Noise-Robust Radio Frequency Fingerprint Identification Using Denoise Diffusion Model

Guolin Yin*, Junqing Zhang*, Yuan Ding†, and Simon Cotton‡
* Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 3GJ, United Kingdom
† Institute of Sensors, Signals and Systems (ISSS), Heriot-Watt University, Edinburgh, EH14 4AS, United Kingdom
‡ Centre for Wireless Innovation (CWI), Queen's University Belfast, Belfast, BT3 9DT, United Kingdom

***Abstract*—Securing Internet of Things (IoT) devices presents increasing challenges due to their limited computational and energy resources. Radio Frequency Fingerprint Identification (RFFI) emerges as a promising authentication technique to identify wireless devices through hardware impairments. RFFI performance under low signal-to-noise ratio (SNR) scenarios is significantly degraded because the minute hardware features can be easily swamped in noise. In this paper, we leveraged the diffusion model to effectively restore the RFF under low SNR scenarios. Specifically, we trained a powerful noise predictor and tailored a noise removal algorithm to effectively reduce the noise level in the received signal and restore the device fingerprints. We used Wi-Fi as a case study and created a testbed involving 6 commercial off-the-shelf Wi-Fi dongles and a USRP N210 software-defined radio (SDR) platform. We conducted experimental evaluations on various SNR scenarios. The experimental results show that the proposed algorithm can improve the classification accuracy by up to 34.9%.**

***Index Terms*—Denoising, Diffusion Model, Radio Frequency Fingerprint Identification (RFFI), Transformer Wi-Fi**

I. Introduction

RFFI is an emerging authentication approach by using implicit hardware impairments in wireless transmitters, such as oscillators, mixers, and power amplifiers [1], [2]. These impairments come from the manufacturing process which will deviate the nominal values of the hardware components slightly from their specifications. The impairments are unique and can be extracted as device identifiers, in a similar manner to the biometric fingerprint authentication. As RFFI exploits the existing hardware impairments, this technique can be implemented solely at the receiver end but does not require any modification to the devices under test (DUTs). Therefore, it can be readily applied to any IoT networks.

Deep learning has been widely adopted for RFFI [3], [4] thanks to its superior feature extraction capability and classification capability. As indicated in [5]–[7], the convolutional neural network (CNN) can significantly enhance the fingerprinting performance, outperforming the traditional method which uses hand-crafted features. Various deep learning techniques have been proposed to address challenges in RFFI. In [8], the authors proposed a supervised contrastive learning-based method to address the channel variation. In [9], the authors proposed a Transformer-based structure for overcoming the variable input length problem. The adversarial training method was proposed in [10] to remove the impact of the receiver variations in RFFI.

SNR plays an essential role in RFFI. As the hardware fingerprints are subtle, they can be easily buried in noise. For example, the RFFI accuracy dropped from 97.1% to 23.12% in [8] when SNR drops from 40 dB to 5 dB. Also, as reported in [11], the RFFI performance dropped below 20% when the SNR was around 20 dB. The noise problem poses significant challenges for reliable RFFI. Data augmentation and collaborative identification [9] can be employed to improve the noise robustness of RFFI systems. Data augmentation improves the diversity of the training datasets and collaborative identification integrates the deep learning predictions from multiple packets [9] or receivers [10]. However, the RFFI accuracy under low SNR is still limited.

Diffusion models (DMs) [12], [13] have recently emerged as a powerful class of generative models, particularly effective for data generation tasks. They can generate new data by iteratively denoising Gaussian noise in a probabilistic manner. In addition, DMs can also be employed for denoising. By initializing the denoising process with a noisy input instead of pure Gaussian noise, the trained model can iteratively remove noise in a deterministic fashion. However, there is no research on using DMs for denoising in RFFI.

In this paper, we employed the DMs to train a powerful noise predictor for restoring the RFF. To adapt the pretrained noise predictor to the RFFI system, we proposed an SNR mapping algorithm that enables the noise predictor to accurately remove noise and restore RFF from noisy observations. We created a testbed consisting of commercial off-the-shelf Wi-Fi dongles as DUTs and a USRP N210 SDR platform as the receiver. Experimental evaluation was carried out. The technical contributions of this paper are summarized as follows:

- We adapted the DM for denoising in the RFFI system to enable effective noise removal and restore RFF degraded by noise. To the best of the authors' knowledge, this is the first work to apply DM for denoising within RFF.
- We proposed a Transformed-based RFFI approach by integrating the noise predictor. We designed an SNR mapping method, which allows the noise predictor to remove noise accurately and recover the underlying RFF in a computationally efficient manner.
- We experimentally evaluated the performance of the pro-

J. Zhang is the corresponding author (email: Junqing.Zhang@liverpool.ac.uk).

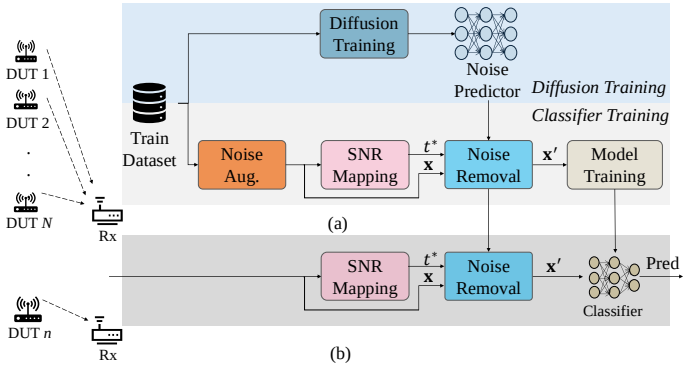


Fig. 1: System overview. (a) Model training stage. (b) Inference stage.

posed system across a range of SNR levels. The experimental results demonstrated that the proposed system significantly enhanced identification accuracy, particularly under low SNR conditions. Specifically, the classification performance improved by 34.9% at an SNR of 0 dB.

The rest of this paper is organised as follows. Section II presents the overview of the proposed RFFI system. Section III presents the technical details of the proposed diffusion model based denoising method. The details of the classifier used in this work are presented in Section IV. The experimental evaluation is presented in Section V. Finally, Section VI concludes this paper.

II. System Overview

Figure 1 depicts the system overview, which includes the model training and inference stages. During the training stage, each DUT sends packets that are captured by the receiver, forming a training dataset. This dataset is then utilized for DM training which yields a noise predictor model for noise removal. A classifier is also trained to identify the DUTs.

A. Model Training Stage

1) Diffusion Model Training: The objective is to develop a robust noise predictor capable of predicting noise contained in signal and then use it to recover the RFF features of transmitted signals distorted by noise. The DM training involves two steps, i.e., forward process and reverse process. The forward process involves adding noise to the training signals through a series of timesteps, whereas in the reverse process, a deep learning model is trained to learn how to remove the noise from the noisy observation and recover the original clean signals. The technical details will be presented in Section III.

2) Classifier Training: The objective is to develop a model capable of extracting unique RFF features from the denoised signals produced by the DM trained in the previous step. First, we incorporate noise augmentation to increase the diversity of the training dataset. Second, we propose SNR mapping to determine the optimal denoising step t^* based on the noise conditions. Then, in the noise removal step, the noise predictor receives the noisy signal $\mathbf{x}$ along with the optimal timestep t^*. It processes these inputs to produce a denoised signal, which is then used to train a classifier to identify the RFF. The details of the classifier training will be presented in Section IV.

B. Inference Stage

In the inference stage, the system processes the incoming signals to identify the DUT. The receiver will first capture the packet that is sent from the DUT. The signal will be extracted along with the SNR value of the current signal. The SNR mapping module will estimate the optimal step for the subsequent noise removal procedure. Upon the received signal and SNR value, the noise predictor will remove noises and restore the essential RFF. The classifier will make a prediction on the denoised signal to predict the identity of the DUT.

III. Restore RFF Feature with Diffusion Model

In this section, we first explain how to train the DM to obtain a noise predictor. Next, we detail the noise removal block, which uses the noise predictor to restore signals from noisy observations.

A. Diffusion Model Training

1) Forward Process: The forward process [12] of the DM training aims to emulate the representation of a signal $\mathbf{x}$ in different levels of SNR by progressively adding Gaussian noise to the original data over a series of discrete timesteps.

As shown in Fig. 2(a), the forward process will produce a series of noisy versions, $\{\mathbf{x}_t\}_{t=1}^{T}$, of the original input, $\mathbf{x_0}$. The noise power at each step is controlled by the variance schedule denoted as $\{\alpha_t \in (0,1)\}_{t=1}^{T}$ and $\alpha_t = 1-\beta_t$, then the forward process is defined as:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)I\right), \tag{1}$$

where $\mathcal{N}$ denotes a Gaussian distribution with mean $\sqrt{\alpha_t}\mathbf{x}_{t-1}$ and variance $1-\alpha_t$. Furthermore, the forward process can be expressed in a closed-form equation that directly computes the noisy signal $\mathbf{x}_t$ from the original signal $\mathbf{x}_0$, given as

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \tag{2}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ represents the cumulative product of noise scheduling parameters α_i. The SNR of the signal at the timestep t, γ_t, can be calculated as

$$\gamma_t = \frac{\bar{\alpha}_t\, P_{\mathrm{s},0}}{\bar{\alpha}_t\, P_{\mathrm{n},0} + (1-\bar{\alpha}_t)}, \tag{3}$$

where $P_{\mathrm{s},0}$ is the signal power of $\mathbf{x}_0$ and $P_{\mathrm{n},0}$ is the power of noise in the original signal $\mathbf{x}_0$.

This process is crucial for subsequent reverse process to train robust noise predictor. By understanding how noise progressively distorts the signal, the model learns to effectively reverse this process, removing noise while preserving the essential RF features necessary for accurate device identification.

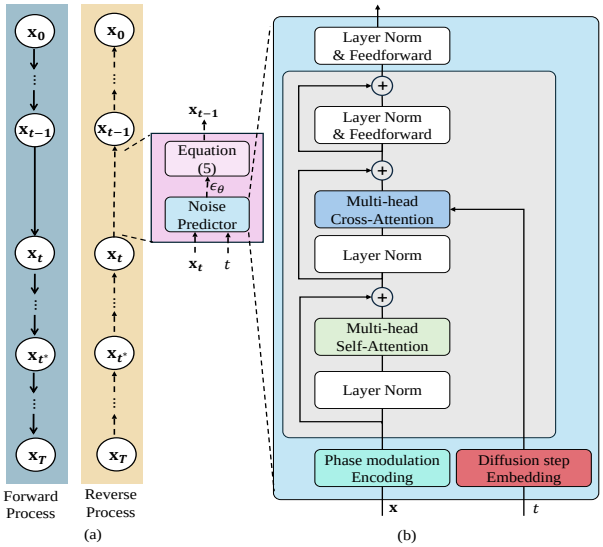


Fig. 2: (a) The forward and reverse process. (b) The structure of noise predictor.

2) Reverse Process: The objective of the reverse process is to systematically remove the noise added during the forward diffusion process, thereby reconstructing the original clean signal from its noisy counterparts. In the reverse process, a deep neural network is employed to predict the noise added in the previous timestep.

In this work, we modified the Hierarchical Diffusion Transformer (HDT) proposed in [14], originally tailored for RF signal generation, to serve as the backbone model for the noise predictor, f_θ, parameterized by θ. The model structure is shown in Fig. 2(a). Different from the original HDT, we removed class embedding as the class information during the denoising stage is not available in RFFI. The core block is a Transformer-based structure shown in the gray box in Fig. 2(a). This differs from the original Transformer encoder [15], as it introduced multi-head cross-attention that takes the embedded t as input and allows the model to estimate the current noise level. The multi-head self-attention (MHSA) captures autocorrelation features from the noisy input and extracts high-level representations implicit in the signal. The gray box in the diffusion step embedding encodes the current diffusion step t which will be used as input of MHSA. The Phase Modulation Encoding takes a complex-valued signal as input and encodes positional information in the complex domain.

As shown in the Fig 2(b). The reverse process involves learning a series of denoising steps that progressively refine the noisy data $\mathbf{x}_T$ to the original clean signal $\mathbf{x}_0$. Mathematically, the reverse process is modeled as a sequence of conditional probability distributions $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, t)$, where each distribution aims to predict the noise ϵ added from the previous noisy signal $\mathbf{x}_{t-1}$ during the forward process. The noise predicted by the noise predictor at step t is $\epsilon_\theta(\mathbf{x}_t, t)$, which is denoted as ϵ_θ too for simplicity. During the DM training, the parameter of noise predictor, i.e., θ, is updated by minimizing the mean squared error between the actual noise ϵ and the predicted noise ϵ_θ. The loss function is given as

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2\right]. \tag{4}$$

The previous signal after the denoising can be computed as

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\left(\frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon_\theta}}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\epsilon_\theta}. \tag{5}$$

B. Noise Removal

The noise predictor obtained during the DM training phase is used to remove noise from the received noisy signal. The generative task starts by removing the noise from pure Gaussian noise. Differently, in our task, we start from a noisy signal which is from intermediate, denoted as state "x_{t^*}", as shown in the Fig. 2(a). Denoted optimal intermediate step as t^*, which will be detailed in Section IV-B. In this phase, the received signal is refined iteratively over t' steps. In the DDPM framework, the denoising process, also referred to as the sampling process, is performed iteratively from step t^* to step t_0, encompassing the total steps t^*, i.e., $t' = t^*$. This method proves to be both time-consuming and computationally intensive, posing challenges for real-time processing required in RFFI systems.

To achieve efficient and precise noise removal, we adopted the step-skipping method, which is originally proposed in Denoising Diffusion Implicit Models (DDIM) [13]. In this approach, the number of timesteps is selected such that $t' < t^*$. The timestep t is determined as follows:

$$t_i = t^* - i \cdot \Delta t \tag{6}$$

where $\Delta t = \frac{t^*}{t'}$ and $i = 1, 2, ..., t'$.

DDPM [12] employs a probabilistic sampling method by adding noise at each denoising step. In contrast, DDIM sampling [13] used in our approach focuses on deterministically recovering the original signal without introducing additional noise. This distinction is crucial for signal recovery tasks, where the aim is to accurately reconstruct the original signal rather than to generate new variations. During the noise removal phase, the f_θ obtained in minimise (5) is used to predict the noise ϵ_θ based on the current noisy observation $\mathbf{x}_t$ and timestep t.

This deterministic denoising method ensures that noise is removed without additional randomness, which is important for refining the signal while preserving its RFF features necessary for accurate device identification. For simplicity, we denote the output of the noise removal as $\mathbf{x}'$.

IV. Classifier Design

A. Noise Augmentation

To enhance the robustness of the classifier against varying noise conditions, we employ the augmentation method proposed in [9]. This approach involves the addition of Gaussian noise to different training batches to increase the diversity of

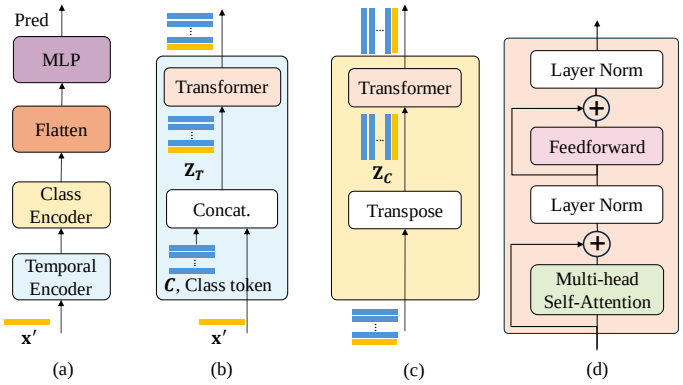


Fig. 3: (a) The proposed Transformer-based classifier structure. (b) The temporal encoder. (c) The class encoder. (d) Backbone transformer encoder.

the entire dataset. Denoting the initial SNR of a training batch as γ_{init}, the addition of Gaussian noise will adjust the batch SNR to a target value γ_{tgt}, where $\gamma_{tgt} < \gamma_{init}$. By increasing the diversity of the training dataset, the noise robustness can be improved.

B. SNR Mapping

The SNR mapping step aims to determine the optimal timestep t^* that corresponds to the current SNR of the received signal. This mapping ensures that the denoising operations are accurately aligned with the specific noise level present in the signal, allowing the model to apply the appropriate denoising step tailored to that noise intensity. Without SNR mapping, the noise predictor may insufficiently reduce the noise or overestimate the noise, obscuring the essential recovery of the RFF feature, which is necessary for accurate device identification.

To accurately determine the optimal t^*, we define the SNR map as $\gamma_{\text{map}} = \{\gamma_0, \gamma_1, ..., \gamma_T\}$ that can be computed using (3). We determine t^* by finding the timestep in which the SNR value of the predefined noise schedule matches the current SNR, given as

$$t^* = \arg\min_t |\gamma_{\text{map}}(t) - \gamma|, \tag{7}$$

where γ is the SNR of a signal input to the noise predictor.

C. Classifier Model Structure

We employed a Transformer-based classifier as shown in Fig. 3(a), the classifier consists of the temporal encoder and the class encoder. For both the temporal and class encoders, the Transformer is used as the backbone.

1) Temporal Encoder: The temporal encoder processes the signal throughout the time dimension, capturing the inherent temporal patterns and dynamics. Inspired by the Vision Transformer (ViT) [16], a learnable class token is added to the input sequence of image patches to provide a global representation, aggregating information through self-attention for classification decisions. Expanding on this, we introduce multiple learnable class tokens, one for each class, as shown in Fig. 3(b). Class tokens enhance a model's capacity to aggregate temporal characteristics, enabling concurrent recognition of patterns unique to specific classes as well as temporal linkages within the signal. The concatenation is given as

$$\mathbf{Z}_T = concat(\mathbf{C}, \mathbf{x}') \in \mathbb{R}^{(N+1)\times M}, \tag{8}$$

where N denotes the total number of classes, $\mathbf{C} \in \mathbb{R}^{N\times M}$ represents the class tokens, $\mathbf{x}' \in \mathbb{R}^{1\times M}$ corresponds to the input vector for the classifier, and M denotes the length of the signal.

2) Class Encoder: Following temporal feature extraction, the class encoder extracts relationships between different classes. The class encoder is designed to refine and aggregate the temporal features extracted by the temporal encoder to facilitate accurate classification. As shown in Fig. 3(c), the class encoder employs the Transformer block with the same structure as the temporal encoder to model the inter-class relationships and the extracted temporal features. Specifically, the class encoder takes the output of the temporal encoder as input and then performs a transpose on the input to obtain $\mathbf{Z}_C \in \mathbb{R}^{M\times(N+1)}$ to enable the Transformer to effectively capture class-wise dependencies.

The transpose operation will enable the Transformer to work over different class tokens to extract key features to distinguish different classes. This hierarchical processing, from temporal to class-specific encoding, enhances the model's ability to capture both temporal dependencies and inter-class relationships.

3) Classification: After the class encoder, the output feature map is passed through a flatten layer, which transforms the multidimensional tensor into a one-dimensional vector. This flattened vector serves as the input to the subsequent Multilayer Perceptron (MLP) responsible for classification. The classifier is trained using the cross-entropy loss.

(a) (b)

Fig. 4: The experiment devices. (a) USRP N210. (b) TP-Link USB dongle.

V. Experimental Evaluation

A. Data Collection and Experiments Setup

1) Experiment Devices: In this work, as shown in Fig. 4, we employed a USRP N210 SDR as the receiver and 6 Wi-Fi dongles as DUTs. During the data collection stage, the DUTs were plugged into a Linux laptop which was connected

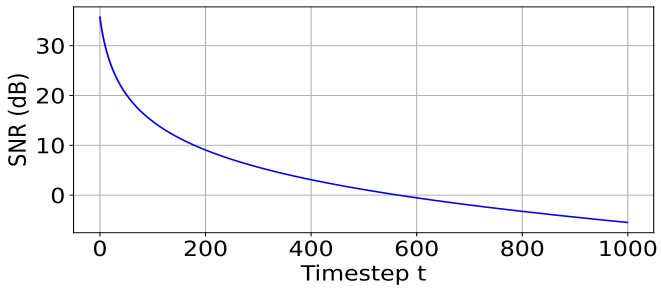


Fig. 5: The noise schedule during diffusion model training.

to a Wi-Fi router to create a packet stream. At the receiver end, the N210 was connected to a Linux laptop installed with Picosence [17] to obtain the baseband signal. The sampling rate was 20 Msamples/s.

2) Datasets: In this work, we use IEEE 802.11 as a case study, specifically leveraging its legacy preamble which consists of Short Training Sequence (STS) and Long Training Sequence (LTS) for RFF extraction, as these standardized training sequences provide consistent and comparable features across all compliant devices.

During the data collection, the USRP N210 and the WiFi dongles were placed about 1 meter apart with line-of-sight (LOS) present. Therefore, the SNR values of the collected packets all exceed 40 dB. The channel was kept stationary to maintain a fixed multipath channel. Under this configuration, we can focus exclusively on noise effects. The different noise levels are emulated by noise augmentation by adding artificial noise in a simulation manner.

We collected 30,000 WiFi packets at 2.4 GHz on channel 13 for each DUT as the training dataset. For testing, we obtained 2,000 packets for each DUT on the same channel.

3) Training Configuration: We conducted deep learning training on a PC equipped with an NVIDIA GeForce RTX 4090 GPU using PyTorch. The neural network parameters were optimized with the Adamax optimizer, starting with an initial learning rate of 0.0001 and a batch size of 32. We choose the linear spaced β_t range from 1×10^{-5} to 1.5×10^{-3}. During training, we halved the learning rate if the validation loss failed to improve for 20 consecutive epochs. Training was stopped when the validation loss showed no improvement for 30 epochs.

B. Denoise Performance

The SNR schedule across timesteps t can be computed using (3) and is illustrated in Fig. 5. This noise schedule ensures that the model is exposed to varying noise levels, allowing it to learn effective noise removal strategies across a wide range of signal quality conditions. The training process is designed to progressively denoise the received signals, preserving the essential features required for accurate RFF.

Fig. 6 exemplifies the waveforms of the original signal (40 dB), the noisy signal (5 dB), and the denoised signal from 5 dB. The noisy signal, shown in Fig. 6(b) shows significant fluctuations and show significant information loss compared to the original signal. The denoised signal, shown in Fig. 6(c) reveals a substantial improvement, with noise significantly removed and the signal shape more closely resembling that of the original signal. The visual comparison highlights the efficacy of the DM in reconstructing the signal, even under moderate noise conditions.

Fig. 7 shows the correlation between the noise signal and original clean signal, as well as the correlation between the denoised signal and the original signal, at different SNR levels (0 dB to 40 dB). When the SNR is below 20 dB, the DM effectively recovers the original signal, maintaining high correlation between the denoised and original signals even under challenging conditions (0 dB). As SNR rises above 20 dB, noisy signal correlation increases and can match or surpass denoised signal correlation. This is because when the SNR is high, the noise is minimal compared to the signal strength, which results in a marginally lower correlation for the denoised signal compared to the noisy one.

The analysis of both waveform comparisons and correlation results indicates that the DM effectively mitigates noise and restores signal quality, especially under low to moderate SNR conditions. As the SNR increases and the noise becomes less significant, the benefit of denoising diminishes slightly, as reflected in the correlation metrics. Nevertheless, the overall results demonstrate that the DM is highly capable of restoring RF signals for fingerprinting purposes, particularly in challenging noise environments.

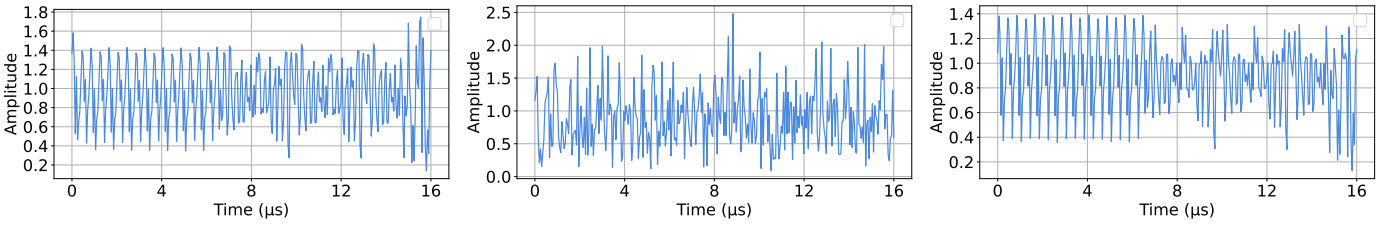
C. Device Identification

In this section, we present the device identification accuracy results of our proposed method. For comparison, we trained a baseline model that incorporates noise augmentation but excludes a noise removal step, meaning no noise removal is performed. The rest of the setup is the same as the proposed method in this paper.

The accuracy comparison is illustrated in Fig. 8. Our proposed approach consistently outperforms the baseline model, when the SNR is below 20 dB. In particular, at an SNR of 0 dB, the proposed model achieves a 34.9% improvement over the baseline. This indicates that the proposed method can significantly enhance the robustness to noise. As the SNR increases beyond 20 dB, the accuracy of both models converges, indicating that the benefit of denoising diminishes when the noise level is low. This convergence is expected, as the impact of noise becomes negligible at higher SNRs, leading to similar performance in both cases. Those observations align with the correlation relationship shown in Fig. 7.

VI. Conclusion

In this paper, we present a noise-robust RFFI system by employing a diffusion model to remove noise and restore RFF features. To adapt the diffusion model for noise mitigation within the RFFI system, we propose an SNR mapping method



(a) (b) (c)

Fig. 6: Exemplification of signal waveforms. (a) Original signal (40 dB). (b) Noised signal (5 dB). (c) Denoised signal from 5 dB.

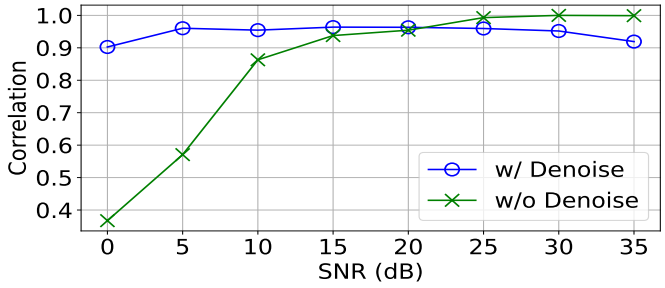


Fig. 7: The comparison of the correlation for "Noised - Original (40dB)" and "Denoised - Original (40dB)"

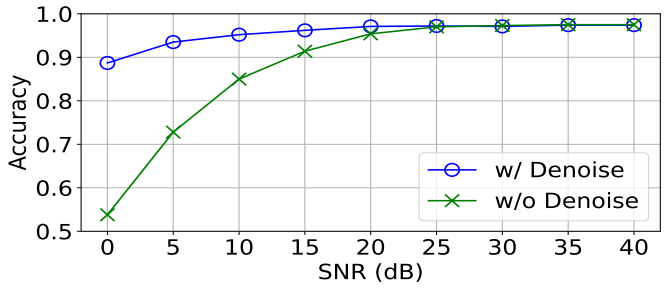


Fig. 8: Comparison of the classification performance with and without denoising.

that enables the pretrained noise predictor to estimate and eliminate noise from the signal. To validate the proposed system, we conducted real-world experiments using Wi-Fi as a case study. The experiments demonstrate that the proposed method effectively removes noise and restores RFF. Compared to the simple noise augmentation method, our approach improves the RFFI classification accuracy by up to 34.9%.

Acknowledgement

The work was supported in part by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant ID EP/Y037197/1 and in part by the UK Royal Society Research Grants RGS\R1\231435.

References

[1] J. Zhang, R. Woods, M. Sandell, M. Valkama, A. Marshall, and J. Cavallaro, "Radio frequency fingerprint identification for narrowband systems, modelling and classification," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3974–3987, 2021.

[2] W. Wang, Z. Sun, S. Piao, B. Zhu, and K. Ren, "Wireless physical-layer identification: Modeling and validation," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 9, pp. 2091–2106, 2016.

[3] G. Shen, J. Zhang, and A. Marshall, "Deep learning-powered radio frequency fingerprint identification: Methodology and case study," *IEEE Commun. Mag.*, vol. 61, no. 9, pp. 170–176, 2024.

[4] J. He, S. Huang, Z. Yang, K. Yu, H. Huan, and Z. Feng, "Channel-agnostic radio frequency fingerprint identification using spectral quotient constellation errors," *IEEE Trans. Wireless Commun.*, vol. 23, no. 1, pp. 158–170, 2023.

[5] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, "Deep learning for RF device fingerprinting in cognitive communication networks," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 160–167, 2018.

[6] M. Cekic, S. Gopalakrishnan, and U. Madhow, "Wireless fingerprinting via deep learning: The impact of confounding factors," in *Proc. Asilomar Conf. Signals, Sys., and Comput.*, 2021, pp. 677–684.

[7] P. Yin, L. Peng, G. Shen, J. Zhang, M. Liu, H. Fu, A. Hu, and X. Wang, "Multi-channel CNN-based open-set RF fingerprint identification for LTE devices," *IEEE Trans. on Cogn. Commun. Netw.*, 2024.

[8] R. Kong and H. Chen, "Towards channel-resilient CSI-based RF fingerprinting using deep learning," in *Proc. IEEE INFOCOM Workshops*, 2024, pp. 1–6.

[9] G. Shen, J. Zhang, A. Marshall, M. Valkama, and J. R. Cavallaro, "Toward length-versatile and noise-robust radio frequency fingerprint identification," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 2355–2367, 2023.

[10] G. Shen, J. Zhang, A. Marshall, R. Woods, J. Cavallaro, and L. Chen, "Towards receiver-agnostic and collaborative radio frequency fingerprint identification," *IEEE Trans. Mobile Comput.*, 2023.

[11] J. He, S. Huang, Y. Chen, S. Chang, Y. Zhang, and Z. Feng, "Radio frequency fingerprint identification for OFDM system considering unknown multipath fading channel," *IEEE Trans. on Cogn. Commun. Netw.*, 2024.

[12] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 6840–6851, 2020.

[13] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[14] G. Chi, Z. Yang, C. Wu, J. Xu, Y. Gao, Y. Liu, and T. X. Han, "RF-diffusion: Radio signal generation via time-frequency diffusion," in *Proc. Annu. Int. Conf. Mob. Comput. Netw. (MOBICOM)*, 2024, pp. 77–92.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[16] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[17] Z. Jiang, T. H. Luan, X. Ren, D. Lv, H. Hao, J. Wang, K. Zhao, W. Xi, Y. Xu, and R. Li, "Eliminating the barriers: Demystifying Wi-Fi baseband design and introducing the picoscenes Wi-Fi sensing platform," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4476–4496, 2021.