymbol{x'}_v)). \quad (26)$$

The evidence $P(\hat{M}(\boldsymbol{x'}_1), \ldots \hat{M}(\boldsymbol{x'}_X))$ of intelligent attacks $\boldsymbol{x'}_X)$ can be estimated by (26).

We summarize the algorithmic procedure of the proposed Dempster–Shafer-based uncertainty-informed evidence creation in Algorithm 2. Line 3 in Algorithm 2 creates sample space from the generated poisoning attracts $\boldsymbol{x}'$. Lines from 4 to 7 estimate evidence of generated attacks by populating the Bayesian Belief Network (BBN) of the wireless network parameters and metrics in Algorithm 2. The plausibility of affected network parameters and metrics is calculated in line 8 of Algorithm 2. Finally, Algorithm 2 provides the evidence and plausibility in line 11. We will further utilize the uncertainty-informed evidence of generated intelligent attacks for developing a Trust-By-Learning framework to establish adaptive mitigation of intelligent attacks in wireless networks.

This algorithm's complexity arises from nested loops over the samples and elements in the dataset, leading to a base complexity of $O(m \times n)$. Further complexity is added through probabilistic calculations (e.g., prior, conditional, and posterior probabilities) that depend on the number of metrics, scaling the overall complexity to $O(m \times n^2)$. The operations grow quadratically with the number of metrics.

**Algorithm 2** Dempster–Shafer-Based Uncertainty-Informed Evidence Creation

**Input:** $\forall \boldsymbol{x'}_{tij} \in \boldsymbol{x}'$, $\mathcal{X}$.
**Output:** $P(\hat{M}(\boldsymbol{x'}_1), \ldots \hat{M}(\boldsymbol{x'}_X))$, $pl(x)$.
**Initialization**: $\hat{\mathcal{X}}$, $\mathcal{X}$
1: **while** $\forall \boldsymbol{x}_{tij} \in \mathcal{X}$ **do**
2: **for** $\forall \boldsymbol{x'}_{tij} \in \boldsymbol{x}'$ **do**
3: **Create sample:** $G(\hat{\mathcal{X}}, \mathcal{X})$ using (23)
4: **Estimate prior probability:** $P(G(\hat{\mathcal{X}}, \mathcal{X}))$ from (23)
5: **Conditional density:** $P(\hat{M}(\boldsymbol{x}'))$ using (25)
6: **Posterior probability:** $P(\hat{M}(\boldsymbol{x}')|G(\hat{\mathcal{X}}, \mathcal{X}))$ using (24)
7: **Estimate evidence:** $P(\hat{M}(\boldsymbol{x'}_1), \ldots \hat{M}(\boldsymbol{x'}_X))$ using (26)
8: **Calculate plausibility:** $Pl(x) = 1 - \hat{M}(\forall x' \in \boldsymbol{x}_{tij}, x \neq \boldsymbol{x}_{tij})$
9: **end for**
10: **end while**
11: **return** $P(\hat{M}(\boldsymbol{x'}_1), \ldots \hat{M}(\boldsymbol{x'}_X))$, $pl(x)$

### C. TRUST-BY-LEARNING FRAMEWORK FOR ADAPTIVE MITIGATION OF INTELLIGENT ATTACK

The goal of the proposed TBL framework is primarily to capture the highly uncertain behavior of intelligent attacks, establish long-term temporal dependencies, and cope with high-dimensional attack spaces for heterogeneous wireless services. There, we design a Meta-RL-based centralized training and decentralized execution-based mechanism to establish trust in network parameters and metrics by enabling adaptive mitigation of heterogeneous intelligent attacks.

We design an uncertainty-informed trust-based reward function (10) to capture the trust during the learning. In particular, the reward shaping is designed by capturing uncertainty from the plausibility of Dempster–Shafer theory (in Algorithm 2) while maximizing the CQI of the network. In this solution design, we consider each heterogeneous service $j \in \mathcal{J}$ acts as a learning agent while a virtual agent plays the role of a centralized meta-agent for training. Therefore, the learning parameters $\Theta$ of the meta-agent can be estimated as follows:

$$\Theta^* = \arg\max_{\Theta} \sum_{j=1}^{|\mathcal{J}|} \mathbf{E}_{\pi_{\Phi_j}}[\Upsilon(t)], \quad (27)$$

where $\Phi_j$ is the learning parameters of each service trust score. In particular, we can capture as a function of evidence $\mathcal{M}_j$ over the distribution of meta-agents parameters $\Theta$, $\Phi_j = H_\Theta(\mathcal{M}_j)$. Then, thus, we design our Trust-By-Learning as a Markov Decision Process (MDP)

$\mathcal{Y}_j = \{\mathcal{X}, \mathcal{A}, \hat{M}(\boldsymbol{x}'), \Upsilon(t)\}$, where $\mathcal{X}$ is network parameters and metrics, $\mathcal{A}$ presents trust action, $\Upsilon(t)$ is trust score, and $\hat{M}(\boldsymbol{x}')$ becomes evidence. For each heterogeneous service agent $j \in \mathcal{J}$, the MDP can present as $\mathcal{Y}_j = \{(\boldsymbol{x}'_{1ij}, \boldsymbol{a}_{1ij}, \boldsymbol{x}'_{2ij}, \Upsilon(1), \ldots, (\boldsymbol{x}'_{tij}, \boldsymbol{a}_{tij}, \boldsymbol{x}'_{jt+1}, \Upsilon(t)\}$, where $\mathcal{Y}_j \sim \hat{M}(\boldsymbol{x}')$.

We define the cumulative discounted trust for adaptive mitigation learning as follows:

$$V^{\pi_{\Phi_j}}(\boldsymbol{x}'_{tij}) = \mathbb{E}_{\boldsymbol{a}_{tij}\sim\pi_{\Phi_j}(\boldsymbol{a}_{tij}|\boldsymbol{x}'_{tij};\Phi_j)} \left[\sum_{t'=t}^{T} \rho^{t'-t}\Upsilon(t+t')|\boldsymbol{x}'_{tij}, \boldsymbol{a}_{tij}\right], \tag{28}$$

where $\rho^{t'-t}$ is a discount factor and ensures the convergence of the state value function $V^{\pi_{\Phi_j}}(\boldsymbol{x}'_{tij})$ over the finite time horizon $T$. By imposing Markovian property, the optimal trusted-value function is written as follows:

$$V^{\pi^*_{\Phi_j}}(\boldsymbol{x}'_{tij}) = \max_{a_{tij}\in\mathcal{A}} \mathbb{E}_{\pi^*_{\Phi_j}} \left[\sum_{j\in\mathcal{J}} \Upsilon(t') + \sum_{t'=t}^{\infty} \rho^{t'-t} V^{\pi_{\Phi_j}}_{t'}(\boldsymbol{x}'_{t'ij})|\boldsymbol{x}'_{tij}; \Phi_j, \boldsymbol{a}_{tij}\right]. \tag{29}$$

Here, the optimal trusted-value function (29) learns a parameterized policy of intelligent attack $\pi_{\Phi_j}(\boldsymbol{a}_{tij}|\boldsymbol{x}'_{tij}; \Phi_j)$ by using a Long short-term memory (LSTM)-based Q-networks for the parameters $\Phi_j$. To employ multi-agent settings for understanding the heterogeneity of intelligent attacks, we define an advantage function as follows [33], [34]:

$$\Lambda^{\pi^*_{\Phi_j}}(\boldsymbol{x}'_{tij}, \boldsymbol{a}_{ti1}, \ldots, \boldsymbol{a}_{tiJ}; \Phi_t) = \Upsilon(t) + \sum_{\boldsymbol{x}'_{t'ij}\in\mathcal{X}, t'=t}^{T} \rho^{t'-t}\mathcal{M}(\boldsymbol{x}'_{t'ij}|\boldsymbol{x}'_{tij}, \boldsymbol{a}_{ti1}, \ldots, \boldsymbol{a}_{tiJ}) V^{\pi_{\Phi_j}}(\boldsymbol{x}'_{t'ij}, \pi^*_{\Phi_1}, \ldots, \pi^*_{\Phi_J}) - V^{\pi_{\Phi_j}}(\boldsymbol{x}'_{tij}, \pi^*_{\Phi_1}, \ldots, \pi^*_{\Phi_J}), \tag{30}$$

where $\pi^*_{\Phi}: (\pi^*_{\Phi 1}, \ldots, \pi^*_{\Phi J})$ is a joint policy and $\Gamma(\boldsymbol{s}_{t'i}|\boldsymbol{x}'_{tij}, \boldsymbol{a}_{ti1}, \ldots, \boldsymbol{a}_{tiJ}) \mapsto [0, 1]$ represents state transition probability. Therefore, the objective of the proposed TBL framework is to minimize the temporal difference [33], [34], [35]. Therefore, the loss function is defined as follows:

$$L(\Phi_j) = \min_{\pi_{\Phi_j}} \frac{1}{|\mathcal{J}|} \sum_{j\in\mathcal{J}} \frac{1}{2} \left(\left(\Upsilon(t) + \sum_{t'=t}^{T} \rho^{t'-t} V^{\pi^*_{\Phi_j}}(\boldsymbol{x}'_{t'ij}|\Phi_t)\right) - V^{\pi^*_{\Phi_j}}(\boldsymbol{x}'_{tij})\right)^2. \tag{31}$$

Thus, we can redefine the trust policy loss function to capture the uncertainty during learning as follows:

$$L(\Phi_j) = -\mathbb{E}_{\boldsymbol{x}'_{tij}, \boldsymbol{a}_{tij}}[\pi_{\Phi_j}(\boldsymbol{a}_{tij}|\boldsymbol{x}'_{tij}) + \underbrace{(\tau h(\pi_{\Phi_j}(\boldsymbol{a}_{tij}|\boldsymbol{x}'_{tij}; \Phi_t)))}_{Entropy}]. \tag{32}$$

Therefore, the trusted policy gradient of the loss function (32) is defined in terms of temporal difference and entropy. Therefore, the policy gradient of the loss function of the Trust-By-Learning framework is defined as follows:

$$\nabla_{\Phi_j} L(\Phi_j) = \frac{1}{|\mathcal{J}|} \sum_{j\in\mathcal{J}} \sum_{t'=t}^{\infty} \nabla_{\Phi_j} \log \pi_{\Phi_j}(\boldsymbol{a}_{tij}|\boldsymbol{x}'_{tij}) \Lambda^{\pi_{\Phi_j}}(\boldsymbol{x}'_{tij}, \boldsymbol{a}_{tij}|\theta_t) + \tau h(\pi_{\Phi_j}(\boldsymbol{a}_{tij}|\boldsymbol{x}'_{tij}; \theta_t)). \tag{33}$$

We train and estimate the parameters $\Phi_j$ of each service $j \in \mathcal{J}_i$ trust-policy $\pi_{\Phi_j}$. We execute meta test $\mathcal{M}_{test} \sim \hat{M}(\boldsymbol{x}')$ of intelligent attack $\boldsymbol{x}'$ and estimate $\Phi_j = H_\Theta(\mathcal{M}_{test})$. During this test, we capture the trust-informed decision from meta policy $\pi_\Theta$ as $a_{tij} \sim \pi_\Theta(a_{tij}|\boldsymbol{x}'_t)$. The objective is to improve policy with MDP experience $\mathcal{Y}_j$, $\{(\boldsymbol{x}_1, a_{ij1}, \boldsymbol{x}'_2, \Upsilon(1), \ldots, (\boldsymbol{x}_t, a_{tij}, \boldsymbol{x}'_{t+1}, \Upsilon(t)\}$ for decentralized execution of adaptive mitigation from intelligent attacks. Thus, each service $j \in \mathcal{J}$ parameters will be updated through LSTM-based RNN states $\Phi_j = [h_j, \Theta_\pi]$, $\pi_{\Phi_j}(a_j|\boldsymbol{x}')$, where $h_j$ is RNN hidden state, and meta learned weight $\Theta_\pi$.

An algorithmic procedure of the proposed uncertainty-informed Trust-By-Learning framework for defense against intelligent attacks of network parameters and metrics is described in Algorithm 3. Lines from 3 to 10 train individuals service agents while lines from 12 to 16 update the meta-policy of each service agent $j \in \mathcal{J}_i$.

## VI. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we present the experimental results and discuss their implications for the effectiveness of our Trust-By-Learning framework in safeguarding 6G service architectures from intelligent cyber attacks. The section is organized as follows: experiment setup and objectives, performance analysis that includes trust evaluation of generated intelligent attacks of the proposed Trust-By-Learning framework.

### A. EXPERIMENT SETUP AND OBJECTIVES

This section outlines the experimental setup and objectives designed to evaluate the effectiveness of our novel Trust-By-Learning framework in mitigating intelligent cyber attacks within 6G service architectures. Initially, we have considered three GenAI such as GAN, Autoencoder, and Variational Autoencoder models for the regeneration of network parameters and metrics for three different services namely Amazon, Netflix, and Download from the open source dataset [25]. By regenerating network parameters and metrics, we can know the intelligent poisoning attack generation capabilities of these GenAI models. The optimal choice among these models will be determined based on training efficiency and quality of intelligent attack generation.

The preprocessing of data is performed for the three services. We have used python platform for performing scientific experiment while we have used open-source tools such as Google Collaboratory (Collab) for data preprocessing, model training, and evaluation. We have utilized the

**Algorithm 3** Uncertainty-Informed Trust-by-Learning for Defense Against Intelligent Attacks

**Input:** $P(\hat{M}(\boldsymbol{x}'_1), \ldots \hat{M}(\boldsymbol{x}'_X))$, $pl(x)$,$\forall \boldsymbol{x}'_{tij} \in \boldsymbol{x}'$, $\mathcal{X}$.
**Output:** $\Theta_\pi$, $a_{tij} \sim \pi_\Theta(a_{tij}|\boldsymbol{x}_t)$

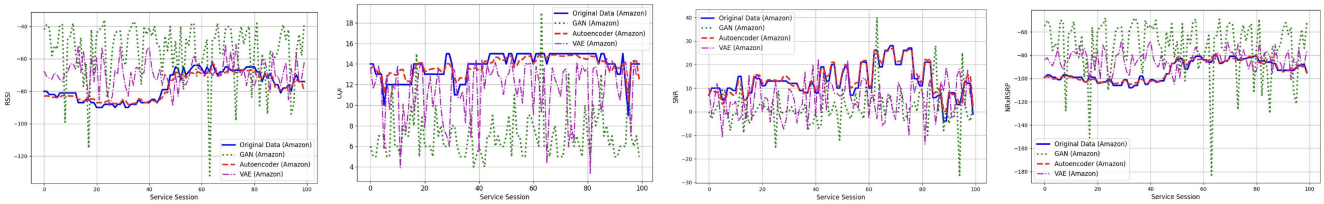
**while** *episode_count* $\leq$ *max_episode* **do**
  **for** $t \leq T$ *and* $\forall j \in \mathcal{J}_i$ **do**
    **Calculate trust score:** $\Upsilon(t)$ using 10
    **Build Markov decision process:** $\mathcal{Y}_j = \left\{\mathcal{X}, \mathcal{A}, \hat{M}(\boldsymbol{x}'), \Upsilon(t)\right\}$
    **Estimate cumulative discounted reward:** $V^{\pi_{\Phi_j}}(\boldsymbol{x}'_{tij})$ using (28)
    **optimal value function by imposing MDP:** $V^{\pi^*_{\Phi_j}}(\boldsymbol{x}'_{tij})$ using (29)
    **Parameterized policy via LSTM learning:** $\pi_{\Phi_j}(\boldsymbol{a}_{tij}|\boldsymbol{x}'_{tij}; \Phi_j)$.
    **Estimate parameterized policy:** $\Lambda^{\pi^*_{\Phi_j}}(\boldsymbol{x}'_{tij}, \boldsymbol{a}_{ti1}, \ldots, \boldsymbol{a}_{tiJ}; \Phi_t)$ using (30)
    **Estimate joint policy:** $\pi^*_\Phi$: $(\pi^*_{\Phi 1}, \ldots, \pi^*_{\Phi J})$
    $t = t + 1$
  **end for**
  **Estimate gradient of loss:** $\nabla_{\Phi_j} L(\Phi_j)$ using (33)
  **Pick:** $a_{tij} \sim \pi_\Theta(a_{tij}|\boldsymbol{x}'_t)$
  **Improve policy with MDP experience:** $\mathcal{Y}_j$, $\left\{(\boldsymbol{x}_1, a_{ij1}, \boldsymbol{x}'_2, \Upsilon(1), \ldots, (\boldsymbol{x}_t, a_{tij}, \boldsymbol{x}'_{t+1}, \Upsilon(t)\right\}$
  **Meta learned weight parameters update:** $\Phi_j = [h_j, \Theta_\pi]$, $\pi_{\Phi_j}(a_j|\boldsymbol{x}')$
  *episode_count* = *episode_count* + 1
**end while**
**return** $\Theta_\pi$, $a_{tij} \sim \pi_\Theta(a_{tij}|\boldsymbol{x}'_t)$

established Python libraries like Pandas for data engineering, NumPy for numerical computations, and scikit-learn for statistical utilities. Additionally, Matplotlib and Seaborn libraries were leveraged for data visualization tasks. In case of intelligent poisoning attack generation, the Autoencoder model has been constructed and trained using the Keras APIs that are provided by TensorFlow. Similarly, the Keras API facilitated the training of a GAN model. Finally, a PyTorch deep learning framework was utilized to implement the VAE model architecture and perform model training and evaluation of attack generation. Table 2 presents a summary of important parameters that are used during the experiment. In Table 2, the network parameters and metrics are selected based on the dataset and the 3GPP standard [1], [25], [26], while the hyperparameters are determined through a trial-and-error method, using the best-performing combination based on the accuracy metric.

The Autoencoder demonstrated a balance between complexity and performance, effectively capturing and replicating the dependencies within the original data. Its computational complexity is moderate, making it a practical choice for real-time applications. The VAE emerged as the most effective algorithm in terms of preserving the integrity of the original data's network relationships, successfully replicating key dependencies with high fidelity despite its higher computational complexity. In contrast, the

**TABLE 2.** Summary of experiment setup.

| Description | Value |
|---|---|
| No of Samples | 2000 |
| Input Dimensions | 11 |
| Autoencoder Encoding Dimensions | 8 |
| Autoencoder Number of Encoder and Decoder Layers | 4 |
| Autoencoder Dense Encoder Layer1 ,Layer2, Layer3, Layer4 | 128 units, 64 units, 32 units, 8 units, |
| Autoencoder Activation function | ReLU activation |
| Autoencoder Dense Decoded Layer1, Layer2, Layer3, Layer4 | 32 units, 64 units, 128 units, 11 units |
| Autoencoder Optimizer | Adam |
| Autoencoder Loss Function | Mean Sqaured Error (MSE) |
| Autoencoder Batch Size | 32 |
| VAE Input Dimensions | 11 |
| VAE First Hidden Layer (H) | 45 units,BatchNorm, ReLU activation |
| VAE Second Hidden Layer (H2) | 15 units, BatchNorm, ReLU activation |
| VAE Third Hidden Layer (H2) | 15 units, BatchNorm, ReLU activation |
| VAE Latent Dimension | 11 |
| VAE Decoder First Hidden Layer (H2) | 15 units, BatchNorm, ReLU activation |
| VAE Decoder Second Hidden Layer (H2) | 15 units, BatchNorm, ReLU activation |
| VAE Decoder Third Hidden Layer (H2) | 45 units, BatchNorm, ReLU activation |
| Learning Rate | 0.001 |
| Optimizer | Adam |
| Activation Function | ReLU |
| GAN Latent Dim | 70 |
| GAN Input dimension | 11 |
| GAN Hidden Layer | 64 |
| GAN Activation<br>GAN Optimizer | ReLU<br>Adam |
| GAN Output Layer Activation<br>Batch Size | Sigmoid<br>18 |
| BBN Input Dimensions | 7 |
| BBN Encoding Dimensions | 1 |
| BBN Learning function | Correlation Matrix with Threshold |
| BBN Prior Type | BDeu, dirichlet, K2 |
| BBN Sample Size | 20 |
| MAMRL Discount factor | 0.9 |
| MAMRL Number of Actions | 2 |
| MAMRL LSTM Units | 48 |
| MAMRL Activation Function (Policy) | Softmax |
| MAMRL Entropy Bonus | 0.05 |
| MAMRL Gradient Clipping Norm | 50.0 |




(a) RSSI of Amazon services. (b) CQI of Amazon services. (c) SNR of Amazon services. (d) NRxRSRP of Amazon services.

**FIGURE 3. Comparison of RSSI, CQI, SNR, NRxRSRP metrics between original and intelligent poisoning attacks by GenAI models such as GAN, VAE, Autoencoder for Amazon services.**

GAN exhibited higher computational complexity due to its adversarial training process but struggled to maintain the intricate network relationships of the original data, resulting in a less complex and less interconnected network. This indicates that while GANs can create realistic data, they may not effectively capture the nuanced dependencies required for robust trust evaluation in 6G architectures.

The performance of each GenAI model was evaluated after training and testing the models. This comparison was performed through both visual inspections and quantitative metrics, including Root Mean Square Error (RMSE) and Mean Squared Error (MSE). The analysis revealed insights into the model's effectiveness in replicating the characteristics of the original dataset. The evaluation process considered both training efficiency and the quality of poisoning network metrics and parameters. we have demonstrated how well our selected GenAI model produces realistic intelligent poisoning attack, we are now focusing on quantifying trust by using a Bayesian Belief Network as a core component of our Trust-By-Learning framework. We have designed a Bayesian Network using regression-based mutual information, leveraging several key Python libraries: numpy, pandas, networkx, matplotlib, pgmpy, and scikit-learn. It begins by importing these libraries, which are essential for data manipulation, network modeling, probabilistic graphical models, and statistical calculations. The code constructs a correlation matrix from the input data to identify potential relationships between variables, represented as nodes in a BayesianModel from pgmpy. Edges between nodes are added based on the correlation values, and Conditional Probability Distributions (CPDs) are estimated using maximum likelihood. It computes the mutual information (mi_value) between each pair of nodes using the mutual_info_regression function from scikit-learn, which quantifies the amount of shared information between two variables, capturing both linear and non-linear dependencies. A directed graph is created to visualize these relationships, with edge weights corresponding to the calculated mutual information values, helping to illustrate the strength of the dependencies in the network. The resulting Bayesian Network can be used for further probabilistic inference with tools like Variable Elimination from pgmpy. However,to effectively compare the evolving nature of intelligent attacks, the framework requires a mechanism for continuous learning and adaptation. Therefore, we compare the proposed TBL framework with a centralized method to justify the effectiveness of the centralized tanning with decentralized execution of the protection scheme.

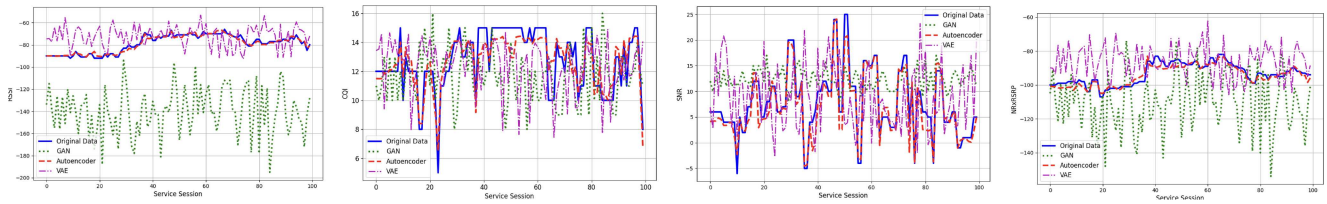
### B. PERFORMANCE ANALYSIS

In this experiment, our goal is to benchmark the proposed framework by comparing with baseline methods. This section describes the experimental findings and technical analysis conducted for the Amazon, Netflix, and Download services, focusing on the comparison between original and intelligent poisoning attacks generated using three different GenAI models. We explore various metrics including CQI, RSSI, SNR, and Neighbor cell Reference signal received power (NRxRSRP) through a sequence of visual representations. These metrics are visualized through a series of Figures. Each Figure provides insights into the effectiveness of GenAI models in replicating the underlying data distributions and characteristics. Moreover, we delve into the implications of these findings for network security, method efficiency, and the significance of intelligent poisoning attack generation for advancing 6G networks. Moreover, we delve into the implications of these findings for network security, method efficiency, and the significance of intelligent poisoning attack generation for advancing 6G networks.

Figures 3, 4, and 5 present a comparison of key communication metrics: RSSI, CQI, SNR, and NRxRSRP between the original data and intelligent poisoning attack for the Amazon, Netflix, and Download services, generated by the three GenAI models such as GAN, VAE, and Autoencoder.

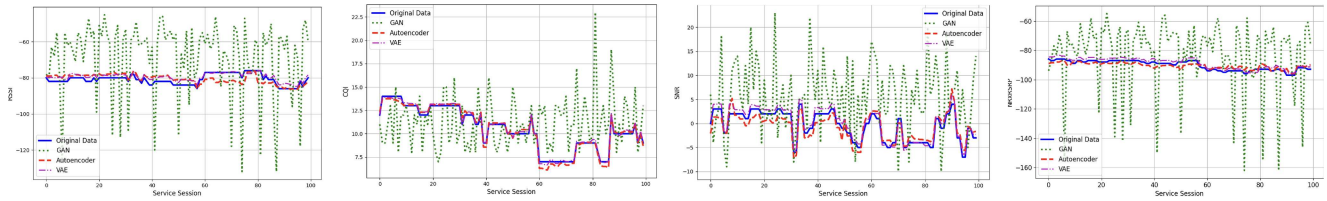
An evaluation of key metrics as mentioned above reveals that the Autoencoder model consistently demonstrated the most accurate replication of the original data distribution. This is evident in Figures 3(a), 3(b), 3(c), and 3(d) for Amazon service, followed by Figures (4(a), 4(b), 4(c)), and 4(d) represents Downlaod service. Further, intelligent attacks on the Netflix service are shown in Figures 5(a), 5(b), 5(c), and 5(d). The VAE model achieved a reasonable level of accuracy, while the GAN model exhibited less fidelity in replicating the original data patterns.

The Autoencoder's superior ability to learn and generalize signal strength patterns from the training data enables accurate replication of RSSI values as shown in Figures 3(a), 4(a), and 5(a) for these services. This fidelity is crucial for understanding and mitigating security vulnerabilities related to signal strength fluctuations, thereby enhancing




(a) RSSI of Download services. (b) CQI of Download services. (c) SNR of Download services. (d) NRxRSRP of Download services.

**FIGURE 4.** **Comparison of RSSI, CQI, SNR, NRxRSRP metrics between original and intelligent poisoning attacks by GenAI models such as GAN, VAE, Autoencoder for download services.**




(a) RSSI of Netflix service. (b) CQI of Netflix service. (c) SNR of Netflix service. (d) NRxRSRP of Netflix service.

**FIGURE 5.** **Comparison of RSSI, CQI, SNR, NRxRSRP metrics between original and intelligent poisoning attacks by GenAI models such as GAN, VAE, Autoencoder for Netflix services.**

the network's defense mechanisms. High-quality intelligent poisoning attacks RSSI data, aids in developing and refining 6G network algorithms and infrastructure, contributing to better network reliability and efficiency.

Similarly, by accurately replicating CQI values as shown in Figures 3(b), 4(b), and 5(b), the models can help simulate various network conditions, aiding in the development of robust security mechanisms against intelligent attacks that exploit CQI variations. The ability to generate realistic intelligent poisoning attack, demonstrates the value of GenAI for data augmentation and testing. This is crucial for network optimization, management, and ultimately, the ongoing development and enhancement of 6G technologies. Ensuring high-quality intelligent poisoning attack, especially for critical metrics like CQI, is essential for simulating and testing new 6G network features.

Replicating accurate SNR values as shown in Figure 3(c), is vital for assessing and enhancing the network's resilience to interference and noise-based attacks, which are common intelligent attack vectors. It is crucial for determining signal quality. High-fidelity intelligent poisoning attacks SNR data is significant for the development and testing of advanced 6G communication systems, ensuring they can maintain high performance under various conditions.

The model's ability to accurately replicate NRxRSRP values as shown in Figures 3(d), 4(d), and 5(d) which holds significant implications for network security and 6G development. It plays a critical role in radio resource management and interference coordination within the network. By generating realistic intelligent poisoning attacks data, we can develop robust strategies to detect and mitigate attacks that manipulate signal power levels, ultimately strengthening network security. Furthermore, high-fidelity intelligent poisoning attacks on NRxRSRP data are essential for designing and testing new 6G network features, ensuring optimal performance and efficient resource utilization in future network deployments.

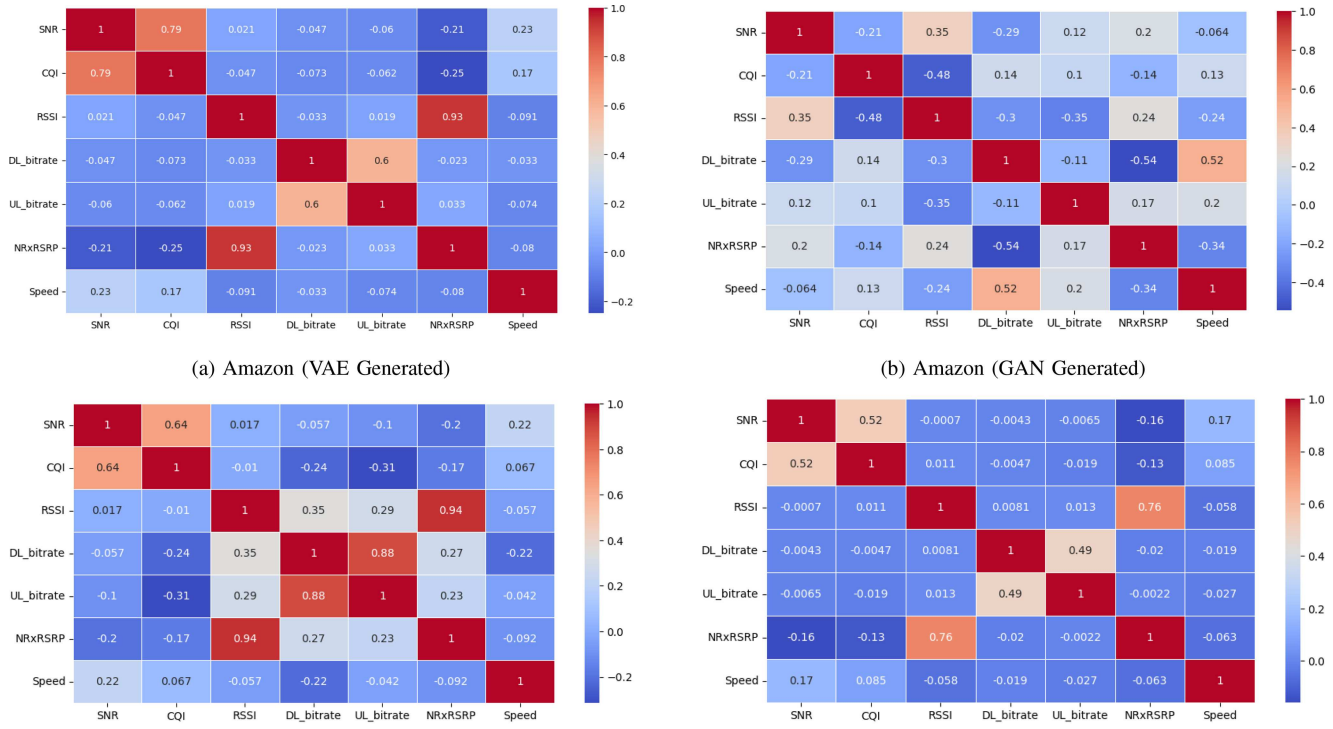
Once we have generated the intelligent poisoning attack for Amazon, Netflix, and Download services, we aim to analyze the differences between intelligent poisoning attacks generated for these services using various GenAI models like VAE, GAN, and Autoencoder. To achieve this, we employed BBNs to understand how the relationships between features differ between the original and generated data.

In our analysis, we constructed a BBN by first calculating the correlation matrix of our dataset to identify pairwise correlations between features. Based on the correlation values, we applied a threshold-based filtering method to determine which features are highly correlated and should be included in the BBN. We then estimated the Conditional Probability Distributions (CPDs) for each node using Maximum Likelihood Estimation (MLE) and Bayesian Estimation methods, except for the target variable. These CPDs represent the probabilities of each node conditioned on its parent nodes in the network.

Figures 6, 7, and 8 illustrate the correlation matrix comparisons for Amazon, Download, and Netflix services, respectively, between the original network parameters and metrics and those generated by the GenAI-based attack vector.

The color intensity in these heatmaps indicates the strength and direction of the correlations, with darker reds representing stronger positive correlations and darker blues representing stronger negative correlations.

Figure 6 illustrates the Amazon service, the original correlation matrix shows moderate correlations between SNR and CQI 0.52 in Figure 6(d) and a high correlation between RSSI and NRxRSRP 0.76. When comparing this to the



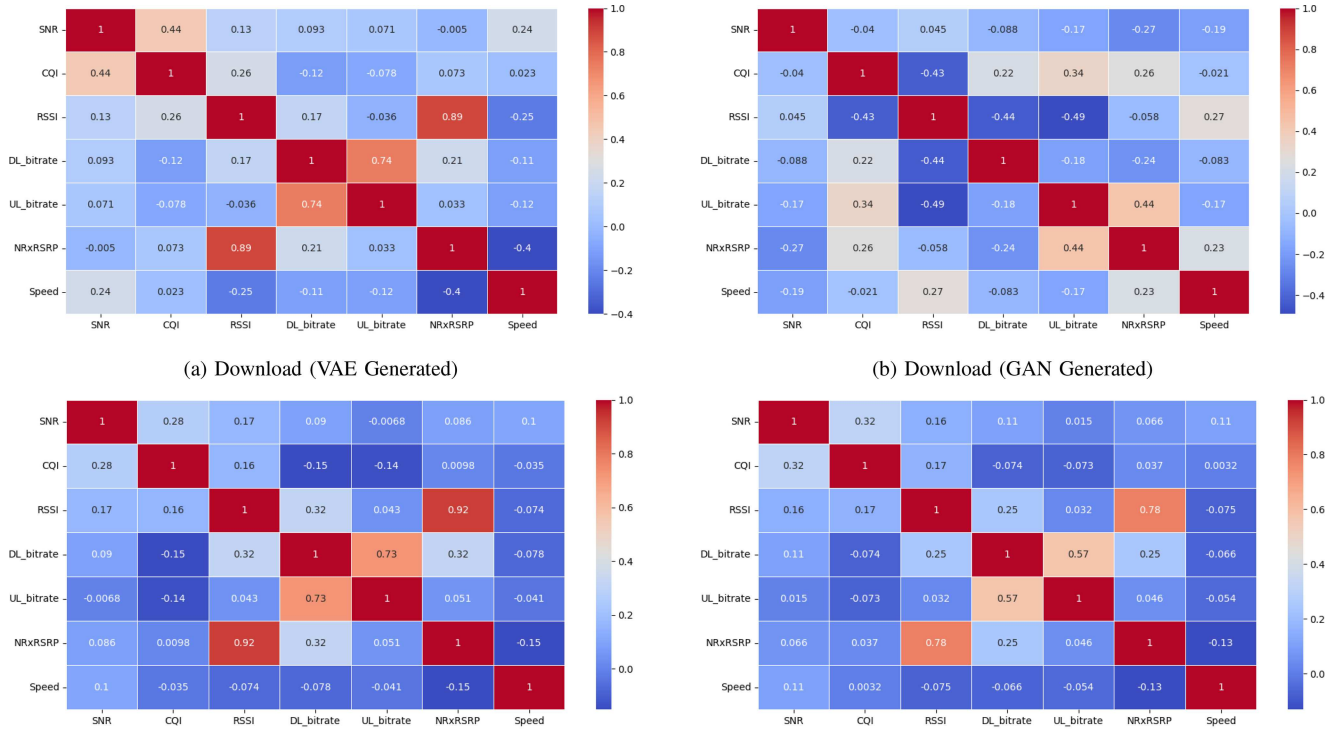

(a) Amazon (VAE Generated)

(b) Amazon (GAN Generated)

(c) Amazon (AE Generated)

(d) Amazon (Original)

**FIGURE 6.** Comparison of different Amazon data generation methods with the original data.




(a) Download (VAE Generated)

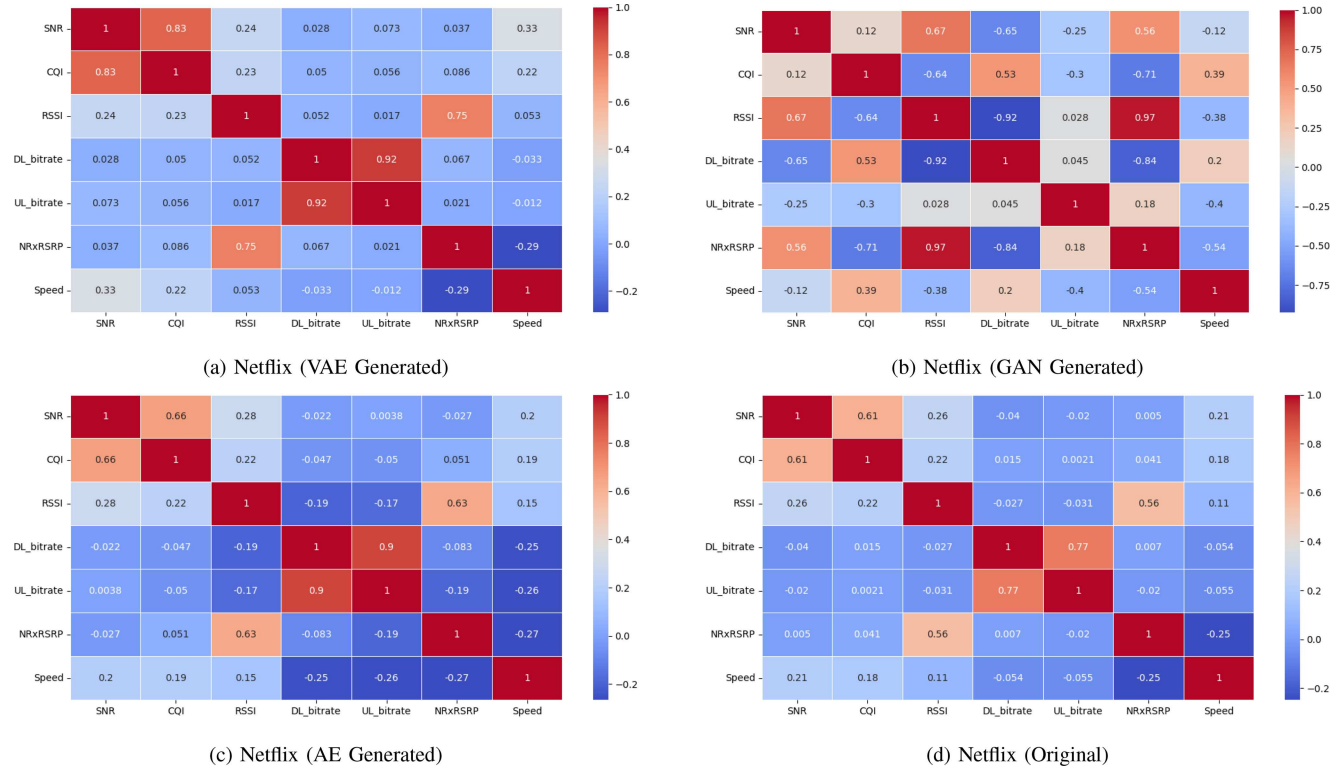
(b) Download (GAN Generated)

(c) Download (AE Generated)

(d) Download (Original)

**FIGURE 7.** Comparison of Download data generation methods with the original data.

VAE-generated intelligent poisoning attack, we observe that the intelligent poisoning attack shows stronger correlations between SNR and CQI 0.79 in Figure 6(a) and a very high correlation between RSSI and NRxRSRP 0.93. This indicates that the VAE model has amplified these relationships. The GAN-generated heatmap reveals similar patterns but with




(a) Netflix (VAE Generated)

(b) Netflix (GAN Generated)

(c) Netflix (AE Generated)

(d) Netflix (Original)

**FIGURE 8.** **Comparison of Netflix data generation methods with the original data.**

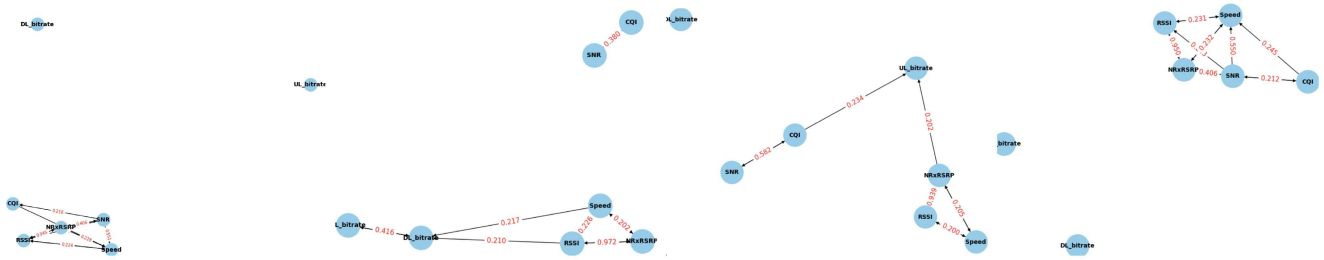
a negative correlation between SNR and CQI −0.21 in Figure 6(b), which is different from the original data pattern and a correlation between RSSI and NRxRSRP 0.24. This indicates that the GAN model has not really amplified these relationships. The Autoencoder-generated data shows an even higher correlation between RSSI and NRxRSRP 0.94 in Figure 6(c), and a high correlation between CQI and SNR 0.64 reflecting a slight exaggeration of feature dependencies.

Figure 7 depicts the Download service, the original data shows moderate correlations between RSSI and NRxRSRP 0.78 in Figure 7(d) and between DL_bitrate and UL_bitrate 0.57. The VAE-generated data shows moderate correlations between DL_bitrate and UL_bitrate 0.74 in Figure 7(a) and between RSSI and NRxRSRP 0.89. The GAN model captures similar trends with high correlation between UL_bitrate and DL_bitrate 0.73 in Figure 7(b). The Autoencoder-generated data displays consistent correlation patterns, though the correlation between SNR and CQI is 0.28 in Figure 7(c) is notably lower.

The original Netflix data shows good correlations between DL_bitrate and UL_bitrate 0.77, SNR and CQI 0.61 in Figure 8(d) and a moderate correlation between RSSI and NRxRSRP 0.56. The VAE-generated data for Netflix shows stronger correlations between SNR and CQI 0.79 in Figure 8(a) and a high correlation between RSSI and NRxRSRP 0.93, indicating stronger dependencies than in the original data. The GAN model captures similar trends with some variations, such as a moderate correlation between DL_bitrate and UL_bitrate 0.84 in Figure 8(b). The Autoencoder-generated data for Netflix shows slightly lower correlations between SNR and CQI 0.67 in Figure 8(c) and between RSSI and NRxRSRP 0.92 compared to the VAE and GAN models.

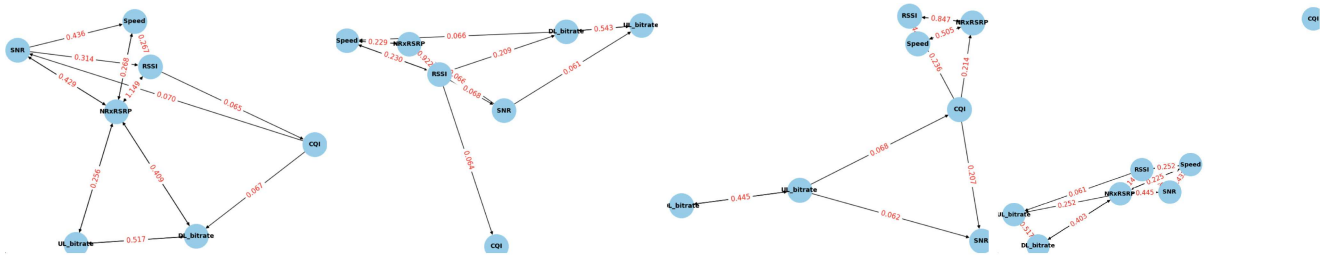
By comparing these intelligent poisoning attacks correlation matrices with those of the original datasets, we can analyze how the relationships between features differ across various generative models. This comparison helps us assess the fidelity of the intelligent poisoning attack and understand any discrepancies in feature dependencies. Across all datasets and models, there is a consistently high correlation between RSSI and NRxRSRP, indicating a strong probabilistic dependency that is accurately captured by all generative models. Some variations in correlations, such as those between DL_bitrate and UL_bitrate or SNR and CQI, suggest that different models capture different aspects of the data relationships. Each generative model seems to emphasize different feature dependencies, with Autoencoders generally showing higher correlation values, while GANs and VAEs show more moderate values.

We explore the effectiveness of generative models through the construction and analysis of Bayesian Network graphs. Using mutual information as a measure of dependency between variables, we compare the original data with data generated by Autoencoder, VAE, and GAN models across various datasets, including Amazon, Netflix, and Download services. The Bayesian Networks are constructed by adding nodes for each variable and the relationship between them based on a significant conditional probability



(a) Benign parameters & metrics. (b) Attack by Autoencoder. (c) Attack by VAE. (d) Attack by GAN.

**FIGURE 9.** Mutual information-based trust dependencies analysis of intelligent poisoning attacks on network metrics and parameters in wireless networks of Amazon services.



(a) Benign parameters & metrics. (b) Attack by Autoencoder. (c) Attack by VAE. (d) Attack by GAN.

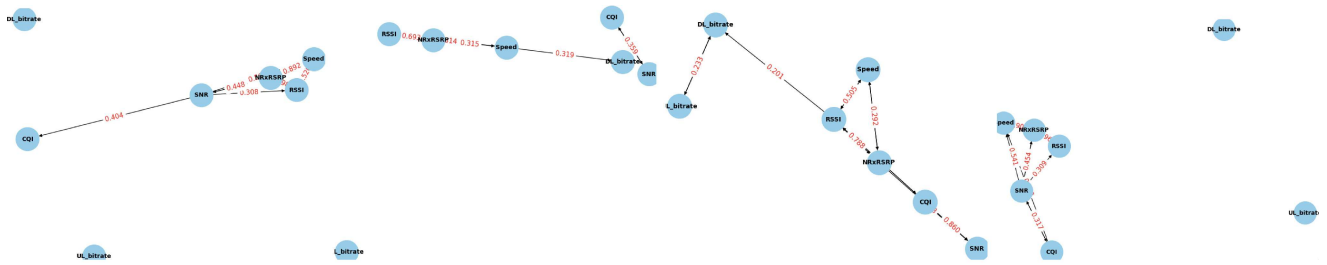
**FIGURE 10.** Mutual information-based trust dependencies analysis of intelligent poisoning attacks on network metrics and parameters in wireless networks of download services.

matrix, ensuring the most meaningful connections. The edges were introduced in the BBN graphs based on a threshold for mutual information greater than 0.2, indicating significant relationships between variables. This approach allows us to assess how well each generative model preserves the structure and dependencies of the original data, providing insights into their capabilities and limitations in data replication.

By constructing Bayesian Network models for each dataset type of the Amazon dataset involves comparing the original data with data generated from Autoencoder, VAE, and GAN models, focusing on mutual information values to understand how well each model captures the relationships between key variables. The original data's Bayesian Network Figure 9(a) displays a dense network with strong dependencies among metrics such as CQI, RSSI, SNR, and NRxRSRP. For instance, the mutual information between RSSI and SNR is particularly high at 0.766, indicating a strong correlation, while the relationship between NRxRSRP and SNR shows an even higher mutual information value of 0.944, reflecting a significant dependency. These values highlight the intricate and interdependent nature of the metrics in the original dataset.

The Autoencoder model closely replicates the original parameters network structure as in Figure 9(b), preserving key relationships with mutual information values that are similar to those in the original data. For example, the mutual information between RSSI and NRxRSRP in the Autoencoder-generated data is 0.972, almost mirroring the original, and the relationship between SNR and CQI retains a mutual information value of 0.380. The VAE model, Figure 9(c), also maintains key relationships effectively, with the mutual information between RSSI and NRxRSRP being 0.939, closely matching the original. However, the VAE tends to simplify the network slightly, resulting in weaker but still significant relationships. On the other hand, the GAN model, Figure 9(d) demonstrates a notable reduction in the strength of relationships, with mutual information values such as 0.212 between SNR and CQI, significantly lower than those in the original dataset, indicating that the GAN struggles to replicate the network's complexity for Amazon service parameters and metrics.

The Download service parameters and metrics analysis reveals a significant difference between the original data and the data produced by the GAN, VAE, and Autoencoder models. The original dataset shows a highly interconnected network as in Figure 10(a) with high values of mutual information amongst important variables. For instance, the mutual information value of 1.149 for the link between RSSI and NRxRSRP indicates a very strong dependency, whereas the value of 0.314 for the relationship between RSSI and SNR indicates robustness. In contrast, the Autoencoder-generated data retains significant relationships 10(b), such as the mutual information between RSSI and NRxRSRP at 0.922 and between NRxRSRP and speed at 0.230, demonstrating that it captures the essential dependencies of the original data, though with a slight reduction in the strength of some relationships. The VAE model (10(c)) also effectively preserves key relationships, maintaining mutual information of 0.847 between RSSI and NRxRSRP, closely aligning with the original, while slightly simplifying the network by reducing mutual information in some relationships. On the other hand, the GAN model in Figure 10(d)



(a) Benign parameters & metrics. (b) Attack by Autoencoder. (c) Attack by VAE. (d) Attack by GAN.

**FIGURE 11.** Mutual information-based trust dependencies analysis of intelligent poisoning attacks on network metrics and parameters in wireless networks of Netflix services.

shows a considerable drop in the strength and number of relationships, with mutual information values like 0.445 for RSSI and NRxRSRP, indicating weaker replication of the original network's complexity. This analysis highlights that the Autoencoder and VAE models are more successful in capturing the original data's intricate dependencies compared to the GAN model, which shows a less accurate replication of the downloaded data.

As per the BBN analysis for Netflix parameters and metrics, the original data exhibits strong mutual information values among key variables in Figure 11(a), such as mutual information of 0.892 between RSSI and NRxRSRP and 0.521 between RSSI and Speed, indicating significant dependencies. The Autoencoder maintains these relationships well 11(b), with mutual information values like 0.699 for RSSI and NRxRSRP, closely mirroring the original data, though some relationships are slightly weaker, such as the 0.315 mutual information between NRxRSRP and Speed. The VAE model in Figure (11(c)) captures the original data's dependencies even more effectively, with mutual information values that closely match the original, such as 0.892 for RSSI and NRxRSRP and 0.404 for SNR and CQI. In contrast, the GAN model shows a considerable reduction in the strength and number of relationships as in Figure 11(d), with mutual information values like 0.454 for RSSI and NRxRSRP, indicating a much less complex and less connected network structure compared to the original parameters and metrics.

Across the Amazon, Download, and Netflix parameters and metrics datasets, the analysis demonstrates that Autoencoder and VAE models consistently outperform the GAN model in replicating the intricate relationships present in the original data. The Autoencoder maintains significant dependencies with mutual information values closely matching those of the original data, though it tends to slightly weaken some relationships. The VAE model not only preserves the network complexity but also replicates key dependencies with high fidelity, making it the most effective in capturing the original data's structure. In contrast, the GAN model exhibits a marked reduction in the strength and number of relationships, leading to a less complex and less interconnected network. This trend is evident across all three datasets, indicating that while GANs may generate plausible data, they struggle to accurately replicate the nuanced dependencies of the original datasets.

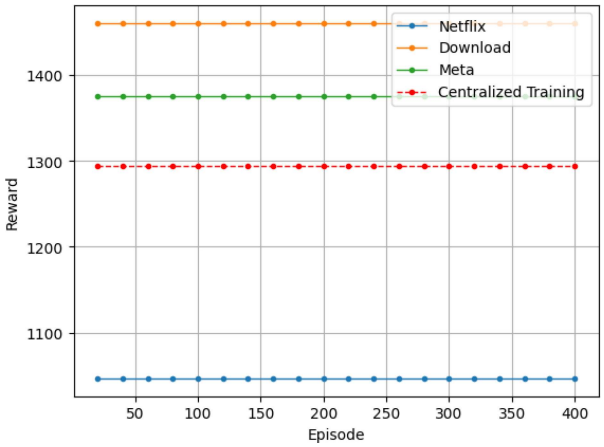



**FIGURE 12.** Cumulative training rewards based on accumulating CQI-based trust score.

In our experiment, we evaluate the performance of three agents, Amazon, Download, and Netflix, as well as a centralized training approach, all trained using the Multi-Agent Meta Reinforcement Learning (MAMRL) framework. Each agent is trained with a dataset that combines both original and intelligent poisoning attacks, the intelligent poisoning attack, which is regenerated using an autoencoder to ensure that the data retains significant characteristics of the original while introducing variability. These datasets are shuffled during training to ensure robust learning and generalization. The integration of both original and autoencoder-regenerated intelligent poisoning attack aims to enhance the agents' ability to adapt to varied network conditions and improve their overall performance in dynamic environments.

CQI is the key variable in calculating the trust score, which serves as a comprehensive metric to evaluate the agents' performance, as explained in 10. The trust score is used to evaluate the reliability of both original and intelligent poisoning attack in maintaining robust network performance and guiding the agents' decision-making processes.

The centralized learning scenario, as depicted in Figure 12, shows a high level of trust score stability, maintaining around 1294.3 per episode. This performance underscores the centralized approach's capability to handle a diverse dataset and maintain optimal network conditions effectively. The moderate reward levels suggest that centralized training is particularly adept at balancing original and

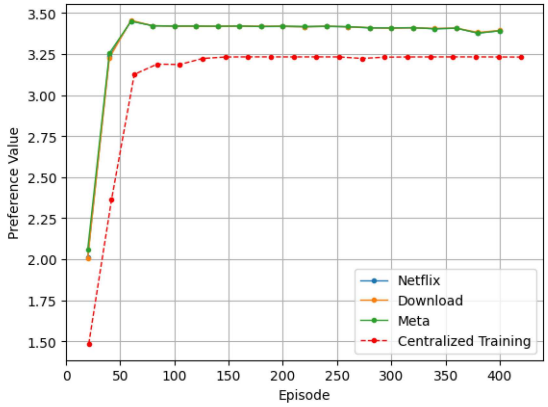



FIGURE 13. Preference value during Trust-By-Learning.

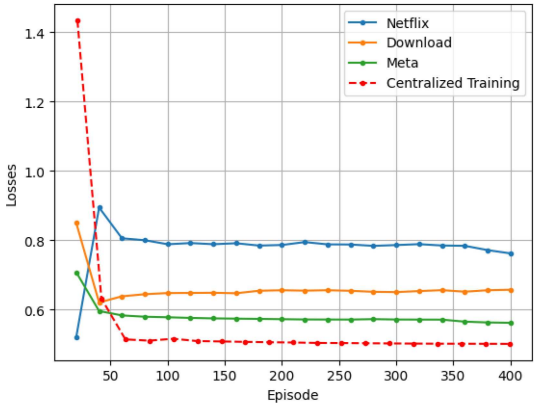



FIGURE 14. Convergence analysis by value loss.

intelligent poisoning attack. The Amazon agent performs moderately well with a stable trust score around 1375.65, Figure 12 showcasing its robust learning mechanism and superior ability to generalize from the diversified data to optimize performance. This agent's consistent trust score highlight its proficiency in managing network conditions and integrating varied data inputs effectively. On the other hand, the Netflix agent, while maintaining reward stability, achieves the lowest performance with a trust score of around 1046.79. This indicates potential deficiencies in its learning or variation in the dataset, which hinders model performance. Despite its stable rewards, the Netflix agent struggles to match the performance of the other agents. The Download agent has high performance with a stable reward of approximately 1460.45, surpassing the Netflix agent and Amazon agent. The stability in the Download agent's trust scores suggests effective management of network conditions and successful utilization of the mixed dataset for consistent performance.

The trust score plots reveal significant disparities in how each agent manages the mixed dataset of original and intelligent poisoning attack. The Download training approach achieves the highest and most stable trust scores, indicating its efficacy in maintaining balanced network conditions and optimizing performance. The Amazon agent also displays high and stable performance, demonstrating effective generalization and robust learning. The Centralized agent, while maintaining stability, achieves moderate trust scores, reflecting its ability to handle data diversity effectively. The Netflix agent, despite its stability, records the lowest trust scores, suggesting potential areas for improvement in its learning approach and data handling strategies. These findings underscore the importance of a well-structured trust score function and diverse training data in achieving stable and robust performance across different training scenarios.

In Figure 13, we compare the value per episode across all agents, illustrating a clear differentiation in performance. The Amzon agent demonstrates the highest and most stable value, approaching 3.5, which signifies strong learning capabilities and effective integration of diverse data inputs. The centralized training approach follows closely, maintaining a steady value of around 3.3, reflecting its proficiency in leveraging the comprehensive dataset for robust performance. The Download agent achieves a moderate value of approximately 3.2, indicating effective learning but slightly less optimization compared to the Meta agent. The Netflix agent records the lowest value, just above 2.5, suggesting challenges in handling the mixed dataset and potential areas for improvement in learning strategies.

Overall, the graphs collectively highlight significant differences in how each agent handles and learns from the mixed dataset. The centralized approach and Meta agent show superior performance with high and stable values, indicating effective learning and data management. The Download agent achieves moderate success, while the Netflix agent's lower value highlights opportunities for improvement in data handling and learning strategy.

The graph in Figure 14 presents a comparative analysis of losses per episode for all agents. The Amazon agent achieves the lowest loss values, rapidly decreasing to around 0.5 within the first 50 episodes and maintaining stability thereafter. This indicates effective learning and quick convergence to optimal policies. The centralized training approach also exhibits a significant reduction in losses, stabilizing at around 0.6, reflecting its ability to leverage a comprehensive dataset effectively. The Download agent's losses stabilize around 0.6 as well, demonstrating moderate efficiency in managing the mixed dataset. The Netflix agent, however, records the highest initial loss values, peaking at 1.4, and gradually reduces to around 0.7. This suggests challenges in learning from the combined dataset, highlighting areas for potential improvement.

Figure 15 illustrates the comparative performance of four agents—Netflix, Download, Amazon, and a centralized training approach—across 400 training episodes. The graph highlights that the Netflix agent experiences a substantial decrease in policy loss, stabilizing at the lowest values around $-0.6$, indicating effective policy optimization and learning. The centralized training method also performs robustly, showing a rapid decrease in policy loss to near-zero

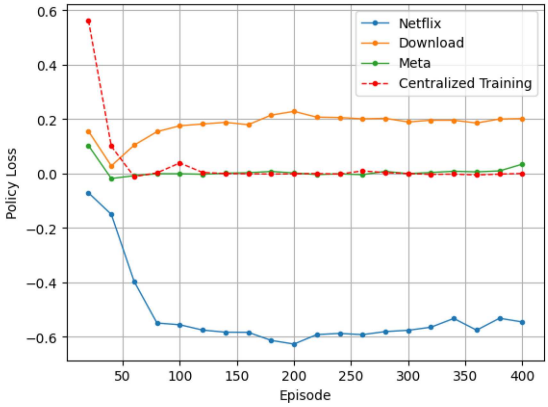



**FIGURE 15.** Policy loss during agents' training.

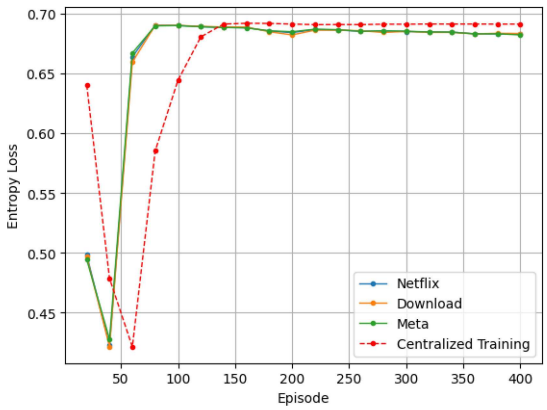



**FIGURE 16.** Entropy loss during agents' training.

levels, reflecting its efficiency in achieving optimal policy quickly. The Amazon agent maintains a low and stable policy loss, demonstrating consistent and effective learning. In contrast, the Download agent's policy loss initially rises and then stabilizes at a relatively higher value around 0.15, suggesting less effective adaptation and optimization compared to the other agents. Overall, the centralized and Netflix agents lead in minimizing policy loss, showcasing superior learning capabilities and performance in aligning actions with the optimal policy.

Figure 16 further complements the analysis of policy loss by shedding light on the exploration-exploitation dynamics across the agents. Initially, the centralized training approach exhibits a higher entropy loss around 0.7, indicative of a significant exploratory phase, which aligns with its rapid initial adjustments seen in the policy loss Figure. This entropy quickly stabilizes at approximately 0.68, highlighting a balanced shift towards more deterministic policies as training progresses. Similarly, the Amazon and Download agents begin with lower entropy, around 0.45, suggesting a controlled exploration strategy from the outset. They both follow a trajectory that sees a rapid increase and subsequent stabilization at around 0.68, mirroring the centralized method's trend towards a balanced exploration-exploitation trade-off. The Netflix agent, however, starts with the lowest entropy, reflecting minimal initial exploration, and undergoes significant fluctuations before settling near 0.67. This initial low entropy, combined with the substantial decrease in policy loss, suggests that Netflix's policy initially focused more on refining a deterministic strategy rather than exploring new actions. Overall, the entropy loss Figure indicates that while the agents adopted varying levels of exploration initially, they all converge towards a similar level of policy stability, reflecting an effective learning process that balances exploration with policy refinement, thus complementing the trends observed in the policy loss metrics.

In summary, our evaluation of the Amzon, Download, and Netflix agents, alongside a centralized training approach using the MAMRL framework, highlights significant differences in their performance and learning dynamics. The centralized method and Amzon agent emerged as the top performers, demonstrating high and stable trust scores and values, indicating effective learning and data management from a mixed dataset of original and intelligent poisoning attack. The Download agent also showed competent performance, albeit with room for improvement in data handling strategies. The Netflix agent, despite achieving reward stability, exhibited the lowest trust scores and struggled with policy optimization, suggesting challenges in processing the combined dataset effectively. The analysis of entropy loss further underscored the importance of a balanced exploration-exploitation strategy, with all agents eventually converging to similar levels of policy stability. These findings underscore the critical role of diverse training data and robust learning mechanisms in enhancing agent performance in dynamic environments, positioning the centralized approach and Amazon agent as exemplary models for future reinforcement learning applications.

## VII. CONCLUSION

In this work, we have proposed a new Trust-By-Learning framework to secure upcoming 6G wireless networks from GenAI-driven intelligent attacks by understanding uncertainty, severity, root cause, and trustworthy service aggregation. To cope uncertain nature of GenAI-driven intelligent attacks in network parameters and metrics, we device a narrow GenAI to produce such poisoning attacks and examine them by deploying Dempster–Shafer–based evidence theory for quantifying trust. Then, we have developed a meta-RL-based Markov Decision Process learning to mitigate intelligent attacks by enabling trustworthy service aggregation in a wireless network. The proposed TBL framework has established a secure and trustworthy wireless communication by capturing the highly uncertain intelligent poisoning attacks while employing long-term temporal dependencies in network parameters and metrics for trust establishment. Our experimental results show that the Autoencoders and VAEs successfully replicate dependencies with scores of 0.972 for Amazon, 0.922 for Download, and 0.892 for Netflix. In contrast, GANs demonstrate weaker

**TABLE 3.** Summary of abbreviation.

| Abbreviation | Meaning |
|---|---|
| 6G | Sixth-generation |
| AWGN | Additive white Gaussian noise |
| CAV | Connected and Autonomous Vehicles |
| CPDs | Conditional Probability Distributions |
| CQI | Channel quality indicator |
| DRL | Deep Reinforcement Learning |
| GANs | Generative Adversarial Networks |
| GenAI | Generative AI |
| gNBs | Next-generation NodeBs |
| IoE | Internet of Everything |
| LLMs | Large Language Models |
| MAMRL | Multi-Agent Meta Reinforcement Learning |
| MLE | Maximum Likelihood Estimation |
| MSE | Mean Squared Error |
| NRxRSRP | Neighbor cell Reference signal received power |
| RMSE | Root Mean Square Error |
| RSRP | Reference signal received power |
| RNN | Recurrent neural network |
| RSSI | Received signal strength indicator |
| SINR | Signal-to-interference-plus-noise ratio |
| TBL | Trust-By-Learning |
| VAE | Variational autoencoder |
| XR | Extended Reality |
| ZTNs | Zero Touch Networks |

performance in replicating dependencies, with scores of 0.212 for Amazon, 0.445 for Download, and 0.454 for Netflix across all network parameters and metrics. In summary, this work investigates the capabilities of native GenAI to generate poisoning attacks in wireless communication systems, while also analyzing the nature of such intelligent threats to facilitate effective mitigation strategies. In the future, the proposed TBL framework will be extended to vertical enablers of 6G wireless networks such as connected autonomous vehicles, critical infrastructure, and smart grids to enhance trustworthy and resilient network operations.

## APPENDIX

See Table 3.

## REFERENCES


[1] M. S. Munir et al., "Securing next-generation wireless networks against native GenAI attacks: An evidence-theoretic approach," in *Proc. 21st Int. Wireless Commun. Mobile Comput. Conf.*, 2025, pp. 805–811.

[2] C. K. Thomas, C. Chaccour, W. Saad, M. Debbah, and C. S. Hong, "Causal reasoning: Charting a revolutionary course for next-generation AI-native wireless networks," *IEEE Veh. Technol. Mag.*, vol. 19, no. 1, pp. 16–31, Mar. 2024.

[3] W. Saad et al., "Artificial general intelligence (AGI)-native wireless systems: A journey beyond 6G," *Proc. IEEE*, early access, Mar. 17, 2025, doi: 10.1109/JPROC.2025.3526887.

[4] P. Porambage, G. Gür, D. P. M. Osorio, M. Liyanage, A. Gurtov, and M. Ylianttila, "The roadmap to 6G security and privacy," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 1094–1122, 2021.

[5] M. Zawish et al., "AI and 6G into the metaverse: Fundamentals, challenges and future research trends," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 730–778, 2024.

[6] A. Celik and A. M. Eltawil, "At the dawn of generative AI era: A tutorial-cum-survey on new frontiers in 6G wireless intelligence," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 2433–2489, 2024.

[7] Q. Luo, Z. Han, and B. Di, "Meta-critic reinforcement learning for intelligent omnidirectional surface assisted multi-user communications," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 9085–9098, Aug. 2024.

[8] R. Chataut, M. Nankya, and R. Akl, "6G networks and the AI revolution—Exploring technologies, applications, and emerging challenges," *Sensors*, vol. 24, no. 6, p. 1888, 2024.

[9] J. Wang et al., "A unified framework for guiding generative AI with wireless perception in resource constrained mobile edge networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 11, pp. 10344–10360, Nov. 2024.

[10] M. S. Munir, S. Proddatoori, M. Muralidhara, W. Saad, Z. Han, and S. Shetty, "A zero trust framework for Realization and defense against generative AI attacks in power grid," in *Proc. IEEE Int. Conf. Commun.*, 2024, pp. 2482–2488.

[11] C. D. Alwis et al., "Survey on 6G frontiers: Trends, applications, requirements, technologies and future research," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 836–886, 2021.

[12] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 84–90, Aug. 2019.

[13] F. Naeem, M. Ali, G. Kaddoum, C. Huang, and C. Yuen, "Security and privacy for reconfigurable intelligent surface in 6G: A review of prospective applications and challenges," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 1196–1217, 2023.

[14] Y.-C. Wang, J. Xue, C. Wei, and C. C. J. Kuo, "An overview on generative AI at scale with edge–cloud computing," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 2952–2971, 2023.

[15] B. D. Son et al., "Adversarial attacks and defenses in 6G network-assisted IoT systems," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 19168–19187, Jun. 2024.

[16] A. S. Ali, D. M. Manias, A. Shami, and S. Muhaidat, "Leveraging large language models for DRL-based anti-jamming strategies in zero touch networks," 2023, *arXiv:2308.09376*.

[17] V. C. Andrei, A. Djuhera, X. Li, U. J. Mönich, H. Boche, and W. Saad, "Resilient-by-design framework for MIMO-OFDM communications under smart jamming," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2024, pp. 1968–1973.

[18] H. Chen, K. Tu, J. Li, S. Tang, T. Li, and Z. Qing, "6G wireless communications: Security technologies and research challenges," in *Proc. Int. Conf. Urban Eng. Manage. Sci. (ICUEMS)*, 2020, pp. 592–595.

[19] S. Ankalaki, A. R. Atmakuri, M. Pallavi, G. S. Hukkeri, T. Jan, and G. R. Naik, "Cyber attack prediction: From traditional machine learning to generative artificial intelligence," *IEEE Access*, vol. 13, pp. 44662–44706, 2025.

[20] R. Mohawesh, M. Ottom, and H. B. Salameh, "A data-driven risk assessment of cybersecurity challenges posed by generative AI," *Decis. Anal. J.*, vol. 15, Jul. 2025, Art. no. 100580.

[21] M. S. Munir, S. Proddatoori, M. Muralidhara, and S. Shetty, "A Bayesian belief network framework for protecting generative AI-driven attack in smart grid communication," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, 2025, pp. 168–173.

[22] C. Chaccour, W. Saad, M. Debbah, Z. Han, and H. V. Poor, "Less data, more knowledge: Building next-generation semantic communication networks," *IEEE Commun. Surveys Tuts.*, vol. 27, no. 1, pp. 37–76, Feb. 2025.

[23] M. S. Munir, S.-B. Park, and C. S. Hong, "An explainable artificial intelligence framework for quality-aware IoE service delivery," in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 4787–4793.

[24] X. Li, Q. Fang, and L. Shi, "A effective SINR link to system mapping method for CQI feedback in TD-LTE system," in *Proc. IEEE 2nd Int. Conf. Comput., Control Ind. Eng.*, vol. 2, 2011, pp. 208–211.

[25] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, "Beyond throughput, the next generation: A 5G dataset with channel and context metrics," in *Proc. 11th ACM Multimedia Syst. Conf.*, 2014, pp. 304–308.

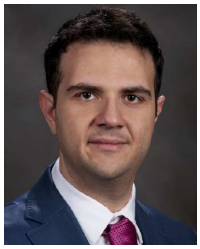
[26] "5G; NR; physical layer procedures for data, version 16.2.0 Release 16," 3GPP, Sophia Antipolis, France, Rep. TS 38.214, Jul. 2020. Accessed: May 5, 2024. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/138200_138299/138214/16.02.00_60/ts_138214v160200p.pdf
[27] M. S. Munir et al., "Neuro-symbolic explainable artificial intelligence twin for zero-touch IoE in wireless network," *IEEE Internet Things J.*, vol. 10, no. 24, pp. 22451–22468, Dec. 2023.
[28] G. Shafer, *A Mathematical Theory of Evidence*, vol. 42. Princeton, NJ, USA: Princeton Univ. Press, 1976.
[29] S. Narayan, "The generalized sigmoid activation function: Competitive supervised learning," *Inf. Sci.*, vol. 99, nos. 1–2, pp. 69–82, 1997.
[30] B. De Moor, "Structured total least squares and L2 approximation problems," *Linear Algebra Appl.*, vols. 188–189, pp. 163–205, Jul./Aug. 1993.
[31] D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–41.
[32] I. Cohen et al., "Pearson correlation coefficient," in *Noise Reduction in Speech Processing*. Heidelberg, Germany: Springer, 2009, pp. 1–4.
[33] M. S. Munir, K. T. Kim, K. Thar, D. Niyato, and C. S. Hong, "Risk adversarial learning system for connected and autonomous vehicle charging," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 15184–15203, Aug. 2022.
[34] M. S. Munir, N. H. Tran, W. Saad, and C. S. Hong, "Multi-agent meta-reinforcement learning for self-powered and sustainable edge computing systems," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 3, pp. 3353–3374, Sep. 2021.
[35] M. S. Munir, S. F. Abedin, D. H. Kim, N. H. Tran, Z. Han, and C. S. Hong, "A multi-agent system toward the green edge computing with microgrid," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–7.

**MD SHIRAJUM MUNIR** (Member, IEEE) received the B.S. degree in computer science and engineering from Khulna University, Khulna, Bangladesh, in 2010, and the Ph.D. degree in computer engineering from Kyung Hee University, South Korea, in 2021, where he is also served as a Postdoctoral Researcher with the Department of Computer Science and Engineering from September 2021 to August 2022. He is currently working as an Assistant Professor of Computer Science with the School of Computing, Analytics, and Modeling, University of West Georgia, Carrollton, GA, USA. Previously, he worked as a Research Assistant Professor with the School of Cybersecurity, Old Dominion University, USA, from July 2023 to August 2024, and performed the role of a Visiting Faculty with the Air Force Research Lab, Rome, NY, USA, from May 2024 to July 2024. He served as a Postdoctoral Research Associate with Virginia Modeling, Analysis, and Simulation Center, Old Dominion University from September 2022 to July 2023. Before his Ph.D., he served as a Software Engineer, a Senior Software Engineer, and a Lead Engineer with the Solution Laboratory, Samsung Research and Development Institute, Dhaka, Bangladesh, from 2010 to 2016. His research interests include machine learning, data science, trustworthy AI and stochastic models, wireless communication, healthcare, industrial IoT, Cyber-physical systems, future internet, and resilient smart grid. He is a recipient of the 2025 Ralph E. Powe Junior Faculty Enhancement Award and the NSF Early-Career CRII Award. He has been serving as an Associate Editor for IEEE INTERNET OF THINGS JOURNAL since April 2023.

**SRAVANTHI PRODDATOORI** received the master's degree in computer science from Old Dominion University, where she is a Software Engineer. Her expertise spans full-stack development, machine learning, and data analysis, with hands-on experience in developing mobile and web applications, performing data analysis, and applying generative AI models to simulate cyberattacks and detect anomalies in smart grid communication systems. She has contributed to enhancing security in smart grid communication systems and has trained various native generative AI models. Her work focuses on Meta-Adversarial Meta-Reinforcement Learning to address cyber threats in 6G networks. With a strong focus on real-world impact, she applies her skills in AI, full-stack development, and system optimization to tackle complex challenges in cybersecurity and intelligent automation.

**MANJUSHREE MURALIDHARA** received the master's degree in computer science from Old Dominion University, where she is a Senior Analyst. Her expertise spans machine learning, data analysis, and network security, with hands-on experience in PLC design and anomaly analysis. She has also worked extensively on optimizing systems and enhancing security, focusing on Meta-Adversarial Meta-Reinforcement Learning to mitigate cyber threats in 6G networks. With a focus on real-world applications, she applies her skills in AI, machine learning, and data-driven systems to solve challenges in security and optimization.

**TRINIDAD MARIO DENA** received the master's degree from the University of West Georgia in 2025, where he is currently pursuing the undergraduate degree in computer science with the School of Computing, Analytics, and Modeling. His research interests include explainable and trustworthy AI, Internet of Things networks, and cyber-physical system design.



**WALID SAAD** (Fellow, IEEE) received the Ph.D. degree from the University of Oslo in 2010. He is currently a Full Professor with the Department of Electrical and Computer Engineering, Virginia Tech, where he leads the Network sciEnce, Wireless, and Security (NEWS@VT) Laboratory, within the Wireless@VT Research Group. His research interests include wireless networks, machine learning, game theory, security, unmanned aerial vehicles, cyber-physical systems, smart grids, and network science. He is also the recipient of the NSF CAREER award in 2013, the AFOSR summer faculty fellowship in 2014, and the Young Investigator Award from the Office of Naval Research (ONR) in 2015. He was an author/co-author of ten conference best paper awards at WiOpt in 2009, ICIMP in 2010, IEEE WCNC in 2012, IEEE PIMRC in 2015, IEEE SmartGridComm in 2015, EuCNC in 2017, IEEE GLOBECOM in 2018, IFIP NTMS in 2019, IEEE ICC in 2020, and IEEE GLOBECOM in 2020. He is the recipient of the 2015 Fred W. Ellersick Prize from the IEEE Communications Society, the 2017 IEEE ComSoc Best Young Professional in Academia award, the 2018 IEEE ComSoc Radio Communications Committee Early Achievement Award, and the 2019 IEEE ComSoc Communication Theory Technical Committee. He was also a co-author of the 2019 IEEE Communications Society Young Author Best Paper. From 2015 to 2017, he was named the Stephen O. Lane Junior Faculty Fellow at Virginia Tech and, in 2017, he was named College of Engineering Faculty Fellow. He received the Dean's award for Research Excellence from Virginia Tech in 2019. He currently serves as an Editor for the IEEE TRANSACTIONS ON MOBILE COMPUTING and the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING. He is an Editor-at-Large of the IEEE TRANSACTIONS ON COMMUNICATIONS. He is an IEEE Distinguished Lecturer.

**ZHU HAN** (Fellow, IEEE) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively. From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, MD, USA. From 2003 to 2006, he was a Research Associate with the University of Maryland. From 2006 to 2008, he was an Assistant Professor with Boise State University, Idaho. He is currently a John and Rebecca Moores Professor with the Electrical and Computer Engineering Department, Computer Science Department, University of Houston, TX, USA. He is a 1% highly cited researcher since 2017 according to Web of Science. His main research targets on the novel game-theory related concepts critical to enabling efficient and distributive use of wireless networks with limited resources. His other research interests include wireless resource allocation and management, wireless communications and networking, quantum computing, data science, smart grid, carbon neutralization, security, and privacy. He received the NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, the IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in IEEE JSAC) in 2016, the IEEE Vehicular Technology Society 2022 Best Land Transportation Paper Award, and several best paper awards in IEEE conferences. He is also the winner of the 2021 IEEE Kiyo Tomiyasu Award (an IEEE Field Award), for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: "for contributions to game theory and distributed management of autonomous communication networks." He was an IEEE Communications Society Distinguished Lecturer from 2015 to 2018 and an ACM Distinguished Speaker from 2022 to 2025. He has been an AAAS Fellow since 2019, and an ACM Fellow since 2024.

**SACHIN SHETTY** (Senior Member, IEEE) received the Ph.D. degree in modeling and simulation from Old Dominion University in 2007 under the supervision of Prof. Song. He is an Executive Director with the Center of Secure and Intelligent Critical Systems, Office of Enterprise Research and Innovation (OERI), Old Dominion University. He holds a joint appointment as a Professor with the Department of Electrical and Computer Engineering. Prior to joining Old Dominion University, he was an Associate Professor with the Electrical and Computer Engineering Department, Tennessee State University. He was also an Associate Director with the Tennessee Interdisciplinary Graduate Engineering Research Institute and directed the Cyber Security Laboratory, Tennessee State University. He also holds a dual appointment as an Engineer with the Naval Surface Warfare Center, Crane Indiana. He has authored and coauthored over 150 research articles in journals and conference proceedings and two books. His research interests lie at the intersection of computer networking, network security, and machine learning. He is the recipient of the Fulbright Specialist Award, the EPRI Cybersecurity Research Challenge Award, the DHS Scientific Leadership Award and has been inducted in Tennessee State University's million dollar club. He has served on the technical program committee for ACM CCS, IEEE INFOCOM, IEEE ICDCN, and IEEE ICCCN. He is a CCI Fellow.