A Defensive Strategy Against Beam Training Attack in 5G mmWave Networks For Manufacturing

Son Dinh-Van, Tiep M. Hoang, Berna Bulut Cebecioglu, Daniel S. Fowler, Yuen Kwan Mo, and Matthew D. Higgins, *Senior Member, IEEE*

***Abstract*—Millimeter-wave (mmWave) carriers are an essential building block of fifth-generation (5G) systems. Satisfactory performance of the communications over the mmWave spectrum requires an alignment between the signal beam of the transmitter and receiver, achieved via beam training protocols. Nevertheless, beam training is vulnerable to jamming attacks, where the attacker intends to send jamming signals over different spatial directions to confuse legitimate nodes. This paper focuses on defending against this attack in smart factories where a moving Automated Guided Vehicle (AGV) communicates with a base station via a mmWave carrier. We introduce a defensive strategy to cope with jamming attacks, including two stages: jamming detection and jamming mitigation. Developed based on autoencoders, both algorithms can learn the characteristics/features of the received signals at the AGV. They can be employed consecutively before performing the downlink data transmission. In particular, once a jamming attack is identified, the jamming mitigation can be utilized to retrieve the corrupted received signal strength vector, allowing a better decision during the beam training operation. In addition, the proposed algorithm is straightforward and fully compliant with the existing beam training protocols in 5G New Radio. The numerical results show that not only the proposed defensive strategy can capture more than 80% of attack events, but it also improves the average signal-to-interference-plus-noise-ratio significantly, i.e., up to 15 dB.**

***Index Terms*—Attack detection, beam training, beam training attack, machine learning, mmWave, PHY-layer security, 5G.**

I. Introduction

The fifth generation (5G) of mobile communications is playing a key role in solving the growing demand for mobile data and future business applications through the delivery of multi-Gbps ultra-low-latency high reliability connections. Primary verticals for deployment include Industry 4.0, health care, tactile entertainment, future finance, and autonomous transportation [1]. In the context of Industry 4.0 and/or manufacturing, process productivity, efficiency and quality gains can be found from automated robotic systems such as AGVs. Such gains, however, are only achievable if the machines are truly flexible and completely mobile. Thus the 5G eco-system is seen as the key connectivity enabler to supersede what was only possible using a wired (typically Ethernet) infrastructure.

Due to the explosive growth of mobile data demand, 5G networks would make use of spectrum in the millimeter-wave (mmWave) bands to improve communication capacities even further [2]. Nevertheless, wireless communications at mmWave frequencies are challenging due to severe free-space path-loss and atmospheric absorption. Owing to the beamforming gain, highly directional transmission links, typically using antenna arrays with proper signal processing techniques, can be used to overcome this issue. Directional links, however, require a fine alignment between the transmitter and receiver beams, achieved via a set of operations known as *beam training* [3]. In particular, beam training allows the transmitter, such as the 5G base station (BS), to transmit reference signals (RSs) sequentially using multiple beams over various spatial directions. Subsequently, the user equipment (UE) measures the received signal strength (RSS) corresponding to each beam before determining the best beam that provides the strongest RSS, followed by a report of beam selection back to the BS [4], [5]. The beam training procedure is essential for wireless communications at mmWave since a beam misalignment might lead to a degradation in the channel gain between transceivers [4].

Despite recent developments, 5G technologies are still under active investigation, and many obstacles and issues remain. One key concern is a secured beam management procedure at the physical (PHY) layer. Due to the broadcast nature of wireless communications, an adversary can overhear and jam transmissions. In the context of 5G mmWave, the adversary can launch a jamming attack targeting the beam training procedure, causing 5G devices to make the incorrect beam selection decision and reducing the data rate enormously.

Machine learning (ML) is anticipated to play a crucial part in the evolution of 5G, the Industrial Internet of Things (IIoTs), and manufacturing. ML also enjoys much promise in improving security for the 5G wireless communication systems at the PHY layer [6], [7]. In this vein, we consider including ML in the beam training procedure to defend against jamming attacks in the context of manufacturing. Given that malicious attacks can cause severe damage to plant equipment, disrupt the mass production process, and the loss of confidential information, the use of ML for securing 5G communications in manufacturing is of salient significance.

Son Dinh-Van, Yuen Kwan Mo and Matthew D. Higgins are with Warwick Manufacturing Group, the School of Engineering, University of Warwick, Coventry, CV4 7AL, U.K. Email: {son.v.dinh, tony.mo, m.higgins}@warwick.ac.uk.

Tiep M. Hoang is with the Department of Electrical Engineering, the University of Colorado Denver, United States. Email: minhtiep.hoang@ucdenver.edu.

Berna Bulut Cebecioglu is with School of Computing and Digital Technology, Birmingham City University, Birmingham B4 7XG, U.K. Email: berna.bulut@bcu.ac.uk.

Daniel S. Fowler is with Warwick Manufacturing Group, University of Warwick, Coventry, CV4 7AL, U.K. Email: dan.fowler@warwick.ac.uk.

This work was supported in part by the WMG Centre High Value Manufacturing Catapult, University of Warwick, Coventry, U.K.

A. Related Research

In mmWave communications, analog beamformers are normally selected from a codebook consisting of beam patterns at different resolutions using a beam training strategy. Each beamforming given in the codebook directs transmission in a particular angular direction. There is various research concentrating on beam training for mmWave communications. One common method is to use an iterative process to measure the RSS over its codebook [4], [5], [8], for discovering the angular directions of the strongest signal between the transmitter and receiver without explicit channel estimation. This training method has also been implemented in standards such as IEEE 802.15.3c, IEEE 802.11ad, and WirelessHD [2]. More sophisticated discovery techniques were also studied such as channel-sparsity-based beam training [9], and context-information-based search [10]. Recently, ML has also been adopted extensively for beam training in 5G mmWave communications [11]–[13].

The 5G PHY-layer vulnerability was studied in [14]–[17]. In particular, the research proposed in [14] investigated the impact of jamming attacks on the physical downlink control channel (PDCCH) and RS in 5G communications. The study showed that jamming attacks are highly effective and can be used to disrupt any 5G UE communications. The authors in [15] developed two adversarial attack strategies to fool the deep learning models built for dynamic spectrum sharing and signal authentication in network slicing. In [16], the authors designed a malicious attack against learning-based beam training for 5G mmWave networks. Herein, a deep neural network was trained to enhance the robustness and latency of the beam selection process. To attack, the jammer adds a perturbation, which is designed carefully, to the neural network input so that the legitimate 5G user fails to classify beam patterns. However, this study did not consider the effects of the wireless channel. Since the wireless channel is stochastic and highly dynamic, the perturbation design should consider the channel state information (CSI). More recently, in [18], the authors also raised security concerns related to deep learning-based beam training for mmWave communications. It stated that despite an increase in cellular performance, the security and privacy issues of ML in 5G and beyond networks are still being ignored. Thus, the study developed a defended ML-based beam prediction against adversarial attacks, which achieves a nearly identical performance to the undefended model without attack.

More relevant to jamming attack detection, in [19], [20], the authors employed a statistical-based technique for jamming detection in a non-coherent massive single-input multiple-output (SIMO) communication and an OFDM-based industrial system. However, there are two main limitations of the statistical method. Firstly, the channel might not follow any specific statistical distribution, particularly in industrial settings where the distribution might undergo a sudden shift. Secondly, fitting the distribution normally is normally complicated, leading to a noticeable latency and significant modification to the existing protocols. In comparison, ML does not rely on any assumptions about channel distribution. Instead, it is capable of learning the general pattern of information and identifying the anomalies. Furthermore, since feeding inputs into a deep neural network is a process that resembles a series of optimized calculations, its online performance is low latency. The research proposed in [21] presented a ML-based (i.e., random forests, support vector machine) jamming detection approach for IEEE 802.11 networks. A detection method was developed in [22] using the time series analysis in which the network measurements were modeled as time series, and the sequential change-point detection algorithm was applied to detect the change of state. The authors in [23] also adopted a ML-based technique to detect pilot contamination attacks in 5G IoT networks. It is designed based on the multipath channel characteristics of mmWave frequencies, i.e., the reflection and refraction are sensitive to the transceiver locations. Recently, more sophisticated methods were introduced. For example, the research in [24] suggested deploying unmanned aerial vehicles (UAVs) to assist in the prevention and detection of attacks in 5G networks since UAVs can locate the jamming signal source faster than ground vehicles and without human intervention.

There are very limited studies ([16], [18]) focusing on the security aspect of beam training procedure in 5G communications. The research proposed in [16] only aimed at designing attack strategies to manipulate the beam determination without considering the effect of channel fading. In addition, the study did not consider jamming detection or mitigation. Wireless attacks are not the main focus of [18]. Instead, the study investigated a scenario in which the beam prediction model is poisoned by mobile malware or copied mobile. It is also important to note that the research proposed in [16], [18] considered a scenario where deep learning is utilized for beam determination. However, this approach has not yet been officially approved by any standards or implemented in industrial settings. Our research, on the other hand, focuses on the beam sweeping method, a well-established and widely-used technique in industrial standards such as IEEE 802.15.3c, IEEE 802.11ad, and WirelessHD. As the channel characteristics vary depending on the locations of the transceiver, the detection method proposed in [23] might be suitable for deterministic scenarios only. Besides the jamming detection capability, a defensive strategy should be able to eliminate or alleviate the impact of jamming attacks. The research proposed in [7], [19], [21]–[23] only focused on jamming detection, which despite its usefulness, is still incapable of thwarting jamming threats.

In this research, we aim to address the issue of overlooked security in beam training. We will show that a simple jamming attack can still severely impact the beam determination process, resulting in a deterioration of signal-to-interference-plus-noise-ratio (SINR) performance. To address this issue, we propose a defensive algorithm that is capable of detecting and mitigating jamming attacks.

B. Main Contributions

The main contributions of our work may now be summarized as follows. **Firstly**, we investigate the security aspect of the beam training procedure in mmWave communications for manufacturing where there is a so-called BS performing a downlink communication with a moving robotic vehicle, so called AGV. Our research takes into account the movement of the AGV, unlike prior studies that only considered deterministic locations. The significant change of the channel distribution during AGV movement poses a challenge in detecting jamming attacks. In practice, this can also be a consequence of the dynamic nature of industrial environments, where the surrounding environment is rapidly changing. To attack, the so-called Attacker confuses AGV during the beam training by transmitting jamming signals over various spatial directions. This attack method is distinct from the adversarial attacks proposed in [16], [18] targeting ML-based beam prediction only. Our **second** contribution is the development of a defensive strategy consisting of two stages, which are: (1) jamming detection and (2) jamming mitigation. For the first stage, we develop a jamming detection autoencoder (so-called JDAE), which can be trained effectively without any prior knowledge of Attacker. In the second stage, we build a jamming mitigation autoencoder (so-called JMAE) to counteract the effects of jamming and recover the RSS vector at the AGV. Both algorithms rely on the RSS information and can be easily integrated into existing protocols used for beam training in 5G, requiring minimal modifications. Our research not only focuses on jamming detection but also extends to the investigation of anti-jamming countermeasures, an aspect that has not been addressed in previous studies such as [19], [21]–[23]. **Finally,** numerical results provided to benchmark the proposed defensive strategy show that with a proper parameter setting, JDAE can obtain a satisfactory detection performance. Furthermore, when jamming attacks are detected, JMAE effectively reduces their destructive impact, resulting in an average improvement in SINR performance of over 15 dB as compared to the scenario in which the jamming mitigation technique is not employed.

Notation: Throughout this paper, we use lowercase and uppercase boldface letters to represent vectors and matrices, respectively. The transpose of $\mathbf{X}$ is denoted by $\mathbf{X}^{\mathrm{T}}$. Furthermore, $\mathbf{X} \sim \mathcal{CN}(\mathbf{M}, \mathbf{V})$ denotes that $\mathbf{X}$ is a complex Gaussian matrix with mean matrix $\mathbf{M}$ and covariance matrix $\mathbf{V}$. The operator $||.||$ is the Euclidean norm.

II. System Model

In this paper, we consider the wireless downlink communication system in a smart factory, which comprises the AGV, one legitimate BS, and one illegitimate Attacker. For the sake of tractability, we use subscripts A and B to represent the terms related to Attacker and BS, respectively. While AGV has only a single antenna, BS and Attacker are equipped with an array of N_{B} and N_{A} antenna elements and a codebook $\mathbf{W}_{\mathrm{B}}$ and $\mathbf{W}_{\mathrm{A}}$, respectively. The detailed design of $\mathbf{W}_{\mathrm{B}}$ and $\mathbf{W}_{\mathrm{A}}$ will be described in Section II-D. We consider a manufacturing environment where AGV follows a predetermined trajectory for tracking and transporting heavy materials. Industrial settings often exhibit unique characteristics affecting the radio wave environment and transmissions due to their physical features such as floor plans, layouts of metallic machines and work cells. Hence, to maintain a good wireless communication link between BS and AGV, the beam alignment needs to be obtained via a set of protocols, so-called *beam training*[1].

A. Beam Training Procedure

Assume that BS has a predefined codebook $\mathbf{W}_{\mathrm{B}} \triangleq [\mathbf{w}_{\mathrm{B}}^1, \mathbf{w}_{\mathrm{B}}^2, \cdots, \mathbf{w}_{\mathrm{B}}^{M_{\mathrm{B}}}] \in \mathbb{C}^{N_{\mathrm{B}} \times M_{\mathrm{B}}}$, where M_{B} is the codebook size and each beamforming vector $\mathbf{w}_{\mathrm{B}}^k$ specifies a transmit pattern. For convenience, we denote the ID of the beamforming vector $\mathbf{w}_{\mathrm{B}}^n$ as n.[2] In general, there are various operations categorized under the term beam training, which is composed of four different operations, as illustrated in Fig. 4(a). These operations are described as follows:

- *Beam sweeping:* In the time slot t, BS broadcasts a Synchronization Signal (SS) burst denoted as $\mathbf{s}_{\mathrm{B}}(t) \triangleq [s_{\mathrm{B}}^1(t), s_{\mathrm{B}}^2(t), \cdots, s_{\mathrm{B}}^{M_{\mathrm{B}}}(t)]$ using each beamforming vector in $\mathbf{W}_{\mathrm{B}}$. Herein, $s_{\mathrm{B}}^n(t)$ is the RS transmitted by BS using the beam n.
- *Beam measurement:* During this phase, the evaluation of the RSS is performed at AGV after receiving the SS burst. In other words, let $\mathbf{r}(t) \triangleq [r^1(t), r^2(t), \cdots, r^{M_{\mathrm{B}}}(t)]$ be the signal sequence received by AGV in the time slot t, its corresponding RSS vector, denoted as $\mathbf{g}(t) \triangleq [g^1(t), g^2(t), \cdots, g^{M_{\mathrm{B}}}(t)]$, is measured during this phase.
- *Beam determination:* Based on the beam measurement, AGV selects the beam which provides the maximum RSS. In other words, during this step, AGV determines a beam m satisfying $m = \arg\max_n \{g^n(t)\}$.
- *Beam reporting:* This is a procedure used by AGV to report beam quality (i.e., $\mathbf{g}(t)$) and beam decision information (i.e., beam m) to BS. Based on this, BS initiates the downlink data transmission with AGV in the time slot t using the beam m.

B. Attack Strategy

Generally speaking, when it comes to PHY-layer security, compared to lower frequencies, wireless communications are more secure at the mmWave frequencies when combined with high directional antennas. If the CSI to the target is not available, it is challenging for Attacker to launch an attack due to the high path-loss at the mmWave frequencies. To address this, it simply breaks through the legitimate communication system by sequentially transmitting jamming signals toward different spatial directions. In other words, Attacker also performs *beam sweeping* simultaneously with BS, by using all of the codewords available in its codebook, $\mathbf{W}_{\mathrm{A}}$. This attack can manipulate the RSS vector at AGV, hence a poor decision in the beam determination step. If Attacker employs a high jamming power for the attack, AGV can be fooled easily and might determine a beam ID that is more favorable for Attacker than BS.

C. Channel Model

We assume the Rician channel fading model for the links from BS or Attacker to AGV, consisting of a line-of-sight

[1] In some research [3], it is also known as beam management.

[2] Henceforth, we refer beam n as the n-th beamforming vector in the codebook, i.e., $\mathbf{w}_{\mathrm{B}}^n$ for BS and $\mathbf{w}_{\mathrm{A}}^n$ for Attacker.

(LoS) path and multiple non-line-of-sight (NLoS) paths. Since AGV might move, we adopt the time-varying geometric channel model when the Doppler effect is introduced [2], [25]. To be more specific, the channel between BS (or Attacker) and AGV in the time slot t are modeled as follows

$$\mathbf{h}_{\mathrm{i}}(t) = \sqrt{\beta_{\mathrm{i}}}\, e^{j2\pi f_{\mathrm{D}} t T_s} \times \widetilde{\mathbf{h}}_{\mathrm{i}}(t), \tag{1}$$

where $\mathrm{i} \in \{\mathrm{A}, \mathrm{B}\}$. Additionally, β_{i} represents the large-scale fading coefficient of the considered link while f_{D} is the Doppler shift and T_s stands for the transmit symbol interval. Finally, the term $\widetilde{\mathbf{h}}_{\mathrm{i}}(t)$, which consists of one LoS path and L_{i} non-LoS paths, can be expressed as follows [2]

$$\widetilde{\mathbf{h}}_{\mathrm{i}}(t) = \sqrt{\frac{\mathcal{K}_{\mathrm{i}}}{\mathcal{K}_{\mathrm{i}}+1}}\, \mathbf{h}_{\mathrm{i}}^{\mathrm{L}} + \sqrt{\frac{1}{\mathcal{K}_{\mathrm{i}}+1}} \sum_{j=1}^{L_{\mathrm{i}}} \alpha_{\mathrm{i}}^{j} \mathbf{a}_{\mathrm{i}}(\theta_{\mathrm{i}}^{j}), \tag{2}$$

where $\mathcal{K}_{\mathrm{i},k}$ stands for the Rician-$\mathcal{K}$ factor of the considered link. Moreover, $\mathbf{h}_{\mathrm{i}}^{\mathrm{L}}$ denotes the deterministic LoS component, α_{i}^{j} is the complex channel gain from BS (or Attacker) to AGV following the j-th scattered NLoS path, $\mathbf{a}_{\mathrm{i}}(\theta)$ is the transmitter steering vector associated with the angle of departure (AoD) θ at BS or Attacker. Finally, $\theta_{\mathrm{i}}^{j} \in [0, 2\pi]$ stands for the AoD of the j-th scattering signal. Herein, for notational convenience, we drop the time index since α_{i}^{j} and θ_{i}^{j} are random variables.

Considering only the azimuth and neglecting elevation imply that all scattering happens in azimuth, thus, BS and Attacker implement horizontal (2D) beamforming only. The steering vector at the BS or Attacker can be expressed as

$$\mathbf{a}_{\mathrm{i}}(\theta) \triangleq \frac{1}{\sqrt{N_{\mathrm{i}}}} \left[1,\ e^{j\frac{2\pi}{\lambda} d_{\mathrm{i}} \sin\theta},\ \cdots,\ e^{j(N_{\mathrm{i}}-1)\frac{2\pi}{\lambda} d_{\mathrm{i}} \sin\theta}\right]^{\mathrm{T}}, \tag{3}$$

with $\forall \theta \in [0, 2\pi]$ and $\mathrm{i} \in \{\mathrm{A}, \mathrm{B}\}$.

D. Codebook Based mmWave Precoding Design

For a phased antenna array, an RF codebook can be represented by a matrix, where each column specifies a transmit pattern. Particularly, let $\mathbf{W}$ be an $N \times M$ predesigned codebook matrix, where M is the codebook size and N is the number of antenna array elements, there are several common RF codebooks, such as the codebook proposed in IEEE 802.15.3c and wireless personal area networks (WPAN) [26] or the discrete Fourier transform (DFT) codebook [27].

The codebook utilized in IEEE 802.15.3c and WPAN is to simplify hardware implementation. The codebook is relatively simple since it is generated with a 90-degree phase resolution without amplitude adjustment to reduce power consumption. In particular, the elements of the codebook are given as [26]

$$\mathbf{W}(n, m) = \frac{1}{\sqrt{N}} j^{\left\lfloor \frac{4n \times \mathrm{mod}(m + \frac{M}{4}, M)}{M} \right\rfloor}, \tag{4}$$

with $\forall n \in \mathcal{N}, \forall m \in \mathcal{M}$ where $\mathcal{N} = \{0, 1, \cdots, N-1\}$ denotes the set of antennas and $\mathcal{M} = \{0, 1, \cdots, M-1\}$ is the set of beam patterns in the codebook. Additionally, $\lfloor \cdot \rfloor$ represents the floor function.

In practice, DFT codebooks are widely used as they can match approximately the optimal beamforming. In addition, they also can achieve higher antenna gains at the beam directions than the codebooks used in IEEE 802.15.3c. The entries of a DFT codebook are defined as [27]

$$\mathbf{W}(n, m) = \frac{1}{\sqrt{N}} e^{-j2\pi \frac{nm}{N}}, \tag{5}$$

with $\forall n \in \mathcal{N}, \forall m \in \mathcal{M}$.

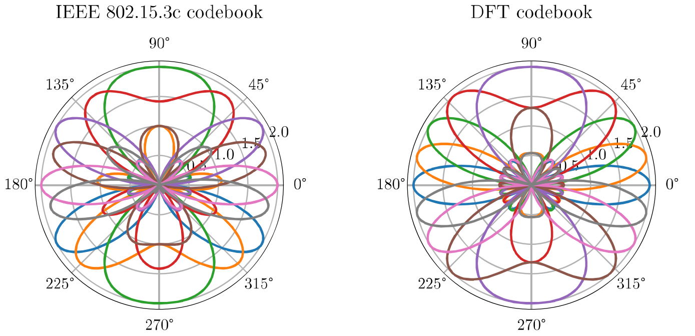


Fig. 1. Polar plots for the array factor of the IEEE 802.15.3c and DFT codebooks with 3-bit resolution. In this simulation, $M = 8$ and $N = 4$.

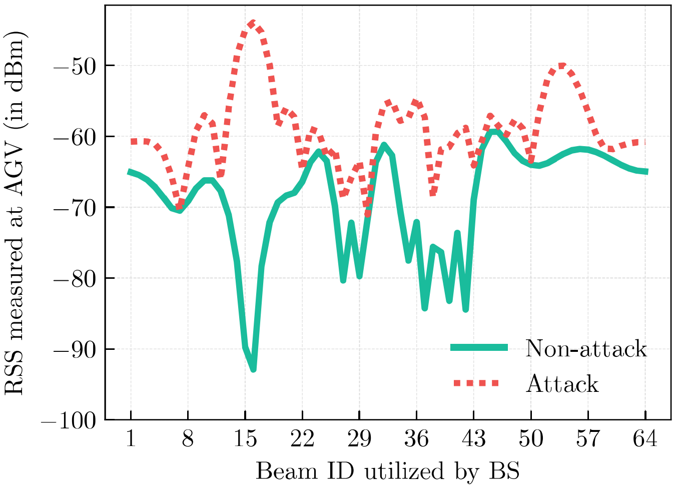


Fig. 2. Beam ID utilized by BS versus corresponding RSS measured at AGV for two scenarios: under attack and non-attack. In this simulation, $N_{\mathrm{A}} = N_{\mathrm{B}} = 16$, $M_{\mathrm{A}} = M_{\mathrm{B}} = 64$, $P_{\mathrm{A}} = 30$ dBm, and $P_{\mathrm{B}} = 20$ dBm.

Fig. 1 demonstrates the polar plots of array factor for two codebooks using (4) and (5) with a resolution of 3 bits, which points out that compared to the IEEE 802.15.3c codebook, the DFT codebook can achieve higher antenna gains at the beam directions with reduced side lobes and a better beam resolution.

E. Received Signal Model at AGV

To model the received SS burst at AGV, we will take into account the path-loss, scattering, and codebook design described in Sections II-C and II-D. In general, the presence of Attacker leads to the following hypotheses:

- $\mathcal{H}_s$: Attacker stays silent.
- $\mathcal{H}_a$: Attacker breaks into the system.

For simplicity, we consider a scenario where Attacker and BS have an identical codebook size M, the extension is possible. In the time slot t, the SS burst received at AGV can be described as $\mathbf{r}(t) = [r^1(t), r^2(t), \cdots, r^M(t)]^{\mathrm{T}} \in \mathbb{C}^{M\times 1}$. Herein, $r^n(t)$ is the RS received at AGV in the case when BS utilizes the beam n. To be more specific, $r^n(t)$ can be modeled as

$$r^n(t) = \begin{cases} \sqrt{P_{\mathrm{B}}}\mathbf{h}_{\mathrm{B}}^{\mathrm{T}}\mathbf{w}_{\mathrm{B}}^n s_{\mathrm{B}}^n + \mathrm{n}(t), & \text{under } \mathcal{H}_s; \\ \sqrt{P_{\mathrm{B}}}\mathbf{h}_{\mathrm{B}}^{\mathrm{T}}\mathbf{w}_{\mathrm{B}}^n s_{\mathrm{B}}^n + \sqrt{P_{\mathrm{A}}}\mathbf{h}_{\mathrm{A}}^{\mathrm{T}}\mathbf{w}_{\mathrm{A}}^n s_{\mathrm{A}}^n + \mathrm{n}(t), & \text{under } \mathcal{H}_a, \end{cases} \tag{6}$$

where s_{A}^n and s_{B}^n are the n-th data symbol in the SS burst transmitted by Attacker and BS, respectively whereas $n(t) \sim \mathcal{CN}(0, N_0)$ with N_0 represents the noise power. In addition, P_{A} and P_{B} denote the transmit power of Attacker and BS, respectively.

During the measurement step, the RSS vector measured by AGV can be achieved as $\mathbf{g}(t) = [g^0(t), g^1(t), \cdots, g^M(t)]^{\mathrm{T}} \in \mathbb{R}^{M\times 1}$ where $g^n(t)$ is the RSS measured for the received signal $r^n(t)$, calculated as

$$g^n(t) = \begin{cases} P_{\mathrm{B}}\left|\mathbf{h}_{\mathrm{B}}^{\mathrm{T}}\mathbf{w}_{\mathrm{B}}^n\right|^2 + N_0, & \text{under } \mathcal{H}_s; \\ P_{\mathrm{B}}\left|\mathbf{h}_{\mathrm{B}}^{\mathrm{T}}\mathbf{w}_{\mathrm{B}}^n\right|^2 + P_{\mathrm{A}}\left|\mathbf{h}_{\mathrm{A}}^{\mathrm{T}}\mathbf{w}_{\mathrm{A}}^n\right|^2 + N_0, & \text{under } \mathcal{H}_a. \end{cases} \tag{7}$$

Let $m = \arg\max_n\{g^n(t)\}$, which means that AGV selects the beam m for the downlink data transmission. As a result, during the primary data transmission, the SINR received at AGV in the time slot t can be written as

$$\mathrm{SINR}(t) = \frac{P_{\mathrm{B}}\left|\mathbf{h}_{\mathrm{B}}^{\mathrm{T}}\mathbf{w}_{\mathrm{B}}^m\right|^2}{P_{\mathrm{A}}\left|\mathbf{h}_{\mathrm{A}}^{\mathrm{T}}\mathbf{w}_{\mathrm{A}}^m\right|^2 + N_0}. \tag{8}$$

Fig. 2 shows an example of the RSS vector measured at AGV during the beam training step in two scenarios: under attack and non-attack. Based on the RSS vector, AGV decides which beam is the best for use in the downlink data transmission. In particular, when there is no attack, beam 44 will be selected as it provides the highest RSS value of nearly -60 dBm. However, when Attacker performs jamming attacks, the RSS vector is manipulated with the highest value of -50 dBm corresponding to the beam 16. As a result, AGV, fooled by the jamming signals transmitted by Attacker, will make a poor decision by selecting the beam 16. Unfortunately, as we can see, if BS utilizes this beam, the desired signal strength will be noticeably low, which is about 34 dB lower than that in the case of no attack. This points out that such attacks, despite relative simplicity, might cause a deterioration of the mmWave communications.

III. Jamming Attack Detection Strategy

This section describes how to train a ML model, the so-called Jamming Detection Autoencoder (JDAE), to learn the underlying patterns of normal data samples when AGV moves along its trajectory, thereby distinguishing them from the abnormal samples. In addition, the detailed mechanism for data normalization, parameter setting, and online jamming detection will also be discussed.

A. Data Generation

For each time slot t, AGV obtains an RSS vector $\mathbf{g}(t)$ which can be utilized not only for beam determination but also for jamming attack detection. For tractability, we use the notations $\mathbf{g}_{\mathcal{H}_s}(t)$ and $\mathbf{g}_{\mathcal{H}_a}(t)$ to indicate the RSS vector achieved at AGV in the event $\mathcal{H}_s$ and $\mathcal{H}_a$ during the time slot t, respectively. This information will enable the legitimate system to build datasets, and then learn the underlying patterns of the RSS vector in legitimate scenarios.

TABLE I
AUTOENCODER ARCHITECTURE OF JDAE & JMAE

Network	Layer type	Number of neurons
Encoder	Input	M
	Fully connected + Elu	$\lfloor M/2^1 \rfloor$
	Fully connected + Elu	$\lfloor M/2^2 \rfloor$
	...	...
	Fully connected + Elu	4
Decoder	Input	4
	...	...
	Fully connected + Elu	$\lfloor M/2^2 \rfloor$
	Fully connected + Elu	$\lfloor M/2^1 \rfloor$
	Fully connected + Linear	M

AGV will perform a moving average in each time slot to reduce the channel variation effect. For instance, the average RSS vector computed by AGV at the time slot t, defined as the average of W RSS vectors measured during W prior consecutive time slots, can be expressed as

$$\mathbf{x}_i(t) \triangleq \frac{1}{\mathrm{W}}\sum_{t+1-\mathrm{W}}^{t}\mathbf{g}_i(t), \tag{9}$$

where $\mathbf{x}_i(t) \in \mathbb{R}^{M\times 1}$, W represents the time window length, and $i \in \{\mathcal{H}_s, \mathcal{H}_a\}$. It is important to note that as we do not have any prior knowledge of Attacker, the training dataset will only be associated with the $\mathcal{H}_s$ events solely. To put it simply, it is possible to construct the training data under the hypothesis $\mathcal{H}_s$, defined as

$$\mathbf{X}_{\mathrm{JD}}^{\mathrm{train}} = \left\{\mathbf{x}_{\mathcal{H}_s}[0], \mathbf{x}_{\mathcal{H}_s}[1], \cdots, \mathbf{x}_{\mathcal{H}_s}[N_{\mathrm{train}}]\right\}, \tag{10}$$

where N_{train} stands for the number of training data samples.

Regarding the test dataset, it is necessary to consider both the case of Attacker being absent and present. As a result, the test dataset can be represented as follows

$$\mathbf{X}_{\mathrm{JD}}^{\mathrm{test}} = \Big\{\underbrace{\mathbf{x}_{\mathcal{H}_s}[N_{\mathrm{train}}+1], \cdots, \mathbf{x}_{\mathcal{H}_s}[N_{\mathrm{train}} + N_{\mathcal{H}_s,\mathrm{test}}]}_{\text{normal data samples}}, \\ \underbrace{\mathbf{x}_{\mathcal{H}_a}[0], \mathbf{x}_{\mathcal{H}_a}[1], \cdots, \mathbf{x}_{\mathcal{H}_a}[N_{\mathcal{H}_a,\mathrm{test}}]}_{\text{abnormal data samples}}\Big\}, \tag{11}$$

where $N_{\mathcal{H}_s,\mathrm{test}}$ and $N_{\mathcal{H}_a,\mathrm{test}}$ indicate the number of data samples associated to the event $\mathcal{H}_s$ and $\mathcal{H}_a$ in the test dataset, respectively. In our experiments, we opt for $N_{\mathcal{H}_s,\mathrm{test}} = N_{\mathcal{H}_a,\mathrm{test}} = N_{\mathrm{test}}$ to ensure the fairness and balance between the normal and abnormal events in the testing phase. Note that the dataset $\mathbf{X}_{\mathrm{JD}}^{\mathrm{test}}$ will not be used for training ML models since Attacker is anonymous.

B. The Architecture of JDAE

Since the identity of Attacker is normally unknown to the legitimate system, only the data samples in $\mathcal{H}_s$ are available for training ML models. Therefore, an unsupervised learning technique is suitable for this scenario. Autoencoders (AEs) are ML architectures in which neural networks are leveraged for the task of representation learning. A typical AE architecture comprises two networks, namely encoder, and decoder. In particular, while the encoder network translates the original high-dimension input into a latent low-dimensional layer located in the middle of the AE, the decoder network computes the data from the latent layer to generate a reconstructed version of the original input data. To put it simply, assume that the AE represents a mapping function $f(.)$, if the input is $\mathbf{x}$, it generates an output $\mathbf{x}' = f(\mathbf{x})$ so that $\mathbf{x}'$ is as identical to $\mathbf{x}$ as possible.[3] During training, as a benefit of the compression process, the AE is capable of filtering out atypical features of the data and retaining only the significant characteristics. In the context of anomaly detection, the AE can learn the pattern of the normal data samples, thereby identifying the anomalous data points.

Regarding the architecture of JDAE, we utilize a typical AE with several fully connected layers, each followed by a layer including Elu activations. JDAE also consists of an encoder and a decoder network whose architectures are symmetric across the latent layer, as illustrated in Table I. The encoder receives the average RSS vector $\mathbf{x}(t)$ as the input and then compresses it at the latent layer. Subsequently, the decoder attempts to map the signal represented at the latent layer back to the original input signal. Note that, unlike other layers in JDAE, the output layer of the decoder has linear activations since the output represents the reconstructed version of the input data.

C. Training, Testing Strategy and Online Jamming Detection

1) Data Normalization: Data normalization plays an essential role in training ML models. In this work, a data sample $\mathbf{x}$ is normalized as follows

$$\mathbf{x}_{\text{normalize}} \triangleq \frac{\mathbf{x} - \mu_{\text{JD}}}{\sigma_{\text{JD}}}, \tag{12}$$

where μ_{JD} and σ_{JD} represent the mean and standard deviation of the RSS values in $\mathbf{X}_{\text{JD}}^{\text{train}}$, respectively. It is also worth noting that μ_{JD} and σ_{JD} are determined by computing the training dataset, prior to the training phase.

2) Training Strategy: To capture the underlying pattern of the normal data samples, JDAE needs to be trained on normal data samples only, i.e., $\mathbf{X}_{\text{JD}}^{\text{train}}$. During the training process, JDAE attempts to reconstruct each data sample in the training dataset by minimizing the following loss function

$$\mathcal{L}(\theta) = \frac{1}{B}\sum_{i=0}^{B} \left\|\mathbf{x}_{\text{normalize}}[i] - \mathbf{x}'_{\text{normalize}}[i]\right\|^2, \tag{13}$$

where θ denotes the "learnable" parameters of the AE (including weights and biases), B is the batch size, i.e., the number of data samples feeding the AE model in each iteration and $\mathbf{x}_{\text{normalize}}[i]$ is the normalized value of the i-th data sample $\mathbf{x}[i]$ in $\mathcal{H}_s$. Note that $\mathcal{L}(\theta)$ is also often termed as the reconstruction loss. The model parameter θ can be updated for each batch of B data samples using stochastic gradient descent (SGD) algorithm as follows

$$\theta := \theta - \eta\ \nabla\mathcal{L}(\theta), \tag{14}$$

where η is the learning rate and $\nabla\mathcal{L}(\theta)$ is the gradient of $\mathcal{L}(\theta)$ with respect to θ. In this work, we adopt an advanced SGD method, so-called the adaptive moment estimation (Adam) for updating θ.

Algorithm 1: Jamming detection algorithm

```
Data: $\mu_{\text{JD}}$ and $\sigma_{\text{JD}}$ determined from the offline
      training phase, a pre-defined window length W.
Input: The RSS vector $\mathbf{g}(t)$ measured at time slot $t$.
Result: outcome, assigned True if there is an attack;
        otherwise False.
Construct the average RSS vector $\mathbf{x}(t)$ using (9);
Compute the normalized average RSS vector
  $\mathbf{x}_{\text{normalized}}(t)$ using (12);
Input $\mathbf{x}_{\text{normalized}}(t)$ into JDAE and compute the
  reconstructed signal $\mathbf{x}'_{\text{normalized}}(t)$;
Compute the reconstruction loss as follows
  loss $= \frac{1}{M}\left\|\mathbf{x}_{\text{normalized}}(t) - \mathbf{x}'_{\text{normalized}}(t)\right\|^2$;
if loss $> L_{\text{th}}$ then
    outcome = True
else
    outcome = False
end
return outcome
```

3) Online Inference and Attack Detection: After training, jamming attacks can be detected by evaluating how well the JDAE can reconstruct the normalized input samples. In this context, since JDAE was trained on $\mathbf{X}_{\text{JD}}^{\text{train}}$ containing the normal RSS vectors only, it captures the characteristics of normal data samples, thereby identifying the jamming attacks if a significant difference is detected. The reconstruction loss can be employed to evaluate this. Particularly, once the reconstruction loss calculated for input is greater than a pre-defined threshold L_{th}, we can infer that an unfamiliar pattern occurs, which can be labeled as a jamming attack. Hence, setting L_{th} has a great impact on detection performance. Since Attacker is anonymous, we introduce a method for setting L_{th} only based on the training data $\mathbf{X}_{\text{JD}}^{\text{train}}$. In particular, after training the AE, it is possible to construct a reconstruction loss vector $\mathbf{r} \in \mathbb{R}^{N_{\text{train}} \times 1}$ whose each element is the reconstruction loss value for each data point in $\mathbf{X}_{\mathcal{H}_s}^{\text{train}}$. Let μ_r and σ_r be the mean and standard deviation of $\mathbf{r}$, respectively, and the detection threshold parameter can be set as

$$L_{\text{th}} = \mu_r + \alpha\sigma_r, \tag{15}$$

where α is a parameter used for adjusting the sensitivity of the detection algorithm.

After the training and offline fine-tuning, JDAE will be employed online at AGV for jamming attack detection. During each time slot t, the average RSS vector $\mathbf{x}(t)$ is computed as in (9), then normalized as in (12), before being fed into JDAE.

[3] We henceforth denote $\mathbf{x}'$ as the output generated by the AE if the input data is $\mathbf{x}$.

TABLE II
THE CONFUSION MATRIX

		Predicted **labels**	
		($\mathcal{H}_\mathrm{s}$)	($\mathcal{H}_\mathrm{a}$)
Actual **labels**	($\mathcal{H}_\mathrm{s}$)	TP N_TP samples	FP N_FP samples
	($\mathcal{H}_\mathrm{a}$)	FN N_FN samples	TN N_TN samples

Subsequently, the reconstruction loss is calculated, followed by a comparison with L_th to determine whether there is a jamming attack. The algorithm utilized for jamming attack detection is now summarized in Algorithm 1.

Remark 1. *The parameter α determines the sensitivity of the detection algorithm toward jamming attacks. On the one hand, when $\alpha \to -\infty$, we have $L_\mathrm{th} \to -\infty$, which means that all events will be identified as jamming attacks. On the other hand, in the case when $\alpha \to +\infty$, we have $L_\mathrm{th} \to +\infty$, which means that all events will be labeled as non-attack. In general, setting a higher value of α reduces the sensitivity toward jamming attacks. Hence, introducing α improves flexibility for the detection algorithm to be suitable for various applications.*

4) Testing Strategy: The testing dataset $\mathbf{X}_\mathrm{JD}^\mathrm{test}$ now can be utilized for evaluating the performance of the trained AE. With two types of data samples available in $\mathbf{X}^\mathrm{test}$, there are four possible detection outcomes listed as follows:

- True positive (TP): $\mathcal{H}_s$ samples are correctly detected.
- True negative (TN): $\mathcal{H}_a$ samples are correctly detected.
- False positive (FP): $\mathcal{H}_s$ samples are incorrectly detected as $\mathcal{H}_a$.
- False negative (FN): $\mathcal{H}_a$ samples are incorrectly detected as $\mathcal{H}_s$.

These events are the elements of the so-called confusion matrix, illustrated in Table II. Herein, $N_\mathrm{TP}, N_\mathrm{TN}, N_\mathrm{FP}$ and N_FN represent the number of data samples identified with the outcome TP, TN, FL, and FN, respectively. Given these metrics and recall that the number of attack and non-attack samples are equal in the testing data, the *accuracy* (ACC), which is the probability of an event classified correctly, can be used to evaluate the performance of the detection algorithm. In particular, ACC can be written as

$$\mathtt{ACC} = \frac{N_\mathrm{TP} + N_\mathrm{TN}}{N_\mathrm{TP} + N_\mathrm{TN} + N_\mathrm{FP} + N_\mathrm{FN}}. \tag{16}$$

In addition, it is also important to investigate the receiver operating characteristic (ROC) curve which characterizes the metrics *true positive rate* (TPR) and *false positive rate* (FPR) for various detection thresholds. In particular, TPR and FPR can be defined as

$$\mathtt{TPR} = \frac{N_\mathrm{TP}}{N_\mathrm{TP} + N_\mathrm{FN}}, \quad \mathtt{FPR} = \frac{N_\mathrm{FP}}{N_\mathrm{FP} + N_\mathrm{TN}}. \tag{17}$$

Algorithm 2: Jamming mitigation algorithm

```
Data: $\mu_\mathrm{JM}$ and $\sigma_\mathrm{JM}$ executed from training.
Input: The corrupted RSS vector $\mathbf{g}(t)$ measured at
       time slot $t$.
Result: The reconstructed RSS vector $\hat{\mathbf{g}}(t)$
Compute the normalized average RSS vector
  $\mathbf{g}_\mathrm{normalized}(t)$ using (24);
Input $\mathbf{g}_\mathrm{normalized}(t)$ into JMAE and compute the
  reconstructed signal $\mathbf{g}'_\mathrm{normalized}(t)$;
Compute the denormalization of the output via
  $\hat{\mathbf{g}}(t) = \mathbf{g}'_\mathrm{normalized}(t) \times \sigma_\mathrm{JM} + \mu_\mathrm{JM}$;
return $\hat{\mathbf{g}}(t)$
```

Algorithm 3: Defensive strategy by JDAE and JMAE

```
Perform jamming detection via Algorithm 1;
if outcome = True then
    Perform jamming mitigation via Algorithm 2
else
    Skip
end
return RSS vector
```

IV. JAMMING MITIGATION STRATEGY

Once an attack is detected successfully, one of the most natural questions is how to cope with it. In the context of PHY-layer security, a simple method is to utilize high transmit power on the jammed channels to compete actively with the attackers [28]. Another popular anti-jamming countermeasure is the channel hopping technique which allows the legitimate system to switch to another channel selected either randomly or according to a pre-defined method [29]. This section presents an alternative strategy, known as jamming mitigation, to reduce the impact of jamming signals, and recover the RSS vector so that AGV can make a better decision in the beam determination step. This strategy employs an ML model, developed based on denoising autoencoders (DAEs) and JMAE.

A. Jamming Attack Mitigation Based on DAEs

DAEs enjoy a plethora of applications in image processing and computer vision. Unlike the conventional AEs, which copy the input to the output, DAEs use corrupted data as the input and uncorrupted data as the output. By doing this, not only can the DAEs learn to compress data like the AEs, but it also can remove noise from the corrupted input. Inspired by this fundamental, JMAE can also be trained to eliminate the jamming signals from the corrupted RSS vector. JMAE has an identical architecture to JDAE, as illustrated in Table I. Note that, however, the data used for training JMAE is different from that for training JDAE.

B. Training Strategy

During the training, the RSS vectors corrupted by jamming attacks will be used as the input, while the corresponding uncorrupted RSS vector (also known as desired RSS vector) will

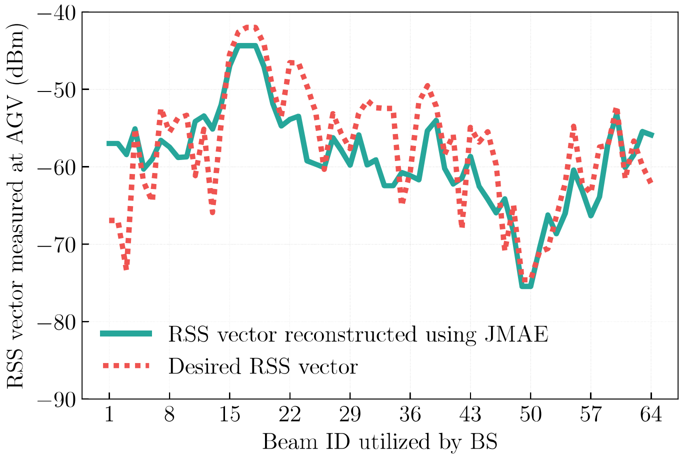


Fig. 3. The RSS vector recovered after jamming mitigation using JMAE vs the desired RSS vector. In this simulation, $N_{\text{A}} = N_{\text{B}} = 16$, $M_{\text{A}} = M_{\text{B}} = 64$ and $P_{\text{A}} = 30$ dBm, $P_{\text{B}} = 20$ dBm.

be used as the output. However, due to the unknown identity of Attacker, the training data, including the corrupted RSS and the corresponding desired RSS vectors, is not available yet. To circumvent this issue, we build *synthetic* data for training JMAE. The synthetic data is the combination of two datasets, namely *desired* and *jamming* datasets, defined as follows:

$$\mathbf{X}_{\text{desired}} = \left\{\mathbf{g}_{\mathcal{H}_s}[0], \mathbf{g}_{\mathcal{H}_s}[1], \cdots, \mathbf{g}_{\mathcal{H}_s}[N_{\text{desired}}]\right\}, \tag{18}$$

$$\mathbf{X}_{\text{jam}} = \left\{\mathbf{jam}[0], \mathbf{jam}[1], \cdots, \mathbf{jam}[N_{\text{jam}}]\right\}, \tag{19}$$

where N_{desired} and N_{jam} denote the number of data samples in the desired and jamming datasets, respectively. Moreover, $\mathbf{jam}[i]$ stands for the i-th jamming signal recorded at AGV, not including the desired signal. While $\mathbf{X}_{\text{desired}}$ is readily available at AGV, $\mathbf{X}_{\text{jam}}$ needs to be obtained via a *channel sensing* technique.

To be more specific, the training data is built via two steps:

- *Channel sensing:* This phase is performed once a jamming attack is detected, in which BS stops transmitting the reference signals temporarily while AGV attempts to collect the jamming signals transmitted by Attacker. By doing this, $\mathbf{X}_{\text{jam}}$ can be achieved. Note that the size of the jamming dataset depends on the channel sensing period.
- *Data synthesis:* During this step, a synthetic training dataset can be built by combining $\mathbf{X}_{\text{jam}}$ and $\mathbf{X}_{\text{desired}}$. The synthetic training data can be formulated as

$$\mathbf{X}_{\text{JM}}^{\text{train}} = \left\{\hat{\mathbf{rss}}[0], \hat{\mathbf{rss}}[1], \cdots, \hat{\mathbf{rss}}[N_{\text{JM}}^{\text{train}}]\right\}, \tag{20}$$

$$\mathbf{Y}_{\text{JM}}^{\text{train}} = \left\{\mathbf{rss}[0], \mathbf{rss}[1], \cdots, \mathbf{rss}[N_{\text{JM}}^{\text{train}}]\right\}, \tag{21}$$

where $\mathbf{X}_{\text{JM}}^{\text{train}}$ and $\mathbf{Y}_{\text{JM}}^{\text{train}}$ are used for the input and output of JMAE during training, respectively. In addition, $N_{\text{JM}}^{\text{train}}$ is the number of samples in the synthetic training data, while $\hat{\mathbf{rss}}[i]$ and $\mathbf{rss}[i]$ represent the i-th corrupted RSS vector and its corresponding uncorrupted signal, respectively. These are constructed as follows

$$\hat{\mathbf{rss}}[i] = \mathtt{ds} + \mathtt{jm}, \tag{22}$$

$$\mathbf{rss}[i] = \mathtt{ds}, \tag{23}$$

where $\mathtt{ds}$ and $\mathtt{jm}$ are a data sample drawn randomly from $\mathbf{X}_{\text{desired}}$ and $\mathbf{X}_{\text{jam}}$, respectively.

The channel sensing procedure enables the AGV to gather information about the Attacker, while the data synthesis process is carried out to construct a dataset where the desired signal can be suppressed by different jamming signals. It is important to note that the JMAE only needs to be trained once prior to deployment. Regarding the data normalization, we normalize a RSS vector as follows

$$\mathbf{g}_{\text{normalized}} \triangleq \frac{\mathbf{g} - \mu_{\text{JM}}}{\sigma_{\text{JM}}}, \tag{24}$$

where μ_{JM} and σ_{JM} represent the mean and standard deviation of the elements in $\mathbf{X}_{\text{JM}}^{\text{train}}$, respectively. With the synthetic dataset, we now can train JMAE using a similar training technique to JDAE.

C. Online Jamming Mitigation and Defensive Strategy

After training, JMAE can be used to retrieve the original RSS vector once the legitimate system is under attack. The algorithm employing JMAE for jamming mitigation is summarized in Algorithm 2. Fig. 3 demonstrates an example of the RSS vector obtained by utilizing JMAE, contrasted with the desired RSS vector. Herein, $N_{\text{JM}}^{\text{train}} = 10^4$, $N_{\text{desired}} = 10^4$ and $N_{\text{jam}} = 10^3$. As we can see, despite a reduced variation, the reconstructed RSS vector shows a similar pattern to the desired RSS signal. In other words, after performing jamming mitigation via JMAE, the most significant features of the desired RSS vector are successfully recovered. The most striking point is that the highest RSS value of both vectors corresponds to an identical beam, i.e., beam 16. Thus, JMAE effectively reduces the impact of jamming attacks, allowing a better decision about the beam used for the downlink data transmission.

By incorporating both JDAE and JMAE, we achieve a comprehensive countermeasure against jamming threats. In particular, Algorithm 1 is utilized for jamming attack detection first. If an attack is detected, Algorithm 2, in turn, will be responsible for mitigating the jamming effect and retrieving the desired RSS vector. Based on this, AGV can then perform the beam determination. Subsequently, utilizing the best beam obtained from the beam determination step, BS and AGV can establish a downlink data transmission. The defensive strategy, as summarized in Algorithm 3, can now be included as an additional step in the beam training protocol, as illustrated in Fig. 4(b).

Remark 2. *In 5G NR, the periodicity of SS burst transmission is configurable, which can be up to 160 ms and set by default to 20 ms [3]. Therefore, the proposed defensive strategy should be sufficiently fast to compute the jamming detection and jamming mitigation within one periodicity of SS burst transmission. Thanks to the AE architecture, both JDAE and JMAE are remarkably fast regarding online inference (just several ms per inference). Thus, the defensive strategy potentially meets the aforementioned computational requirement of 5G NR. This will be investigated in Section V.*

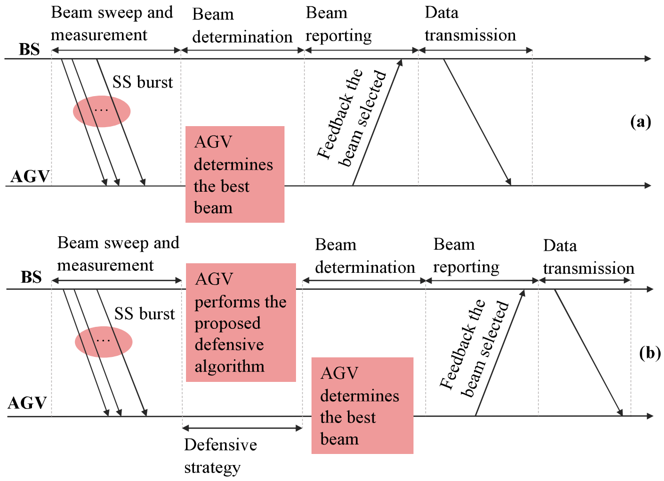

Beam sweep and measurement
Beam determination
Beam reporting
Data transmission
BS
SS burst
...
AGV determines the best beam
Feedback the beam selected
(a)
AGV
Beam sweep and measurement
Beam determination
Beam reporting
Data transmission
BS
AGV performs the proposed defensive algorithm
SS burst
...
AGV determines the best beam
Feedback the beam selected
(b)
AGV
Defensive strategy

Fig. 4. The illustration of the beam training protocol for: (a) 5G mmWave communications; and (b) 5G mmWave communications with the proposed defensive strategy.

Remark 3. *Since the proposed defensive strategy relies mainly on the RSS information, it requires minimal modification from the existing beam training protocol in 5G NR [3]. In particular, the current protocol only needs an additional block without modifying the other steps, as illustrated in Fig. 4.*

V. Numerical Results

A. Simulation Setup

1) Large-scale Fading Model and Rician Factor: The large-scale fading coefficient $\beta_{\rm i}$ contains the path-loss and shadow fading, according to

$$\beta_{\rm i} = \mathrm{PL}_{\rm i} \cdot 10^{\frac{\sigma_{\rm sh} z_{\rm i}}{10}}, \tag{25}$$

where $\mathrm{PL}_{\rm i}$ represents the path-loss while $10^{\frac{\sigma_{\rm sh} z_{\rm i}}{10}}$ stands for the shadow fading with the standard deviation $\sigma_{\rm sh}$, and $z_{\rm i} \sim \mathcal{N}(0,1)$.

Regarding the path-loss component, we employ an industrial indoor model proposed in 3GPP [30] to simulate the path-loss in the indoor factory environment. In particular, the path-loss (measured in dB) is calculated as follows

$$\mathrm{PL}_{\rm i} = \begin{cases} 31.84 + 21.5\log(D_{\rm i}) + 19.0\log(f_c), & \text{if LOS;} \\ 33.0 + 25.5\log(D_{\rm i}) + 20.0\log(f_c), & \text{if NLOS,} \end{cases} \tag{26}$$

where $D_{\rm i}$ denotes the length of the considered link (measured in meters) while f_c is the center frequency (measured in GHz). In addition, the shadowing standard deviation is equal to 4.0 dB and 5.7 dB in the case of LOS and NLOS channels, respectively, according to [30].

It is worth noting that the Rician $\mathcal{K}$-factor and large-scale fading coefficients vary depending on the location of transceivers. To reflect the realistic environment in the industrial factory where a direct communication link might be blocked, we use the following formulation as in [31]:

$$\mathcal{K}_{\rm i} = \frac{P_{\rm LOS}(D_{\rm i})}{1 - P_{\rm LOS}(D_{\rm i})}, \tag{27}$$

TABLE III
SYSTEM PARAMETERS USED FOR THE SIMULATIONS

Parameters	**Value**
Operating frequency (f_c)	28.0 GHz
Bandwidth (B)	10.0 MHz
Transmit power of BS ($P_{\rm B}$)	+20 dBm
Transmit power of Attacker ($P_{\rm A}$)	+30 dBm
Number of NLOS paths ($L_{\rm i}$)	10
The typical clutter size ($d_{\rm clutter}$) [30]	10
The clutter density ($r_{\rm dens}$)	0.4
Velocity of AGV (v)	1.0 m/s
Transmit symbol interval (T_s)	$1/3 \times 10^{-6}$ seconds

where $P_{\rm LOS}$ is defined as in [30]:

$$P_{\rm LOS}(D_{\rm i}) = \exp\left(-\frac{D_{\rm i}}{k_{\rm subsce}}\right), \tag{28}$$

where $D_{\rm i}$ is the link distance measured in meters, and $k_{\rm subsce} = -d_{\rm clutter}/\ln(1 - r_{\rm dens})$ with $d_{\rm clutter}$ being the typical clutter size and $r_{\rm dens}$ standing for the clutter density. In addition, the noise power is given by

$$N_0 = \text{bandwidth} \times k_B \times T_0 \times \text{noise figure}, \tag{29}$$

where $k_B = 1.381 \times 10^{-23}$ (Joule per Kelvin) is the Boltzmann constant and $T_0 = 290$ (Kelvin) is the noise temperature. Other parameters can be found in Table III, unless otherwise specified.

2) System Parameters: For all simulations, we consider a deployment area which is a square of 50×50 m^2, where BS locates at the center of $(0,0)$. In addition, the location of Attacker is uniformly distributed at random within the area due to its unknown identity. BS and Attacker utilize an IEEE 802.15.3c and a DFT codebook for the transmission with 3-bit resolution, respectively. Meanwhile, AGV moves along a horizontal line connecting $(-20,10)$ and $(+20,10)$ with a velocity of 1.0 m/s.

3) Machine Learning Parameters: In this paper, all AEs were trained using a computer with system specifications including a 2.9 GHz dual-core Intel Core i5 and 8 Gb of RAM. Regarding JDAE, it was trained using training datasets with $N_{\rm train} = 10^3$ and tested on testing datasets with $N_{\mathcal{H}_s,\rm test} + N_{\mathcal{H}_a,\rm test} = 2 \times N_{\rm test} = 2 \times 10^4$. Meanwhile, JMAE was trained using synthetic datasets with $N_{\rm JM}^{\rm train} = 10^4$, which is constructed from desired and jamming datasets: $N_{\rm desired} = 10^4$, $N_{\rm jam} = 10^3$, unless otherwise specified.

For both AEs, during training, the batch size is 40 while the learning rate is set to 0.005. The early stopping technique was also utilized to avoid overfitting.

B. Numerical Results for Jamming Detection

1) The impact of codebook size: Fig. 5(a) shows the accuracy achieved with various values of α while Fig. 5(b) illustrates the ROC curve achieved under JDAE in three cases: $M = 8, 16$ and 32. As we can see, when α is very low, the accuracy is also low since JDAE will be more sensitive to jamming attacks. In this case, the majority of events might

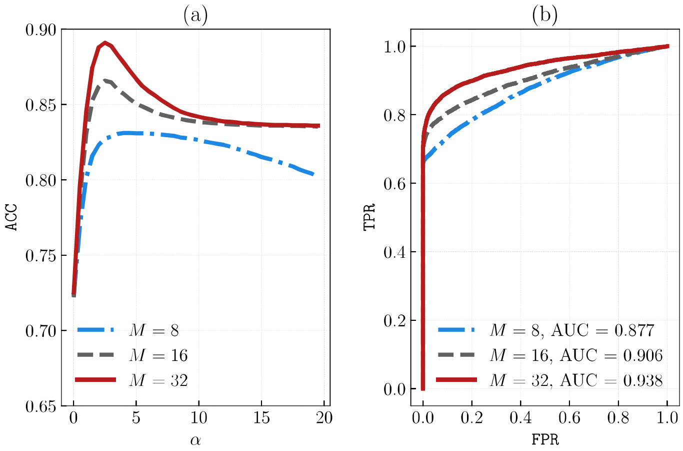


Fig. 5. The performance achieved under JDAE with various values of $M = 8, 12, 16$ in terms of (a) accuracy, and (b) the ROC curve. In this simulation, $N_{\text{A}} = N_{\text{B}} = 8$ and $\text{W} = 30$.

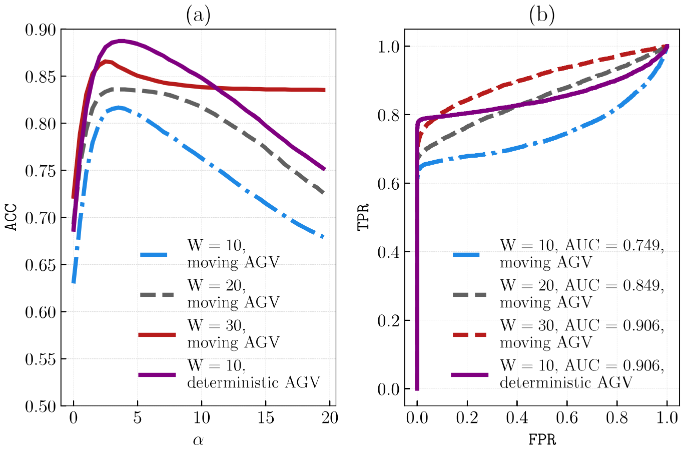


Fig. 6. The performance achieved under JDAE with various values of $\text{W} = 10, 20, 30$ in terms of (a) accuracy, and (b) the ROC curve. In this simulation, $N_{\text{A}} = N_{\text{B}} = 8$ and $M = 16$. This simulation considers 2 cases: moving AGV and deterministic AGV.

be classified as $\mathcal{H}_a$ events. By contrast, when α is very high, the detection algorithm ignores the majority of the jamming attacks. For example, considering $M = 8$, setting α to 0.0 and 19.5 obtain an accuracy of just 0.72 and 0.80, respectively. As a result, an appropriate setting of α is necessary to achieve a satisfactory detection performance. In addition, we also can see that a higher value of M leads to improved accuracy. For instance, considering $\alpha = 2.5$, the accuracy obtained in the case when $M = 8, 16, 32$ is approximately $0.82, 0.86, 0.89$, respectively. This is because a higher resolution of the codebook allows JDAE to learn the underlying pattern more efficiently. The insight is also confirmed by Fig. 5 indicating that the AUC score is improved when M increases. Moreover, Fig. 5(b) also demonstrates the trade-off between the achieved $\texttt{FPR}$ and $\texttt{TPR}$. In the context of classification, it is ideal to obtain a high value of $\texttt{TPR}$ and a low value of $\texttt{FPR}$ simultaneously. Nevertheless, increasing the detection threshold will improve $\texttt{TPR}$ while degrading $\texttt{FPR}$. We also observe that the trade-off becomes more disadvantageous when M decreases. However, when $M = 16$, we can still guarantee $\texttt{TPR} = 0.8$ if we accept $\texttt{FPR} = 0.2$.

2) The impact of window length: Fig. 6 demonstrates the impact of window length on the accuracy performance as well as the ROC curve acquired under JDAE for 3 cases $\text{W} = 10, 20$ and 30. This simulation considers 2 scenarios: AGV moves and AGV's location is deterministic. Regarding the case when AGV moves, the observation is that increasing the window length will substantially improve both the accuracy and the AUC score of the detection algorithm. To be specific, the obtained AUC score is equal to $0.749, 0.849$ and 0.906 in the case when $\text{W} = 10, 20$ and 30, respectively. The explanation is that increasing the window length will reduce the effect of small-scale fading variation. However, AGV might need additional memory to store the RSS vectors. Again, Fig. 6(b) also illustrates the trade-off between $\texttt{TPR}$ and $\texttt{FPR}$. When $\text{W} = 30$, a satisfactory performance is still obtained since $\texttt{TPR} = 0.83$ if we accept $\texttt{FPR} = 0.2$. The notable point is that the both $\texttt{ACC}$ and AUC score show an improvement when the location of AGV is deterministic. When $\text{W} = 10$ and $\alpha = 5$, JDAE achieves $\texttt{ACC} = 0.89$, which is even greater compared to the scenario where AGV moves with $\text{W} = 30$ and $\alpha = 5$. This is because when AGV does not move, the channel distribution does not change significantly, leading to easier jamming detection.

3) The impact of the number of antennas: Fig. 7(a) shows the accuracy of the detection algorithm whereas Fig. 7(b) demonstrates the trade-off between $\texttt{TPR}$ and $\texttt{FPR}$ under a different number of antennas at Attacker, i.e., $N_{\text{A}} = 8, 16$ and 32. Herein, the number of antennas at BS remains unchanged ($N_{\text{B}} = 8$). The striking point is that the detection performance is enhanced if Attacker utilizes more antennas. To explain, utilizing more antennas allows Attacker to focus more energy toward AGV, leading to more destructive, but more easily noticeable attacks. For example, in the case when $N_{\text{A}} = 32$, the proposed algorithm achieved a very high AUC, i.e., 0.953. Also, by setting $\alpha = 2.5$, nearly 90% of the testing jamming attacks were classified correctly.

4) The impact of the distance between Attacker and BS: Next, we investigate the impact of the distance between Attacker and BS on the achieved accuracy of JDAE, as depicted in Fig. 8. This simulation considers 4 different average distances, which are $1.25, 2.5, 5.0$, and 10.0 m. To do this, Attacker's location was randomly generated within a circular area centered on BS and with radii of $2.5, 5.5, 10.0$, and 20.0 m, respectively. As can be seen, the accuracy of JDAE increases when Attacker is closer to BS. For example, when the average distance between the Attacker and BS is 1.25 m and $\alpha = 5.0$, an accuracy of nearly 100% is achieved. By contrast, when the average distance between Attacker and BS is 10.0 m, the detection accuracy drops below 90%. This is because when Attacker and BS are in close proximity, there is a possibility of having an identical strongest beam, resulting in a significant change in the RSS vector, thus enabling easy detection of the attack. Nevertheless, when Attacker is further away from BS, AGV may not perceive significant changes in the RSS vector if its peak remains unchanged, causing false

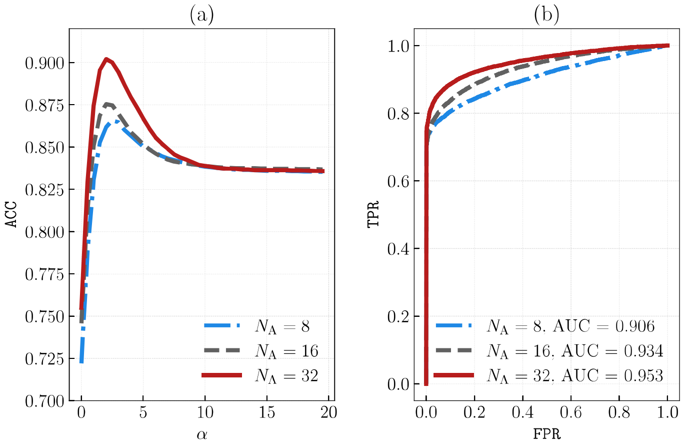


Fig. 7. The performance achieved under JDAE with various values of $N_{\mathrm{A}} = 8, 16, 32$ in terms of (a) accuracy, and (b) the ROC curve. In this simulation, $N_{\mathrm{B}} = 8$, $M = 16$ and W = 30.

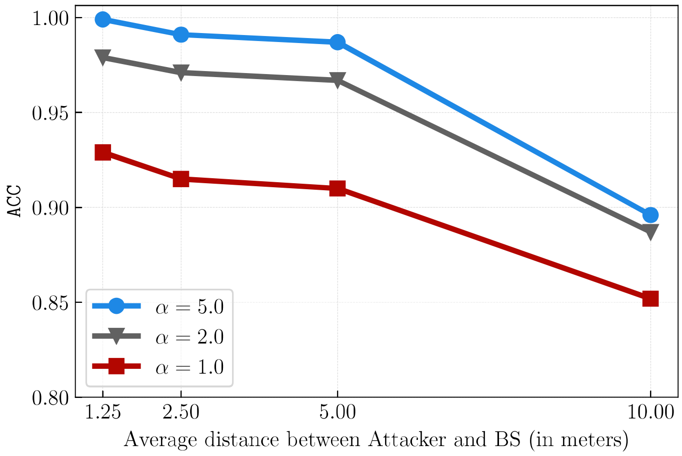


Fig. 8. The accuracy achieved under JDAE versus the average distance between Attacker and BS with various values of $\alpha = 1.0, 2.0$ and 5.0. In this simulation, $N_{\mathrm{A}} = N_{\mathrm{B}} = 8, \mathrm{W} = 20$ and $M = 32$.

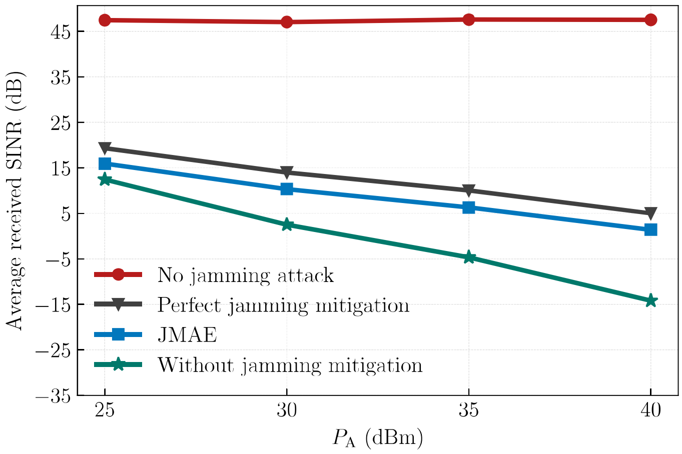


Fig. 9. The average SINR received at AGV with various values of P_{A}. In this simulation, $N_{\mathrm{A}} = N_{\mathrm{B}} = 16$, $M = 32$, and $P_{\mathrm{B}} = 20$ dBm.

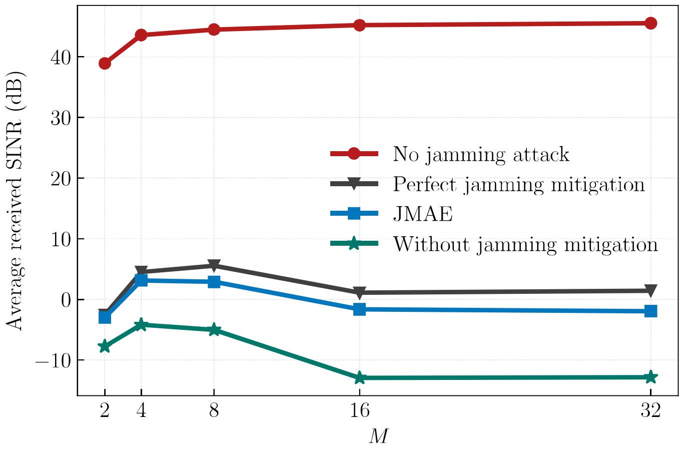


Fig. 10. The average SINR received at AGV with various values of M. In this simulation, $N_{\mathrm{A}} = N_{\mathrm{B}} = 8$, $P_{\mathrm{A}} = 35$ dBm, and $P_{\mathrm{B}} = 20$ dBm.

detections. It is important to note that despite the very high accuracy, it is anticipated that the mitigation performance will be low when Attacker is close to BS.

C. Numerical Results for Jamming Mitigation

In this simulation, we fix the location of Attacker at $(20, 15)$ as JMAE is only employed once an attack is captured. There were 10000 realizations and shadowing profiles generated during the evaluation. As a benchmark, we consider the following schemes:

- *No jamming attack:* There are no jamming signals.
- *Perfect jamming mitigation:* The RSS vector is reconstructed perfectly.
- *JMAE:* The RSS vector is reconstructed by using JMAE.
- *Without jamming mitigation:* No jamming mitigation is applied. Therefore, AGV performs the beam determination based on the corrupted RSS vector.

1) The impact of transmit power utilized by Attacker: Fig. 9 shows the average SINR received at AGV in the case when Attacker utilizes various transmit powers. As can be seen, the proposed attack strategy causes an enormous deterioration in the average received SINR, which tends to increase proportionally with P_{A}. For example, when $P_{\mathrm{A}} = 25$ dBm, the average received SINR is equal to less than 15 dB, which is nearly 35 dB lower than that in the case when there is no jamming attack. If Attacker increases the transmit power to $P_{\mathrm{A}} = 40$ dBm, this gap is even more significant, i.e., more than 60 dB. This result indicates that by confusing AGV during the beam training procedure, the proposed attack strategy can break the downlink data transmission. In addition, Fig. 9 also demonstrates that employing JMAE for jamming mitigation can improve the average received SINR remarkably. For instance, when $P_{\mathrm{A}} = 40$ dBm, utilizing JMAE can achieve an average received SINR of approximately 2 dB, whereas this figure is just -15 dB if the jamming mitigation is not applied. Last but not least, the performance achieved under JMAE is just around 2 dB lower than that in the case of perfect jamming mitigation, which proves the effectiveness of the proposed JMAE algorithm.

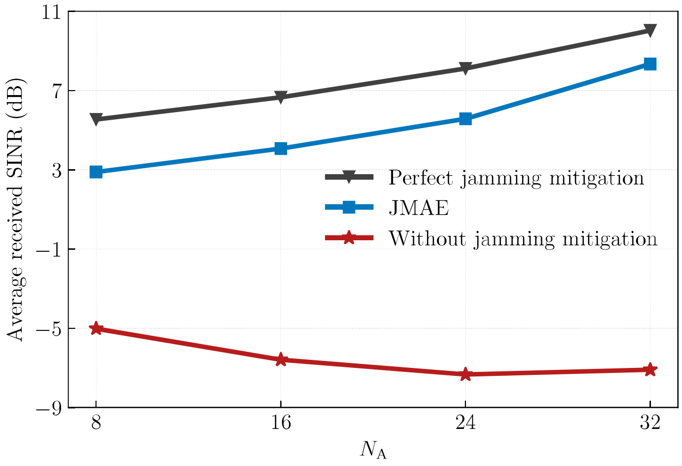


Fig. 11. The average SINR received at AGV with various values of $N_{\rm A}$. In this simulation, $N_{\rm B} = 8$, $M = 8$, $P_{\rm A} = 35$ dBm, and $P_{\rm B} = 20$ dBm.

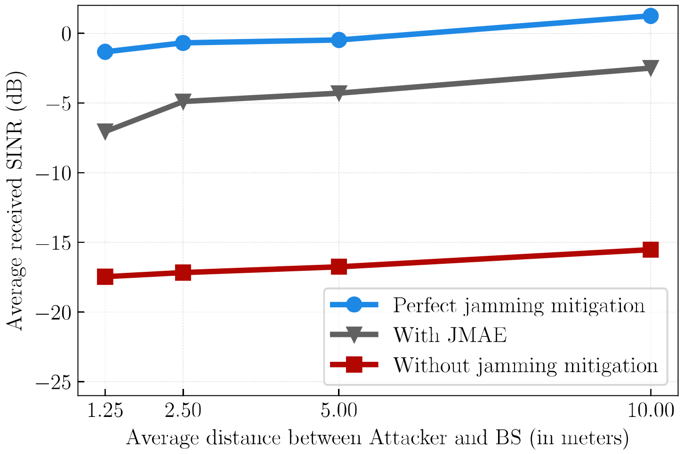


Fig. 12. The average SINR received at AGV versus average distance between Attacker and BS. In this simulation, $N_{\rm A} = N_{\rm B} = 8$, $M = 32$, $P_{\rm B} = 20$ dBm.

2) The impact of codebook size: Fig. 10 demonstrates the impact of codebook resolution on the SINR performance. It can be seen that Attacker can launch more destructive attacks on the legitimate system by using a higher-resolution codebook. The explanation is that when M increases, Attacker can beam the jamming signals toward AGV more accurately. Moreover, for all the considered values of M, despite an insignificant reduction compared to the case of perfect jamming mitigation, JMAE still shows a noticeably improved performance in comparison with the case of no jamming mitigation, i.e., up to 15 dB when $M = 16, 32$.

3) The impact of the number of antennas: Fig. 11 represents the average SINR performance achieved at AGV in the case when Attacker utilizes a different number of antennas. As employing more antennas allows Attacker to focus the jamming signals in a narrower beam with a higher gain, the average received SINR decreases when $N_{\rm A}$ increases. Due to the narrower jamming beam, if AGV still can select the advantageous beam for BS, the destructive impact of jamming signals will be less severe. As a result, when $N_{\rm A}$ increases, the

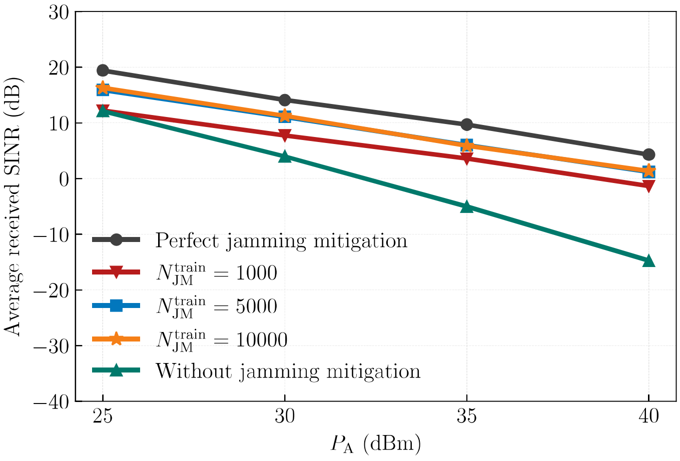


Fig. 13. The average SINR received at AGV with various values of $P_{\rm A}$. In this simulation, $N_{\rm A} = N_{\rm B} = 16$, $M = 32$, $P_{\rm B} = 20$ dBm.

TABLE IV
BENCHMARK OF AVERAGE LATENCY (IN MILLISECONDS) OF JDAE AND JMAE OVER 10000 REALIZATIONS.

M	8	16	32	64
Algorithm 1	1.90	2.10	2.89	3.7
Algorithm 2	1.34	1.66	2.46	3.0

average received SINR enhances under the perfect jamming mitigation. Interestingly, the performance achieved by JMAE still improves when $N_{\rm A}$ increases, which is equal to 7 dB when $N_{\rm A} = 16$. This indicates that the more antennas Attacker can utilize to launch more destructive attacks, the more effectively JMAE can perform.

4) The impact of the distance between Attacker and BS: Fig. 12 illustrates the relationship between the average SINR received at AGV and the 4 average distances between Attacker and BS. As previously mentioned, JDAE achieves a very high detection accuracy when Attacker is close to BS. However, this simulation points out that the average received SINR is still very low in this scenario. This is because even though JMAE can effectively mitigate the jamming signal and AGV can accurately select the optimal beam, this beam is still advantageous for Attacker to launch attacks. In particular, the average received SINR is just -7.0 dB when the average distance is 1.25m, but it improves to 7 dB when Attacker is located 10.0 m away from BS.

5) The impact of $N_{\rm JM}^{\rm train}$: Fig. 13 illustrates the achieved average SINR when JMAE is trained on a different number of synthetic data samples. Considering $P_{\rm A} = 30$ dBm, using $N_{\rm JM}^{\rm train} = 1000$ achieves an average received SINR of approximately 8 dB, which is still noticeably lower than that for the case when $N_{\rm JM}^{\rm train} = 5000$. However, the performance does not improve further when more synthetic data samples are used for training, i.e., $N_{\rm JM}^{\rm train} = 5000, 10000$. This points out that using an insufficient number of synthetic data samples might lead to unsatisfactory performance, but it is still not necessary to use a very large value of $N_{\rm JM}^{\rm train}$.

D. Computational Latency Analysis

Table IV shows the average latency (in milliseconds) that Algorithm 1 and Algorithm 2 require to compute one inference for various codebook sizes. As we can see, increasing M leads to a higher latency for the online inference since utilizing a higher-resolution codebook means that the AE architectures employ more layers and neurons. For example, when $M = 8$, Algorithm 2 only needs an average computational time of 1.34 ms to execute one jamming mitigation while Algorithm 1 spends 1.9 ms for detecting one jamming signal. These figures nearly double when $M = 64$, which are 3.0 and 3.7 ms on average for Algorithm 2 and Algorithm 1, respectively. In the case when a jamming attack is identified, the latency for Algorithm 3 is 3.2 ms for $M = 8$, and more than 6 ms for $M = 64$. Since the periodicity of SS burst transmission can be up to 160 ms in 5G NR, Algorithm 3 shows much promise for meeting the latency requirement.

VI. Conclusions

We investigated the security aspect of the beam training procedure in 5G mmWave communications. We considered a scenario normally encountered in industrial factories where an AGV and a BS perform beam training to maintain a wireless connection using a mmWave frequency. We introduced a simple attack strategy targeting the beam training procedure to manipulate the RSS vector at AGV, resulting in a very large deterioration, up to 60 dB, for the average received SINR. To defend against this attack, we propose an anti-jamming countermeasure that comprises two AE-based ML models, namely JDAE and JMAE, for jamming detection and jamming mitigation, respectively. The numerical results showed that JDAE can identify the jamming attacks with an accuracy of more than 80%. In addition, once JDAE captures an attack, JMAE, in turn, can be employed to alleviate the jamming effects, leading to a significant improvement of the average SINR, i.e., more than 15 dB compared to the case of no jamming mitigation. A potential future extension is to investigate the case when multiple AGVs are present and BS utilizes a hybrid beamforming. It is anticipated that the autoencoder architectures will be more complicated, resulting in an improved detection accuracy with a cost of increasing latency during the online executation.

Acknowledgment

This work was supported in part by the WMG Centre High Value Manufacturing Catapult, University of Warwick, Coventry, U.K.

References

[1] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "A survey on 5G usage scenarios and traffic models," *IEEE Commun. Surv. Tutor.*, vol. 22, no. 2, pp. 905–929, 2020.

[2] R. W. Heath, N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 436–453, 2016.

[3] M. Giordani, M. Polese, A. Roy, D. Castor, and M. Zorzi, "A tutorial on beam management for 3GPP NR at mmWave frequencies," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 1, pp. 173–196, 2018.

[4] C. Jeong, J. Park, and H. Yu, "Random access in millimeter-wave beamforming cellular networks: issues and approaches," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 180–185, 2015.

[5] C. N. Barati, S. A. Hosseini, S. Rangan, P. Liu, T. Korakis, S. S. Panwar, and T. S. Rappaport, "Directional cell discovery in millimeter wave cellular networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 12, pp. 6664–6678, 2015.

[6] K. Ma, Z. Wang, W. Tian, S. Chen, and L. Hanzo, "Deep learning for mmwave beam-management: State-of-the-art, opportunities and challenges," *IEEE Wirel. Commun.*, 2022.

[7] T. M. Hoang, T. Van Chien, T. Van Luong, S. Chatzinotas, B. Ottersten, and L. Hanzo, "Detection of spoofing attacks in aeronautical ad-hoc networks using deep autoencoders," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 1010–1023, 2022.

[8] S. Hur, T. Kim, D. J. Love, J. V. Krogmeier, T. A. Thomas, and A. Ghosh, "Millimeter wave beamforming for wireless backhaul and access in small cell networks," *IEEE Trans. Commun.*, vol. 61, no. 10, pp. 4391–4403, 2013.

[9] X. Cheng, M. Wang, and S. Li, "Compressive sensing-based beamforming for millimeter-wave OFDM systems," *IEEE Trans. Commun.*, vol. 65, no. 1, pp. 371–386, 2016.

[10] A. Abdelreheem, E. M. Mohamed, and H. Esmaiel, "Location-based millimeter wave multi-level beamforming using compressive sensing," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 185–188, 2017.

[11] M. Alrabeiah and A. Alkhateeb, "Deep learning for mmwave beam and blockage prediction using sub-6 GHz channels," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5504–5518, 2020.

[12] K. Ma, D. He, H. Sun, Z. Wang, and S. Chen, "Deep learning assisted calibrated beam training for millimeter-wave communication systems," *IEEE Trans. Commun.*, vol. 69, no. 10, pp. 6706–6721, 2021.

[13] J. Zhang and C. Masouros, "Learning-based predictive transmitter-receiver beam alignment in millimeter wave fixed wireless access links," *IEEE Trans. Signal Process.*, vol. 69, pp. 3268–3282, 2021.

[14] M. Lichtman, R. Rao, V. Marojevic, J. Reed, and R. P. Jover, "5G NR jamming, spoofing, and sniffing: Threat assessment and mitigation," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2018, pp. 1–6.

[15] Y. E. Sagduyu, T. Erpek, and Y. Shi, "Adversarial machine learning for 5G communications security," *Game Theory and Machine Learning for Cyber Security*, pp. 270–288, 2021.

[16] B. Kim, Y. Sagduyu, T. Erpek, and S. Ulukus, "Adversarial attacks on deep learning based mmWave beam prediction in 5G and beyond," in *2021 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE, 2021, pp. 590–594.

[17] H. Pirayesh and H. Zeng, "Jamming attacks and anti-jamming strategies in wireless networks: A comprehensive survey," *IEEE Commun. Surv. Tutor.*, 2022.

[18] F. O. Catak, M. Kuzlu, E. Catak, U. Cali, and D. Unal, "Security concerns on machine learning solutions for 6G networks in mmWave beam prediction," *Physical Communication*, p. 101626, 2022.

[19] S. Xu, W. Xu, C. Pan, and M. Elkashlan, "Detection of jamming attack in non-coherent massive SIMO systems," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 9, pp. 2387–2399, 2019.

[20] L. Chiarello, P. Baracca, K. Upadhya, S. R. Khosravirad, and T. Wild, "Jamming detection with subcarrier blanking for 5G and beyond in industry 4.0 scenarios," in *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2021, pp. 758–764.

[21] O. Puñal, I. Aktaş, C.-J. Schnelke, G. Abidin, K. Wehrle, and J. Gross, "Machine learning-based jamming detection for IEEE 802.11: Design and experimental evaluation," in *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*. IEEE, 2014, pp. 1–10.

[22] M. Cheng, Y. Ling, and W. B. Wu, "Time series analysis for jamming attack detection in wireless networks," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–7.

[23] N. Wang, W. Li, A. Alipour-Fanid, L. Jiao, M. Dabaghchian, and K. Zeng, "Pilot contamination attack detection for 5G mmWave grant-free IoT networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 658–670, 2020.

[24] A. S. Abdalla, K. Powell, V. Marojevic, and G. Geraci, "UAV-assisted attack prevention, detection, and recovery of 5G networks," *IEEE Wirel. Commun.*, vol. 27, no. 4, pp. 40–47, 2020.

[25] Q. Qin, L. Gui, P. Cheng, and B. Gong, “Time-varying channel estimation for millimeter wave multiuser MIMO systems,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9435–9448, 2018.

[26] S. Kutty and D. Sen, “Beamforming for millimeter wave communications: An inclusive survey,” *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 949–973, 2015.

[27] W. Hao, M. Zeng, G. Sun, O. Muta, O. A. Dobre, S. Yang, and H. Gacanin, “Codebook-based max–min energy-efficient resource allocation for uplink mmWave MIMO-NOMA systems,” *IEEE Trans. Commun.*, vol. 67, no. 12, pp. 8303–8314, 2019.

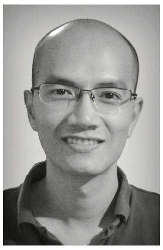
[28] K. Pelechrinis, I. Broustis, S. V. Krishnamurthy, and C. Gkantsidis, “A measurement-driven anti-jamming system for 802.11 networks,” *IEEE/ACM Trans. Netw.*, vol. 19, no. 4, pp. 1208–1222, 2011.

[29] J. Jeung, S. Jeong, and J. Lim, “Adaptive rapid channel-hopping scheme mitigating smart jammer attacks in secure WLAN,” in *MILCOM 2011 Military Communications Conference*. IEEE, 2011, pp. 1231–1236.

[30] 3GPP, “3GPP TR 38.901 version 16.1.0 Release 16: Study on channel model for frequencies from 0.5 to 100 GHz,” 2020.

[31] S. D. Van, H. Q. Ngo, and S. L. Cotton, “Wireless powered wearables using distributed massive MIMO,” *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2156–2172, 2020.

Son Dinh-Van received the B.S. degree from Hanoi University of Science and Technology, Vietnam, in 2013, the M.S. degree from Soongsil University, Seoul, South Korea, in 2015, and the Ph.D. degree from Queen’s University of Belfast, Belfast, U.K., in 2019, all in electrical engineering. He worked as a Data Scientist in 2020 and a Visiting Researcher with Middlesex University, U.K in 2021. His current research interests include 5G/6G wireless communications, wireless power transfer, millimeter wave and Terahertz communications, reconfigurable intelligent surface, and machine learning.



Tiep. M. Hoang received the B.Eng. degree from the HCMC University of Technology, Vietnam, in 2012, the M.Eng. degree from Kyung Hee University, South Korea, in 2014, and the Ph.D. degree from the Queen’s University of Belfast, United Kingdom, in 2019. From 2020 to 2022, he was a (postdoctoral) Research Fellow with the School of Electronics and Computer Science, the University of Southampton, United Kingdom. Since May 2022, he has been a Postdoctoral Fellow with the Department of Electrical Engineering, the University of Colorado Denver, United States. His current research interests include 5G/6G wireless communications, wireless security and authentication, reconfigurable intelligent surface (RIS), convex optimization, and machine learning.

Berna Bulut Cebecioglu received the B.Sc. degree in electrical engineering from Kocaeli University, Kocaeli, Turkey, in 2007, the M.Sc. degree in communication networks and signal processing, and the Ph.D. degree in electrical and electronic engineering from the University of Bristol, Bristol, U.K., in 2011 and 2016, respectively. She worked as a Senior Research Associate in electrical and electronic engineering at the University of Bristol and a Senior Research Fellow at WMG, University of Warwick, U.K. She is an Associate Professor in Turkey and currently working as a Senior Research Fellow in Future Communication Systems, Birmingham City University, U.K. Her research interests include 5G and beyond networks, cross-layer design and optimisations of wireless networks, multimedia multicasting and broadcasting services, application layer forward error correction codes, propagation modeling, millimeter wave, and vehicular communications.

Daniel S. Fowler has a B.Eng. (Hons.) in computer and control system engineering, a M.S. in forensic computing, and a Ph.D. in automotive cybersecurity, all degrees from Coventry University, UK. He is a Research Fellow in the Secure Cyber Systems Research Group at WMG. His research field is secure and resilient system design. He has aided the engineering and delivery of several Innovate UK funded projects. He contributes to the open source community through software and has written over 300 articles. He is a Chartered Engineer, and a member of the IET and ACM.

Yuen Kwan Mo has a Ph.D. in communications and network engineering from The University of Warwick, Coventry, UK. He is a Project Engineer with the Connectivity and Communications Technology Research Group, WMG, The University of Warwick. His specialisms include: 5G and cellular communications for Industry 4.0 applications, connected and autonomous vehicles, millimeter-wave communications, massive MIMO, precoding techniques and optimization algorithm.

Matthew D. Higgins (Senior Member, IEEE) is a Reader at the University of Warwick, where he leads WMG’s Connectivity and Communications Technology Research Group within its Intelligent Vehicles Directorate. His research interests span 5G and Beyond, Core Networking, IEEE 802.3xx, GNSS, and Timing, with applications to both the Automotive and Manufacturing domains. Coupled with an overarching motivation to ensure ongoing resilience of the domain is considered, Matthew leads many high-value collaborative projects funded through EPSRC, Innovate UK, and HVMC, as well as also leading multiple projects funded directly by industry.