DeepSpoof: Deep Reinforcement Learning-based Spoofing Attack in Cross-Technology Multimedia Communication

Demin Gao, *Member, IEEE,* Liyuan Ou, *Member, IEEE,* Ye Liu *Member, IEEE,* Qing Yang, *Senior Member, IEEE,* and Honggang Wang, *Fellow, IEEE*

***Abstract*—Cross-technology communication is essential for the Internet of Multimedia Things (IoMT) applications, enabling seamless integration of diverse media formats, optimized data transmission, and improved user experiences across devices and platforms. This integration drives innovative and efficient IoMT solutions in areas like smart homes, smart cities, and healthcare monitoring. However, this integration of diverse wireless standards within cross-technology multimedia communication increases the susceptibility of wireless networks to attacks. Current methods lack robust authentication mechanisms, leaving them vulnerable to spoofing attacks. To mitigate this concern, we introduce DeepSpoof, a spoofing system that utilizes deep learning to analyze historical wireless traffic and anticipate future patterns in the IoMT context. This innovative approach significantly boosts an attacker's impersonation capabilities and offers a higher degree of covertness compared to traditional spoofing methods. Rigorous evaluations, leveraging both simulated and real-world data, confirm that DeepSpoof significantly elevates the average success rate of attacks.**

***Index Terms*—Cross-Technology Communication, Internet of Multimedia Things, Spoofing Attack, Deep Learning.**

I. Introduction

CROSS-technology communication (CTC) [1] for the Internet of Multimedia Things (IoMT) [2], [3] refers to the seamless integration of multimedia content across diverse technologies and platforms within the IoMT ecosystem. This concept enables the exchange and processing of audio, video, and other media types between heterogeneous devices, enhancing the user experience and functionality of IoMT applications [4]–[6]. Applications of cross-technology multimedia communication in the IoMT range from smart homes, where it facilitates multimedia sharing and control across devices, to smart cities, where it supports real-time monitoring and data analysis for public services. In healthcare, it enables remote patient monitoring, diagnosis, and surgical assistance. In the field of Internet of Vehicles (IoV), CTC can be utilized to monitor the status of important vehicle components in real time for heterogeneous wireless devices employed in IoV, predict possible failures, perform maintenance in advance, and reduce the incidence of failures.

Entertainment and social media platforms also benefit from this technology, delivering high-quality audio and video experiences. The advantages of CTC for the IoMT include seamless integration, which breaks down technological barriers and allows for the free flow of multimedia content. It also offers efficient data transmission, minimizing delays and bandwidth consumption, crucial for real-time applications. Additionally, this communication enhances user experiences by delivering high-quality media content and enabling rich, interactive experiences. Finally, it fosters innovation, enabling the development of new and innovative IoMT applications that leverage the power of cross-technology multimedia communication.

In the IoMT context, security has become a top priority [7], [8]. The complexity of wireless environment and the proliferation of smart multimedia devices have created significant threats to wireless security [9]–[11]. Among various malicious attacks, spoofing is particularly concerning. Spoofing attacks exploit the open and shared nature of the communication medium to transmit fraudulent data to victims, falsely posing as a trusted source. These attacks can lead to network failure due to misleading information from the attacker [12], [13]. So far, most attack strategies have been designed for homogeneous networks [14], [15], where it is not feasible for a WiFi device to spoof ZigBee devices in traditional designs. However, the emergence of CTC techniques has removed this constraint.

Like other wireless communication methods [16], [17], CTC is vulnerable to malicious attacks, including sniffing and spoofing. However, the spoofing techniques used in CTC differ from traditional designs. While most conventional spoofing strategies are designed for homogeneous networks where signals from devices of the same type, such as ZigBee, are spoofed by other signals in the same channel, with CTC, it is feasible to use a WiFi signal to forge ZigBee information and subsequently attack ZigBee devices. In the event of such attacks, the applications developed using CTC may be severely impacted, leading to fraudulent data or deceptive urgent events. Therefore, it is essential to investigate the security of CTC to ensure reliable and effective communication via CTC links.

This work was supported by the Future Network Scientific Research Fund Project (Grant No. FNSRFP-2021-YB-17) and the Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institutions. *(Corresponding author: Ye Liu)*

Demin Gao and Liyuan Ou are with the College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China (e-mail: dmgao@njfu.edu.cn; liyuanou@njfu.edu.cn).

Ye Liu is with the College of Artificial Intelligence, Nanjing Agricultural University, Nanjing 210095, China (e-mail: yeliu@njau.edu.cn).

Qing Yang is with the Department of Computer Science and Engineering, University of North Texas, Denton, TX 76205 USA (e-mail: qing.yang@unt.edu).

Honggang Wang is with the Department of Graduate Computer Science and Engineering, Katz School of Science and Health, Yeshiva University, New York, NY 10016 USA (e-mail: Honggang.wang@yu.edu).

CTC techniques introduce new vulnerabilities that malicious actors can exploit through spoofing, disrupting regular wireless communication. However, pinpointing the exact moments for spoofing attacks is a considerable challenge. This is due to the unpredictable nature of wireless traffic patterns caused by random backoff periods. In this paper, we introduce a cutting-edge spoofing attack strategy that combines deep learning and CTC techniques, named as DeepSpoof (Deep Learning Spoofing). This strategy leverages the robust transmission power and extended range of WiFi signals to target and spoof ZigBee-embedded devices effectively. It's worth mentioning that our proposed approach is adaptable and can be applied to other scenarios where multiple wireless technologies coexist, highlighting its versatility and potential impact.

Our contributions are summarized as follows:

- We introduce a novel attack strategy called DeepSpoof, which integrates deep learning techniques based on Long Short-Term Memory (LSTM) with CTC technology. This strategy transforms the spoofing attack into a time series process by meticulously selecting the slot duration and defining the status. This design facilitates easy implementation without requiring any modifications to the firmware or hardware of both WiFi and ZigBee devices.
- A method is presented to enable parallel cross-technology communications with DeepSpoof in ZigBee channels. Specifically, a WiFi device is capable of simultaneously spoofing the communication in two independent channels, demonstrating the versatility of our approach.
- We develop a spoofing attack scheme that incorporates deep learning to capture temporal patterns and predict future wireless traffic. Our proposed scheme is evaluated using a hybrid platform consisting of a USRP-N210 and MICAz, providing empirical evidence of its feasibility.

The remainder of this paper is organized as follows. Section II provides a review of the related work. Next, we delves into the preliminaries of spoofing attack strategies in Section III. In Section IV, we present the problem formulation of this work. Section V provides the detailed information of DeepSpoof design. Then, performance evaluation is presented in Section VI. Finally, we conclude our paper in Section VII.

II. Related Work

This section reviews the related work of our study with respect to the following domains.

Spoofing Schemes for IoMT Networks. IoMT networks have seen the emergence of numerous spoofing schemes in the literature. Specifically, in [18], [19], the authors comprehensively examine spoofing attacks targeting wireless sensor networks, reviewing detection techniques and countermeasures against such threats. In [20], the authors propose a new attack method called adversarial laser beam (AdvLB), which can perform adversarial attacks on autonomous vehicles to recognize traffic signs by manipulating the physical parameters of the laser beam. FooLoc [21] fools WiFi CSI fingerprinting DNNs over the realistic wireless channel between the attacker and the victim access point (AP). In another study [22], an adversarial transmitter-receiver pair assumes the roles of generator and discriminator in a Generative Adversarial Network, engaging in a minimax game to craft optimal spoofing signals. Researchers proposed a novel adversarial attack framework [23], and designed to generate adversarial malicious traffic capable of deceiving ML-based traffic classification systems. It exhibits a high evasion growth rate across multiple models and datasets Furthermore, [24] delves into the theoretical implications of ACK spoofing on rate control and transport-layer protocols, constructing mathematical models to assess throughput performance under attack conditions for Minstrel.

Anti-Spoofing in Homogeneous Networks. A considerable amount of research has been devoted to crafting anti-spoofing solutions for secure wireless device connectivity. For example, Tomic et al. [25] introduced a mechanism that utilizes physical data, including MAC addresses and signal strength values, to identify and mitigate harmful spoofing attacks. Meanwhile, Mahmood et al. [26] proposed a more secure user authentication scheme using elliptic curve cryptography for multimedia IoT. In [27], the authors designed an autoencoder deep neural network (AENN) that minimizes unauthorized access by predicting data transmission outcomes. An traffic obfuscation method based on neural networks was proposed in [28], which generates traffic distortions with minimal overhead and computational cost but attains comparable obfuscation performance. Such obfuscation can effectively defend eavesdropping or traffic analysis attacks. In another study, Madani et al. [29] presented an innovative method utilizing multi-model Long Short-Term Memory (LSTM) for spotting MAC-layer spoofing attempts. Additional strategies include PHY-layer authentication [30], Node Identification [31], and the Beacon-Trap Approach [32]. It's worth noting that these works primarily focus on spoofing attacks executed within homogeneous networks using a single communication technique.

Cross-Technology Communications. ESence [33] introduced the CTC technique, which aims to create an alphabet of implicit messages by encoding CTC symbols using packet lengths and duration information [34]. FreeBee [35] utilizes packet timing for bidirectional communication between WiFi and ZigBee devices. HoWiES [36] is designed to enable WiFi radios to convey different messages to ZigBee radios while using less energy. ZigFi [37] employs RSSI to capture interference signatures and facilitate communication from ZigBee to WiFi. WEBee [38] and BlueBee [39] are practical CTC designs that simulate through the physical layer [40], along with other techniques like WiFi-to-Bluetooth [41] and LTE-to-ZigBee [42]. WibZig [43] can accurately simulate any given ZigBee symbol by selecting CCK codewords that exhibit similar phase characteristics to the ZigBee chip. Waves [44] utilizes WiFi to ZigBee CTC and adaptive transmit power control technology of WiFi access points to achieve reliable and fast data transmission in low duty cycle ZigBee networks. In [45], the authors provide a comprehensive survey of CTC techniques utilized in heterogeneous IoT networks.

Anti-Spoofing in CTC Networks. To counter spoofing attacks in CTC, a collaborative mechanism between WiFi and ZigBee devices is introduced in [46], leveraging physical layer information and the One-Class Support Vector Machine

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2024.3414660

(OSVM) algorithm for attack detection. MuZi [47] introduces three mechanisms to mitigate WiFi interference: interference assessment, channel switching, and connectivity maintenance. However, these common security measures may not be sufficient to thwart certain attacks. Given the growing popularity of deep learning, this paper introduces a novel approach for launching spoofing attacks using deep neural networks, aiming to achieve the highest attack success rate.

In traditional methods [48], reactive attacks only launch attacks when specific signals (such as ZigBee MAC frame headers) are detected, which require extremely high real-time performance. Therefore, in complex network environments, this attack method is less efficient; Random attacks waste a lot of time and energy trying various possibilities, only a few of which may succeed, which results in a huge waste of resources and is extremely inefficient. In this paper, we introduce a cutting-edge spoofing attack strategy that combines deep learning and CTC techniques. This strategy leverages the robust transmission power and extended range of WiFi signals to target and spoof ZigBee-embedded devices effectively. Compared with existing research [49], the probability of successful spoofing attacks is increased by 21.8%.

III. Preliminaries

A. Bidirectional CTC between WiFi and ZigBee

ZigFi [37] carefully overlaps ZigBee packets with WiFi packets and convey data from ZigBee to WiFi without modifying the WiFi or ZigBee physical layer by encoding the data by different WiFi Channel State Information chirps with different indexes of the starting channel. WEBee [38] is the first work based on physical-layer to achieve direct communication from WiFi to ZigBee. One possible approach is for a WiFi device to intricately construct its frame payload, resulting in an RF waveform that mimics ZigBee signals. In this scenario, the preamble, header, and trailer of the WiFi frame will be disregarded by the ZigBee receiver and treated as noise. NetCTC, as presented in the study by [50], proposes a real-time interaction mechanism that enables dependable, transmission-efficient, and simultaneous interactive communication among diverse devices. Besides many others that concurrent interactive communication among heterogeneous devices [50]. Fig. 1 plots the networking support for bidirectional CTC.

According to Cisco's predictions, the number of WiFi hotspots will reach approximately 600 million, with a 53% accessibility rate in major urban centers. WiFi technology can leverage its extensive bandwidth to acquire sufficient computational resources for processing and analyzing the vast and intricate data received from ZigBee networks. It can also perform specialized operations, such as spoofing attacks, which are unique to this communication technology. Since all nodes in the ZigBee network are homogeneous, a ZigBee device can only spoof other ZigBee devices. The WiFi device's transmission range extends nearly 300 meters with a transmission power of 100 milliwatts (-20 dBm), which is significantly greater than that of the ZigBee device (MICAz), which has a range of less than 70 meters and a transmission power of 1 milliwatt (0 dBm).

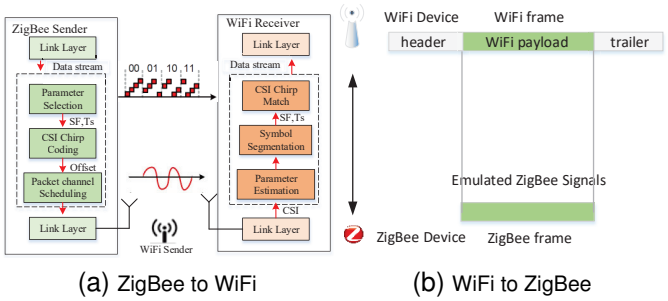


Fig. 1. Cross-Technology Communication between WiFi and ZigBee.

B. The Motivation of DeepSpoof

In the IoMT networks, CTC communication is vulnerable to malicious attacks. It is feasible to use WiFi signals to forge ZigBee information and conduct spoofing attacks. Once such attacks occur, applications developed using CTC will be seriously affected. In the multimedia Internet of Vehicles, malicious attacks may lead to incorrect vehicle decisions and endanger life safety. In multimedia healthcare, malicious attacks may interrupt medical services and cause serious harm to patients. In multimedia smart homes, there are some security threats that can Affects home privacy, security and comfort.

Although CTC is a relatively new technology, there has been limited research on its security vulnerabilities. This paper introduces a novel approach for launching spoofing attacks using deep neural networks in IoMT networks, aiming to achieve the highest attack success probability. This approach highlights the need for more robust security measures in CTC networks and aims to contribute to the growing body of literature on the security of this technology. By utilizing deep learning algorithms, we hope to assist in the development of more effective security measures that can keep pace with the evolving threat landscape.

C. Trade-Off between the Benefits and Costs of DeepSpoof

In this study, we propose utilizing a WiFi device to launch a powerful spoofing attack on ZigBee devices. However, this method has the potential to disrupt normal WiFi communication. The decision to employ WiFi communication in pursuit of a powerful spoofing attack on ZigBee devices must be carefully balanced against the potential costs. On one hand, this approach offers the advantage of a high-powered attack capable of targeting multiple ZigBee devices. This can be particularly beneficial in scenarios where the attacker aims to compromise a large number of ZigBee devices or to execute attacks over a wide area. On the other hand, disrupting WiFi communication may pose challenges if it is essential for the operation of other devices within the network.

It's important to note that existing WiFi infrastructure can be efficiently leveraged to manipulate and interfere with sensor networks using standard WiFi devices. This eliminates the need to acquire a ZigBee device solely for spoofing purposes, as any WiFi device can be exploited using cutting-edge technologies. In other words, we don't have to deploy dedicated WiFi devices for spoofing ZigBee devices; we can instead

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2024.3414660

utilize existing devices for malicious intent. Consequently, the decision to employ WiFi communication in exchange for a powerful spoofing attack on ZigBee devices depends largely on the specific circumstances and the attacker's priorities. A thorough assessment of the trade-offs involved is essential before proceeding with such an approach.

IV. Problem Formulation

In this section, we first present the fundamental concepts of the time-slot model for ZigBee and WiFi devices. We then delve into an analysis of the slot status for both ZigBee and WiFi devices, as well as an overview of the DeepSpoof basics.

A. Time-Slot Model for ZigBee and WiFi Devices

In energy-constrained sensor networks, the utilization of a time-slot model is a common practice for commercial off-the-shelf (COTS) devices, serving to conserve energy, minimize energy consumption, and prevent signal interference. A WiFi Access Point (AP), being a wired device with a constant power supply, has the capability to frequently broadcast messages within its coverage area without any energy constraints. However, in our design, we must consider the coexistence of WiFi devices with other COTS devices, such as Bluetooth and ZigBee, within the same ISM band. The potential for WiFi signal interference with ZigBee communication is significant, leading to disruptions or intrusions in legitimate data transmissions of ZigBee devices when the WiFi device occupies the channel. Consequently, we propose the implementation of a time-slot model for WiFi devices to facilitate spectrum allocation analysis.

In [51], the authors introduce the time-slot scheme to WiFi networks. It's noteworthy that, despite its random channel access mechanism for exceptional spectral efficiency, traditional WiFi technology does not incorporate time-slotting. In specific scenarios, time-slot models can be applied in WiFi technology to achieve specific objectives. One such scenario is the communication between IoT heterogeneous devices, where WiFi and ZigBee devices coexist. The authors in [52] propose time-slot schemes that enable WiFi to predict the status of ZigBee devices, thus enabling more effective management of network resources. These time-slot schemes are based on the observation that ZigBee devices utilize a duty-cycle strategy, alternating between active and dormant states to regulate energy consumption. By predicting the status of ZigBee devices, WiFi can avoid collisions and enhance network performance.

B. Slot Status for WiFi and ZigBee Devices

In our model, we make an assumption for the wireless network without any loss of generality. Firstly, we consider the duration of the periodic working schedule of a device as t, for example, one minute, etc. Secondly, for each working schedule, it is generally split into a sequence of time instances with length τ, which is the unit of working time for an activity. We emphasize that the slot duration τ is equivalent to half of the time required to transmit the maximum MAC frame length, where the maximum size is 127 bytes as specified in 802.15.4. Fig. 2 illustrates the time slots for WiFi and ZigBee devices.

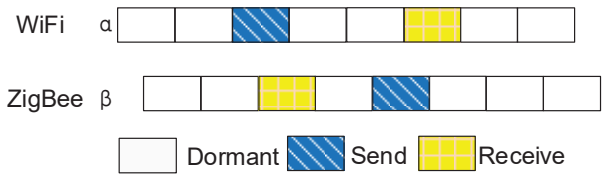


Fig. 2. Time slots for WiFi and ZigBee device.

In the time-slot model, we initially focus on the time-slot division of ZigBee devices. In wireless sensor networks, devices with a duty cycle that regulates energy consumption can only be in two states: active and dormant. In the active state, a device can transmit or receive packets, while in the dormant state, it turns off all its modules except for a timer that wakes it up. Therefore, all devices must regularly alternate between active and dormant states depending on task requirements or preparatory arrangements. In synchronous MAC protocols, each sensor knows the working schedules of other sensors and instructs them to join the neighbours' working-schedule table before establishing communication paths. As a result, for a sensor device, its activities are not random, and its working-schedule statuses are relatively stable, aligning with our DeepSpoof concept that predicts the optimal action in the future slot based on past slot observations.

The assumption that both WiFi and ZigBee share the same time-slot model with identical slot durations is not only practical but also feasible, despite their varying data transmission rates and duty cycles. It is crucial to emphasize that this time-slot model, with its identical slot duration, serves as a simple yet effective means of dividing up communication time into distinct time intervals or slots. Both WiFi and ZigBee employ this model for data transmission, enabling multiple devices to share the same communication channel without causing any interference. In fact, the distinct data transmission rates and duty cycles of WiFi and ZigBee do not prevent their utilization of the same time-slot model. On the contrary, the time-slot model offers a distinct advantage: it enables devices with different data transmission rates and duty cycles to coexist on the same channel without generating any interference.

In practical wireless networks, WiFi and ZigBee devices often exhibit distinct data transmission rates and duty cycles. Despite these inherent differences, it remains both feasible and practical for both types of devices to adopt a unified time-slot model with identical slot durations. This strategic approach effectively divides the available communication time into discrete intervals or slots, which can then be seamlessly utilized by both WiFi and ZigBee for data transmission. While WiFi typically boasts a higher data transmission rate and duty cycle in comparison to ZigBee, the time-slot model ensures the harmonious coexistence of devices with diverse characteristics on a shared communication channel, without any mutual interference. Consequently, the assumption that WiFi and ZigBee can operate within the same time-slot framework is not only intuitive, but also conducive to the efficient and optimized use of the common communication channel.

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2024.3414660

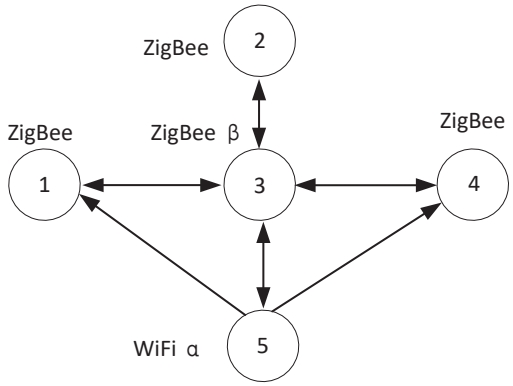


Fig. 3. An example of DeepSpoof basics from a WiFi to ZigBee devices.

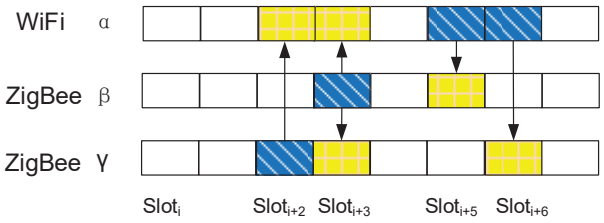


Fig. 4. The WiFi α attempts to capture the ZigBee signal and initiates spoofing attacks.

C. DeepSpoof Basics

Despite WiFi devices' impressive transmission range enabled by their high transmission power, the power of ZigBee signals is insufficient to cause significant interference to WiFi device. Consequently, WiFi device effectively captures only the signals of neighboring ZigBee devices operating within ZigBee frequency band. At the same time, it receives the signal of WiFi α. However, only the signal of ZigBee β can be picked up by WiFi α, as illustrated in Fig. 3. This arrangement allows the WiFi device to monitor the wireless channel and capture a limited signal from a segment of the ZigBee network. By employing deep learning techniques, it can process these captured signals to predict future actions.

As the attacker, the WiFi device is capable of executing three distinct actions in each time slot: Dormant, Receive, or Spoof (Send). When the WiFi device receives instructions to impersonate the ZigBee devices, it initially shifts to a receiving mode and promptly attempts to capture the ZigBee signal. Subsequently, the WiFi device utilizes the observed data to predict the optimal action to take in the subsequent slot. Finally, the WiFi device transitions to a sending state and broadcasts the spoofing signal, as illustrated in Fig. 4. Specifically, if the WiFi device anticipates that the upcoming slot of ZigBee devices will be a receive slot, it takes the spoofing action by sending spoofing signals in this slot. Otherwise, the WiFi device opts for the receiving action to capture the ZigBee signal as much as possible or remains dormant to conserve energy.

It is crucial to acknowledge that in certain instances, the attacker may encounter uncertainty regarding the status of a slot. For instance, if a ZigBee device is in the dormant state instead of the receive state, the attacker may misinterpret the situation and attempt to spoof, leading to a failed attempt as the ZigBee device cannot accept the spoofing signal. In such scenarios, the WiFi device can restore the original signal and evaluate whether its prediction for the completed slot was accurate, thereby enhancing the prediction model. Although the computational capabilities of WiFi devices utilized in these attacks are typically limited, they are wired devices with power cords, and therefore do not face energy constraints. To execute the deep reinforcement learning for spoofing attacks, a system consisting primarily of a cloud computing server cluster and an information service platform is utilized. The WiFi device sends its captured data to the system and receives the results calculated by the system based on the DeepSpoof algorithm.

V. DeepSpoof Design

In this section, we first delve into the fundamental principles of reinforcement learning (RL). We convert DeepSpoof into a typical RL problem and use Q-Learning (QL) to determine the best attack action in the future. Traditional QL obtains the Q function in a table, resulting in a long convergence time. Therefore, we use Deep Q Learning Network (DQN) to handle the large state space. To capture the wireless traffic status over long periods of time, we use an LSTM-based neural network (DNN) to approximate the Q-function. Finally, we analyze the utilization of DeepSpoof in parallel spoofing attacks.

A. Reinforcement Learning

Reinforcement learning involves learning how to maximize rewards in the interaction between an agent and its environment [53]. In this study, we reformulate the DeepSpoof problem as a classic RL problem consisting of a four-tuple $< s_t, a_t, r_t, s_{t+1} >$. At time t, the agent assesses the current state s_t and selects the optimal action to execute. After performing the chosen action, the agent receives a reward r_t and transitions to the next state s_{t+1} through its interaction with the environment. We define the slot states, attacker's actions, and rewards as follows:

1) Slot States: For each slot, the WiFi and ZigBee devices can be in three potential states: Send, receive, or dormant. The WiFi device lacks immediate knowledge of a slot's status and must predict it at the start of the next slot. The optimal action for a given slot depends on the previous slots.

2) Slot States: The WiFi's and ZigBee's actions in slot i are denoted by α_i and β_i, where $\alpha_i, \beta_i \in \{SPOOF/SEND, RECEIVER, DORMANT\}$.

3) Actions: The reward for action α_i is defined based on the channel status of slot i, which is expressed in Eq.1. The reward is $R_x(s_t, a_t)$. If α_i=SPOOF and β_i=RECEIVE, the WiFi receives a reward $R_i = 1$ for successfully spoofing a MAC frame of the victim. If α_i=RECEIVE and β_i=SEND, the WiFi gets a reward $R_j = 1$ for capturing the signal of ZigBee devices.

However, if α_i=SPOOF and β_i=SEND, the WiFi device wastes energy on unnecessary spoofing and receives a negative reward $R_l = -1$. If α_i=RECEIVE and β_i=RECEIVE, the WiFi device misses a successfully transmitted MAC frame of

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2024.3414660

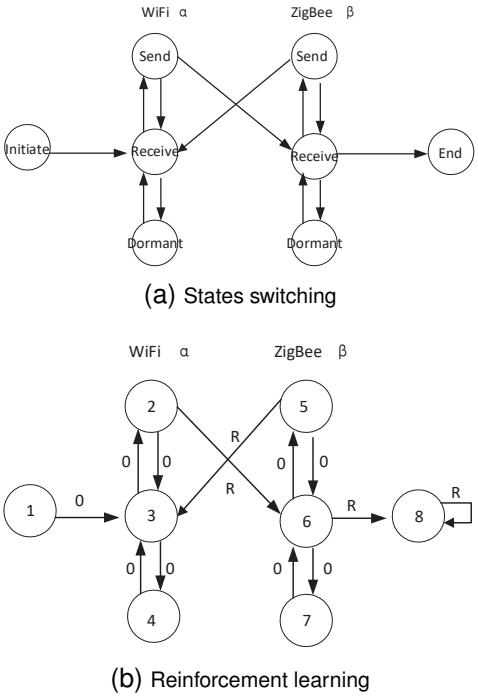


Fig. 5. DeepSpoof test environment with a WiFi and a ZigBee device switching in RECEIVE/SEDN/DORMANT states.

the victim and receives a negative reward $R_m = -1$. Other cases for negative rewards are listed in Eq.1. If the signal is delivered to the receiver, the reward value is $\xi, \xi > 1$.

$$R_x(s_t, a_t) = \begin{cases} \xi \\ +1 \;\; x \in \{i, j, s, t, m, w\} \\ -1 \end{cases} \tag{1}$$

To optimize the spoofing attack, the WiFi device can adjust the values of R_i, R_j, R_s, R_l, R_m, and R_w based on its specific constraints. For instance, as the WiFi device typically has access to a dependable power source, it can maintain its radio receiver in operation for extended periods to capture signals prior to launching the attack. Consequently, it may adopt large absolute values for R_j. The reinforcement learning approach for the spoofing attack is illustrated in Fig. 5, which visualizes how the WiFi and ZigBee devices transition between the RECEIVE/SEND/DORMANT states and employ deep learning techniques to execute the attack.

The WiFi's ability to determine if a prediction is accurate suggests there is some form of data exchange between the WiFi and other devices. To create distinct rewards for various combined actions of WiFi and ZigBee, the authors likely utilized a communication protocol that allowed these devices to share information. One effective approach for WiFi to detect the specific action of a ZigBee device involves the utilization of the ZigFi [37] protocol. This protocol empowers ZigBee devices to communicate not only with each other but also with other devices, including WiFi routers.

When a ZigBee device executes a particular action, it sends a command message to the other ZigBee devices in the network. These messages contain details about the action being taken, such as the type of device and the specific action

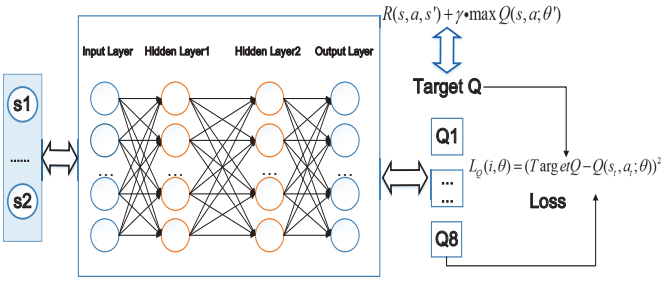


Fig. 6. The architecture of Deep Q-learning for DeepSpoof.

being executed. When a WiFi router receives a command message from a ZigBee device, it can analyze the information enclosed to determine the specific action being implemented. The router then utilizes this information to compute the appropriate reward for that action, based on the combined actions of the WiFi and ZigBee devices. Consequently, WiFi's ability to determine the specific action of a ZigBee device relies on the utilization of a wireless communication protocol, such as ZigFi, that enables devices to exchange information and interact with each other.

B. Deep RL of DeepSpoof

The tabular representation of the Q-function offers a straightforward yet highly effective programming approach for traversing all connections between time-slots and actions, aiding in the identification of optimal actions. While Q-learning excels in relatively simple scenarios, as the number of devices increases and the network complexity escalates, achieving the desired outcome becomes increasingly time-consuming. To address this challenge, we employ deep learning as a function approximator, expediting convergence to the maximum Q-value. This approach is illustrated in Fig. 6.

In our design, we employ a Deep Q-learning network (DQN) in reinforcement learning for DeepSpoof, utilizing the maximum output to determine the optimal actions for spoofing attacks. To address the challenge posed by the extensive state space, we employ the Deep Q-Learning (DQL) technique. To estimate the Q-function, we utilize a specialized Deep Neural Network (DNN). The slot state serves as the input to the DNN, while the Q-values resulting from the SPOOF and WAIT actions in the slot constitute the outputs. Fig. 7 illustrates the configuration of the DeepSpoof DNN architecture, which consists of a single Long Short-Term Memory (LSTM) cell and two Fully Connected (FC) Layers. Additionally, the figure highlights the input and output dimensions for each layer.

The LSTM unit has the ability to capture long-term patterns in wireless traffic. The state of a given time slot s_i incorporates the channel conditions of the preceding slots. Given that earlier slots can significantly impact the optimal action in slot i, we have chosen to utilize the LSTM unit to memorize historical slot information without increasing the dimensionality of the slot state. The LSTM unit consists of four gates that maintain a hidden state and compute the output. The hidden state is influenced by the extensive input history and is continually updated based on the most recent input.

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2024.3414660

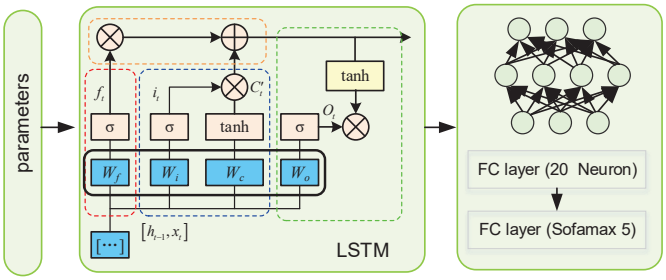


Fig. 7. DeepSpoof DNN architecture based on LSTM.

The four gates control how the hidden state is influenced by the most recent input and how the output is affected by the hidden state. As a result, the LSTM unit preserves the long-term historical information of wireless traffic within its hidden state, aiding in the prediction of optimal actions. The output of the LSTM unit is fed into two fully connected layers, with 20 and 5 neurons, utilizing the Rectified Linear Unit (ReLU) as the activation function. The final output layer is a linear fully connected layer that generates the Q-values for SPOOF and WAIT actions. The input to the DQN is the slot states of all sensor devices, and the outputs are the Q-values, indicating which action sensors should take: SPOOF, RECEIVE or DORMANT. The target or actual output value must be continually calculated and updated until it achieves a maximum output based on the Bellman equation. Thus,

$$Q(s_t,a_t) = Q(s_t,a_t) + \alpha[R_{t+1} + \gamma \cdot \max_{\tilde{a}}\{Q(\tilde{s},\tilde{a})\} - Q(s_t,a_t)] \quad (2)$$

Eq. 2 represents the iterative process of Q learning. $Q(s_t,a_t)$ on the left side of the formula is the new Q value after taking action at in state s_t. $Q(s_t,a_t)$ on the right side is the original Q value. R_{t+1} is the reward obtained after executing the action a_t. $\max_{\tilde{a}}\{Q(\tilde{s},\tilde{a})\}$ represents the maximum Q value of taking action $\tilde{a}$ in the next state $\tilde{s}$. Since the sequence of experiences generated by the interaction between the agent and the environment has a high degree of temporal correlation, and using the same agent network to generate the target Q value for the following state and update the current state Q value simultaneously can easily lead to network instability and non-convergence. Based on the DQN method, we first establish an experience replay pool, and store each Markov decision process at each time step as an experience to update this replay pool. This process combines past and current knowledge, reducing sample differences and ensuring that the training samples can be fully utilized.

During training, only a certain amount of experience is randomly selected from the experience replay pool as samples. This approach effectively reduces data correlation, enhancing learning efficiency by reusing experiences. Additionally, a target network $Q(s_t,a_t;\theta)$ is introduced, which is identical to the agent network, for estimating target Q values. The parameter θ of the target network is updated every certain number of steps, $\theta^{'}$ represents the updated parameter. This allows the Q values in the training process to be temporarily fixed, making the agent learning process more stable. The final agent network is trained by minimizing the loss function. The formula for calculating the loss function is:

$$L_Q(i,\theta) = (R_{t+1} + \gamma \cdot \max_{\tilde{a}}\{Q(\tilde{s},\tilde{a};\theta^{'})\} - Q(s_t,a_t;\theta))^2 \quad (3)$$

To prevent the agent from getting trapped in local minima, we employ the ϵ greedy strategy during the training process. This strategy allows the agent to explore its environment while also exploiting its current knowledge. The value of ϵ starts at 0.1 and is gradually decreased by 0.1 at each iteration until it reaches 0.0001. The calculation formula for ϵ is as follows:

$$a|s = \begin{cases} argmax \quad Q(s,a) & with \quad probability \quad 1-\epsilon \\ any \quad action \quad a & with \quad probability \quad \epsilon \end{cases} \quad (4)$$

where, $argmax \quad Q(s,a)$ represents the maximum output of the function $Q(s,a)$. We then calculate this loss and employ backpropagation, or stochastic gradient descent, to pass it through the network and update the weights accordingly. The algorithm is illustrated in the Algorithm 1.

Algorithm 1: Deep Q-Learning for DeepSpoof

input : Initial $Q(s,a)$,R, replay buffer D
output: Max Q-value
for *episode* $\leftarrow 1$ **to** M **do**
  Initialize s;
  for $t \leftarrow 1$ **to** T **do**
    with probability ϵ select a random action a_t;
    execute action a_t in emulator and observe reward r_t;
    set s_{t+1}=s_t;
    store transition $< s_t, a_t, r_t, s_{t+1} >$ in D;
    sample random minibath of transitions $< s_t, a_t, r_t, s_{t+1} >$ from D;
    if *next station is terminal* **then**
      Target Q = $R(s,a,s')$;
    else
      Update critic by minimizing the loss:;
      $L_Q(i,\theta)$= $(R_{t+1}$+ $\gamma \cdot \max_{\tilde{a}}\{Q(\tilde{s},\tilde{a};\theta^{'})\}$ - $Q(s_t,a_t;\theta))^2$;
      Update the target network:;
      $\alpha[R_{t+1} + \gamma \cdot \max_{\tilde{a}}\{Q(\tilde{s},\tilde{a})\} - Q(s_t,a_t)]$;
    end
  end
end

C. DeepSpoof for Parallel Spoofing Attack

We conduct an analysis of the 802.11 a/g OFDM PHY architecture. Both IEEE 802.11a/g and HIPERLAN/2 signals are characterized as pulsed (or burst) type signals. These signals occupy a total channel bandwidth of 20 MHz, with an effective bandwidth utilization of 16.6 MHz. Each OFDM symbol comprises 52 subcarriers, including 48 data subcarriers

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2024.3414660

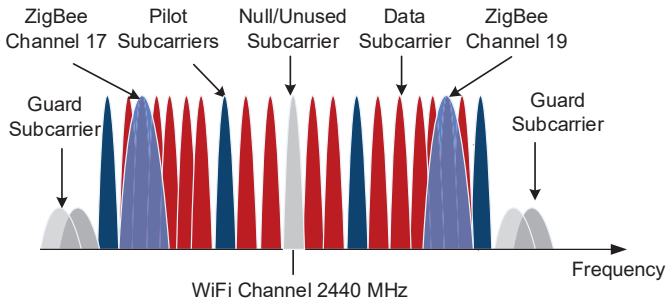


Fig. 8. Channels Mapping for Pilot Avoidance, where only channel 17, 19 can be spoofed by a WiFi signal.

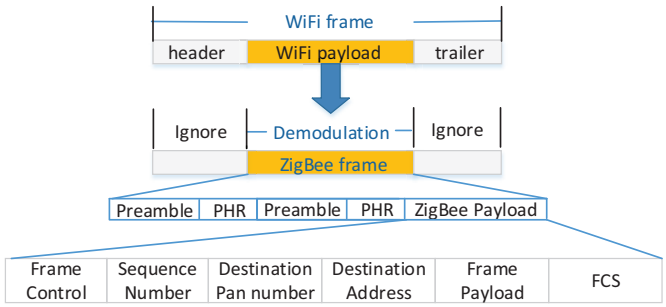


Fig. 9. WiFi frame format.

and 4 pilot subcarriers. It is noteworthy that without hardware modifications, the software alone cannot manipulate WiFi signals transmitted on the pilot, null, or unused subcarriers. As a result, when ZigBee devices operate within frequency bands overlapping with these pilot subcarriers, their proper functionality cannot be guaranteed.

To mitigate interference with these critical subcarriers in WiFi OFDM, DeepSpoof introduces a channel mapping technique. This approach is visualized in the channel mapping scheme depicted in Fig. 8. As an example, when the central frequency of a channel is set to 2440 MHz, DeepSpoof enables two parallel cross-technology communication streams with standard ZigBee channels 17 and 19. This is achieved by utilizing the WiFi OFDM data subcarriers within the ranges [-21, -7] and [7, 21]. By configuring the center frequency of a WiFi device to 2440 MHz, it can simultaneously transmit packets to ZigBee devices operating in channels 17 and 19, leveraging cross-technology communication techniques.

Importantly, many commodity WiFi radios, such as the Atheros AR9485, AR5112, and AR2425, offer the flexibility to adjust their central frequencies. This capability facilitates the implementation of this approach on real-world hardware equipment. Consequently, a WiFi device equipped with this functionality can spoof communications in two independent channels concurrently.

VI. Performance Evaluation

A. System Setting for WiFi and ZigBee Networks

Our DeepSpoof prototype leverages the USRP-N210 platform equipped with 802.11 b/g PHY, as cited in [54], to serve as the spoofer. Meanwhile, a MICAz receiver featuring 802.15.4 PHY, referenced in [55], assumes the role of the victim. The fundamental physical message spans a length of 25 bytes, encompassing the following components: a 4-byte preamble (00000000), a 1-byte SFD (7A), a 1-byte PHR, another set of 4-byte preamble, a 1-byte SFD, a 1-byte PHR, a 7-byte MAC header (consisting of 2-byte Frame Control, 1-byte Sequence number, 2-byte Destination Pan Number, and 2-byte Destination Address), a 4-byte payload, and a 2-byte CRC. Fig. 9 illustrates the WiFi frame structure, where PHR signifies the length of the MAC frame, set to 11001 (representing 25 bytes) in our experimental setup. Since beacons originate from WiFi devices, the MPDU of the MAC frame omits the source address and source pan number. Instead, the destination address is set to 0xFFFF, indicating a broadcast to all ZigBee devices. Furthermore, the MAC payloads incorporate a 64-bit timestamp encoding.

Fig. 10 illustrates the actual experimental setup we conducted to evaluate our proposed system. This experiment involved 20 MICAz nodes and a USRP N210 device equipped with 802.11g PHY, all deployed within an office building spanning approximately 150m x 100m. The wireless environment was highly complex. The USRP-N210 platform, along with its 802.11 b/g PHY, was employed as a WiFi access point (AP) for simulation purposes. The USRP-N210 devices were solely used for evaluation, enabling us to measure low-level PHY information. The packets in this study were encoded using the CTC technique, as detailed in [56]. Table I summarizes the simulation parameters we utilized in our experiments.

In our simulations, the USRP-N210 device was only used for evaluation purposes, but it could be replaced with a commercial WiFi card, such as the Atheros AR2425, in a real-world scenario. We assess our technique using both synthetic and real measurements to demonstrate its practicality across various datasets. Table I lists the parameter values utilized in our simulations, including the energy consumption required for running the circuitry and the energy per bit required for transmission, , denoted as ε_{elec} and ε_{fs} or ε_{amp}, respectively. We set ε_{elec} to be 50 nJ and ε_{fs} or ε_{amp} to be 10 pJ/bit/m^2 or 0.0013 pJ/bit/m^4 for the transmitting amplifier.

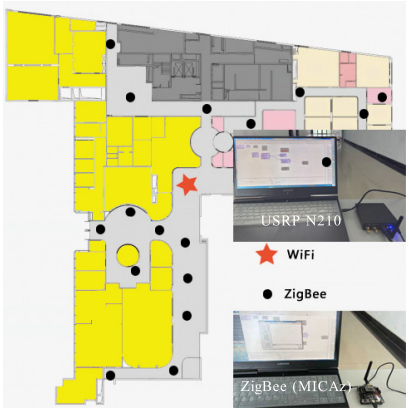


Fig. 10. Experiment setting.

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2024.3414660

TABLE I
SIMULATION PARAMETERS

Parameter Name	Value
Network area	$150 \times 100\ meter^2$
Number of sensor nodes	20
Beacon size	25 $bytes$
E_{elec}	50 nJ/bit
ε_{fs}	10 $pJ/bit/m^2$
ε_{amp}	0.0013 $pJ/bit/m^4$
d for ZigBee	50 $meters$
d for WiFi	300 $meters$
MAC protocol	BoX-MAC [57]

TABLE II
EXPERIMENTAL PARAMETER CONFIGURATION

Parameter Name	Value
Training steps	300000
Learning rate	0.001
λ	0.9
ϵ	0.01
Target network parameter update rate	0.0001
Experience playback pool unit size	300000

B. Evaluation Setup for Hardware and Software

We implemented DeepSpoof using TensorFlow 2.4. Tensorflow is a general deep learning framework that can effectively process complex models, but may require a certain amount of memory and processing power. To ensure minimal assumptions regarding the attacker's capabilities, all experiments were conducted on a commercial off-the-shelf (COTS) personal computer featuring an Intel Core i7-3770 3.4 GHz CPU, where all computations took place. For storage resources, the experience replay pool needs to store a large number of state-action-reward transfer records, which has certain requirements for storage resources. This article uses a memory size of 32GB. However, as the attacker, WiFi can easily connect to a remote cloud computing center, and the demand for computing complexity and computing resources can be achieved.

During the training process of the Deep Q-Network (DQN), the neural network utilized the Adam optimizer and the Rectified Linear Unit (ReLU) activation function. he computational complexity of DeepSpoof is mainly reflected in the use of neural networks to calculate the Q value. The state space size is S, the action space size is A, and B samples are drawn from the experience pool for training to obtain the maximum Q value. Considering that the total number of iterative updates is N and the total number of neural network parameters is P, the final computational complexity is $O(N \cdot S \cdot A \cdot B \cdot P)$. Key parameters involved in this process included the number of algorithm training steps, learning rate, target network parameter update rate, and the size of the experience replay pool. The specific configuration details are outlined in Table II.

This study aims to develop a method for promptly detecting and recognizing deception attack signals. To achieve this, we trained the average reward values to reflect the level of convergence during training. We used different learning rates to obtain additional average reward values. As shown in Fig. 11,

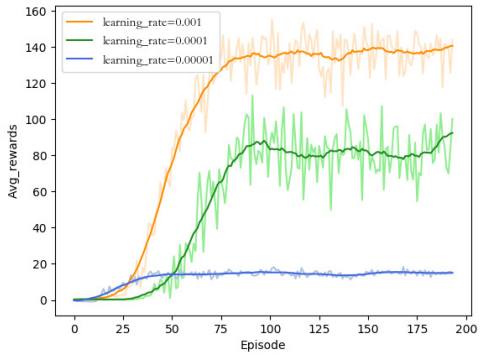


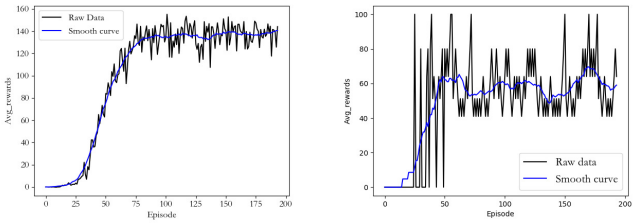
Fig. 11. Convergence of the model under different learning rates.

we examined the convergence of the deep reinforcement learning model using learning rates of 0.001, 0.0001, and 0.00001, respectively. The observed convergence indicates that the DQN algorithm is capable of detecting spoofing attack signals from WiFi devices. When a spoofing attack signal is detected, ZigBee captures the signal through alternative channels and relays it to the receiver. If the receiver successfully receives the signal, the model is awarded a reward. Higher reward values indicate greater robustness in detecting spoofing attack signals.

The broken line in the figure represents the actual reward values obtained during each learning iteration, while the curve represents the smoothed data. Fig. 11 demonstrates that distinct learning rates correlate with varying reward trajectories. Specifically, with a learning rate of 0.001, the reward value stabilizes at its peak and tends to converge after approximately 75 generations. Conversely, a learning rate of 0.00001 is too low, resulting in the smallest reward values obtained. In the case of a learning rate of 0.0001, the reward values fluctuate within the range of 60 to 100 and converge after nearly 100 generations. Therefore, it takes approximately 100 steps for a learning rate of 0.0001 to achieve convergence.

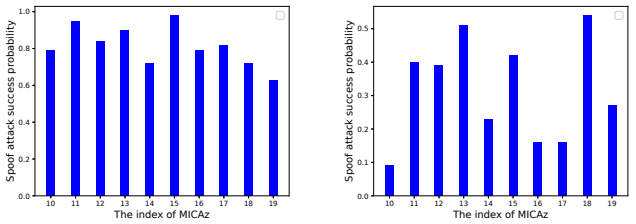
In deep reinforcement learning, the average training reward is commonly used as an indicator of convergence during training. Fig. 12 illustrates the convergence of average rewards in both reinforcement learning and deep reinforcement learning throughout the training process. As shown in Fig. 12(a), the rewards obtained by the DQN model remained relatively stable during the initial 80 training iterations. However, between 80 and 100 iterations, the model underwent a phase of intense exploration and learning, leading to a significant surge in rewards. After 100 iterations, the cumulative reward per training episode stabilized, while the reward per step gradually decreased. Despite the occurrence of jumps in the cumulative reward of each trial, these jumps indicate that the model is indeed converging. Fig. 12(b) reveals that the DL model exhibits a substantial range of reward fluctuations. Although a convergence trend is visible in its smoothed curve, it is relatively minor and the reward values remain low. In contrast, the DQN model demonstrates superior performance when compared to deep reinforcement learning.

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2024.3414660



(a) Reward curve of DQN (b) Reward curve of DL

Fig. 12. Average reward curve for DQN and DL.



(a) From WiFi within 50 m (b) From WiFi beyond 50 m

Fig. 13. The spoofing attack success probability for WiFi device to spoof 20 ZigBee devices.

C. Experimental Results

We conducted a comprehensive experimental evaluation, encompassing 100 rounds, to assess the success probability of spoofing attacks. Fig. 13 illustrates the spoofing attack success probability when the average duty-cycle of ZigBee nodes is set to 10%. Our observations revealed that, with 10 ZigBee devices within a 50m radius of the WiFi device, the average attack success probability was approximately 81.8%. There are some inherent flaws in CTC, so there are still cases where spoofing attacks fail.

There are signal simulation errors in the implementation of cross-technology communication from WiFi to ZigBee [43]. On the one hand, the error comes from WiFi cyclic prefix (CP), which is a technology to eliminate inter-symbol interference (ISI), that is, there will be a guard interval lasting 0.8us in each WiFi symbol that is copied from the right side of WiFi and covers the symbol on the left, so that the front end and back end of the WiFi signal are the same, however the ZigBee I/Q signal has no such duplication. On the other hand, the duration of ZigBee symbol is 16us, while the duration of WiFi Symbol is 4us. Therefore, a ZigBee symbol needs to be divided into four parts for simulation respectively. This division will further increase the simulation error. These are all uncontrollable in signal simulation, which results in the preamble sent by WiFi devices not being 100% recognized by ZigBee devices, further reducing the success rate of spoofing attacks.

Nowadays, WiFi device energy usually has a stable power supply, and the energy budget can be considered to be unlimited. Therefore, WiFi devices can continue to attack without interruption, posing a great threat to ZigBee secure communication. However, this probability decreased significantly to around 38.1% when the other 10 ZigBee devices were

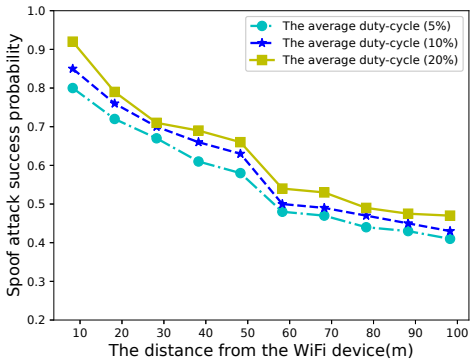


Fig. 14. The spoofing attack success probability with the distance to WiFi device improved.

positioned beyond the 50m range of the WiFi device. This analysis underscores that the WiFi device can only capture signals from nearby ZigBee devices within its transmission range. Consequently, direct analysis of information from the 10 neighboring ZigBee devices by the WiFi device enhances the spoofing attack success probability. In contrast, predicting the behaviors of the other 10 ZigBee nodes solely based on received packets yielded relatively poorer attack performance.

Fig. 14 presents the average success probability of the spoofing attack considering different positions of ZigBee devices. The results indicate that the average success probability of the attack increased with the average duty-cycle, reaching approximately 83.6%, 88.5%, and 90.7% for duty-cycles of 5%, 10%, and 15%, respectively. These findings highlight that ZigBee devices located further away from the WiFi device have a lower probability of being spoofed, as their communication remains unaffected by the WiFi device's eavesdropping or interference capabilities. Specifically, ZigBee devices beyond the transmission radius of the WiFi device are immune to its spoofing attempts. Additionally, while DeepSpoof exhibits robust performance in both busy and idle WiFi networks, it may encounter challenges in low-duty-cycle networks where training samples, particularly slots containing fraudulent transmissions, are limited. This scarcity can lead to prolonged convergence times and subsequently inferior performance.

D. Spoofing Attack Comparison

To assess the effectiveness of DeepSpoof, we conducted a comparative analysis of its spoofing attack success probabilities with those of traditional spoofing attacks, as described in Gao and Ning's study [58]. The success probabilities of WiFi devices executing spoofing attacks are influenced by the proximity between the WiFi and ZigBee devices. Fig. 15 illustrates the average success probabilities in spoofing attacks across varying distances between ZigBee devices. It's worth noting that the maximum transmission power of the WiFi device can reach up to 100 mW (-20 dBm), enabling a transmission range of nearly 300 m. Conversely, the ZigBee device (specifically, the MICAz model) has a maximum transmission power of 1 mW (0 dBm) and a transmission range of less than 70 m.

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2024.3414660

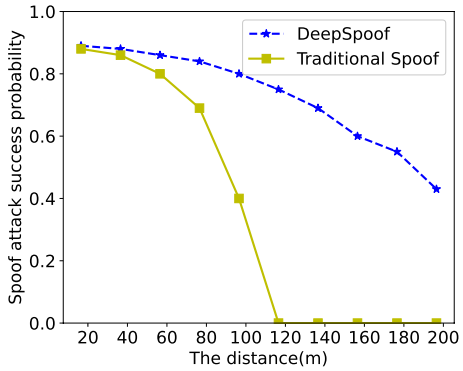


Fig. 15. The average success probabilities of DeepSpoof and traditional spoofing methods in scenarios where WiFi and ZigBee devices are located at varying distances from each other.

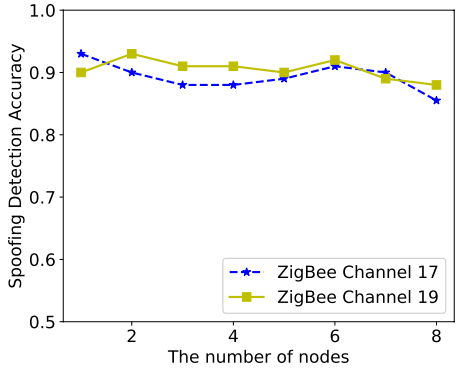


Fig. 16. The average spoofing success probabilities of WiFi for 10 minutes.

Given the significant disparity in transmission ranges between WiFi and ZigBee, a single WiFi device can scan a region with multiple ZigBee nodes and execute a spoofing attack on numerous sensors simultaneously.

In traditional wireless sensor networks that lack CTC technology, spoofer attacks are typically launched by exploiting a captured common ZigBee node to deceive sensors. However, the limited transmission range of the ZigBee device necessitates attackers to approach it as closely as possible. This close proximity poses a greater risk of exposure for the attacker and may not be feasible in certain applications where physical access to the sensor network is restricted or impractical. These constraints highlight the need for innovative spoofer attack techniques that overcome distance limitations and provide increased efficiency.

E. Parallel Spoofing Attack

Fig. 16 illustrates the probabilities of successfully spoofing ZigBee devices using WiFi signals, considering identity numbers ranging from 1 to 8. The figure also shows a 5% duty cycle between ZigBee channels 17 and 19. Our analysis of this figure reveals an average probability of approximately 90% for successful spoofing, with a slightly lower probability observed on channel 19 compared to other channels. This variation can be attributed to the differences in channel quality across different frequencies. Additionally, as WiFi signals occupy channels 16-20, ZigBee devices operating on channels 16, 18, and 20 are unable to recognize or decode the received signals, resulting in them being interpreted as noise. It's worth noting that WiFi devices have sufficient energy reserves and can operate with a 100% duty cycle, making them suitable candidates for executing spoofing attacks against ZigBee devices.

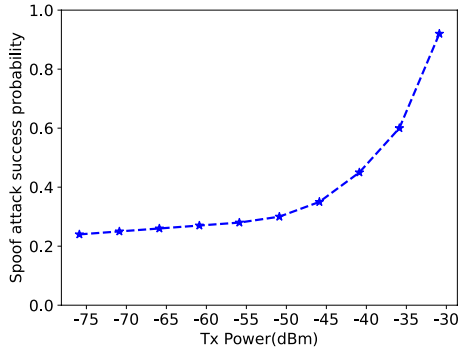


Fig. 17. Impact of Tx power setting for the spoofing attack success probability.

F. WiFi Adjust TX Power Reference

We present an assessment of the attack efficiency of WiFi devices across varying transmit power levels. Typically, attackers strive for a high sending power and a raised reference power to ensure a successful attack. The received signal strength (RSS) is a metric that offers insight into the relative strength and quality of the WiFi signal. It's crucial to note that RSS readings obtained at distinct physical locations are unique and are influenced by various factors, including random noise, environmental factors, and multipath effects. Consequently, RSS readings exhibit strong spatial correlation characteristics. The RSS value measured by a ZigBee device from a WiFi access point (AP) can be mathematically expressed as:

$$P(d)[dBm] = p(d_0)[dBm] - 10\lambda \log_{10}(d) \tag{5}$$

where $p(d_0)$ represents the received signal strength in dBm at a distance of d_0 meters, with d_0 typically set to 1 meter. The variable d represents the distance between the transmitter and receiver, while λ represents the propagation constant or path-loss exponent.

To maximize the number of affected ZigBee devices, the WiFi access point (AP) can enhance its coverage by increasing the transmit power. We conducted a study to assess the impact of transmit power on the likelihood of successful spoofing attacks. We calculated the average probability of success across various transmit (TX) power levels. Fig. 17 clearly illustrates that the probability of successfully executing spoofing attacks increases exponentially with increasing TX power. It's worth noting that when the reference power is set to a high value, such as -40, the probability of success significantly improves. Conversely, with a lower reference power, such as -75, the probability of success remains consistently low. Our analysis indicates that a higher reference power is preferred, but the attacker must carefully balance the potential benefits against the risk of detection. Setting the reference power too high can lead to noticeable disruptions and raise the chances of being detected. Fig. 17 supports our observations.

VII. Conclusion

In this study, our primary focus is on the issue of spoofing attacks in the heterogeneous Internet of Multimedia Things. We introduce a novel approach called DeepSpoof, which employs deep learning techniques based on LSTM and the CTC method to address this challenge in complex wireless environments. Initially, we provide a comprehensive overview of the background and theory behind DeepSpoof. Subsequently, we present a time-slot model tailored for ZigBee and WiFi devices, along with the fundamental principles of DeepSpoof. Then, we implement an LSTM-based Deep Q-Learning approach to handle the vast state space and approximate the Q-function for executing spoofing attacks. Finally, we validate our entire methodology by simulating advanced malicious spoofing attacks on a legitimate dataset.

References

[1] X. Wei, Y. Shi, and L. Zhou, "Haptic Signal Reconstruction for Cross-Modal Communications," *IEEE Transactions on Multimedia*, vol. 24, pp. 4514–4525, 2022.

[2] X. Wei, Y. Yao, H. Wang, and L. Zhou, "Perception-Aware Cross-Modal Signal Reconstruction: From Audio-Haptic to Visual," *IEEE Transactions on Multimedia*, vol. 25, pp. 5527–5538, 2023.

[3] C. M. Lentisco, L. Bellido, A. Cárdenas, R. F. Moyano, and D. Fernández, "Design of a 5G Multimedia Broadcast Application Function Supporting Adaptive Error Recovery," *IEEE Transactions on Multimedia*, vol. 25, pp. 378–388, 2023.

[4] L. Wang, J. Zhang, J. Chuan, R. Ma, and A. Fei, "Edge Intelligence for Mission Cognitive Wireless Emergency Networks," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 103–109, 2020.

[5] Y. Feng, F. Chen, J. Yu, Y. Ji, F. Wu, T. Liu, S. Liu, X.-Y. Jing, and J. Luo, "Cross-Modality Spatial-Temporal Transformer for Video-Based Visible-Infrared Person Re-Identification," *IEEE Transactions on Multimedia*, pp. 1–13, 2024.

[6] J. Guo, X. Ma, A. Sansom, M. McGuire, A. Kalaani, Q. Chen, S. Tang, Q. Yang, and S. Fu, "Spanet: Spatial Pyramid Attention Network for Enhanced Image Recognition," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, 2020, pp. 1–6.

[7] Y. Li, Y.-S. Jeong, B.-S. Shin, and J. H. Park, "Crowdsensing Multimedia Data: Security and Privacy Issues," *IEEE MultiMedia*, vol. 24, no. 4, pp. 58–66, 2017.

[8] X. Liu, Y. Lin, Q. Yang, and H. Fan, "Transferable Adversarial Attack on 3D Object Tracking in Point Cloud," in *the 29th International Conference on MultiMedia Modeling (MMM)*, 2023, pp. 445–458.

[9] P. V. B. Bayari, A. Lakshman, G. Bhatnagar, and C. Chattopadhyay, "A Novel Security Framework for Medical Data in IoT Ecosystems," *IEEE MultiMedia*, vol. 29, no. 2, pp. 34–44, 2022.

[10] N. Xie, Q. Zhang, Y. Chen, J. Hu, G. Luo, and C. Chen, "Low-Cost Anti-Copying 2D Barcode by Exploiting Channel Noise Characteristics," *IEEE Transactions on Multimedia*, vol. 23, pp. 3752–3767, 2021.

[11] W. Liu, D. Sun, Y. Wang, Z. Chen, X. Han, and H. Yang, "ABTD-Net: Autonomous Baggage Threat Detection Networks for X-ray Images," in *2023 IEEE International Conference on Multimedia and Expo (ICME)*, 2023, pp. 1229–1234.

[12] M. R. Nosouhi, K. Sood, M. Grobler, and R. Doss, "Towards Spoofing Resistant Next Generation IoT Networks," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1669–1683, 2022.

[13] X. Huan, K. S. Kim, and J. Zhang, "NISA: Node Identification and Spoofing Attack Detection Based on Clock Features and Radio Information for Wireless Sensor Networks," *IEEE Transactions on Communications*, vol. 69, no. 7, pp. 4691–4703, 2021.

[14] S. Dinh-Van, T. M. Hoang, B. B. Cebecioglu, D. S. Fowler, Y. K. Mo, and M. D. Higgins, "A Defensive Strategy Against Beam Training Attack in 5G mmWave Networks for Manufacturing," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 2204–2217, 2023.

[15] 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 Transactions on Information Forensics and Security*, vol. 17, pp. 1010–1023, 2022.

[16] J. Yan, D. Wu, H. Wang, and R. Wang, "Multipoint Cooperative Transmission for Virtual Reality in 5G New Radio," *IEEE MultiMedia*, vol. 26, no. 1, pp. 51–58, 2019.

[17] D. Wu, Q. Liu, H. Wang, Q. Yang, and R. Wang, "Cache Less for More: Exploiting Cooperative Video Caching and Delivery in D2D Communications," *IEEE Transactions on Multimedia*, vol. 21, no. 7, pp. 1788–1798, 2019.

[18] "Defense Techniques Against Spoofing Attacks in Wireless Sensor Networks," *Materials Today: Proceedings*, 2023.

[19] M. H. Yilmaz and H. Arslan, "A Survey: Spoofing Attacks in Physical Layer Security," in *Local Computer Networks Conference Workshops*, 2015.

[20] R. Duan, X. Mao, A. K. Qin, Y. Chen, S. Ye, Y. He, and Y. Yang, "Adversarial Laser Beam: Effective Physical-World Attack to DNNs in a Blink," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 062–16 071.

[21] F. Xiao, Y. Huang, Y. Zuo, W. Kuang, and W. Wang, "Over-the-Air Adversarial Attacks on Deep Learning Wi-Fi Fingerprinting," *IEEE Internet of Things Journal*, 2023.

[22] J. Kotak and Y. Elovici, "Adversarial Attacks Against IoT Identification Systems," *IEEE Internet of Things Journal*, vol. 10, no. 9, pp. 7868–7883, 2023.

[23] P. Sun, S. Li, J. Xie, H. Xu, Z. Cheng, and R. Yang, "GPMT: Generating Practical Malicious Traffic based on Adversarial Attacks with Little Prior Knowledge," *Computers & Security*, vol. 130, p. 103257, 2023.

[24] W. Yin, P. Hu, J. Wen, and H. Zhou, "ACK Spoofing on MAC-layer Rate Control: Attacks and Defenses," *Computer Networks*, vol. 171, p. 107133, 2020.

[25] S. Tomic and M. Beko, "Detecting Distance-Spoofing Attacks in Arbitrarily-Deployed Wireless Networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 4383–4395, 2022.

[26] K. Mahmood, W. Akram, A. Shafiq, I. Altaf, M. A. Lodhi, and S. H. Islam, "An Enhanced and Provably Secure Multi-Factor Authentication Scheme for Internet-of-Multimedia-Things Environments," *Computers & Electrical Engineering*, vol. 88, p. 106888, 2020.

[27] S. M. Ali, A. S. Elameer, and M. M. Jaber, "IoT Network Security using Autoencoder Deep Neural Network and Channel Access Algorithm," *Journal of Intelligent Systems*, vol. 31, no. 1, pp. 95–103, 2021.

[28] F. Yang, B. Wen, C. Comaniciu, K. P. Subbalakshmi, and R. Chandramouli, "TONet: A Fast and Efficient Method for Traffic Obfuscation Using Adversarial Machine Learning," *IEEE Communications Letters*, vol. 26, no. 11, pp. 2537–2541.

[29] P. Madani and N. Vlajic, "RSSI-Based MAC-Layer Spoofing Detection: Deep Learning Approach," *Multidisciplinary Digital Publishing Institute*, no. 3, 2021.

[30] J. Huang, B. Liu, C. Miao, X. Zhang, J. Liu, L. Su, Z. Liu, and Y. Gu, "PhyFinAtt: An Undetectable Attack Framework Against PHY Layer Fingerprint-based WiFi Authentication," *IEEE Transactions on Mobile Computing*, pp. 1–18, 2023.

[31] A. P. R. R. A. D. K. Gopalakrishnan Subburayalu, Hemanand Duraivelu and C. Thangavel, "Cluster Based Malicious Node Detection System for Mobile Ad-Hoc Network Using ANFIS Classifier," *Journal of Applied Security Research*, vol. 18, no. 3, pp. 402–420, 2023.

[32] H. Shen, J. Xu, T. Wang, and G. Bai, "Detecting Link Correlation Spoofing Attack: A Beacon-Trap Approach," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019.

[33] K. Chebrolu and A. Dhekne, "Esense: Communication through Energy Sensing," in *International Conference on Mobile Computing & Networking*, 2009.

[34] S. Wang, Z. Yin, Y. Chen, Z. Li, and T. He, "Networking Support For Bidirectional Cross-Technology Communication," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2019.

[35] S. M. Kim and T. He, "FreeBee: Cross-Technology Communication via Free Side-channel," in *International Conference on Mobile Computing & Networking*, 2015.

[36] Y. Zhang and Q. Li, "HoWiES: A Holistic Approach to ZigBee Assisted WiFi Energy Savings in Mobile Devices," in *Infocom, IEEE*, 2013.

[37] X. Guo, Y. He, X. Zheng, L. Yu, and O. Gnawali, "ZigFi: Harnessing Channel State Information for Cross-Technology Communication," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1, pp. 301–311, 2020.

[38] Z. Li and T. He, "Webee: Physical-Layer Cross-Technology Communication via Emulation," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, 2017, pp. 2–14.

[39] W. Jiang, Z. Yin, R. Liu, Z. Li, S. M. Kim, and T. He, "BlueBee: a 10,000x Faster Cross-Technology Communication via PHY Emulation," in *Proceedings of the 15th ACM Conference on Embedded Network*

Sensor Systems, ser. SenSys '17. New York, NY, USA: Association for Computing Machinery, 2017.
[40] W. Jiang, S. M. Kim, Z. Li, and T. He, "Achieving Receiver-Side Cross-Technology Communication with Cross-Decoding," in *the 24th Annual International Conference*, 2018.
[41] S. Wang, S. M. Kim, and T. He, "Symbol-level cross-technology communication via payload encoding," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2018, pp. 500–510.
[42] S. Wang, W. Jeong, J. Jung, and S. M. Kim, "X-mimo: Cross-technology multi-user mimo," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 218–231.
[43] T. Cheng, S. Li, F. Jiao, and Y. Pan, "WibZig: Reliable and Commodity-device Compatible PHY-CTC via Chip Emulation in Phase," in *Proceedings of the 22nd International Conference on Information Processing in Sensor Networks*, 2023, pp. 191–204.
[44] S. Wang, J. Guo, P. Wang, K. Parsons, P. Orlik, Y. Nagai, T. Sumi, and P. Pathak, "X-disco: Cross-technology neighbor discovery," in *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2022, pp. 163–171.
[45] S. Xia, Y. Chen, M. Li, and P. Chen, "A Survey of Cross-Technology Communication for IoT Heterogeneous Devices," *IET Communications*, vol. 13, no. 12, pp. 1709–1720, 2019.
[46] B. Lu, Z. Qin, M. Yang, X. Xia, and L. Wang, "Spoofing Attack Detection using Physical Layer Information in Cross-Technology Communication," in *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2018.
[47] R. Xu, G. Shi, J. Luo, Z. Zhao, and Y. Shu, "Muzi: Multi-Channel Zigbee Networks for Avoiding WiFi Interference," in *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*. IEEE, 2011, pp. 323–329.
[48] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, "Denial of Service Attacks in Wireless Networks: The Case of Jammers," *IEEE Communications surveys & tutorials*, vol. 13, no. 2, pp. 245–257, 2010.
[49] D. Gao, S. Wang, Y. Liu, W. Jiang, Z. Li, and T. He, "Spoofing-Jamming Attack based on Cross-Technology Communication for Wireless Networks," *Computer Communications*, vol. 177, pp. 86–95, 2021.
[50] S. Wang, Z. Yin, S. Wang, Z. Li, Y. Chen, S. M. Kim, and T. He, "Networking Support for Bidirectional Cross-Technology Communication," *IEEE Transactions on Mobile Computing*, vol. 20, no. 1, pp. 204–216, 2021.
[51] D. Han, A. Li, L. Zhang, Y. Zhang, J. Li, T. Li, T. Zhu, and Y. Zhang, "Deep Learning-Guided Jamming for Cross-Technology Wireless Networks: Attack and Defense," *IEEE/ACM Transactions on Networking*, vol. 29, no. 5, pp. 1922–1932, 2021.
[52] L. Bracciale, P. Loreti, and G. Bianchi, "Human Time-Scale Duty Cycle for Opportunistic WiFi based Mobile Networks," in *2013 24th Tyrrhenian International Workshop on Digital Communications - Green ICT (TIWDC)*, 2013, pp. 1–6.
[53] T. Huang, R.-X. Zhang, and L. Sun, "Zwei: A Self-Play Reinforcement Learning Framework for Video Transmission Services," *IEEE Transactions on Multimedia*, vol. 24, pp. 1350–1365, 2022.
[54] B. Bloessl, M. Segata, C. Sommer, and F. Dressler, "An IEEE 802.11a/g/p OFDM Receiver for GNU Radio," in *Proceedings of the Second Workshop on Software Radio Implementation Forum*, ser. SRIF '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 9–16.
[55] T. Schmid, "GNU Radio 802.15.4 En- and Decoding," 2006.
[56] Y. Chen, S. Wang, Z. Li, and T. He, "Reliable Physical-Layer Cross-Technology Communication With Emulation Error Correction," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 612–624, 2020.
[57] D. Moss and P. Levis, "BoX-MACs: Exploiting Physical and Link Layer Boundaries in LowPower Networking," 2008.
[58] N. Gao, Q. Ni, D. Feng, X. Jing, and Y. Cao, "Physical Layer Authentication Under Intelligent Spoofing in Wireless Sensor Networks," *Signal Processing*, vol. 166, p. 107272, 2019.

Demin Gao received the Ph.D. degree from Nanjing University of Science and Technology in 2012. He was a joint Ph.D. student and attended the research lab of Kwan-Wu Chin with University of Wollongong. He joined with Nanjing Forestry University, as a Lecturer and an Associate Professor in 2012 and 2016. He was a Visiting Scholar with University of Minnesota Twin Cities. His research interests include WSN and energy harvesting.

Liyuan Ou is studying for a master's degree at the School of Information Science and Technology of Nanjing Forestry University. He received a bachelor's degree from Nanjing University of Science and Technology Zijin College.His main research interests include the Internet of Things and wireless networks.

Ye Liu received the M.S. and Ph.D. degrees from Southeast University in 2013 and 2018. From 2014 to 2015, he was a Visiting Scholar with Montana State University. From 2017 to 2018, he was a joint Ph.D. Student with RISE Swedish Institute of Computer Science. He is currently an Associate Professor with Nanjing Agricultural University. His current research interests include the IoT, energy harvesting, and tiny machine learning.

Qing Yang is an Associate Professor in the Department of Computer Science and Engineering, University of North Texas. He received a Ph.D. degree in Computer Science from Auburn University, in 2011. His current research interests include connected and autonomous vehicles, cooperative perception, and the Internet of Things. His research is funded by the U.S. National Science Foundation, U.S. Federal Highway Administration, PACCAR, and Toyota InfoTech Inc.

Honggang Wang is the founding Chair and Professor of the Department of Graduate Computer Science and Engineering, Katz School of Science and Health, Yeshiva University in New York City. He was a Professor at UMass Dartmouth (UMassD). He is an alumnus of NAE Frontiers of Engineering program. He graduated 30 MS/Ph.D. students and produced high-quality publications in prestigious journals and conferences in his research areas, winning several prestigious best paper awards. His research interests include Internet of Things and its applications in health and transportation (e.g., autonomous vehicles) domains, Machine Learning and Big Data, Multimedia and Cyber Security, Smart and Connected Health, Wireless Networks, and Multimedia Communications. He is an IEEE distinguished lecturer and a Fellow of AAIA. He has served as the Editor in Chief (EiC) for IEEE Internet of Things Journal during 2020–2022. He was the past Chair (2018–2020) of IEEE Multimedia Communications Technical Committee and the past IEEE eHealth Technical Committee Chair (2020–2021).