Towards a Lightweight Edge AI-based Radio Frequency Fingerprinting

Ahmed Hussain*, Nada Abughanam†, and Panos Papadimitratos*
*Networked Systems Security (NSS) Group – KTH Royal Institute of Technology, Stockholm, Sweden
†Electrical Engineering Department – Qatar University, Doha, Qatar
ahmed.hussain@ieee.org, nada.abughanam@qu.edu.qa, papadim@kth.se

***Abstract*—The deployment of Internet of Things (IoT) devices requires efficient security mechanisms. However, cryptographic solutions often prove resource-intensive. Radio Frequency Fingerprinting (RFF) enables device authentication through the intrinsic characteristics of RF signals at the Physical (PHY)-layer. Deploying RFF presents two challenges: ensuring operational efficiency and scalability in resource-constrained environments. This paper presents a lightweight Edge AI-based RFF model for device authentication using PHY-layer characteristics. Our approach implements a Deep Learning (DL) model to extract device-specific features from IQ samples, converted using TensorFlow Lite for edge deployment. Evaluation on Raspberry Pi demonstrates high accuracy ($>$ 0.95) and Receiver Operating Characteristic-Area Under the Curve (ROC-AUC) scores ($>$ 0.90), while maintaining a compact model size suitable for resource-constrained environments.**

***Index Terms*—Artificial Intelligence, Authentication, Convolutional Neural Networks, Edge AI, Internet of Things, Physical Layer Security, Radio Frequency Fingerprinting, Smart Cities, TensorFlow, TinyML, Wireless Security**

I. Introduction

The growth of the Internet of Things (IoT) [1], [2], supported by the evolution of networks from 4G to 5G and beyond, has fundamentally transformed sectors through pervasive connectivity among heterogeneous devices. IoT devices, deployed at the network Edge for smart home, healthcare, industrial automation, and smart city applications [3], face significant security challenges due to their inherent resource constraints and massive scale deployment. Device-generated data legitimacy, authenticity, and integrity remain critical in this landscape, with the vast number of interconnected devices being an extending attack surface, with data breaches, tampering, and unauthorized access. While cryptographic protocols provide authentication and integrity, they can be hard to implement in small-footprint, resource-constrained devices, particularly for IoT devices such as backscatter tags [4].

Radio Frequency (RF) Fingerprinting (RFF) emerges as a promising alternative, leveraging edge device capabilities to implement security mechanisms. RFF utilizes the Physical (PHY)-layer signal characteristics for wireless device authentication [5], exploiting unique transmitter imperfections as device fingerprints. These fingerprints, captured through In-phase (I) and Quadrature (Q) components (together termed IQ samples), enable reliable authentication without the cryptographic overhead [6], even for devices operating on identical protocols and frequency bands. Artificial Intelligence (AI)-based RFF enhances traditional approaches by leveraging Machine Learning (ML), specifically Deep Learning (DL), to improve feature extraction and identification accuracy [7].

Although RFF emerges as an alternative to cryptographic protocols, the fundamental challenge remains to efficiently deploy RFF within resource-constrained environments. While edge devices surpass basic IoT endpoints in capability, they lack the computational resources of cloud infrastructure. Despite their capabilities, devices such as Raspberry Pi or NVIDIA Jetson face constraints in processing power, memory, and energy consumption, challenging the deployment of resource-intensive AI-based RFF systems.

Recent contributions ([8], [9]) explored RFF-based authentication at the edge, with [8] employing transfer learning through ResNet50. However, these are not lightweight architectures specifically designed for RFF applications. To address this limitation:

1) We present optimized AI-based RFF implementations tailored for edge deployment, focusing on efficient training and robust authentication without performance degradation.
2) We introduce a lightweight DL architecture specifically designed for PHY-layer authentication on edge devices. Unlike [8], [9], our model achieves high accuracy without the need to use transfer learning or pre-trained models.
3) We utilize TensorFlow Lite optimization to demonstrate deployment feasibility on resource-constrained edge devices while maintaining performance.
4) We evaluate our implementation across 28 transmitters using an open source dataset [10], demonstrating high accuracy and Receiver Operating Characteristic-Area Under the Curve (ROC-AUC).

Paper Organization. Section II provides an overview of the preliminaries and related work. Section III details the considered system model and states the assumptions. Section IV discusses the implementation of the proposed lightweight framework. Section V presents a comprehensive performance evaluation. Finally, Section VI concludes the paper with a summary of our findings and potential future research directions.

II. Preliminaries and Related Work

This section discusses the concepts essential for understanding the rest of the paper, specifically Deep Learning, TinyML,

and RF Fingerprinting.

A. Deep Learning and TinyML

Deep Learning (DL) is a subset of Machine Learning (ML) utilizing multi-layered Neural Networks (NNs) for complex task processing, including Natural Language Processing (NLP) and signal analysis. Convolutional Neural Networks (CNNs), a specialized NN architecture, excels in image processing through convolutional layers where parametrized filters extract hierarchical features [11]. This architecture enables progressive pattern learning, from basic features to complex representations, making CNNs particularly effective for recognition and detection tasks.

TinyML enables the deployment of DL models on resource-constrained edge devices, shifting from traditional server-based processing to localized computation. This paradigm proves essential for smart applications, including healthcare, urban systems, and mobility solutions. Edge-AI implementation facilitates real-time inference without server dependency [12]. Model optimization techniques, including quantization and pruning [13], ensure efficient deployment while maintaining performance.

B. Radio Frequency Fingerprinting

RF Fingerprinting (RFF) leverages inherent hardware imperfections in RF devices for identification. Device-specific signal characteristics [14] enable device authentication through three key phases [7]: signal acquisition and preprocessing, model training, and deployment. The implementation process begins with controlled signal acquisition, capturing device-specific IQ samples. Preprocessing encompasses feature selection, data augmentation, and noise reduction. The DL model training phase focuses on extracting distinctive RF fingerprints from preprocessed samples. Recent research [14], [15] advanced feature extraction methodologies for effective DL-based RFF implementation.

C. Related Work

Edge security can be enhanced by Artificial Intelligence (AI) that enables real-time attack detection, anomaly detection, and automated recovery mechanisms [16]. AI-based security solutions facilitate rapid analysis of IoT-generated data to identify attack patterns [17]. Tiny Machine Learning (TinyML) [18] advances this capability by enabling AI models deployment on resource-constrained devices, effectively bringing intelligence closer to data sources while enhancing overall IoT network security. In the RFF domain, several approaches demonstrated promising results. Sun et al. [19] developed a transformer-based multi-feature extraction network for Bluetooth device identification, achieving 0.93 accuracy through their multi-scale feature analysis approach. [20] proposed a lightweight CNN for mobile RFF, reaching 0.85 accuracy across 16 USRP devices using IQ signal inputs and time-domain feature extraction. [21] combined coherent integration, multiresolution analysis, and Gaussian Support Vector Machine (SVM) to optimize RF fingerprint classification, focusing on Signal-to-Noise Ratio (SNR) improvement and dimensionality reduction.

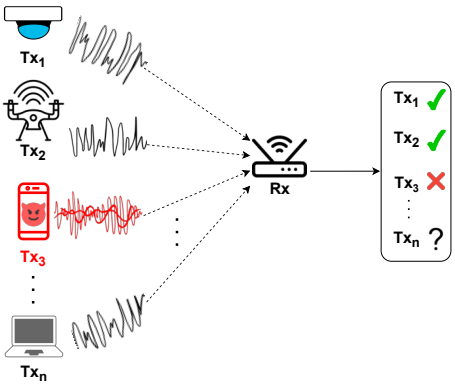


Fig. 1: RFF-based device identification system model, illustrating the interaction between multiple IoT transmitters (Tx_1,...,Tx_n) and an edge-deployed access point (AP), the receiver (Rx). Rx processes unique RF signal characteristics to authenticate', i.e., identify transmissions by legitimate devices (✓) while rejecting unauthorized transmission attempts (✗).

Edge deployment strategies have received particular attention in recent research. [8] introduced structured pruning techniques to compress CNN layers, enabling implementation on resource-constrained devices while maintaining authentication accuracy. [9] explored federated learning approaches for RFF authentication, addressing resource allocation challenges across local and edge computation scenarios, focused on optimizing network delay in various computational configurations. [22] presented a transformer-based DL model for LoRA device signal classification, specifically addressing variable-length signal challenges and low SNR conditions, incorporating data augmentation and multi-packet inference to enhance model robustness. Building on this work, [23] developed a scalable deep metric learning approach for LoRa authentication, utilizing channel-independent spectrograms for RF fingerprint extraction and implemented k-Nearest Neighbor (k-NN) algorithms for device classification.

III. System Model and Problem Statement

We consider a system model, depicted in Fig. 1, where multiple IoT transmitters ($\text{Tx}_1, \text{Tx}_2, \ldots, \text{Tx}_n$) interact with an edge-deployed Access Point (AP) equipped with receiver Rx. The AP implements RF fingerprinting through a pre-trained DL model, processing IQ samples to authenticate devices based on their PHY-layer characteristics. The model evaluates incoming signals against known authorized device fingerprints, rejecting unauthorized access attempts when RF signatures do not match those the model was trained on. This authentication framework faces two fundamental challenges. First, the open-set classification problem ([24], [25]): traditional RFF approaches struggle with unknown device generalization. The system requires comprehensive training data from

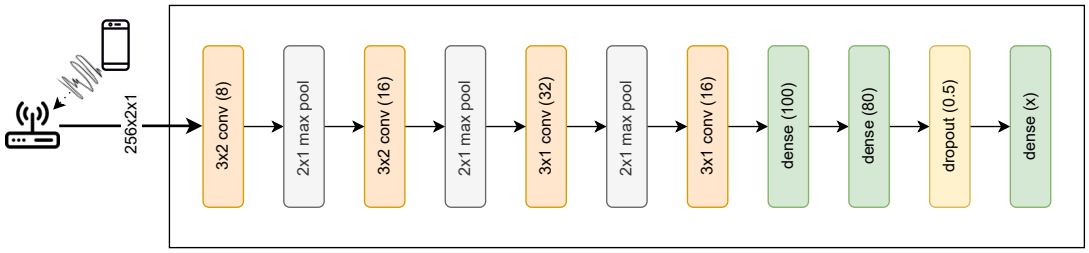


Fig. 2: Structure and details of the implemented CNN.

authorized devices, with regular model updates to maintain authentication robustness. While advanced techniques, such as contrastive learning, offer potential solutions for unknown device handling [25], our work focuses on closed-set scenarios: device classes are known during training. Second, edge deployment presents specific constraints: (a) Model size limitations affecting inference time performance, and (b) Accuracy degradation during model optimization, particularly through quantization of weights and activations. These constraints are addressed in Section IV and Section V, respectively, focusing on practical implementation strategies for resource-constrained environments.

IV. Lightweight Edge AI-based RFF

This section discusses the Deep Learning architecture implementation and the TinyML conversion process.

A. Deep Learning

Fig. 2 illustrates the structure and details of the implemented CNN. The implementation is done using the TensorFlow and Keras libraries. The input layer reshapes the data to (256, 2, 1) to accommodate the 2D convolution operations. The model includes a structured series of convolutional and max-pooling layers arranged in a pattern of decreasing spatial dimensions and increasing feature depth. The first convolutional block employs 8 filters with a kernel size of (3, 2), followed by a max-pooling layer with a pool size of (2, 1). This is followed by the second convolutional block with 16 filters of the same kernel size and pooling configuration. The third block features 32 filters with a kernel size of (3, 1) and identical pooling. The fourth block reduces the filter count to 16, maintaining the kernel size (3, 1) without subsequent pooling to retain sufficient resolution for dense layers. Post convolution, a flattening layer converts the 3D tensor output to a 1D vector. This vector is processed through two dense layers: (i) with 100 units and (ii) with 80 units, both employing Rectified Linear Unit (ReLU) activation and L2 regularization with a factor of 0.0001 to mitigate overfitting. A dropout layer with a rate of 0.5 is also incorporated to further prevent overfitting. The output layer consists of a dense layer with units equal to the number of device classes, utilizing the softmax activation function and L2 regularization. Finally, the model is compiled using the Adam optimizer, with sparse categorical cross-entropy as the loss function.

This lightweight CNN architecture effectively captures spatial and temporal features from the input signals through the convolutional and pooling layers, while the dense layers and the dropout layers ensure robust feature learning and generalization. Table I lists all the parameters and specifications for the developed CNN model. Several optimization techniques, including quantization, pruning, and clustering, can be applied to minimize model size and latency while maintaining precision. We adopted the same steps as in [26] to convert the regular CNN model to a deployable and lightweight model on edge devices using TensorFlow Lite (TFLite).

TABLE I: CNN model parameters summary.

Parameter	Specs
Input Shape	(256, 2, 1)
Convolution Layers	4 layers (8, 16, 32, 16 filters respectively)
Kernel Size	(3, 2) and (3, 1)
Pooling Layers	3 Max Pooling (pool size: (2, 1))
Dense Layers	3 layers (100, 80 units, 1 output layer)
Dropout Rate	0.5
Output Activation	Softmax
Loss Function	Sparse Categorical Crossentropy
Optimizer	Adam
Activation Function	ReLU

Table II shows the sizes of all the models developed in this work: trained, TFLite, and Quantized. The CNN model size is 1437.17 KB, with the TFLite model reduced to 462.02 KB and the Quantized TFLite model even further reduced to 123.80 KB. The TFLite model is 3.11 times smaller, and the TFLite Quantized model is 11.61 times smaller than the original model.

TABLE II: Model size for CNN architecture.

Type	CNN [KB]
Trained Model (Not converted)	1437.17
TFLite	462.02
TFLite Quantized	123.8

V. Performance Evaluation

Training. We utilize the *SingleDay* dataset provided by WiSig [10] to train the presented DL architecture. The dataset consists of 800 Wi-Fi signals generated by 28 transmitters collected over a one-day period and contains the IQ samples of the signals emitted by each device. For a detailed description of the full dataset and how the data is collected, please refer to [10]. The training was performed on a workstation with

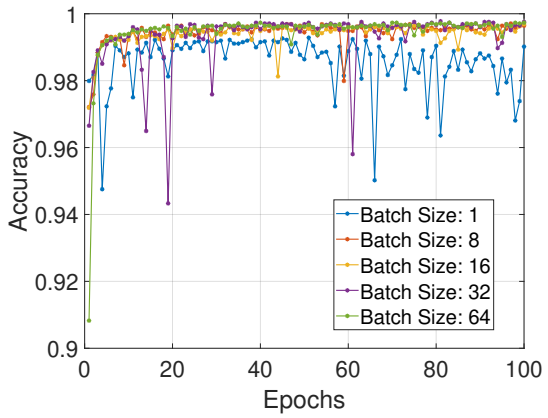


Fig. 3: Model training accuracy with different batch sizes, as a function of the number of training epochs.

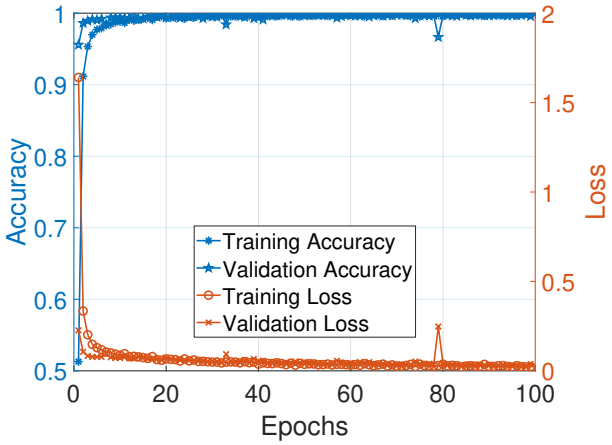


Fig. 4: Model training/validation accuracy and loss, as a function of the number of epochs.

the following configuration: 12th Gen Intel(R) Core(TM) i9-12900K with a clock speed of 3.20 GHz, memory size of 64.0 GB, and an NVIDIA RTX A4000 GPU with a memory size of 32 GB. The model training parameters are summarized in Table III. We considered the default learning rate, as it provides an acceptable performance and accuracy convergence for the DL model. The number of epochs is 100, providing a reasonable balance between underfitting (insufficient learning) and overfitting (model learning noise instead of patterns). The training data is shuffled before each training epoch, preventing the model from learning patterns based on the order of training samples.

TABLE III: Training parameters.

Parameter	Value
Learning Rate	0.001 (Default)
Epochs	100
Batch Size	32
Shuffle	Every Epoch
Validation Data	Random 20% of the data

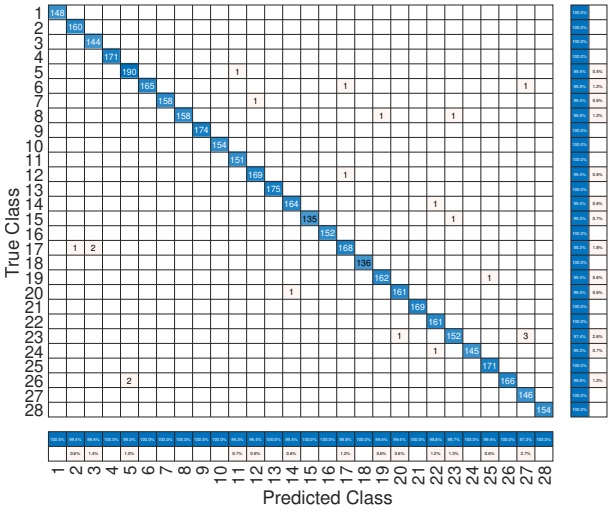


Fig. 5: Trained model confusion matrix.

Data Preprocessing: We preprocess the RF signals to ensure uniform scaling across all samples, thereby mitigating potential biases towards higher magnitude signals. This enhances the convergence and results in faster and stabler model learning. Furthermore, it prevents overflow and underflow issues arising from large variations in signal magnitudes.

Data Validation: The dataset is divided into training and validation sets with a ratio of 80/20. The chosen split ratio is common for the training and validation of DL models. The validation set is used to assess the model performance after each epoch during the tuning of the hyperparameters.

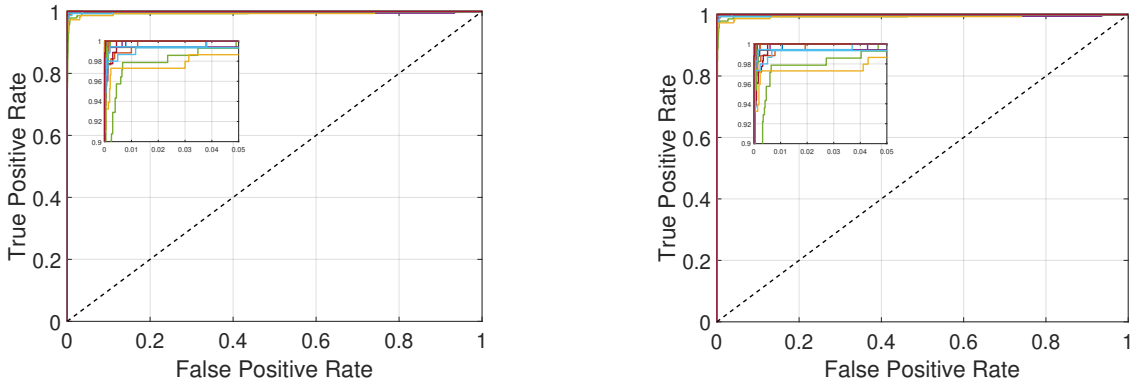
Batch Size: An important training hyperparameter is the batch size, that is, the number of samples utilized in training the network in every epoch. A batch size of 32 is used for training, providing a good trade-off between stability and training time. As illustrated in Fig. 3, we compare batch size efficiency (lower fluctuation in validation accuracy) and training time (faster than smaller batches but not too large to cause accuracy drops).

TABLE IV: Predication accuracy of the non-converted (original) CNN, converted TFLite, and Quantized TFLite models.

Model Type	Accuracy
Trained Model	0.99
TFLite	0.99
Quantized TFLite	0.99

In Fig. 4, the CNN model training and validation accuracy start to converge, with an accuracy > 0.90, from the third epoch on. In the last epoch, the training accuracy is 0.99, while the validation accuracy is approximately 0.99. This shows that the model is able to learn by capturing the unique features of each transmitter in just a few training epochs. Similarly, the training and validation loss of the model decreases with the increase in the number of epochs, illustrating that the model does not overfit.

Evaluation. We evaluated the trained and both converted



(a) (b)

Fig. 6: ROC-AUC curves for the CNN model (a) TFLite and (b) TFLite Quantized, highlighting the model performance.

TFLite and Quantized TFLite models using the raw IQ data. Table IV compares the accuracy for the trained and converted models. Fig. 5 illustrates the confusion matrix when performing the prediction for the trained CNN model. Furthermore, Fig. 6 shows the ROC-AUC for TFLite (Fig. 6a) and Quantized TFLite (Fig. 6b) models. For all 28 transmitters, both models retain very high AUC values (> 0.90), indicating that both models are able to effectively distinguish between true positives and false positives across all the classes.

Inference on the Edge. To evaluate the performance of the Keras converted and optimized models, we run 1000 inferences on a Raspberry Pi 4 and compute the average inference time, that is, the time to perform the prediction. The Raspberry Pi is equipped with 8GB of RAM. Table V depicts the average inference times of the Keras-converted CNN models, i.e., TFLite and Quantized TFLite. The Quantized model significantly reduces the inference time. This is due to quantization, which reduces the numerical precision of the model weights and activations from 32-bit floats to 8-bit integers. This reduces the model size, making it more memory-efficient, and speeds up processing, particularly on hardware optimized for integer arithmetic. Hence, Quantized TFLite models provide faster and more efficient inference without substantially compromising accuracy.

TABLE V: Mean inference time for the CNN model using TFLite and Quantized TFLite on Raspberry Pi 4.

Model Type	Time [ms]	95% CI [ms]
CNN – TFLite	10.9049	0.0072
CNN – Quantized TFLite	0.3983	0.0010

VI. Conclusion

In this paper, we presented the methodology for deploying RFF on edge devices to enhance the security of IoT wireless networks. By leveraging the unique device characteristics extracted from the raw IQ data, our lightweight CNN implementation, with TensorFlow Lite optimization applied, demonstrates robust device identification (authentication) capabilities within resource-constrained environments. Experimental evaluation reveals significant achievements in both performance and practicality. The implementation consistently achieves accuracy exceeding 0.95 and ROC-AUC scores greater than 0.90 in device identification tasks. Through strategic model quantization, we successfully address the computational constraints of edge deployment while maintaining authentication reliability.

Real-world validation on Raspberry Pi hardware confirms the framework viability for practical applications. The developed methodology proves particularly relevant for emerging, 5G and Beyond, applications, offering robust authentication solutions for Internet of Drones (IoD), Internet of Vehicles (IoV), and Internet of Medical Things (IoMT) deployments. Our implementation framework establishes a foundation for efficient edge-based authentication in resource-constrained environments. Future work will focus on exploring different DL architectures, expanding dataset diversity across IoT devices and environmental conditions, optimizing model architectures and hyperparameters, and implementing advanced efficiency enhancement techniques.

Acknowledgment

This work was supported in parts by the Swedish Research Council and the Knut and Alice Wallenberg Foundation.

References

[1] A. Ghasempour, "Internet of Things in Smart Grid: Architecture, Applications, Services, Key Technologies, and Challenges," *Inventions*, vol. 4, no. 1, p. 22, 2019.

[2] IBM, "What is the Internet of Things (IoT)?" 2023, accessed: 2025-04-10. [Online]. Available: https://www.ibm.com/topics/internet-of-things

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[4] R. Narayanan, A. Varshney, and P. Papadimitratos, "HarvestPrint: Securing Battery-Free Backscatter Tags through Fingerprinting," in *ACM Workshop on Hot Topics in Networks (ACM HotNets)*, ser. HotNets '21. New York, NY, USA: Association for Computing Machinery, November 2021, pp. 178 – 184. [Online]. Available: https://doi.org/10.1145/3484266.3487388

[5] N. Soltanieh, Y. Norouzi, Y. Yang, and N. C. Karmakar, "A review of radio frequency fingerprinting techniques," *IEEE Journal of Radio Frequency Identification*, vol. 4, no. 3, pp. 222–233, 2020.

[6] C. Zhu, K. Li, J. Hong, C. Hua, and F. Zou, "A Secure and Private Authentication Based on Radio Frequency Fingerprinting," in *ICC 2024-IEEE International Conference on Communications*. IEEE, 2024, pp. 2210–2215.

[7] S. Al-Hazbi, A. Hussain, S. Sciancalepore, G. Oligeri, and P. Papadimitratos, "Radio Frequency Fingerprinting via Deep Learning: Challenges and Opportunities," in *2024 International Wireless Communications and Mobile Computing (IWCMC)*, 2024, pp. 0824–0829.

[8] T. Jian, Y. Gong, Z. Zhan, R. Shi, N. Soltani, Z. Wang, J. Dy, K. Chowdhury, Y. Wang, and S. Ioannidis, "Radio Frequency Fingerprinting on the Edge," *IEEE Transactions on Mobile Computing*, vol. 21, no. 11, pp. 4078–4093, 2021.

[9] W. Wu, S. Hu, D. Lin, and T. Yang, "Radio-Frequency Fingerprinting for Distributed IoT Networks: Authentication and QoS Optimization," *IEEE Systems Journal*, vol. 17, no. 3, pp. 4440–4451, 2023.

[10] S. Hanna, S. Karunaratne, and D. Cabric, "WiSig: A Large-Scale WiFi Signal Dataset for Receiver and Channel Agnostic RF Fingerprinting," *IEEE Access*, vol. 10, p. 22808–22818, 2022.

[11] K. O'shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[12] R. Singh and S. S. Gill, "Edge AI: a survey," *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 71–92, 2023.

[13] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: A survey," *Neurocomputing*, vol. 461, pp. 370–403, 2021.

[14] A. Elmaghbub and B. Hamdaoui, "Distinguishable IQ Feature Representation for Domain-Adaptation Learning of WiFi Device Fingerprints," *IEEE Transactions on Machine Learning in Communications and Networking*, pp. 1404–1423, August 2024.

[15] ——, "A Needle in a Haystack: Distinguishable Deep Neural Network Features for Domain-Agnostic Device Fingerprinting," in *2023 IEEE Conference on Communications and Network Security (CNS)*, 2023, pp. 1–9.

[16] İ. Kök, F. Y. Okay, and S. Özdemir, "FogAI: An AI-supported fog controller for next generation IoT," *Internet of Things*, vol. 19, p. 100572, 2022.

[17] S. Zaman, K. Alhazmi, M. A. Aseeri, M. R. Ahmed, R. T. Khan, M. S. Kaiser, and M. Mahmud, "Security Threats and Artificial Intelligence Based Countermeasures for Internet of Things Networks: A Comprehensive Survey," *IEEE Access*, vol. 9, pp. 94 668–94 690, 2021.

[18] P. P. Ray, "A review on TinyML: State-of-the-art and prospects," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 4, pp. 1595–1623, 2022.

[19] C. Sun, X. Chen, W. Wang, and H. Yin, "A Transformer-Based Multi-Feature Extraction Neural Network for Bluetooth Devices Identification," in *2023 9th International Conference on Computer and Communications (ICCC)*. IEEE, 2023, pp. 1469–1473.

[20] J. Feng, X. Tang, B. Zhang, and Y. Ren, "Lightweight CNN-Based RF Fingerprint Recognition Method," in *2023 8th International Conference on Computer and Communication Systems (ICCCS)*. IEEE, 2023, pp. 1031–1035.

[21] F. Xie, H. Wen, Y. Li, S. Chen, L. Hu, Y. Chen, and H. Song, "Optimized coherent integration-based radio frequency fingerprinting in Internet of Things," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3967–3977, 2018.

[22] 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*. IEEE, 2021, pp. 309–313.

[23] G. Shen, J. Zhang, A. Marshall, and J. R. Cavallaro, "Towards scalable and channel-robust radio frequency fingerprint identification for LoRa," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 774–787, 2022.

[24] C. Geng, S.-J. Huang, and S. Chen, "Recent Advances in Open Set Recognition: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3614–3631, 2021.

[25] X. Wang, Q. Wang, L. Fang, M. Hua, Y. Jiang, and Y. Hu, "Radio frequency fingerprint authentication based on feature fusion and contrastive learning," *Expert Systems with Applications*, vol. 255, p. 124537, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417424014040

[26] A. Hussain, N. Abughanam, J. Qadir, and A. Mohamed, "Jamming Detection in IoT Wireless Networks: An Edge-AI Based Approach," in *Proceedings of the 12th International Conference on the Internet of Things*, ser. IoT '22. New York, NY, USA: Association for Computing Machinery, 2023, p. 57–64. [Online]. Available: https://doi.org/10.1145/3567445.3567456