electronics

MDPI

Article

P-A Scheme: A Robust and Lightweight Wi-Fi Device Identification Approach for Enhancing Industrial Security

Zaiting Xu, Qian Lu *, Fei Chen, Hanlin Zhang and Hequn Xian

Department of Computer Science and Technology, Qingdao University, Qingdao 266071, China; zaitingxu@qdu.edu.cn (Z.X.); feic@qdu.edu.cn (F.C.); hanlin@qdu.edu.cn (H.Z.); xianhq@qdu.edu.cn (H.X.)
* Correspondence: luqian@qdu.edu.cn

Abstract: The increasing dependence on Wi-Fi for device-to-device communication in industrial environments has introduced significant security and privacy challenges. In such wireless networks, rogue access point (RAP) attacks have become more common, exploiting the openness of wireless communication to intercept sensitive operational data, compromise privacy, and disrupt industrial processes. Existing mitigation schemes often rely on dedicated hardware and cryptographic methods for authentication, which are computationally expensive and impractical for the diverse and resource-limited devices commonly found in industrial networks. To address these challenges, this paper introduces a robust and lightweight Wi-Fi device identification scheme, named the P-A scheme, specifically designed for industrial settings. By extracting hardware fingerprints from the phase and amplitude characteristics of channel state information (CSI), the P-A scheme offers an efficient and scalable solution for identifying devices and detecting rogue access points. A lightweight neural network ensures fast and accurate identification, making the scheme suitable for real-time industrial applications. Extensive experiments in real-world scenarios demonstrate the effectiveness of the scheme, achieving 95% identification accuracy within 0.5 s. The P-A scheme offers a practical pathway to safeguard data integrity and privacy in complex industrial networks against evolving cyber threats.

Keywords: Wi-Fi device identification; Wi-Fi security; rogue access point detection; channel state information

Academic Editor: Djuradj Budimir

Received: 12 December 2024
Revised: 17 January 2025
Accepted: 24 January 2025
Published: 27 January 2025

Citation: Xu, Z.; Lu, Q.; Chen, F.; Zhang, H.; Xian, H. P-A Scheme: A Robust and Lightweight Wi-Fi Device Identification Approach for Enhancing Industrial Security. *Electronics* **2025**, *14*, 513. https://doi.org/10.3390/electronics14030513

1. Introduction

The rapid application of Wi-Fi technology in industrial environments has introduced significant security and privacy risks. As industrial systems increasingly rely on Wi-Fi for device-to-device communication, the attack surface continues to expand, exposing sensitive operational data to cyber threats. Among these threats, Rogue Access Point (RAP) attacks stand out as one of the most persistent and damaging. As illustrated in Figure 1, attackers can create a RAP with the same service set identifier (SSID) and basic service set identifier (BSSID) as a legitimate access point (AP), tricking devices into connecting and enabling subsequent attacks such as DNS spoofing and keystroke inference. These attacks not only compromise data integrity but also harm the privacy and operational reliability of industrial systems. RAP attack detection has, therefore, become a critical component in securing industrial wireless networks and the broader Internet of Things (IoT) system.

Although various wireless device identification schemes have been proposed to address RAP attacks, current schemes face critical limitations, such as low detection rates, limited scalability, and poor adaptability to industrial IoT environments. Traditional cryptography-based schemes, which rely on digital certificates to authenticate devices [1],

become increasingly impractical as network size and complexity grow. Given the inherent consistency, invariance, and difficulty of forgery, device hardware fingerprints extracted from channel state information (CSI) present a promising alternative for wireless device identification. However, existing research on hardware fingerprinting has not fully addressed the challenges in industrial environments. For example, Liu et al. [2] suggested using phase errors as fingerprints, but Lu et al. [3,4] discovered the drift in phase errors, which compromises detection accuracy. Another study [5] proposed mapping CSI to images for deep-learning-based detection, but this approach increases computational and storage costs, limiting its practical applicability in resource-constrained industrial environments. In summary, current hardware fingerprinting methods face challenges such as poor stability, insufficient accuracy, and excessive computational overhead.

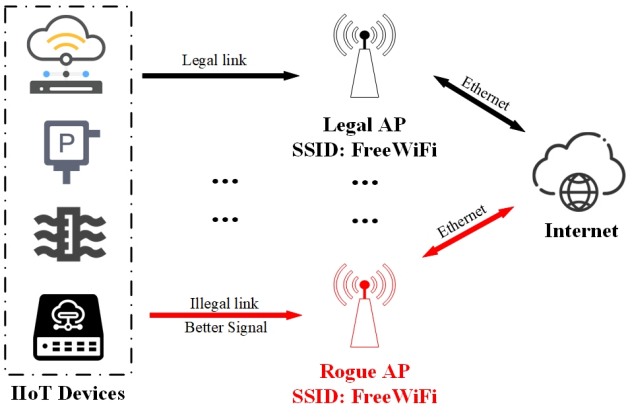


Figure 1. Illustration of RAPs and freeloading.

In this paper, we propose a novel CSI-based RAP detection scheme, termed the P-A scheme, suitable for industrial Wi-Fi networks. The P-A scheme leverages a joint P-A fingerprint derived from phase and amplitude characteristics of CSI as the device hardware fingerprint and incorporates a lightweight neural network, called the P-A network, for accurate device identification and RAP attack detection. Specifically, we introduce two new filtering techniques to extract phase error fingerprints (P fingerprints) and amplitude matrix fingerprints (A fingerprints) from CSI, which are combined to form the joint P-A fingerprint. Our scheme is designed to meet the stringent requirements of industrial applications, including scalability, efficiency, and resilience to dynamic conditions. Extensive real-world experiments demonstrate the superior performance of the proposed scheme. In summary, we have made the following contributions:

1. In this paper, we propose two novel Wi-Fi device identification fingerprints tailored for industrial environments: the P and A fingerprints, extracted from the phase and amplitude characteristics of CSI. Extensive experiments conducted in real-world scenarios demonstrate that combining these fingerprints into a joint P-A fingerprint significantly enhances device identification accuracy, addressing challenges in detecting RAP attacks.
2. We introduce the P-A scheme, which leverages the joint P-A fingerprint as a hardware-based identifier and employs a lightweight convolutional neural network, referred to as P-A networks, to achieve precise and efficient device identification. This approach is particularly suitable for resource-constrained industrial settings.
3. Comprehensive experiments under various real-world conditions validate the robustness and reliability of the P-A scheme. The scheme achieves an identification accuracy of up to 95% within a detection time of just 0.5 s. These results underscore the scheme's robustness and practicality in dynamic and heterogeneous industrial environments.

The rest of this paper is structured as follows. In Section 2, related work is introduced. Section 3 describes the background knowledge. Next, Section 4 outlines the setup and execution of our replication experiment. Then, the proposed mechanism is described in Section 5. In Section 6, we evaluate the proposed method through extensive experiments. Finally, we conclude this paper in Section 8.

2. Related Work

The use of physical layer features, such as radio frequency (RF) signals, for wireless device identification has gained significant attention in recent years. Shen et al. [6] analyzed fine-grained time-frequency features of long-range signals using RF signals and frequency spectrum. They discovered that drift in the instantaneous CFO could lead to classification errors, reducing system performance. To address this, they developed a hybrid classifier based on a convolutional neural network (CNN) to distinguish signals from different devices. Das et al. [7] proposed an LSTM-based deep learning classifier that leverages physical layer I/Q sampling to learn the hardware flaws of low-power wireless devices, effectively identifying high-power attack devices as well. Baldini et al. [8] proposed a device identification scheme for wireless IoT devices based on RF signals. These schemes require specialized collection devices to gather sufficient RF signals for device authentication, leading to cumbersome operations and high costs, which limits their practical application.

Many researchers have investigated using CSI signal characteristics for device authentication. Hua et al. [9] proposed a scheme that evaluates the CFO from the CSI and uses the stripe slope as a device fingerprint. However, this approach is highly sensitive to environmental factors and requires the device to remain stationary for 10 s during CSI collection to minimize environmental impact. Liu et al. [2] proposed a real-time device identification approach that uses nonlinear phase errors from the CSI as device fingerprints. The phase error is caused by I/Q imbalance and suboptimal crystal oscillators in each device's Wi-Fi NIC. However, the phase error exhibits a drift phenomenon [3,4], which significantly impacts the accuracy of phase error-based approaches.

Several CSI-based device fingerprinting methods have been proposed for IoT device identification and RAP detection. However, most of these methods have limitations that hinder their practical application. For instance, Kandel et al. [10] proposed using the relative phase error between the RF chains of the device as a fingerprint, but this method is limited to stable environments and specific hardware. Similarly, Liu et al. [11] proposed two fingerprint features: the non-linear power amplifier fingerprint and the frame interval distribution fingerprint, but both require additional collection and stationary detection equipment. Meanwhile, Yu et al. [12] ensured communication security between the central gateway and wireless sensor nodes by using the three parameters in the CFO stripe slope image as fingerprints. However, existing wireless device fingerprint detection technologies have greatly limited their practical applications due to high cost, poor stability, and the need for additional collection equipment.

Manually extracting hardware features is labor-intensive. To address this challenge, some researchers have integrated CSI with machine learning techniques. For example, Zaman et al. [13] employed an LSTM network to process CSI and authenticate device identities. Wang et al. [14] trained a CNN to extract local features from CSI and a regression neural network (RNN) to capture continuous features across different carrier frequencies. They combined the two networks to capture both local and continuous features, which were then applied to user authentication. Chen et al. [15] identified mobile devices at the packet level using a neural network with real-time update functionality. Germain et al. [16] introduced extra noise into CSI measurements and employed generative adversarial networks for identification. Liao et al. [17] obtained CSI through minimum mean variance

and identified it using a CNN. Germain et al. [18] processed CSI data using two variants of RNN and conditional generative adversarial networks. Dai et al. [19] employed SVM for identification and divided the Wi-Fi signal coverage area to create a fingerprint library for devices in different regions. Wang et al. [5] authenticated wireless devices based on their location. They divided the device's CSI by region and treated the real and imaginary components of the CSI element as two channels input into a deep learning network for training and identification.

However, the above schemes present two main issues [20,21]. Firstly, directly using CSI to train the network model is inefficient and resource-intensive, making it unsuitable for resource-constrained devices. Secondly, using regional differences in signal coverage as fingerprints is limited to fixed equipment, whereas most network terminals are now mobile. In conclusion, designing a practical, robust, and lightweight Wi-Fi device identification approach remains a significant challenge.

3. Background

This section provides background on the proposed scheme, including CSI, nonlinear phase errors and non-linear power amplifiers.

3.1. CSI

Today, Multiple Input Multiple Output (MIMO) and Orthogonal Frequency Division Multiplexing (OFDM) technologies are widely used in wireless networks to enhance bandwidth and transmission rates [22]. However, these technologies can also introduce issues such as mutual signal interference and multipath effects. To address these challenges, the IEEE 802.11 protocol family introduced a physical layer property known as CSI. CSI is used to assess the characteristics of wireless communication links, offering a comprehensive understanding of the effects of scattering, fading, and power decay on the transmission of wireless signals between the transmitter and receiver.

Daniel Halperin et al. [23] introduced a tool called CSI Tool, which modifies the driver of a wireless network card. This tool allows commercial Wi-Fi devices to capture CSI. As a result, CSI can now be used for indoor positioning, motion recognition, device identification, and other applications [24]. In this paper, we modify the driver of the Intel5300 wireless network card to collect CSI, which is then applied to our proposed Wi-Fi device identification scheme. We assume that the received and transmitted signal vectors on both communication sides are denoted as Y and X, respectively. The channel model can then be expressed as follows:

$$Y = H \times X + N \tag{1}$$

where H is the CSI matrix and N is the Gaussian noise. The CSI measurement value represents sampled data for Channel Frequency Response on different subcarriers. H can be represented by the following model:

$$H(f_k) = |H(f_k)| \cdot \mathbf{e}^{j\angle H(f_k)} \tag{2}$$

where $H(f_k)$ represents the CSI of the subcarrier whose center frequency is f_k. The absolute value $|H(f_k)|$ represents the amplitude on the k_{th} subcarrier. $\angle H(f_k)$ represents the phase value on the k_{th} subcarrier.

3.2. Nonlinear Phase Errors

Under ideal conditions, the carrier frequency between the Wi-Fi transmitter and receiver should remain consistent. However, a slight offset arises due to imperfections in the device hardware. This frequency offset alters the signal phase. Thus, Liu et al. [2]

emphasized that nonlinear phase errors caused by I/Q imbalance and suboptimal crystal oscillators can act as fingerprints for real-time device authentication. The phase error formula is given as follows:

$$\boldsymbol{E} = \boldsymbol{\Phi} - (2\pi\lambda \cdot \boldsymbol{K} + \boldsymbol{Z}^*) \tag{3}$$

where $\boldsymbol{\Phi}$ is the phase that satisfies flat gradient filtering $\nabla\boldsymbol{\Phi} = [\boldsymbol{\Phi}_2 - \boldsymbol{\Phi}_1/\boldsymbol{K}_2 - \boldsymbol{K}_1, \ldots, \boldsymbol{\Phi}_{i+1} - \boldsymbol{\Phi}_i/\boldsymbol{K}_{i+1} - \boldsymbol{K}_i \ldots]$, choose λ to make $\boldsymbol{E}_{-28} + \boldsymbol{E}_{28} = 0$, $\boldsymbol{K}$ is the subcarrier index, $\boldsymbol{Z}^* \approx (\boldsymbol{\Phi}_1 + \boldsymbol{\Phi}_{-1})/2$. According to the research [3,4], the phase error fingerprint exhibits a drift phenomenon. Our empirical experiments indicate that some Wi-Fi devices exhibit a tomographic phenomenon in their phase error fingerprints, which may negatively affect prior schemes.

3.3. Power Amplifier Non-Linear Fingerprint

Wireless signals possess two fundamental attributes: phase and amplitude. Amplitude refers to the signal strength or power, which can decrease after channel transmission. To adjust the amplitude, the receiver is equipped with amplifiers. Liu et al. [11] found that the instability of the power amplifier in wireless devices generates a nonlinear fingerprint, which can serve as a unique identifier for the device. We denote the CSI value at the subcarrier index f at time t as $H(f,t)$, and the signal amplitude model measured at the receiving terminal can be expressed as follows (4):

$$\left|\grave{H}(f,t)\right| = |H(f,t)| + G_{PA}(t) + G_{LNA} + G_{VGA}(t) + n \tag{4}$$

where $\left|\grave{H}(f,t)\right|$ are the sum of the gains of the amplifiers. $|H(f,t)|$ represents the propagation fading. $G_{PA}(t)$, G_{LNA}, G_{VGA}, respectively, represent the power gain of the Power Amplifier (PA), Low-Noise Amplifier (LNA), and Variable-Gain Amplifier (VGA). To obtain the power amplifier non-linear fingerprint, we can use the following formula after filtering out irrelevant influencing factors by calculating the weighted mean and variance.

$$\sigma^2\overline{(G_{PA})} = \sigma^2(\overline{\left|\hat{H}\right|}) - \sigma^2\overline{(G_{VGA})} \tag{5}$$

Nevertheless, our experiment reveals that the PA fingerprint is susceptible to fluctuations in the dynamic environment, leading to an unstable fingerprint.

4. Preliminaries

This section outlines the setup and execution of our replication experiment, detailing the preparation, and the results observed.

4.1. Experiment Preparation

For the empirical study, data collection is conducted using a Lenovo T430 laptop. This device, equipped with an Intel5300 wireless network card and running on Ubuntu 14.04, enables CSI collection through modifications to the Intel5300 driver and configuration of the CSI Tool.

To simulate wireless access environments, we randomly select four commercial Wi-Fi routers as access points. These routers include the Mi R1CL, Tplink WR842N, Tenda F3, and Mercury MW325R, which feature MT7628, QCA9531, RTL8192ER, and MT7628KN chipsets, respectively.

Data collection was conducted in a 150-square-meter (10 m $\times$ 15 m) enclosed student laboratory, arranged with four rows and five columns of desks. The data collection setup required positioning the laptop and Wi-Fi units approximately 3 m apart on adjacent tables, ensuring no physical obstructions and maintaining a controlled environment throughout the process.

4.2. Experiment Procedure

CSI Collection: To collect CSI data, we connect the collection device to the tested Wi-Fi devices and initiate two separate terminal sessions. The first terminal sends a ping command to the Wi-Fi device at 0.02-s intervals, while the second terminal executes the data capture command within the CSI Tool (version: 7.3.1.28). A complete CSI collection cycle is completed upon the initiation and conclusion of the log_to_file program. During this process, the collection device is set to receive response frames and systematically log the corresponding CSI data. This procedure is carefully managed to ensure the collection phase lasts approximately 15 s.

CSI Analyzation: The CSI data contain vital information, including the number of transmitting (T_x) and receiving (R_x) antennas, as well as the crucial CSI matrix. The dimensions of the CSI matrix are expressed as $T_x \times R_x \times 30$. Given the variability of T_x and R_x, the matrix dimensions are not uniform. We specifically filter and retain matrices of size [2, 3, 30]. Each element in these matrices is complex, and we calculate the phase value using the arctangent of each element, while concurrently deriving the amplitude by determining the absolute value.

Fingerprint Reproduction: Using the phase and amplitude values derived in the previous step, we apply Formula (3) to calculate the nonlinear phase error fingerprint and Formula (5) for the power amplifier nonlinear fingerprint. We then generate visual representations in the form of a subcarrier-phase error image and a distribution map for the power amplifier nonlinear fingerprint, as shown in Figures 2 and 3.

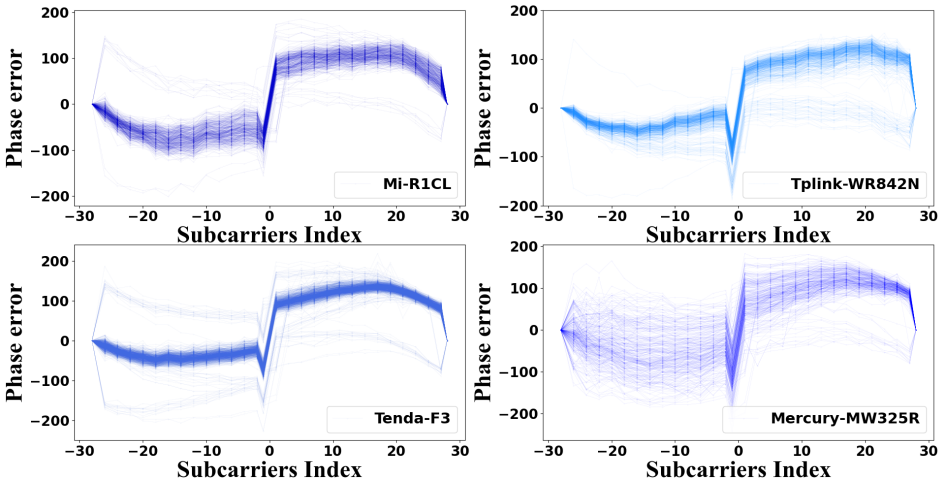


Figure 2. Phase errors of four tested devices.

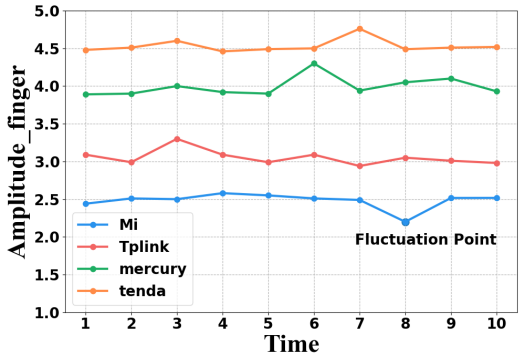


Figure 3. Amplitude nonlinear fingerprints distribution of four tested devices.

4.3. Experimental Results

Figure 2 illustrates the phase error diagram across subcarriers for four Wi-Fi devices under evaluation. In each diagram, the vertical axis represents the phase error, while the horizontal axis indicates the subcarrier index. The observed variability in phase errors within specific bounds supports the findings of prior research [3,4]. Additionally, a notable tomographic pattern emerges in the phase error data, challenging traditional function fitting methods due to its non-derivative nature. Using average or normalized phase errors as distinctive identifiers would overlook critical data characteristics, potentially reducing identification accuracy and limiting the technology's practical applications.

Figure 3 depicts the distribution of averaged power amplifier nonlinear fingerprints for the same Wi-Fi devices. Following the methodology proposed by Liu et al. [11], we combined two sets to derive a single fingerprint for each device. The x-axis corresponds to the data batch, with a total of ten batches of Figure 3 CSI analyzed. The y-axis displays the nonlinear fingerprint, revealing significant stability issues and unpredictable fluctuations. These irregularities could impair the fingerprint's reliability due to the unstructured nature of the fluctuations.

In summary, to more effectively exploit *nonlinear phase errors* and *power amplifier nonlinear fingerprints*, we introduce an innovative identification strategy for Wi-Fi devices, called the P-A scheme. This approach involves distinct processing of phase errors and amplitude matrices to create P and A fingerprints, respectively. We also propose a CNN model, called P-A Networks, designed specifically for training and identifying these fingerprints.

5. The Proposed Mechanism

In this section, we introduce a novel identification scheme for Wi-Fi device identification, termed the P-A scheme. This scheme leverages two key signal attributes—phase and amplitude—from which it derives two distinct fingerprints: the P (phase) and A (amplitude) fingerprints. An efficient Convolutional Neural Network, the P-A network, is proposed as the classifier of these fingerprints. This approach is adept not only at detecting rogue access point attacks but also at identifying conventional Wi-Fi devices.

The benefits of the P-A scheme are twofold: First, it encompasses two effective fingerprint generation algorithms along with a lightweight CNN, minimizing computational and memory demands. Second, it is designed for cloud server execution, requiring detection clients only to acquire and upload CSI. Acquiring CSI merely requires a software update to the client's device driver, without the need for hardware modifications or additional computational load. Therefore, this scheme is particularly well-suited for deployment on resource-constrained devices, with the computational burden being largely shouldered by the server.

The deployment architecture of the scheme is shown in Figure 4. To emulate the server environment, a personal computer with standard performance is employed to host the P-A scheme (The server is optional, meaning data collection and processing can be conducted entirely locally). The client device role is assumed by a separate collection laptop. When device identification is requested, the client gathers and forwards the CSI of the Wi-Fi signal to the server. The server delivers the identification outcome within approximately 0.5 s. If the target device is legitimate, it is allowed access to the network; otherwise, unauthorized devices trigger an alert to the client.

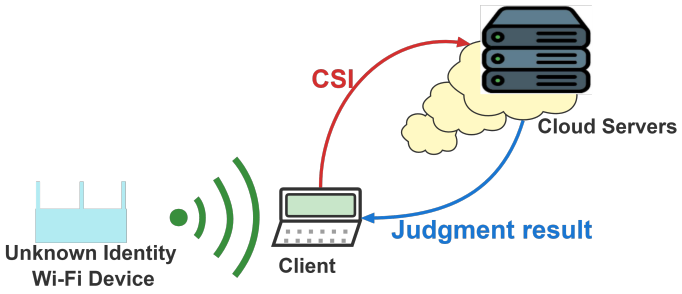


Figure 4. The P-A scheme structure.

The P-A scheme consists of four primary modules, as illustrated in Figure 5. The first module handles data collection and preprocessing. The second module processes the phase error and amplitude matrix to extract the P and A fingerprints. The third module adapts these fingerprints for CNN analysis, culminating in the construction and training of the P-A networks to yield identification results. The final module makes the final judgment.

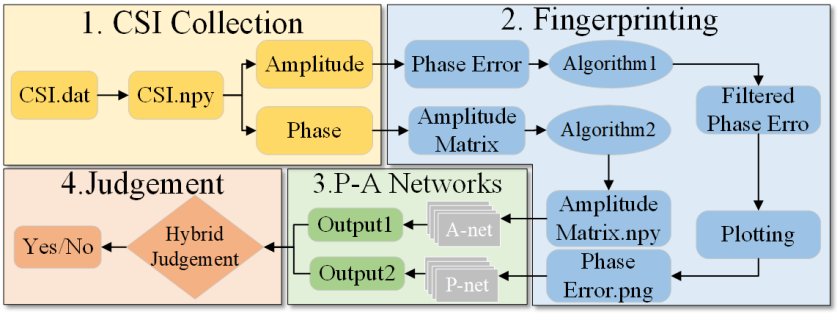


Figure 5. Main modules of the P-A scheme.

5.1. Data Collection and Preprocessing Module

This module primarily performs three core functions: CSI collection, CSI analysis, and phase and amplitude calculation.

In this module, the first step is CSI collection. A detection client connects to the target Wi-Fi device. The detection client then initializes two terminals. One terminal sends a ping command to the target Wi-Fi device at 0.02-s intervals, while the other executes the *log_to_file* function from the CSI Tool to facilitate automatic CSI data acquisition.

The second step involves CSI analysis. The CSI includes various sampling details, such as the number of transmitting antennas (T_x), receiving antennas (R_x), and the CSI matrix itself. Subsequently, a filtration process is applied to the CSI data based on matrix dimensions. This process isolates and extracts matrices with dimensions [2, 3, 30], which are then stored in an *npy* file format. This binary file format, designed for data storage, is generated using Python scripts. Our scheme leverages Python, which is efficient in loading *npy* files during network training, significantly enhancing the speed of training. Additionally, storing processed data in the *npy* format eliminates the need for repetitive recalculations in each identification phase, thereby reducing computational requirements.

The final step involves calculating the phase and amplitude from the *npy* files. CSI matrices are consolidated into a list, which is traversed to calculate the phase of each complex element using the arc tangent function, and the amplitude by determining the absolute value of the complex numbers. This methodology ensures the preservation of original phase and amplitude data positions relative to their slots in the CSI matrix, an essential consideration since phase error depends on the arrangement of the 30 sub-carriers, and the amplitude matrix exhibits spatial characteristics aligned with the CSI matrix. The

processed phase and amplitude data are then converted into P and A fingerprints by the subsequent module, which are used as training inputs for the P-A networks.

5.2. Fingerprint Generation Module

This module is a pivotal element within the system, employing a CNN to identify fingerprints. Its significance lies in the need to derive fingerprints with valid characteristics from CSI. This module applies distinct processing techniques for phase error and amplitude matrices, ultimately generating effective P and A fingerprints.

5.2.1. P-Fingerprint

The P fingerprint is the filtered phase error image. Its production is divided into three steps: calculating the filtering threshold, filtering the phase error interference, drawing the image, and storing it as the P fingerprint.

The first step is calculating the filtering threshold. Upon observation, interference items in the same set of phase errors often fluctuate considerably, and their distribution differs from that of a typical fingerprint. Therefore, standard deviation and median are used for filtering. First, the phase matrix is traversed, and phase error is calculated according to Formula (1). This results in L groups of phase error matrices, denoted as the E matrix, with dimensions [L, 30]. Next, the standard deviation of each group of fingerprints along the first dimension is calculated and stored as a list, denoted as D. Finally, the median of all phase errors over the 30 subcarriers is calculated along the second dimension and stored as the median list M.

The second step involves filtering the phase error interference term. First, list D is traversed. If the standard deviation D_i is less than the overall mean $average(D)$, the maximum absolute difference between $E_{i,j}$ and the median M_j is calculated. If the maximum value is less than the threshold y, the phase error is considered qualified and retained for subsequent identification. Extensive experiments have shown that when y is 0.4, Algorithm 1 yields the best filtering effect. Values may vary under different experimental conditions.

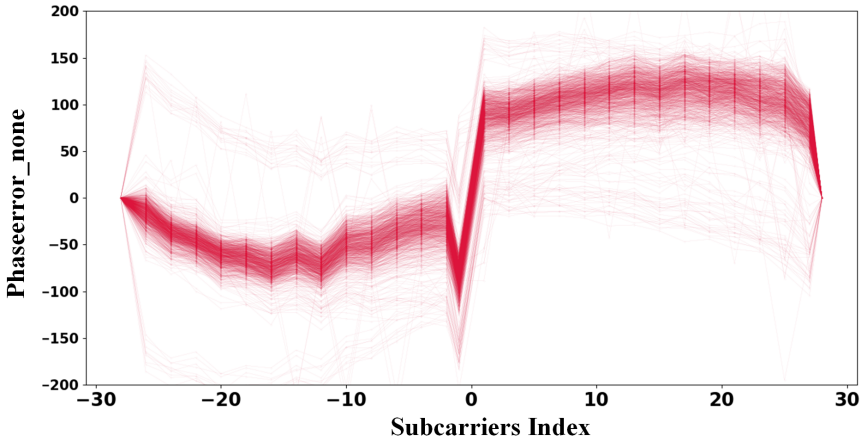
Algorithm 1 Filtering the phase errors

Require: Phase error matrix E; Standard Deviation D; Median M
Ensure: Filtered Phase error E
for i $\leftarrow$ 0 to L **do**
  if $D_i < averaged(D)$ **then**
    for j $\leftarrow$ 0 to 30 **do**
      $maxdifference = max(abs(E_{i,j} - M_j))$
    end for
    if $maxdifference < y$ **then**
      $Filtered.add(E_{i,j})$
    end if
  end if
end for
Output Filtered Phase error E;

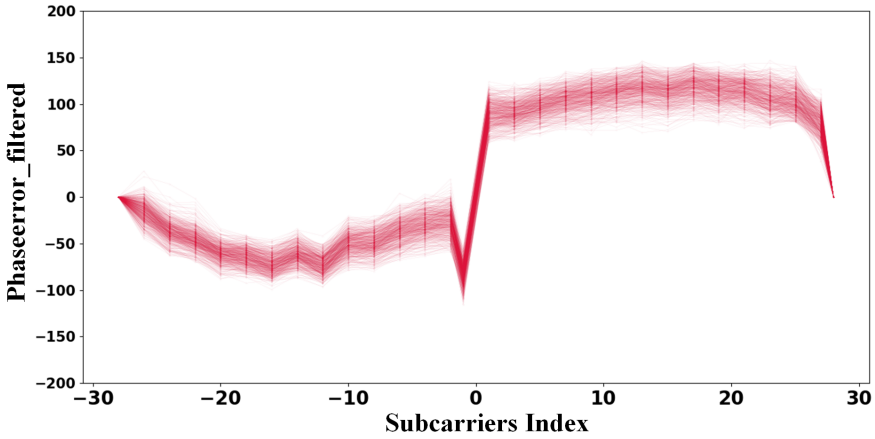
Unfiltered interfering terms in the phase error can affect the sharpness of the phase error image, thereby reducing accuracy. Therefore, accurately filtering these interference items is crucial to minimize their impact on the accuracy of the P fingerprint. The filtering algorithm is presented in Algorithm 1, and the filtered image is shown in Figure 6. As shown in Figure 6, the filtered phase error image appears clearer.

The third step is drawing the image and storing it as a P fingerprint. We use the matplotlib drawing tool to draw all the filtered phase errors in a single graph to form a P

fingerprint. The width of each phase error curve in the graph is 0.08 cm, and the color is uniformly red, the image is saved as 900 × 700 pixel size in *png* format.



(**a**) Non-filtered phase error of tested Tenda device



(**b**) Filtered phase error of tested Tenda device

Figure 6. Phase errors filtered by Algorithm 1.

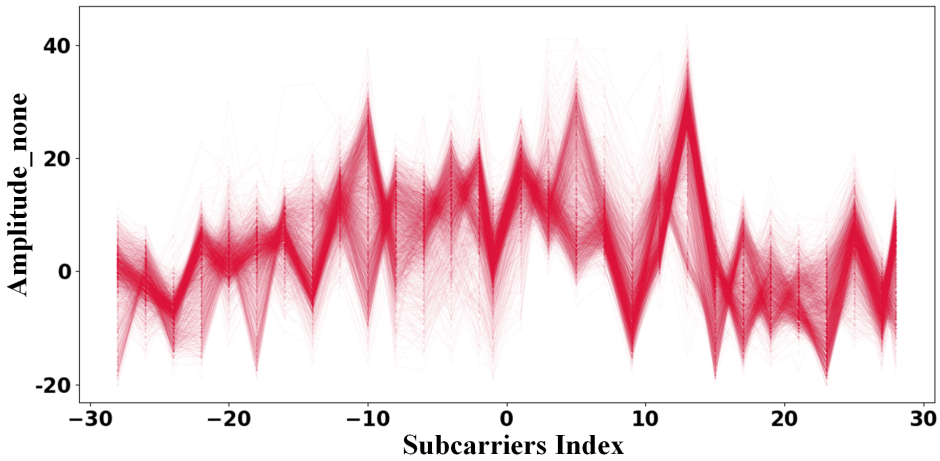
5.2.2. A Fingerprint

The A fingerprint replaces the distractor items in the amplitude matrix with the median and the *power amplifier non-linear fingerprint* in the study [11], to form a matrix with spatial characteristics. The process consists of three steps: calculating the filtering threshold and eigenvalues, filtering and constructing the amplitude matrix, and storing the amplitude matrix as the A fingerprint. The amplitude on the same receiving antenna remains consistent for signals from a pair of wireless transmitting-receiving devices but varies between adjacent antennas. Therefore, the amplitude matrix exhibits specific spatial characteristics, denoted as matrix S, with dimensions [2, 3, 30], and there are L groups in total. Due to the heavy disturbance in the raw amplitude data, we must exclude distractor items and enhance the spatial features as much as possible.

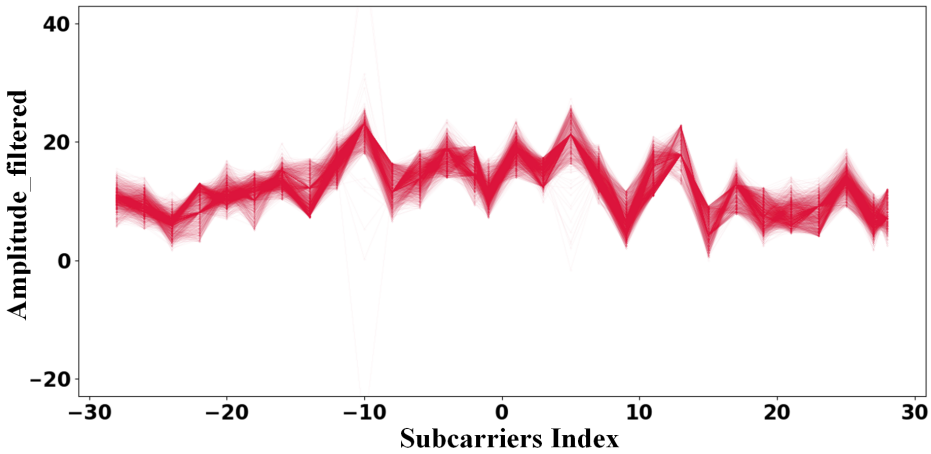
The first step involves calculating the filtering threshold and eigenvalues. The filtering threshold includes the mean value matrix A, the standard deviation matrix V, and the median matrix M. The eigenvalues consist of the amplitude non-linear fingerprint Q and the median matrix M. The median matrix M serves as both the filtering threshold and the eigenvalue. For each [2, 3, 30] amplitude matrix S, the first dimension represents the number of transmit antennas, the second represents the number of receive antennas, and the third corresponds to 30 subcarriers. We traverse the matrix S along the first and second dimensions, calculate the weighted mean of the amplitudes corresponding to subcarrier

indices along the third dimension, and obtain an array of [2, 3]. L groups form a matrix Q with dimensions [L, 2, 3]. Similarly, the mean matrix A and the standard deviation matrix V can be calculated, with dimensions [L, 2, 3]. The matrix M corresponds to the median of all amplitude data for each subcarrier. The median matrix M is obtained by traversing the amplitude matrix, with dimensions [2, 3, 30].

The second step involves filtering and constructing the amplitude matrix. The amplitude matrix S is traversed, and when the absolute value of the difference between $S_{i,j,k,l}$ and the median $M_{j,k,l}$ exceeds the threshold y_1, it is identified as a distractor and replaced with $M_{j,k,l}$. When the absolute value of the difference between $S_{i,j,k,l}$ and the average value $A_{i,j,k}$ exceeds y_2 times the average standard deviation, it is considered a distractor and replaced with $Q_{i,j,k}$. The values of y_1 and y_2 are set to 5 and 0.8, respectively. Empirical experiments have shown that these values optimize the performance of the scheme. These values may vary under different experimental conditions. After the replacement is complete, the new amplitude matrix is constructed. The filtering algorithm is shown in Algorithm 2. The amplitude values for different antenna pairs of Tplink were checked, as shown in Figure 7, demonstrating that the filtering method performs effectively.



(**a**) Non-filtered amplitude of tested Tplink device



(**b**) Filtered amplitude of tested Tplink device

Figure 7. Amplitude filtered by Algorithm 2.

Algorithm 2 Filtering the amplitude matrix

Require: Amplitude matrix S; Weight means Q; Standard deviation V; Average A; Median M;
Ensure: Filtered amplitude matrix S

for i $\leftarrow$ 0 to L **do**
  for j $\leftarrow$ 0 to 2 **do**
    for K $\leftarrow$ 0 to 3 **do**
      for l $\leftarrow$ 0 to 30 **do**
        if $abs(S_{i,j,k,l} - M_{i,j,k}) > y1$ **then**
          $S_{i,j,k,l} = M_{j,k,l}$
        end if
        if $abs(S_{i,j,k,l} - A_{i,j,k}) > y2 \times average(V)$ **then**
          $S_{i,j,k,l} = Q_{i,j,k}$
        end if
      end for
    end for
  end for
end for
Output Filtered amplitude matrix S;

The third step involves storing the amplitude matrix as the A fingerprint. The dimensions of the filtered amplitude matrix remain unchanged; it consists of L groups of [2, 3, 30] matrices, all of which together form the A fingerprint. All matrices are saved as a npy file, which offers advantages such as fast reading, avoiding repeated calculations, and reducing computational costs.

5.3. P-A Networks Module

This section introduces the standardization process for the two fingerprints and the structure of the P-A networks. Two sets of models are responsible for processing the P fingerprint and A fingerprint, respectively, with slightly distinguishable structures.

The P fingerprint and A fingerprint are converted into a data type suitable for processing by the convolutional network. An index is created for all fingerprint data along with a unique identification tag for the device, allowing for easy access to the data and corresponding tags. First, the P fingerprint is read and opened, converted into a three-channel model, uniformly resized to 128 $\times$ 128 pixels, and then transformed into a tensor vector for orthogonalization. Next, the A fingerprint is read and opened, converted into tensor vectors, orthogonalization is performed, and finally transformed into a vector of [2 $\times$ L, 30, 3].

The P-A networks model used is illustrated in Figure 8. Two networks are employed: the P-network and the A-network. The P fingerprint is input into the P-network for training and identification, while the A fingerprint is input into the A-network. Both networks consist of convolutional and linear layer modules. Each convolution unit in the convolution module sequentially applies four functions: convolution, batch normalization, ReLU activation, and max pooling. Five such units are included in the total. The P-network and A-network share the structure of the convolution module. A linear layer unit consists of a linear layer followed by a ReLU activation function. The linear layer module of the P-network contains two linear units and one linear layer, while the A-network's linear layer module consists of four linear units and one linear layer.

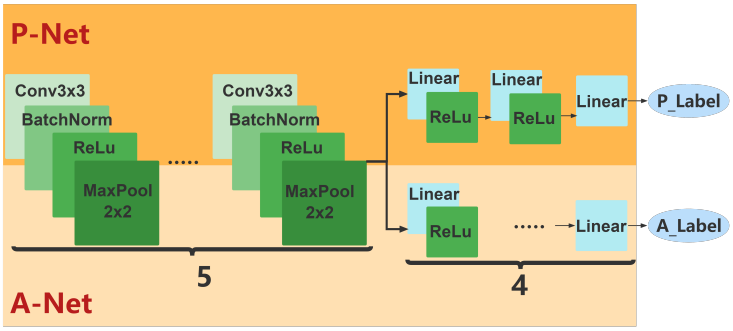


Figure 8. The structure of the P-A networks.

The proposed scheme employs a CNN as the fingerprint classifier, primarily for two reasons. First, the P fingerprint is an image type, whereas the A fingerprint is a three-dimensional matrix. The second dimension of the matrix is treated as the three-channel attribute of the image, thus making it possible to regard it as an image type. Second, convolutional networks are particularly well-suited for handling time series and image data [25], making them ideal for this identification task.

5.4. Judgment Module

This module primarily makes judgments based on the outputs of the P-A networks. In the proposed P-A scheme, the P-network and A-network identify the P fingerprint and A fingerprint, respectively, and output the predicted device label. Approximately 300 P fingerprint images and 1000 A fingerprint matrices can be calculated from the collected CSI data of a single Wi-Fi device to train the P-A networks. Experimental results have shown that only 20 P fingerprints and 100 A fingerprints are needed to complete the identification process. The predicted device label is output after being identified by the P-A network. The number of correctly predicted labels is divided by the total number of fingerprints to calculate the prediction accuracy of the two fingerprints, and the prediction rates of the two fingerprints are combined to determine whether the device is legitimate. Based on our experience, a device is judged as legitimate when the sum of the prediction rates of the P and A fingerprints is greater than 1.7, and the prediction rate of a single fingerprint is greater than 0.85. If the prediction rate of one fingerprint is lower than 0.85, the device will be regarded as illegal, regardless of the prediction rate of the other fingerprint, as shown in Table 1.

Table 1. Judgment Rules.

P-Finger	A-Finger	P-A Finger	Result
>0.85	>0.85	>1.7	permission
any	<0.85	any	warning
<0.85	any	any	warning

6. Evaluation

This part first introduces the setup of the evaluation experiment and then evaluates the performance of the proposed scheme from the four aspects of accuracy, speed, stability, and system overhead. Stability includes time stability, location stability, and environmental stability. Finally, the P-A scheme is compared with several mainstream schemes.

6.1. Setup of the Evaluation Experiment

For the experiment hardware, we used a laptop equipped with the CSI Tool, the model LenovoT430, equipped with a wireless network card Intel5300, for the data collection. We

used a PC, the model is OptiPlex 5060-China HDD Protection, the processor is Intel(R) Core(TM) i5-8500T, and the running memory RAM is 8 GB, to simulate the cloud server. A total of 22 wireless devices were used as testing devices. The brands, models, and quantities are shown in Table 2.

Table 2. Wi-Fi Device Information.

Brand	Model	Quantity
360	T2	4
Tplink	WR842N	4
Honor	XD16	2
Tenda	F3	4
Fast	FW325R	4
Mercury	MW325R	4

For the experiment environment, a 10 m × 15 m student laboratory is selected as the experimental site, and there are four rows and five columns of tables and chairs, as shown in Figure 9. We keep the collection device and the wireless device at a distance of three meters and place them on the desktop with no obstruction in the middle. The collection time is 8 to 10 s, and the collection process keeps the environment stable.

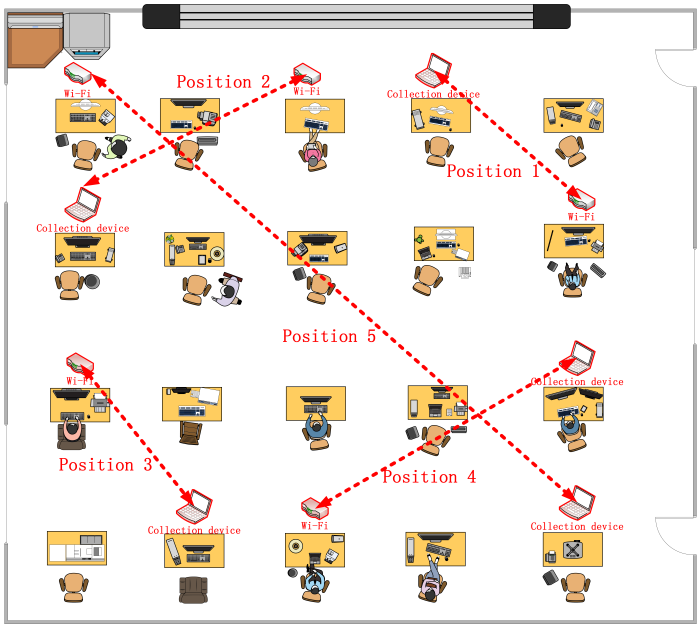


Figure 9. Five position relationships.

For the experimental evaluation, the collection device simulates a client and uploads the collected CSI to the server running the P-A scheme. The server caches the CSI data, calculates the P and A fingerprints, processes them with the P-A networks, and outputs the accuracies. Finally, the accuracies of the two fingerprints are combined to generate a judgment result, which is then returned to the client.

6.2. Accuracy

To verify the performance of the P-A scheme in identifying rogue devices, a total of 30 rounds of evaluation experiments were conducted. In each round, the 22 wireless devices were randomly divided into two groups. One group simulates legitimate devices, while the other simulates malicious devices. In the simulation experiment, the scheme

stores the fingerprint information of the legitimate group, without recording the fingerprint of the illegal group. The scheme then randomly connects to each device's signal and identifies it. If the identity of the legitimate or illegal device is successfully identified, it is marked as a success; otherwise, it is marked as a failure. First, the prediction rates of the P and A fingerprints are used individually for identification. If the prediction rate is greater than 0.85, the identification is deemed successful. Second, the two prediction rates of the combined P-A fingerprints (Table 1) are used for identification. To calculate the device detection accuracy, the following formula is used:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{N}_{\boldsymbol{correctly\_identified}}}{\boldsymbol{N}_{\boldsymbol{total\_devices}}} \times 100\% \tag{6}$$

where $\boldsymbol{N}_{\boldsymbol{correctly\_identified}}$ represents the number of devices successfully identified as either legitimate or malicious, and $\boldsymbol{N}_{\boldsymbol{total\_devices}}$ represents the total number of devices involved in the detection process.

In each round of the evaluation, the P-A scheme determines the accuracy by dividing the number of successfully identified devices by the total number of devices. The accuracy from 10 detection trials is averaged to obtain the final accuracy for each round. The results are presented in Figure 10. As shown in this figure, the accuracy of using the combined P-A fingerprints is significantly higher than that of using a single fingerprint. The accuracy of the combined P-A fingerprints can reach 95%, while the accuracies of the P and A fingerprints are stable at 89.5% and 91.5%, respectively.

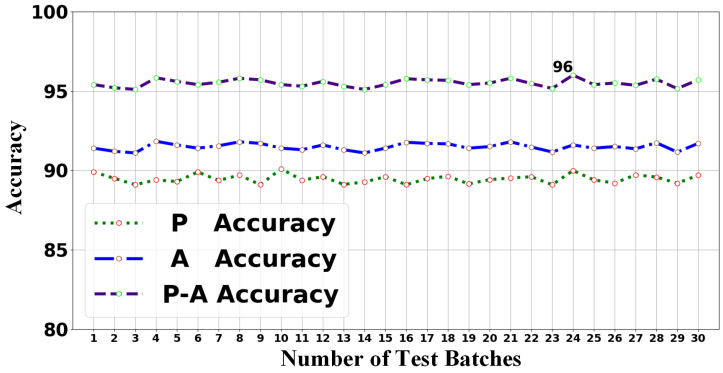


Figure 10. The accuracy of the P-A scheme.

6.3. Speed

We input CSI data of different volumes into the P-A scheme and evaluate the running speed through the time spent during the identification. We directly recorded the detection time during the experiments using a timer in the identification process. This approach ensures an accurate measurement of the time required for device identification.

The durations considered are 2 s, 4 s, 6 s, 8 s, 10 s, 15 s, and 20 s. CSI data of different volumes are input into the P-A scheme, and the running speed is evaluated based on the time spent during identification. It is assumed that the P-A networks have already been trained, so the training process does not account for the time spent. The identification time and accuracy for 22 devices are shown in Table 3. As the time for collecting CSI increases, the identification accuracy of the devices increases, and the corresponding system identification time also increases. When the CSI collection time reaches 8 to 10 s, the P-A scheme achieves its optimal accuracy of about 95.9%. At this point, the identification time of the system is approximately 0.56 s. Therefore, the shortest data collection time for the P-A scheme is about 8 s, and the optimal identification speed is approximately 0.56 s per device.

Table 3. Identification time and corresponding accuracy of P-A scheme under different CSI collection times.

CSI Collection Time (s)	Identification Time (s)	Accuracy Rate (%)
2	0.17	42.0
4	0.19	67.0
6	0.37	86.0
8	0.56	95.9
10	0.66	95.9
15	0.82	94.9
20	1.12	95.0

6.4. Stability

This section evaluates the stability of the proposed scheme from three aspects: time, location, and environment. Different scenarios are arranged to collect CSI in order to evaluate the stability across various factors. To verify the stability of both fingerprints simultaneously, the trained P-A networks are used to identify the joint P-A fingerprints in the CSI data from different scenarios. The reliability of the fingerprint, and thus the stability of the P-A system, is assessed based on the identification accuracy.

6.4.1. Time Stability

We selected five time periods and collected five batches of data to verify the time stability of the P-A system. To eliminate influencing factors other than time, CSI was collected every three days, resulting in a total of five batches, recorded as T1, T2, T3, T4, and T5. Using the P-A scheme, we conducted device identification for the five batches of data. First, we used the P and A fingerprints separately for identification and then combined them for joint identification.

The results, as shown in Figure 11, demonstrate that the accuracy of the P-A system in identifying joint P-A fingerprints across five time periods remained stable at approximately 95.5%, while the accuracy of P and A fingerprints stabilized at 89.5% and 91.5%, respectively. This indicates that the scheme is not significantly affected by the time factor.

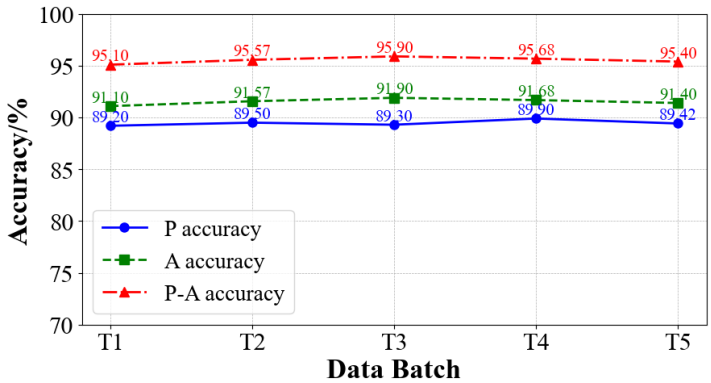


Figure 11. The time stability of the P-A scheme.

In addition, we further analyzed the impact of CSI collection time on the identification accuracy and time efficiency of the P-A scheme. Table 3 illustrates that as the CSI collection time increases, the accuracy of the P-A scheme improves significantly, reaching 95.9% at 8 s. However, when the collection time is extended beyond 10 s, the accuracy remains stable, with a slight fluctuation observed between 94.9% and 95.9%. Meanwhile, the identification time spent increases linearly from 0.17 s at 2 s of collection time to 1.12 s at 20 s. This suggests that 8 to 10 s is the optimal range for achieving high accuracy and time efficiency.

6.4.2. Position Stability

We determined five positional relationships in a 10 m × 15 m laboratory to verify the position stability of P-A. The five position relationships are shown in Figure 9, marked as L1, L2, L3, L4, and L5, respectively. We collect CSI at these five positions and keep the laboratory steady to ensure that the position condition is the unique variable. We use the P-A scheme to identify CSI from each position and output the accuracy of the 22 devices. First, we use P and A fingerprints to verify the device's identity. Then, combine the two fingerprints to verify. The result is shown in Figure 12. It can be found that the accuracy of the joint P-A fingerprints is stable at 95%, while the accuracy of the P and A fingerprints is stable at 89.5% and 91.5%, respectively, which demonstrates that the detection accuracy of the P-A scheme would not be significantly reduced.

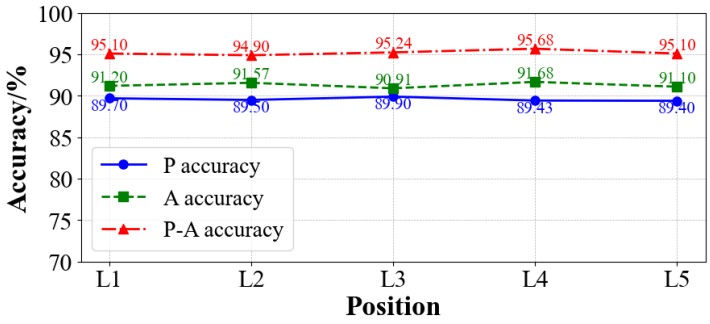


Figure 12. The position stability of the P-A scheme.

6.4.3. Environment Stability

We simulated two scenarios to verify the environmental stability of the P-A scheme, as shown in Figure 13. Two scenarios were conducted in a closed laboratory. In the stable environment, the collection device and the tested Wi-Fi device maintained a fixed distance of 5 m, and the indoor environment remained steady during the collection process. In the dynamic environment, the collection device and the tested device were kept at the same position and distance, but the experimenters randomly walked among the devices, moved tables and chairs, and frequently opened and closed the doors and windows to simulate real-world application scenarios such as human movement and vehicle activity [26,27].

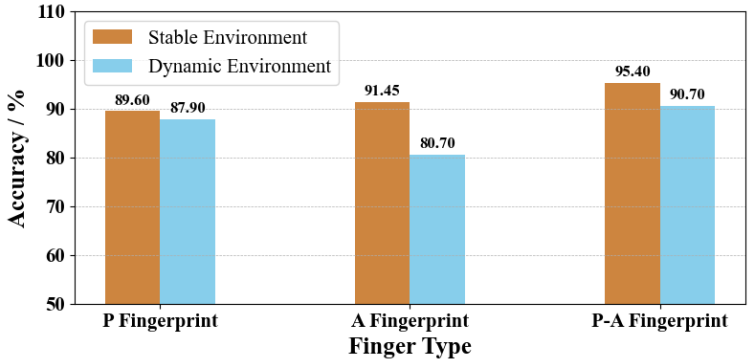


Figure 13. The environmental stability of the P-A scheme.

The results indicate that the P fingerprint is weakly affected by environmental changes, with accuracy decreasing slightly from 89.6% to 87.90%. However, the A fingerprint is more sensitive to environmental factors, with accuracy dropping significantly from 91.45% to 80.70%. This environmental impact on the A fingerprint also caused the joint P-A fingerprint accuracy to decrease from 95.40% to 90.70%.

To address this issue, we further analyzed the impact of extending the CSI collection time in suboptimal environmental conditions (E2). As shown in Figure 14, increasing the collection time can effectively mitigate the adverse effects of environmental disturbances. When the collection time is extended to 20 s, the accuracy of the A fingerprint recovers to approximately 90.5%, the P fingerprint accuracy improves to 89%, and the joint P-A fingerprint accuracy resumes at about 94%.

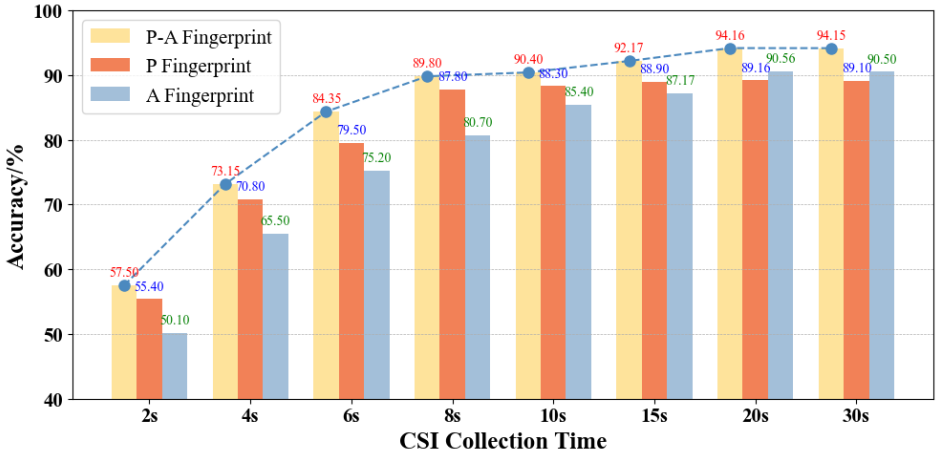


Figure 14. Influence of collection duration on fingerprint accuracy in suboptimal environments.

Combining this with our earlier analysis in Table 3, it is evident that collection time is a critical parameter affecting the detection accuracy. The results indicate that while longer collection times improve accuracy, particularly in challenging environments, there is a trade-off with identification efficiency. For instance, as shown in Table 3, the identification accuracy plateaus at 95.9% when the collection time exceeds 8 s, but the identification time increases linearly, reaching 1.12 s at 20 s of collection time.

These findings highlight that the P-A scheme demonstrates robust performance under various environmental conditions and collection durations. To balance accuracy and efficiency, we recommend a collection time of 8 to 10 s in stable environments and up to 20 s in noisy or dynamic environments to ensure high detection accuracy.

6.5. System Overhead

We evaluate the P-A system overhead of the client device and the server from program space occupancy, cache data occupancy, and CPU occupancy. We use a laptop (Lenovo T430) to simulate the client device and a PC (OptiPlex 5060-China HDD Protection) to simulate the cloud server.

For the program space occupancy, the client device needs to be modified as the driver of the NIC and responsible for collecting and uploading CSI to the server. We accomplish the above functions with a script of less than 50 KB. The server finishes the majority of the P-A scheme, in which the program occupies the server's memory of about 60 KB. Although there may be slight divergences in different operating systems, the program in the client and server will not consume too much storage space.

Regarding cache data occupancy, during identification the client needs to temporarily store the collected CSI for 8 to 10 s, occupying approximately 30 KB. During the training phase of the P-A networks, the client uploads the CSI individually and clears the uploaded data, maintaining a memory cost of approximately 30 KB. The server needs to store a few temporary fingerprints for identification but stores a large quantity of training data during the training phase. The storage cost of a phase error fingerprint image is approximately 32 KB, and every ten amplitude matrices occupy approximately 15 KB. Through extensive

experiments, we have determined that the P-A networks require at least 300 phase error images and 1000 sets of amplitude matrices to learn a device's features, occupying approximately 9.5 MB and 1.5 MB of storage, respectively. We assume that a client has ten wireless devices that need to be authenticated, resulting in a total cached data size of approximately 200 MB. In practical applications, clients rarely need to authenticate more than ten wireless devices, and typically, only a limited number of datasets need to be collected. Furthermore, after the training of the P-A network is completed, the space occupied by the cached data will be freed. The identification stage requires only 20 phase error images and 100 sets of amplitude matrices, occupying approximately 790 KB of cache, which imposes minimal resource constraints on the server.

Regarding CPU occupancy, the client device is a Lenovo T430 laptop from 2012, with an i5-3210M processor and 2 GB of RAM, simulating the client responsible for collecting and updating CSI to the server. The server is a PC model OptiPlex 5060-China HDD Protection, with an Intel(R) Core(TM) i5-8500T processor and 8 GB of RAM. During CSI collection, the client device's CPU utilization is approximately 15%. We collected CSI data from 22 devices, with a total size of 240 MB, and the data were fed into the P-A scheme. During the training process, the server's CPU utilization ranges from 28% to 30%, with a training time of 30 min. Identifying a device takes approximately 0.5 s. The experimental results show that the scheme does not require enormous computational power, thus reducing the server's burden. The server performs most of the work, leaving the lighter task of collecting and uploading CSI to the client. Thus, the proposed scheme can be deployed on resource-constrained entities.

6.6. Comparison with Related Works

Since we obtained the source code from [9,11], we conducted a lot of experiments comparing the P-A scheme with these two approaches. Specifically, a Lenovo T430 laptop was used to simulate the client. When the client is connected to the tested Wi-Fi devices, we collect their CSI and use phase error, amplitude error, and the P-A scheme for identification in a stable environment. Next, we re-experiment by opening and closing doors and windows while the experimenter moves around. A large number of experimental results were statistically analyzed, as shown in Figure 15. Compared to phase error and amplitude error, the P-A scheme has stronger anti-interference capabilities and integrates advanced convolutional networks. The P-A scheme can achieve ideal identification performance in dynamic environments, while the performance of the other two methods decreases significantly. Three mobile devices were then included in the experiment: Honor P30 Pro, iPhone NNGX2CH, and iPad MYLD2CH. The Lenovo T430 laptop simulates a hotspot, allowing mobile devices to connect to it and collect CSI. The P-A scheme was used to detect the identity of three mobile devices, and the experimental results show that it can also identify mobile devices with an accuracy of 90%, preventing freeloading attacks.

In addition to [9,11], we also analyzed and compared the performance of the P-A scheme with the recent four works; the result is shown in Table 4. Pu et al. utilized an improved Client Server-based sparse recovery method to achieve rogue AP localization and anomaly detection. However, their validation was conducted only under a single noise condition and failed to achieve fine-grained AP identification, merely providing a rough estimation of device location. Compared to our approach, we conducted validations under noisy environments, human movement, and device mobility, with the accuracy dropping by about 4% at most (as shown in Figure 13). Dara et al.'s approach requires pre-establishing a database for the intrusion detection system, which not only introduces additional database maintenance overhead but also demands higher computational power from the deployed devices to support the system. In contrast, our approach eliminates the need for a database;

P-A fingerprints are extracted and identified using a lightweight network. Lin et al.'s method experienced a significant 10% accuracy drop in dynamic environments with human movement and lacked systematic evaluation under noisy conditions. In comparison, our approach, as shown in Figure 13, demonstrates comprehensive performance under various noisy environments, with minor accuracy drops easily mitigated by extending the CSI collection time (as shown in Figure 14). Liu et al.'s method relies on device location information to detect RAPs, which becomes challenging when malicious and benign APs are in close proximity. In contrast, as shown in Figure 12, our approach maintains an approximately 95% detection accuracy for P-A fingerprints across different locations, without noticeable fluctuations, highlighting the robustness of our scheme.

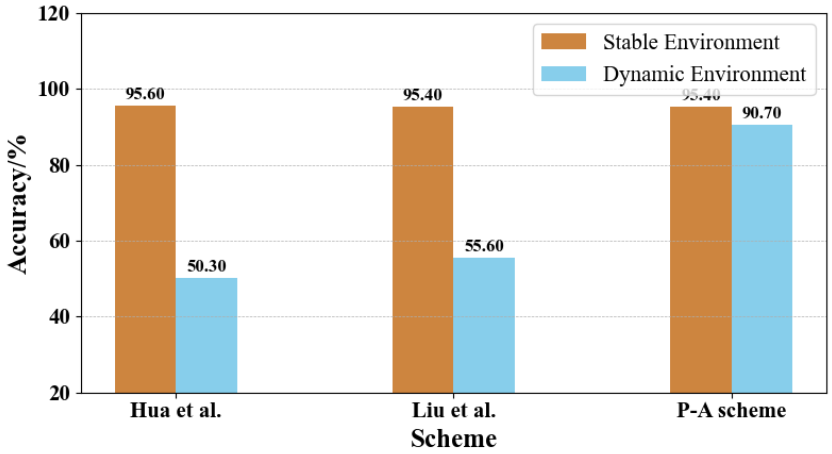


Figure 15. Comparison of Schemes [9,11].

Table 4. Comparison between P-A scheme and existing scheme.

Related Works	Framework	Input Data	Lightweight	Fine-Grained	Robust	Accuracy
Pu et al. [28]	OWAP clustering	RSS value			✓	82%
Dara et al. [29]	Isolation forest	Traffic		✓		87%
Lin et al. [30]	Two-step AP	CSI and FID	✓	✓		96.55%
Liu et al. [31]	RAIM	RSSI				67–97%
P-A scheme	P-A network	P-A value	✓	✓	✓	95%

7. Limitations and Future Work

The proposed P-A scheme, while demonstrating commendable performance in terms of accuracy and efficiency, does present several limitations that could hinder its applicability in certain environments. Firstly, the accuracy of the scheme is dependent on the duration of the CSI collection. While extending the collection time improves the accuracy, it also introduces a trade-off with system efficiency. In noisy environmental conditions, such as those with significant interference or dynamic movements, the A fingerprint is notably sensitive to environmental changes. This vulnerability is less pronounced in stable environments but still affects overall performance in real-world settings. Consequently, to maintain both accuracy and efficiency, the collection time needs to be carefully adjusted based on the specific environmental context. Additionally, while the P-A scheme performs adequately in stable conditions, its robustness in highly variable or noisy environments remains a challenge. Therefore, the scheme's environmental sensitivity calls for further refinement to ensure consistent performance in a broader range of real-world scenarios.

To address these challenges, our future work will focus on incorporating additional signal features such as Received Signal Strength Indicator (RSSI) and Time of Flight (ToF) to further enhance the robustness and accuracy of our approach. Additionally, to reduce

the reliance on CSI, we aim to explore multi-sensor fusion techniques that combine data from multiple sources. Then, we plan to investigate advanced CSI processing techniques to improve the stability and quality of the data, enabling our system to perform effectively across a broader range of environments. In our future work, we will explore the possibility of designing new hardware configurations and signal processing methodologies that can address these limitations and further extend the capabilities of our device identification framework.

8. Conclusions

In this paper, we have introduced a novel CSI-based RAP detection scheme, called the P-A scheme, designed specifically for the challenging conditions of industrial Wi-Fi networks. Our approach leverages two innovative hardware fingerprints—P fingerprints derived from phase errors and A fingerprints from amplitude characteristics—combined into a joint P-A fingerprint. This unique combination addresses the critical limitations of existing methods by enhancing stability, accuracy, and computational efficiency in dynamic and resource-constrained industrial environments.

The P-A scheme introduces a lightweight convolutional neural network, the P-A network, which ensures precise and efficient device identification and RAP attack detection, even under varying real-world conditions. Extensive experimental results demonstrate that our scheme achieves remarkable identification accuracy of up to 95% with a detection time of just 0.5 s. In addition, the process from data collection to fingerprint recognition only takes about 170 KB of memory. In the future, we will focus on studying more promising multi-sensor fusion technologies to extend the P-A scheme. In summary, our work provides a scalable and adaptable framework for securing industrial IoT systems against evolving cyber threats.

Author Contributions: Conceptualization, Z.X. and Q.L.; methodology, Z.X.; software, Z.X.; validation, Q.L.; formal analysis, Q.L.; investigation, Z.X.; resources, F.C.; data curation, H.X.; writing—original draft preparation, Z.X.; writing—review and editing, Q.L. and H.Z.; visualization, H.X.; supervision, F.C.; project administration, H.X.; funding acquisition, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China [62102212]; Shandong Province Youth Innovation and Technology Program Innovation Team [2022KJ296]; Natural Science Foundation of Shandong [ZR202102190210]; Nanchang Major Science and Technology Project [2023137] and Postdoctoral Funding Program of Qingdao [QDBSH20230201012].

Data Availability Statement: The data presented in this study are available on request from the corresponding author Qian Lu.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mushtaq, M.F.; Jamel, S.; Disina, A.H.; Pindar, Z.A.; Shakir, N.S.A.; Deris, M.M. A survey on the cryptographic encryption algorithms. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 333–344.
2. Liu, P.; Yang, P.; Song, W.Z.; Yan, Y.; Li, X.Y. Real-time Identification of Rogue WiFi Connections Using Environment-Independent Physical Features. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 190–198. [CrossRef]
3. Lu, Q.; Li, S.; Zhang, J.; Jiang, R. PEDR: Exploiting phase error drift range to detect full-model rogue access point attacks. *Comput. Secur.* **2022**, *114*, 102581. [CrossRef]
4. Zhang, J.; Lu, Q.; Jiang, R.; Qu, H. PEDR: A Novel Evil Twin Attack Detection Scheme Based on Phase Error Drift Range. In Proceedings of the SecureComm, Washington, DC, USA, 21–23 October 2020.
5. Wang, S.; Huang, K.; Xu, X.; Zhong, Z.; Zhou, Y. CSI-Based Physical Layer Authentication via Deep Learning. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 1748–1752. [CrossRef]

6. Shen, G.; Zhang, J.; Marshall, A.; Peng, L.; Wang, X. Radio Frequency Fingerprint Identification for LoRa Using Spectrogram and CNN. *IEEE Conf. Comput. Commun.* **2021**, *39*, 2604–2616. [CrossRef]
7. Das, R.; Gadre, A.; Zhang, S.; Kumar, S.; Moura, J.M.F. A Deep Learning Approach to IoT Authentication. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6. [CrossRef]
8. Baldini, G.; Giuliani, R.; Dimc, F. Physical layer authentication of Internet of Things wireless devices using Convolutional Neural Networks and Recurrence Plots. *Internet Technol. Lett.* **2018**, *2*, e81. [CrossRef]
9. Hua, J.; Sun, H.; Shen, Z.; Qian, Z.; Zhong, S. Accurate and Efficient Wireless Device Fingerprinting Using Channel State Information. In Proceedings of the IEEE INFOCOM 2018—IEEE Conference on Computer Communications, Honolulu, HI, USA, 16–19 April 2018; pp. 1700–1708. [CrossRef]
10. Kandel, L.N.; Zhang, Z.; Yu, S. Exploiting CSI-MIMO for Accurate and Efficient Device Identification. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6. [CrossRef]
11. Liu, R.; Li, Y.; Zhang, M.; Ding, Z.; Yang, S.; Zhu, S. The Wireless IoT Device Identification based on Channel State Information Fingerprinting. In Proceedings of the 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 11–13 December 2020; Volume 9, pp. 534–541. [CrossRef]
12. Yu, B.; Yang, C.; Ma, J. Continuous Authentication for the Internet of Things Using Channel State Information. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6. [CrossRef]
13. Zaman, S.; Chakraborty, C.; Mehajabin, N.; Mamun-Or-Rashid, M.; Razzaque, M.A. A Deep Learning Based Device Authentication Scheme Using Channel State Information. In Proceedings of the 2018 International Conference on Innovation in Engineering and Technology (ICIET), Dhaka, Bangladesh, 27–28 December 2018; pp. 1–5. [CrossRef]
14. Wang, Q.; Li, H.; Zhao, D.; Chen, Z.; Ye, S.; Cai, J. Deep Neural Networks for CSI-Based Authentication. *IEEE Access* **2019**, *7*, 123026–123034. [CrossRef]
15. Chen, B.; Song, Y.; Zhu, Z.; Gao, S.; Wang, J.; Hu, A. Authenticating Mobile Wireless Device Through Per-packet Channel State Information. In Proceedings of the 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), Taipei, Taiwan, 21–24 June 2021; pp. 78–84. [CrossRef]
16. St. Germain, K.; Kragh, F. Multi-Transmitter Physical Layer Authentication Using Channel State Information and Deep Learning. In Proceedings of the 2020 14th International Conference on Signal Processing and Communication Systems (ICSPCS), Adelaide, SA, Australia, 14–16 December 2020; pp. 1–8. [CrossRef]
17. Liao, R.; Wen, H.; Pan, F.; Song, H.; Xu, A.; Jiang, Y. A Novel Physical Layer Authentication Method with Convolutional Neural Network. In Proceedings of the 2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 29–31 March 2019; pp. 231–235. [CrossRef]
18. Germain, K.S.; Kragh, F. Mobile Physical-Layer Authentication Using Channel State Information and Conditional Recurrent Neural Networks. In Proceedings of the 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring), Helsinki, Finland, 25–28 April 2021; pp. 1–6. [CrossRef]
19. Dai, H.; Shi, W.; Zhou, Z.; Jiang, J. Authentication Method for WiFi Connection of Devices Based on Channel State Information. In Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC), Chengdu, China, 11–14 December 2020; pp. 37–43. [CrossRef]
20. Xia, H.; Shao, S.; Hu, C.; Zhang, R.; Qiu, T.; Xiao, F. Robust clustering model based on attention mechanism and graph convolutional network. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 5203–5215. [CrossRef]
21. Xia, H.; Zhang, R.; Cheng, X.; Qiu, T.; Wu, D.O. Two-stage game design of payoff decision-making scheme for crowdsourcing dilemmas. *IEEE/ACM Trans. Netw.* **2020**, *28*, 2741–2754. [CrossRef]
22. Patil, P.; Patil, M.R.; Itraj, S.; Bomble, U.L. A Review on MIMO OFDM Technology Basics and More. In Proceedings of the 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), Mysore, India, 8–9 September 2017; pp. 119–124. [CrossRef]
23. Halperin, D.; Hu, W.; Sheth, A.; Wetherall, D. Tool Release: Gathering 802.11n Traces with Channel State Information. *SIGCOMM Comput. Commun. Rev.* **2011**, *41*, 53. [CrossRef]
24. Irawan, A.; Putra, A.M.; Ramadhan, H. A DenseNet Model for Joint Activity Recognition and Indoor Localization. In Proceedings of the 2022 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), Bali, Indonesia, 28–30 July 2022; pp. 61–65. [CrossRef]
25. Pelletier, C.; Webb, G.I.; Petitjean, F. Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series. *Remote Sens.* **2019**, *11*, 523. [CrossRef]
26. Cheng, Z.; Rashidi, T.H.; Jian, S.; Maghrebi, M.; Waller, S.T.; Dixit, V. A Spatio-Temporal autocorrelation model for designing a carshare system using historical heterogeneous Data: Policy suggestion. *Transp. Res. Part C Emerg. Technol.* **2022**, *141*, 103758. [CrossRef]

27. Chen, X.; Ma, C.; Allegue, M.; Liu, X. Taming the inconsistency of Wi-Fi fingerprints for device-free passive indoor localization. In Proceedings of the IEEE INFOCOM 2017—IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017; pp. 1–9. [CrossRef]
28. Pu, Q.; Ng, J.K.Y.; Zhou, M.; Wang, J. A joint rogue access point localization and outlier detection scheme leveraging sparse recovery technique. *IEEE Trans. Veh. Technol.* **2021**, *70*, 1866–1877. [CrossRef]
29. Dara, N.; Shankar, P.; Arvind, P.V.; Singh, V. Intelligent Insight into IoT Threats: Leveraging Advanced Analytics with Honeypots for Anomaly Detection. In Proceedings of the 2024 IEEE 9th International Conference for Convergence in Technology (I2CT), Pune, India, 5–7 April 2024; pp. 1–6.
30. Lin, Y.; Gao, Y.; Li, B.; Dong, W. Accurate and robust rogue access point detection with client-agnostic wireless fingerprinting. In Proceedings of the 2020 IEEE International Conference on Pervasive Computing and Communications (PerCom), Austin, TX, USA, 23–27 March 2020; pp. 1–10.
31. Liu, W.; Papadimitratos, P. Position-based Rogue Access Point Detection. *arXiv* **2024**, arXiv:2406.01927.