IET Wireless Sensor Systems

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

Hybrid Model-Based RF Fingerprinting and Spiking Neural Networks for IoT Device Classification

Nadia Adnan Shiltagh Al-Jamali[1] | Ahmed R. Zarzoor[2] | Hamed S. Al-Raweshidy[3] | Talib Mohammed Jawad Abbas[4]

[1]University of Baghdad, Baghdad, Iraq | [2]Ministry of Health, Baghdad, Iraq | [3]Brunel University of London, Uxbridge, UK | [4]Ashur University, Baghdad, Iraq

Correspondence: Hamed S. Al-Raweshidy (Hamed.Al-Raweshidy@brunel.ac.uk)

Received: 15 July 2025 | **Revised:** 8 October 2025 | **Accepted:** 11 October 2025

Handling Editor: Phu H. Phung

Keywords: Internet of Things | neural nets | wireless sensor networks

ABSTRACT

Radio frequency fingerprinting identification (RFFI) leverages the unique features of communication transmitter signals to classify Internet of Things (IoT) devices, enabling individual recognition through waveform analysis. Traditional RFFI methods face challenges in extracting nonlinear features, which machine learning (ML) techniques help overcome by providing advanced wave characteristic analysis. This study introduces RFFI-SCNN, a hybrid model integrating RFFI with a spiking conventional neural network (SCNN) to enhance IoT device authentication within networks. The model operates in two phases: signal processing, where wave data are collected and preprocessed, and SCNN-based classification, where features are extracted and devices are authenticated. The proposed model's performance is evaluated against three ML-based models—1SNN, 1CNN and DCNN—based on accuracy, execution time and memory usage. Experimental results, conducted using a publicly available dataset from the Institute for the Wireless Internet of Things at Northeastern University, indicate that RFFI-SCNN achieves superior accuracy in classifying communication devices compared to 1CNN and 1SNN while also requiring less memory and shorter execution time than DCNN and 1CNN. These findings highlight the effectiveness of RFFI-SCNN in secure and efficient IoT device identification.

1 | Introduction

Information security problems such as reply attacks, hardware cloning and unauthorised user accounts have become a big challenge in confronting how to exactly authenticate and recognise a device in the Internet of Things networks (IoTs). Besides, the rules are applied by the IoT [1, 2]. The conventional authentication methods are performed at the application layer (AP), utilising cryptographic techniques to produce numerical results that are hard for the attacker to fraud. However, the technique has the hazard of key leaks and protocol security gaps. Physical layer (PH) authentication is one of the essential methods to guarantee the security of wireless communication (WC). The PH authentication method provides a wide dais for handling WC security problems. Currently, the research on the PH security authentication method is still incapable of keeping up with the rapid development of other WC authentication methods. Moreover, the extensive PH resources remain largely untapped, despite their considerable potential for further exploration and practical application [3–5]. So, to handle this issue, some researchers proposed a 'radio frequency fingerprinting identification' [6–9] RFFI technique to authenticate and recognise IoT devices. The RFFI is a promising method that exploits substantial features and singular hardware malfunctions (such as power amplifiers, clock skew, filter clock, etc.) as an identifier for the object in the network [10]. The RFF system consists of three steps: feature specification, feature extraction and device authentication [11]. In the first step, the eavesdropping of the wireless device is authenticated, and the fingerprint features are extracted via wave analysis and processing. In the device identification step, the device is authorised based on a matching and identification process performed on the fingerprint features database [12]. These features are generated due to impairment process differences, which cannot be discarded even with sophisticated impairment mechanisms. The hardware features are swerving from ordinary values that lightly affect the signal for a wireless transmission, unless the swerve is in a small range, which cannot affect the ordinary communication operation. The main disadvantage of the traditional RFFI method is that it depends on the hardware quality designed to characterise the extraction method. Machine learning (ML) techniques (such as deep learning) are broadly applied in RFFI because of their powerful capabilities for extracting hardware features. Besides, the ML-based RFFI method can provide superior identification performance [13]. It can be utilised to promptly process the wave to be identified (i.e., it does not need manual 'features' design). Furthermore, ML is used to optimise the RFFI scheme by increasing the accuracy of the device classification.

Unlike traditional neural networks, SCNNs offer biologically inspired temporal processing capabilities that are well-suited for RF signal dynamics. Moreover, RF signals contain rich temporal dynamics, and SCNNs are inherently capable of capturing these features through biologically inspired spike-based processing. Therefore, in this study, we introduce architectural adaptations to SCNNs—including custom spike encoding and convolutional configurations—designed specifically to capture the unique temporal and spectral features of I/Q waveform data. These innovations enable more efficient and accurate device identification in IoT environments. The proposed method, called RFFI, is based on the spiking conventional neural network (RFFI-SCNN) to identify devices in IoT. The RFFI-SCNN model consists of two phases: signal processing and the SCNN classifier. In signal processing, the waves are collected for transmitter parameters of I & Q from the dataset [14], utilising the sliding window technique, to divide the incoming I & Q data to use as inputs for SCNN. The SCNN phase includes two fully connected layers, each layer consists of the leaky integrate-and-fire (LIF) neurons, which perform the classification and recognition process. Also, the model performance has been evaluated by using three metrics (accuracy device authentication, memory usage and execution time) in comparison with the three models: RFFI-based one, conventional neural network (RFFI-1CNN) and RFFI-based one SNN (RFFI-1SNN) model and RFFI-based on deep CNN (RFFI-DCNN). The rest of this paper is organised as follows: Section 2 explores the related works of using ML-based RFFI, Section 3 presents the RFFI-SCNN model and Section 4 demonstrates the study results and discussions. Finally, Section 5 includes the study conclusion.

2 | Related Works

Recent studies in WC have highlighted the distinctiveness and effectiveness of machine learning by demonstrating its ability to learn based on wave classification and specific transmitter identification [15]. For instance, the authors in ref. [16] employed a one-dimensional convolutional neural network (1-CNN) to extract features from RF waveforms, aiming to reduce training time and improve classification accuracy. However, this approach lacks temporal encoding mechanisms, which limits its ability to capture dynamic signal patterns. The authors in ref. [17] introduced a deep learning model based on autoencoders to generate device authentication codes (DACs) by minimising the reconstruction error in RF tracks. Although effective for error reduction, this method does not leverage biologically inspired processing or energy-efficient architectures. The authors in ref. [18] proposed a CNN-based RFFI model to prevent unauthorised access to wireless resources. Their architecture includes two convolutional layers, two pooling layers and a fully connected layer, using I/Q data as inputs. Although the model achieves reliable classification, it relies on dense activation patterns and conventional deep learning structures. The authors in ref. [19] applied support vector data description (SVDD) to mobile IoT devices, identifying unique RF features by enclosing training samples within a minimal-radius hypersphere. Although SVDD offers geometric interpretability, it lacks scalability and adaptability to noisy RF environments. The authors in ref. [20] developed a deep complex residual (DCR) network to enhance RF fingerprinting by extracting correlation features from baseband waveforms. Although powerful, the DCR model is computationally intensive and less suited for low-power IoT applications. The authors in ref. [21] explored a multi-DCNN approach using six different convolutional schemes to optimise identification performance across varying I/Q sample times. This method improves accuracy but increases model complexity and training overhead. The authors in ref. [22] introduced a spiking neural network (SNN) for Wi-Fi frame detection, utilising leaky integrate-and-fire (LIF) neurons and spike timing-dependent plasticity (STDP) learning. Although this work demonstrates the potential of neuromorphic computing, it does not incorporate convolutional layers or spike encoding tailored to RF fingerprinting.

In summary, although prior works have explored various deep learning and SNN-based approaches for RF fingerprinting, they often suffer from high computational costs, limited temporal modelling or lack of neuromorphic compatibility. Our proposed RFFI-SCNN model addresses these gaps by combining spike-based learning with convolutional feature extraction, optimised for RF signal characteristics. Table 1 provides a comparative overview of these methods.

3 | Study Method

The SCNN classifier comprises two fully connected layers using leaky integrate-and-fire (LIF) neurons. To adapt the SCNN architecture for RF fingerprinting, we introduced several key modifications tailored to the characteristics of I/Q waveform data. First, we implemented a custom spike encoding scheme that transforms amplitude and phase variations into spike trains using threshold-based temporal encoding. This preserves the dynamic structure of RF signals while enabling sparse, event-driven processing. Second, the convolutional layers were configured with kernel sizes and stride values optimised to capture short-term dependencies and spectral features inherent in RF

TABLE 1 | A summary RFFI-based ML comparison of existing approaches.

Study	Year	RFFI-based ML	Summary
[16]	2020	1-CNN learning approach	Uses a one-dimensional convolution kernel to extract features from RF signals, aiming to reduce training time and improve classification accuracy.
[17]	2021	Autoencoder algorithm	Applies deep learning to minimise reconstruction error in RF tracks, generating device authentication codes (DACs) for device identification.
[18]	2022	CNN	Utilises I/Q waveform data as input to a CNN with multiple layers; final output classifies authorised IoT devices.
[19]	2020	Support vector data description	Uses support vector data description to enclose training samples in a minimal-radius hypersphere; features are extracted based on support vectors.
[20]	2020	Deep complex residual (DCR)	Extracts correlation features from RF baseband signals using a DCR network to identify the emitter's fingerprint.
[21]	2022	DCNN	Employs six DCNN architectures with varying configurations to optimise identification accuracy across different I/Q sampling intervals.
[22]	2023	SNN	Uses leaky integrate-and-fire (LIF) neurons and spike timing-dependent plasticity (STDP) learning to detect Wi-Fi frames from RF signals.
Propose study		CSNN	Introduces a spiking convolutional neural network with LIF neurons and optimised convolutional layers for RF fingerprinting; achieves efficient classification with low memory usage and fast execution time.

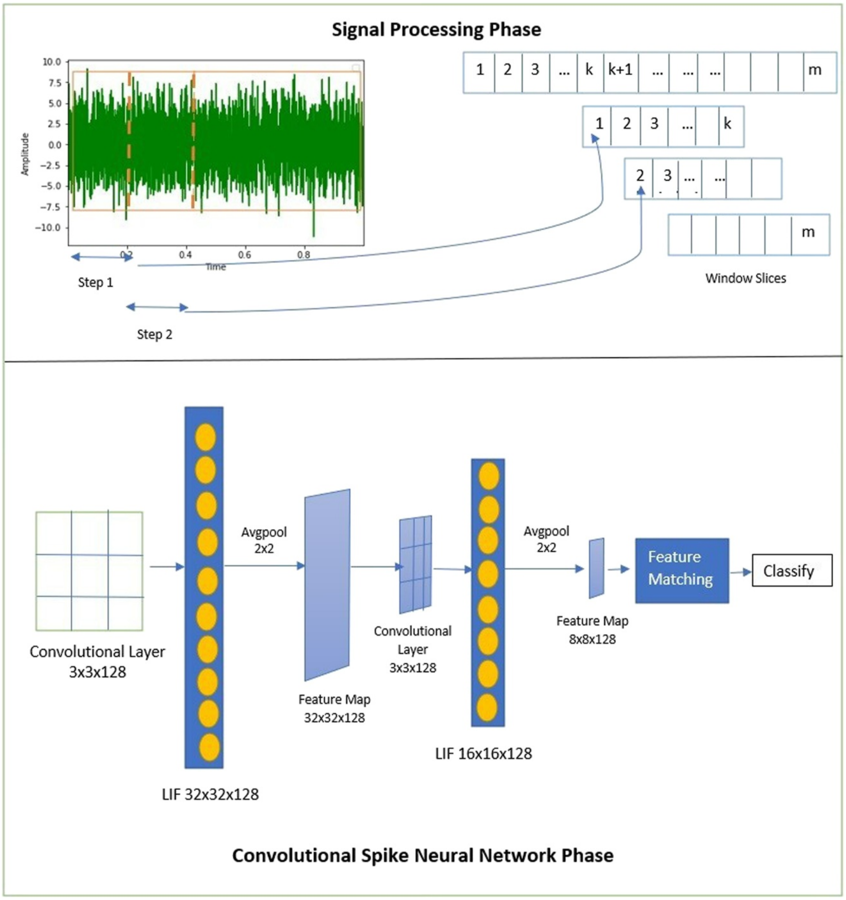


FIGURE 1 | Demonstrates the RFFI-SCNN model phases.

transmissions. These architectural choices enhance the model's ability to extract discriminative features from noisy and variable RF data, improving classification accuracy and robustness in IoT device identification (see Figure 1).

3.1 | Signal Processing Phase

The I/Q wave data used in this study are collected from the Wi-Fi dataset [14]. The emitters consist of 12 NI N20 and 8 NI X310

SDRs, operating with GNU Radio. Each device transmits signals for 30 s using IEEE 802.11 a/g standards at a frequency of 2.432 GHz with a sampling rate of 20 MS/s, utilising BPSK modulation and 20 Ettus VERT2450 antennas. On the receiving end, a single static receiver equipped with an Ettus VERT2450 antenna captures the signal in all cases. The I/Q Wi-Fi data samples are then processed for further analysis.

In this phase, the linear discriminant analysis (LDA) technique is employed to reduce dataset dimensionality and extract relevant features for input into the second stage, CSNN classification. LDA works by projecting training samples onto a linear axis, ensuring that points within the same class are positioned as closely as possible, whereas points from different classes are maximally separated. This approach enhances classification accuracy by improving data differentiation after dimensional reduction. When a new data sample is introduced, it is projected onto the same axis, allowing classification to be determined based on its position relative to existing points. [23]. A Fisher criterion is used with LDA to compute the projection vector (PV) by using Equation (1), where $\acute{\varphi}$ represents the best projection vector (with coefficients φ) that maximises the rate of the S_b between-class scattering (calculated by Equation 2) to S_φ, the within-class-scattering (calculated by Equation 3). In Equation (2), a_i represents samples from a_1, ..., a_n and b_i represents a_i class labels from b_1, ..., b_n. μ_{b_i} represents the mean of the class labels (b_i). In Equation (3), μ_c is the sample mean of the cth class, m is the number of classes, μ is the mean of the entire sample and n_c is the number of samples in c (i.e., data samples in the cth class). Thus, a computed value $\acute{\varphi}$ Equation (1) gives a good PV when the eigenvector has a minimum eigenvalue $S_\varphi = S_b$). However, in most cases, a single PV proves insufficient for accurate recognition across multiple groups.

$$\acute{\varphi} = \underset{\varphi}{\text{argmax}} \frac{\varphi^T \, S_b \, \varphi}{\varphi^T \, S_\varphi \, \varphi}, \tag{1}$$

$$S_b = \sum_{i=1}^{n} \left(a_i - \mu_{b_i}\right)\left(a_i - \mu_{b_i}\right)^T, \tag{2}$$

$$S_\varphi = \sum_{c=1}^{m} n_c(\mu_c - \mu)(\mu_c - \mu)^T, \tag{3}$$

Since SCNN requires a fixed input data length, but the collected receiver data varies in size, the sliding window method [24] is applied to address this issue. This method uses a window of fixed length (L) that moves through the data samples step by step, calculating statistics within each window. The output of each step is a statistical representation of both the current window (L) and the previous step ($L - 1$).

For example, if a wave of length n is received, the sliding window mechanism processes the wave incrementally:

- Step 1: The I/Q wave data are sliced from 1 to k.
- Step 2: The window shifts, slicing from 2 to $k + 1$.
- The process continues until reaching the final step (see Figure 1).

By structuring the wave data into a fixed-length series, the input size remains consistent, preventing gradient disappearance during training and enhancing the stability of SCNN learning features.

3.2 | SCNN Phase

A spiking neural network (SNN) is employed in this study to extract features from the input dataset generated during the signal processing phase. The SNN consists of multiple synaptic neurons, each receiving an input wave and producing an output wave, independent of the actions of other neurons. These neurons exhibit internal dynamics, causing modulation over time. When the neuron's time threshold is exceeded, it resets to an empty state, reducing its membrane potential. Consequently, split input spikes do not trigger a spike or fire [23, 25]. Each neuron is connected to a synapse with an associated weight, which is updated during learning using either supervised or unsupervised methods. The spikes are encoded by converting the input wave into spike trains, a process referred to as 'encoding'. In this study, SCNN consists of two layers, each containing fully interconnected leaky integrate-and-fire (LIF) neurons along with two conventional layers (see Figure 1). The LIF neuron is mathematically defined using Equation (4), where $\tau_m = 10$ represents the time constant, $\text{V}_{\text{mem}}(t)$ and V_{reset} denote the membrane voltage and reset voltage, respectively, whereas $I(t)$ refers to the pre-neuron input current at time t computed using Equation (5). In Equation (5), M represents the number of pre-neurons, w_{xy} is the weight between neuron y in the pre-neuron layer and neuron x in the post-neuron layer and $z_y(t)$ indicates the response of pre-neuron y. The instantaneous membrane voltage $V[t]$ is determined using Equations (6–8) [26, 27], where $V[t]$ describes the voltage before a neuron fires and after it is charged. Additionally, $E[t]$ represents the emitted energy via the neuron, whereas V_{th} is the membrane threshold voltage.

$$\tau_m \frac{dV(t)}{dt} = -(V_{\text{mem}}(t) - V_{\text{reset}}) + I(t), \tag{4}$$

$$I(t) = \sum_{y=1}^{M} w_{xy} z_y(t), \tag{5}$$

$$\begin{aligned} V[t] = f(V[t-1], I[t]) = V[t-1] \\ + \frac{1}{\tau_m}\left(-V[t-1] - V_{\text{reset}}\right) + I[t], \end{aligned} \tag{6}$$

$$[t] = f(V[t-1], I[t]), \tag{7}$$

$$E[t] = s(V[t] - V_{\text{th}}). \tag{8}$$

The first convolutional layer has a size of 3 × 3 × 128, taking input from the signal processing phase and passing it to the first LIF layer (32 × 32 × 128). The spiking output from the LIF layer is gathered along the time dimension and processed using an average pooling layer (2 × 2), reducing the feature map size to 32 × 32 × 128 while preserving essential features. This process is repeated as the output data are fed into the second

LIF layer (16 × 16 × 128). Again, the spiking output is processed using an average pooling layer (2 × 2), further refining the feature map to 8 × 8 × 128. To illustrate the spike time transformation, see Figure 2. The LIF neuron synchronously emits spikes across multiple time steps. The input spike signal tensor (feature map channel size 4 × 4) is divided into four-time steps, ensuring consistent spike processing throughout the network. The kernel size (3 × 3) remains synchronised across all time steps, ensuring that the output tensor captures potentials at every step. Since spikes are stored accumulatively, the potentials are progressively enhanced, improving feature extraction. The final feature map (8 × 8 × 128) is passed to the next layer, which consists of two groups, each containing 128 leaky integrate-and-fire (LIF) neurons, responsible for signal classification (i.e., wave identification). To train the LIF neurons, a surrogate gradient [28, 29] is utilised, with a sigmoid function performing back-propagation, as defined in Equation (9). For forward-propagation, a step function is employed to differentiate binary states (0 and 1s) within the LIF neurons. The step function is computed using Equations (10) and (11), where ∅ represents the Heaviside step function, and φ corresponds to the Dirac–Delta function.

$$a(x) = \text{sigmid}\ [\alpha\, x] = \frac{1}{1 + e^{-\alpha x}}, \tag{9}$$

$$S[t] = \varnothing(V[t] - V_{\text{th}}), \tag{10}$$

$$\frac{\partial S}{\partial V} = \varphi\ (V - V_{\text{th}}) \in \{0, 1\}. \tag{11}$$

4 | Results and Discussions

The RFFI-SCNN model has been implemented on a laptop type Lenovo CPU speed 2.8 GHz Intel Core i7, RAM 8 GB and operating system MS Windows 10. Four experiments have been conducted to evaluate the performance of the model: Since SCNN requires a fixed input length, the first experiment employed the RFFI-based Single-SNN (RFFI-1SNN) model, which used a single fully connected layer with 128 LIF neurons for signal identification. The second experiment adopted the RFFI-DCNN model as proposed in [21], while the third utilized the RFFI-1CNN model from [16]. The four experiments have been implemented by using the Python language (version 3.8). The SnnTorch library has been used to implement RFFI-1SNN and RFFI-SCNN, whereas TensorFlow, Keras and PyTorch libraries have been used to apply RFFI-1CNN and RFFI-DCNN. In the four experiments, a signal-to-noise ratio (SNR) utilised values (0 dB to −15 dB), and time frequency-map (value 200 and 600) is used for testing and training, respectively, and scaled to a size of 32 × 32 utilising bicubic interpolation, see Table 2.

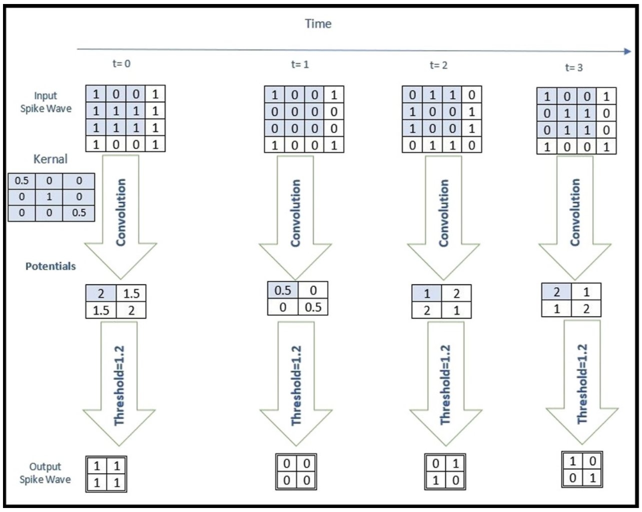
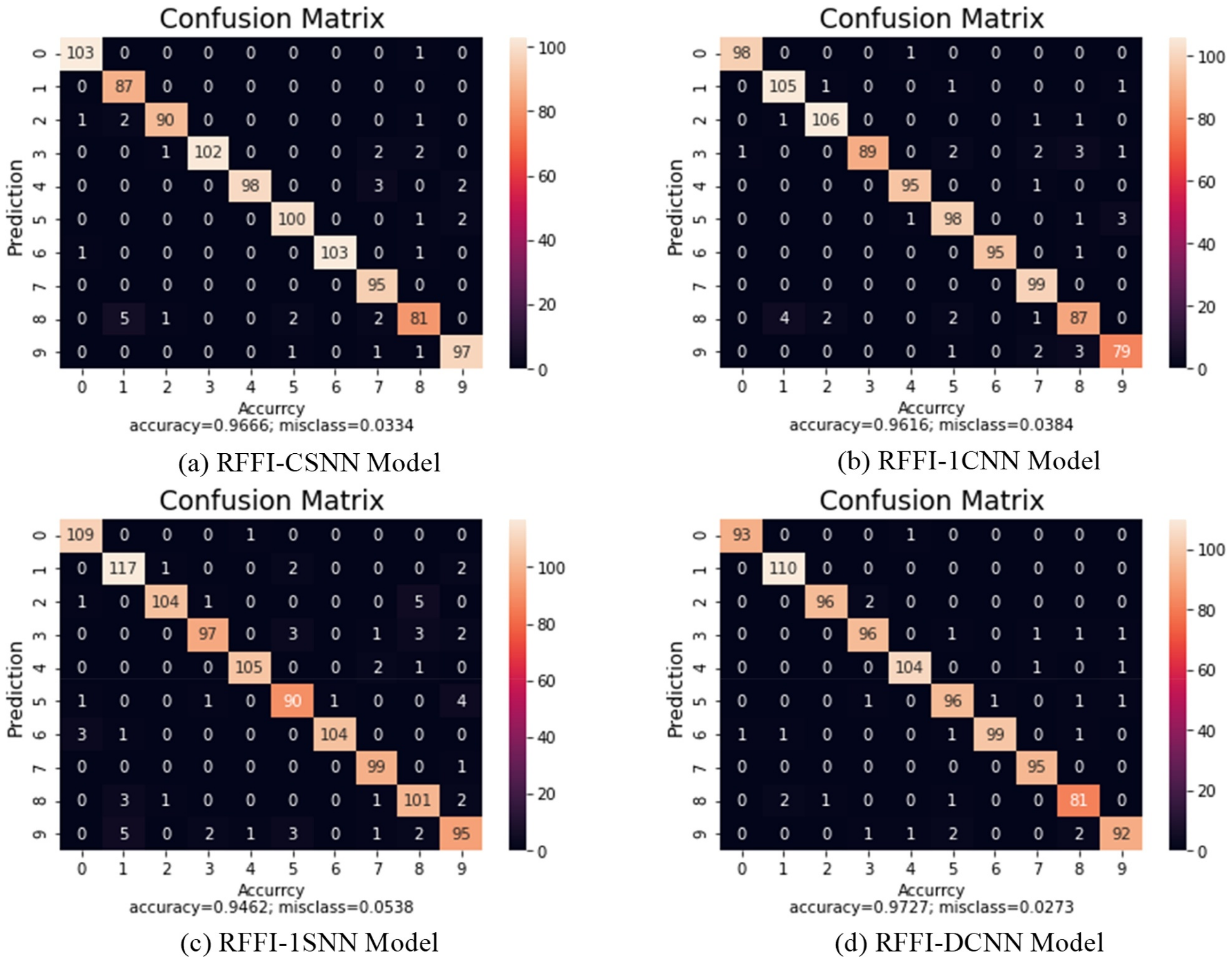


FIGURE 2 | Transforming spike times process for the input spike-wave 4 × 4 channel.

During the signal processing phase, the I/Q sample wave data are collected from the Wi-Fi dataset [14], with 10,000 samples gathered from each radiation source. The window sliding length is set to 128, and each data sample consists of two channels: I and Q. The dataset is split into 75% for training and 25% for testing across the four experiments. To evaluate model performance, three key metrics are used: signal identification accuracy, memory utilisation and execution time. Accuracy is measured using a confusion matrix to assess prediction performance. Among the four experiments, the RFFI-DCNN model achieves the highest accuracy (0.972) and lowest misclassification rate (0.027) compared to other models. The proposed RFFI-SCNN model demonstrates strong performance with 0.966 accuracy and 0.033 misclassification rate, outperforming RFFI-1CNN (accuracy: 0.961, misclass: 0.038) and RFFI-1SNN (accuracy: 0.946, misclass: 0.053; see Figure 3). The resource computation metrics, including memory utilisation and execution time, are assessed during both the training and testing phases. Figure 4 presents the lowest total memory usage (in KiB) for each model: RFFI-SCNN (2800 KiB), RFFI-1CNN (5000 KiB), RFFI-1SNN (2500 KiB) and RFFI-DCNN (5200 KiB). The highest memory usage observed was 4500 KiB (RFFI-SCNN), 6500 KiB (RFFI-1CNN), 4000 KiB (RFFI-1SNN) and 6800 KiB (RFFI-DCNN). Regarding execution time, RFFI-SCNN and RFFI-1SNN performed faster compared to RFFI-1CNN and RFFI-DCNN, as illustrated in Figure 5. During the testing phase, Figure 6 shows the lowest total memory usage: RFFI-SCNN (800 KiB), RFFI-1CNN (1200

TABLE 2 | Summary of the three experiments parameters.

Parameter	Value
SNR	Rang from 0 dB to −15 dB
Time frequency map for training	600 MHz
Time frequency map for testing	200 MHz
Time frequency map size	32 × 32
Sampling rate	2 GHz
Time width	0.4 µs
Bandwidth	60 MHz
Learning rate	0.1



(a) RFFI-CSNN Model

(b) RFFI-1CNN Model

(c) RFFI-1SNN Model

(d) RFFI-DCNN Model

FIGURE 3 | Illustrates the identifying results of the individual communication radiance source based on RFFI-CSNN, RFFI-1CNN, RFFI-1SNN and RFFI-DCNN model, respectively.

KiB), RFFI-1SNN (1000 KiB) and RFFI-DCNN (13,400 KiB). The highest allocated memory values recorded were 1200 KiB (RFFI-SCNN), 1900 KiB (RFFI-1CNN), 1480 KiB (RFFI-1SNN) and 2380 KiB (RFFI-DCNN). Figure 7 further demonstrates that RFFI-SCNN and RFFI-1SNN required less execution time compared to RFFI-1CNN and RFFI-DCNN. Although these results highlight the efficiency of RFFI-SCNN in terms of memory and execution time, it is important to note that we did not perform cross-device generalisation testing. We agree that this is a critical aspect for practical deployment, especially in dynamic IoT environments where new or similar devices may be introduced. We will add a note in the manuscript discussing the importance of evaluating model generalisation on larger and more diverse datasets in future work. However, the superior performance of RFFI-SCNN in terms of accuracy, execution time and memory usage can be attributed to the architectural adaptations made for RF data. The spike encoding scheme and tailored convolutional configurations allowed the SCNN to efficiently process temporal signal features, resulting in more robust classification. These findings validate the effectiveness of our design choices and highlight the potential of SCNNs in neuromorphic RF fingerprinting applications.

5 | Conclusion

This study aims to enhance physical layer (PH) security in IoT device recognition by leveraging transmitter wave features through the development of a hybrid model, RFFI-SCNN. The model operates in two phases: signal processing and SCNN-based classification. In the signal processing phase, I/Q wave data are collected and preprocessed using the sliding window method to ensure consistency before being fed into the next stage. The SCNN phase extracts wave features from the processed data and executes the classification process, enabling accurate device identification within the network. The proposed RFFI-SCNN model has been evaluated against three models—RFFI-1SNN, RFFI-1CNN and RFFI-DCNN—using three performance metrics: accuracy, execution time and memory usage. In terms of accuracy, RFFI-SCNN demonstrates superior performance compared to RFFI-1SNN and RFFI-1CNN. Regarding execution time, RFFI-SCNN achieves the shortest processing duration, outperforming RFFI-1CNN and RFFI-DCNN. Although the results highlight the efficiency and accuracy of RFFI-SCNN, future work is needed to address its resilience against security threats such as spoofing, replay attacks and signal injection. These vulnerabilities are

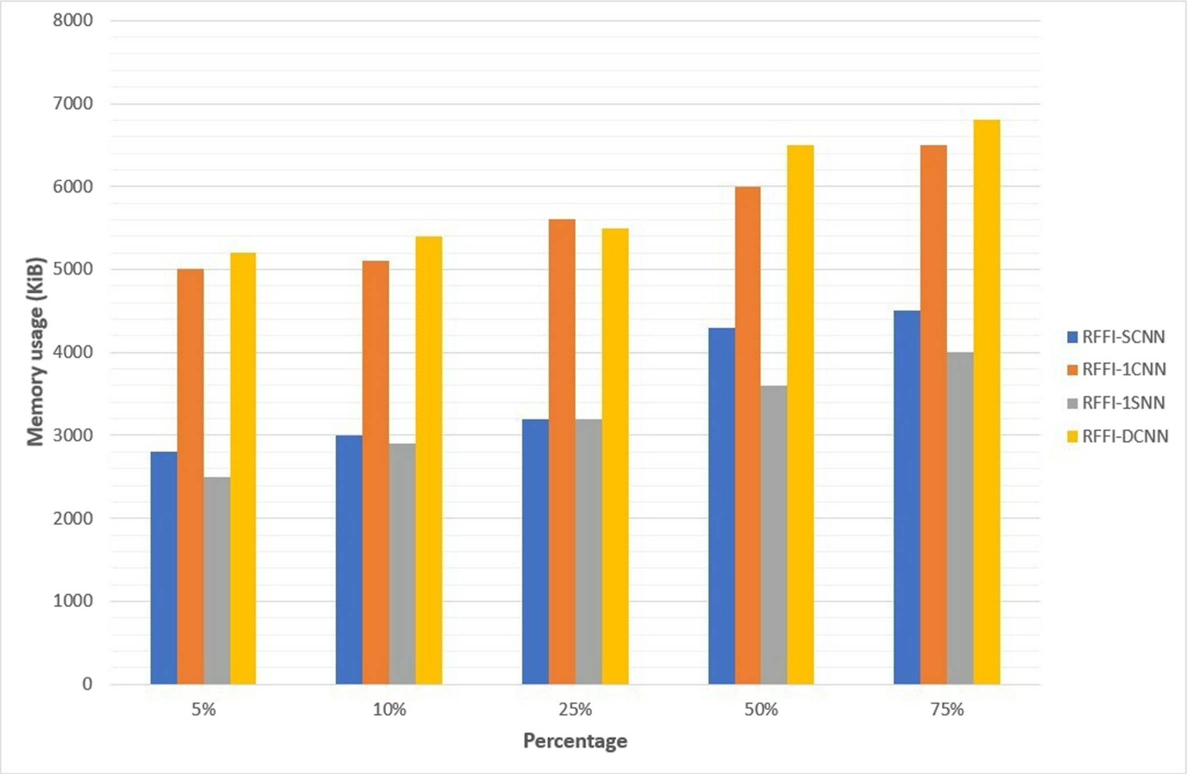


FIGURE 4 | Illustrates memory usage in the training phase for the four experiments.

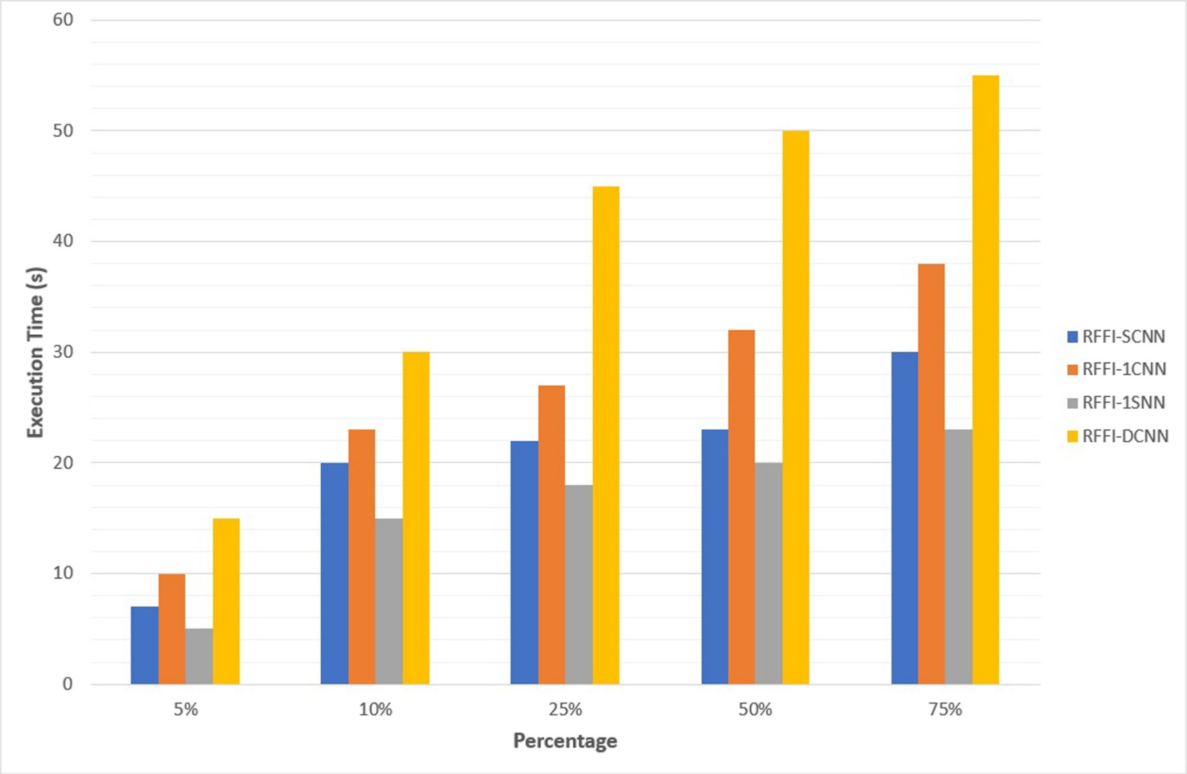


FIGURE 5 | Illustrates the execution time in the training phase for the four experiments.

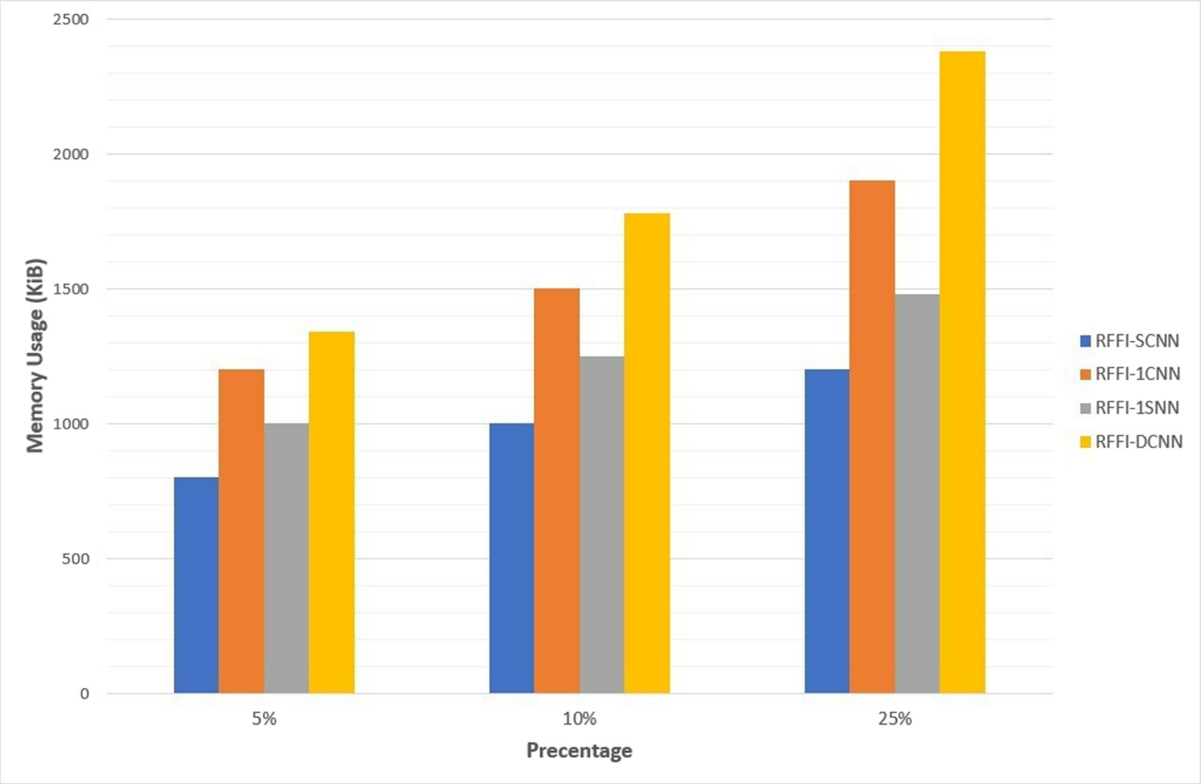


FIGURE 6 | Illustrates memory usage in the testing phase for the four experiments.

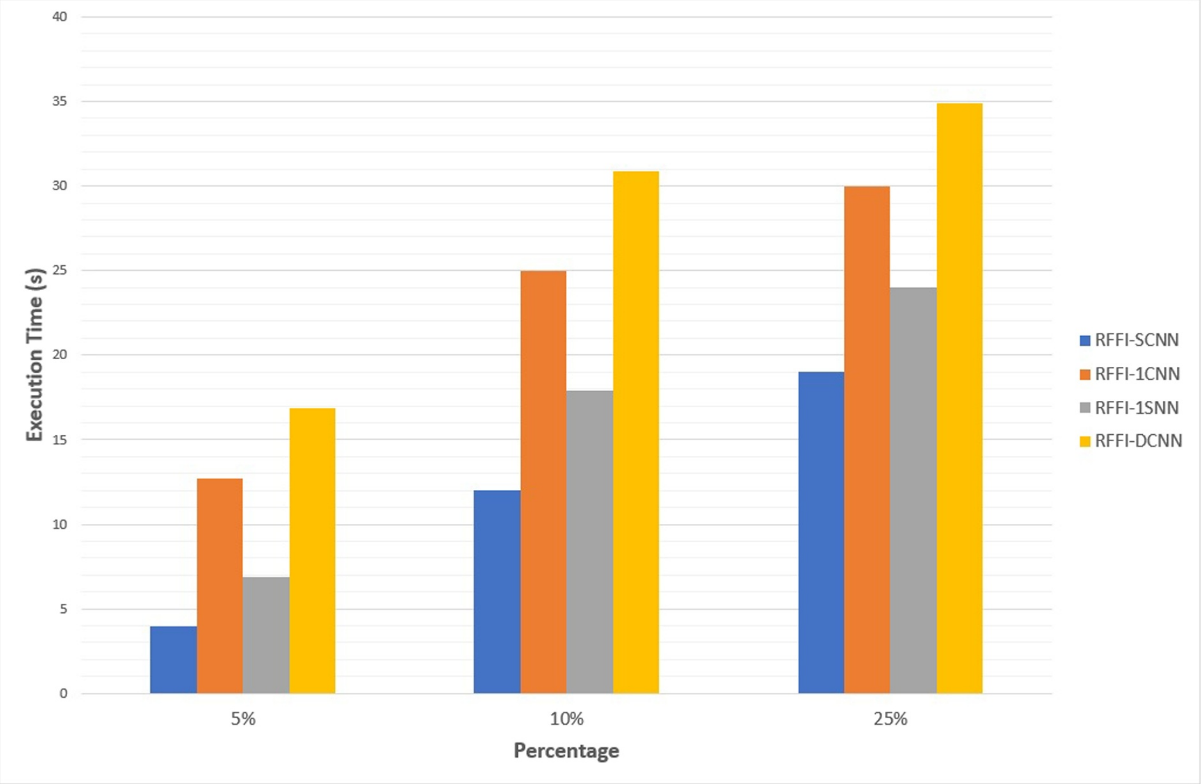


FIGURE 7 | Illustrates the execution time in the testing phase for the four experiments.

critical in practical deployments, and we will include a discussion in the revised manuscript to emphasise the importance of securing RFFI systems against such adversarial scenarios.

Author Contributions

Nadia Adnan Shiltagh Al-Jamali: conceptualization, data curation, investigation, methodology, project administration, software, supervision, validation, visualization, writing – original draft, writing – review and editing. **Ahmed R. Zarzoor:** methodology, project administration, resources, software, validation, writing – original draft, writing – review and editing. **Hamed S. Al-Raweshidy:** formal analysis, funding acquisition, investigation, methodology, project administration, resources, supervision, validation, visualization, writing – original draft, writing – review and editing. **Talib Mohammed Jawad Abbas:** conceptualization, data curation, software, validation, visualization, writing – original draft.

Acknowledgements

The authors would like to appreciate all the excellent suggestions of anonymous reviewers to enhance the quality of this paper.

Funding

This study was supported by Brunel University London.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. S. J. Akuma and E. G. AbdAllah, "Impacts of Radio Frequency Identification (RFID) Technology in Business Continuity," in *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)* (2022), 1544–1549.

2. H. Taherdoost, "Security and Internet of Things: Benefits, Challenges, and Future Perspectives," *Electronics* 12, no. 1901 (2023): 1–22, https://doi.org/10.3390/electronics12081901.

3. K. Cao, H. Ding, B. Wang, et al., "Enhancing Physical-Layer Security for IoT With Nonorthogonal Multiple Access Assisted Semi-Grant-Free Transmission," *IEEE Internet of Things Journal* 9, no. 24 (2022): 24669–24681, https://doi.org/10.1109/jiot.2022.3193189.

4. Q. Wei, B. Wang, and K. Cao, "Physical Layer Security Technology Based on Nonorthogonal Multiple Access Communication," *Mobile Information Systems* 2022 (2022): 1–11: Article ID 6303210, https://doi.org/10.1155/2022/6303210.

5. P. Rojas, S. Alahmadi, and M. Bayoumi, "Physical Layer Security for IoT Communications—A Survey," in *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)* (2021), 95–100.

6. G. Shen, J. Zhang, A. Marshall, M. Valkama, and J. Cavallaro, "Radio Frequency Fingerprint Identification for Security in Low-Cost IoT Devices," in *2021 55th Asilomar Conference on Signals, Systems, and Computers* (2021), 309–313.

7. J. Feng, T. Zhao, S. Sarkar, et al., "Fingerprinting IoT Devices Using Latent Physical Side-Channels", *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, no. 2 (2023): 1–2: Article 54, https://doi.org/10.1145/3596247.

8. W. Wu, S. Hu, D. Lin, and T. Yang, "Radio-Frequency Fingerprinting for Distributed IoT Networks: Authentication and QoS Optimization," *IEEE Systems Journal* 17, no. 3 (September 2023): 4440–4451, https://doi.org/10.1109/jsyst.2023.3248685.

9. C. Shang, J. Cao, T. Zhu, Y. Zhang, B. Niu, and H. Li, "CADFA: A Clock Skew-Based Active Device Fingerprint Authentication Scheme for Class-1 IoT Devices," *IEEE Systems Journal* 18, no. 1 (March 2024): 590–599, https://doi.org/10.1109/jsyst.2024.3351222.

10. J. Zhang, R. Woods, M. Sandell, M. Valkama, A. Marshall, and J. Cavallaro, "Radio Frequency Fingerprint Identification for Narrowband Systems, Modelling and Classification," *IEEE Transactions on Information Forensics and Security* 16 (2021): 3974–3987, https://doi.org/10.1109/tifs.2021.3088008.

11. A. Jagannath, J. Jagannath, P. Sagar, and P. Kumar, "A Comprehensive Survey on Radio Frequency (RF) Fingerprinting: Traditional Approaches, Deep Learning, and Open Challenges," *Computer Networks* 219, no. 109455 (2022): 1–31, https://doi.org/10.1016/j.comnet.2022.109455.

12. N. Soltanieh, Y. Norouzi, Y. Yang, and N. C. Karmakar, "A Review of Radio Frequency Fingerprinting Techniques," *IEEE Journal of Radio Frequency Identification* 4, no. 3 (2020): 222–233, https://doi.org/10.1109/jrfid.2020.2968369.

13. G. Shen, J. Zhang, and A. Marshall, "Deep Learning-Powered Radio Frequency Fingerprint Identification: Methodology and Case Study," in *IEEE Communications Magazine* (2023), 1–7.

14. *Institute for the Wireless Internet of Things* (Northeastern University), Accessed July. 7, 2023 [Online], https://wiot.northeastern.edu/wp-content/uploads/2020/07/dataset_release.pdf.

15. L. Alhoraibi, D. Alghazzawi, R. Alhebshi, and O. Rabie, "Physical Layer Authentication in Wireless Networks-Based Machine Learning Approaches," *Sensors* 23, no. 4 (2023): 1–34, https://doi.org/10.3390/s23041814.

16. F. Xie, H. Wen, J. Wu, S. Chen, W. Hou, and Y. Jiang, "Convolution Based Feature Extraction for Edge Computing Access Authentication," *IEEE Transactions on Network Science and Engineering* 7, no. 4 (2020): 2336–2346, https://doi.org/10.1109/tnse.2019.2957323.

17. J. Bassey, X. Li, and L. Qian, "Device Authentication Codes Based on RF Fingerprinting Using Deep Learning," *EAI Endorsed Transactions on Security and Safety* 8, no. Issue 29 (2021): 1–17, https://doi.org/10.4108/eai.30-11-2021.172305.

18. W. Wu, S. Hu, D. Lin, and G. Wu, "Reliable Resource Allocation With RF Fingerprinting Authentication in Secure IoT Networks," *Science China Information Sciences* 65, no. 7 (2022): 1–16, https://doi.org/10.1007/s11432-021-3284-y.

19. Q. Tian, Y. Lin, X. Guo, J. Wang, O. AlFarraj, and A. Tolba, "An Identity Authentication Method of a MIoT Device Based on Radio Frequency (RF) Fingerprint Technology," *Sensors* 20, no. 1213 (2020): 1–18, https://doi.org/10.3390/s20041213.

20. S. Wang, H. Jiang, X. Fang, Y. Ying, J. Li, and B. Zhang, "Radio Frequency Fingerprint Identification Based on Deep Complex Residual Network," *IEEE Access* 8 (2020): 204417–204424, https://doi.org/10.1109/access.2020.3037206.

21. Y. Wu and Li Hao, "Radio Frequency Fingerprint Recognition Based on Deep Transfer Learning," in *Proceedings of SPIE 12252, International Conference on Biometrics, Microelectronic Sensors, and Artificial Intelligence (BMSAI)*, Vol. 12252 (2022), 1–8.

22. H. Lee, D. Kim, and J. Lim, “Wi-Fi Frame Detection via Spiking Neural Networks With Memristive Synapses,” *Computer Communications* 208 (2023): 256–270, https://doi.org/10.1016/j.comcom.2023.06.006.

23. A. R. Zarzoor, N. A. Al-Jamali, and D. A. Abdul Qader, “Intrusion Detection Method for Internet of Things Based on the Spiking Neural Network and Decision Tree Method,” *International Journal of Electrical and Computer Engineering* 13, no. 2 (2023): 2278–2288, https://doi.org/10.11591/ijece.v13i2.pp2278-2288.

24. J. Xiang, Y. Zhu, R. Wu, R. Xu, Y. Ishiwaka, and C. Zheng, “Dynamic Sliding Window for Realtime Denoising Networks,” in *ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2022), 361–365.

25. A. R. Zarzoor, N. A. Al-Jamali, and I. R. K. Al-Saedi, “Traffic Classification of IoT Devices by Utilizing Spike Neural Network Learning Approach,” *Mathematical Modelling of Engineering Problems* 10, no. 2 (2023): 639–646, https://doi.org/10.18280/mmep.100234.

26. X. Fang, D. Liu, S. Duan, and L. Wang, “Memristive LIF Spiking Neuron Model and Its Application in Morse Code,” *Frontiers in Neuroscience* 16, no. 853010 (2022): 1–17, https://doi.org/10.3389/fnins.2022.853010.

27. S. Xiang, S. Jiang, X. Liu, T. Zhang, and L. Yu, “Spiking VGG7: Deep Convolutional Spiking Neural Network With Direct Training for Object Recognition,” *Electronics* 11, no. 2097 (2022): 1–13, https://doi.org/10.3390/electronics11132097.

28. A. Bittar and P. N. Garner, “A Surrogate Gradient Spiking Baseline for Speech Command Recognition,” *Frontiers in Neuroscience* 16, no. 86589 (2022): 1–18, https://doi.org/10.3389/fnins.2022.865897.16

29. G. O. Young, “Synthetic Structure of Industrial Plastics,” in *Plastics*, ed. J. Peters. 2nd ed., Vol. 3 (McGraw-Hill, 1964), 15–64.