Embedding-Assisted Attentional Deep Learning for Real-World RF Fingerprinting of Bluetooth

Anu Jagannath, *Senior Member, IEEE,* Jithin Jagannath, *Senior Member, IEEE*

Abstract—A scalable and computationally efficient framework is designed to fingerprint real-world Bluetooth devices. We propose an embedding-assisted attentional framework (Mbed-ATN) suitable for fingerprinting actual Bluetooth devices. Its generalization capability is analyzed in different settings and the effect of sample length and anti-aliasing decimation is demonstrated. The embedding module serves as a dimensionality reduction unit that maps the high dimensional 3D input tensor to a 1D feature vector for further processing by the ATN module. Furthermore, unlike the prior research in this field, we closely evaluate the complexity of the model and test its fingerprinting capability with real-world Bluetooth dataset collected under a different time frame and experimental setting while being trained on another. Our study reveals a $9.17\times$ and $65.2\times$ lesser memory usage at a sample length of 100 kS when compared to the benchmark - GRU and Oracle models respectively. Further, the proposed Mbed-ATN showcases a $16.9\times$ fewer FLOPs and $7.5\times$ lesser trainable parameters when compared to Oracle. Finally, we show that when subject to anti-aliasing decimation and at greater input sample lengths of 1 MS, the proposed Mbed-ATN framework results in a $5.32\times$ higher TPR, 37.9% fewer false alarms, and $6.74\times$ higher accuracy under the challenging real-world setting.

Index Terms—RF fingerprinting, Bluetooth, Deep learning, Embedding module, Attention mechanism

I. INTRODUCTION

RADIO frequency (RF) fingerprint based on the hardware imperfections of the emitter circuit serves as an excellent tool or watermark to distinguish between devices manufactured by the same manufacturer even while transmitting the same message. In the present day and evolving Internet of Things (IoT) era where numerous wireless devices emerge everyday, the wireless security and the privacy of data shared across the spectrum accessed by these devices is a growing concern [1], [2].

RF fingerprinting pertaining to the extraction of physical layer-level (hardware circuitry) imperfections is emerging as a passive form of wireless emitter identification, i.e., a type of security scheme that can be implemented in a passive wireless receiver without any *apriori* knowledge of the emissions from the emitter. Wireless fingerprinting enables several security-related applications such as indoor positioning [3]–[5], emitter tracking and localization [6]–[8], device identification and authentication [9], [10], among others. These applications leverage the signals of opportunity (SoOP) [5], i.e., existing RF emissions that hold significant information about their source. For instance, WiFi, Bluetooth (BT), Frequency Modulated (FM) broadcast signals, LoRa, ZigBee, among others are SoOP

Anu Jagannath is with Northeastern University and Marconi-Rosenblatt AI/ML Innovation Laboratory, ANDRO Computational Solutions, LLC.

Jithin Jagannath is with University at Buffalo and Marconi-Rosenblatt AI/ML Innovation Laboratory, ANDRO Computational Solutions, LLC.

signals that are ubiquitously present in the spectrum. A passive listener (receiver radio) can therefore decipher and characterize the emitter without sharing any mutual handshake information. However, extracting the emitter features from the overheard signals hold a plethora of unique challenges.

RF fingerprinting falls under the broader signal intelligence application [11] such as modulation classification [12]–[17], signal/protocol classification [18], [19], etc. but is considered much more challenging due to the minute nature of the hardware-intrinsic features. A few of those challenges that severely affect RF fingerprinting involve the aging of hardware components, effect of wireless standard (complexity of the waveform) on the fingerprint features, effect of multipath propagation on the received waveform contributed by the obstacles, walls, environmental changes, location changes, noise, interference, etc. For instance, the frequency hopping nature of the BT waveform makes it challenging to capture the hardware-intrinsic features. Here, the performance depends significantly on the input preprocessing, input sample length, DL architecture, among others. This difficult nature was demonstrated in [20] where the single task model performance for the BT fingerprinting was poorer than with WiFi. Model generalization challenges were portrayed in [21] where even with WiFi channel equalization, their baseline model could achieve only 23.2% accuracy under the different *day* training and testing (Train One Day and Test on Another - TDTA). These factors consequently affect the performance of the trained and deployed deep learning (DL) models in the real-world operational environment. Hence, studies in this realm *must emphasize validating the generalization capability of the model* such that the effect of these confounding factors is accounted for in the evaluation. To this end, we collect emissions from actual IoT devices in a real-world operational environment to capture the differences in the *time frame, channel, location, and testbed setups* during training and deployment. Hence, the samples are collected under two *different testbed setups with different time frames and locations*.

II. RELATED WORKS

The various approaches towards RF fingerprinting to enhance the security of wireless devices that utilize wireless standards such as WiFi, BT, and LoRa are an actively researched topic [20], [22]–[24]. The earlier works focused on traditional approaches such as wavelet-based [25], I/Q imbalance [26], radio turn-on transient-based [27] among others [28]–[30]. After the recent resurgence of machine learning, DL has been leveraged to overcome some of the challenges [2]. However, the application of DL especially a lightweight deployable framework that improves generalization capability for fingerprinting real-world BT devices is lacking.

A. Real-World Emitters but Without Generalization Test

The authors of [24] study WiFi-based drone detection with actual drone emitters rather than synthetically generated emissions. In [31], a 1D AlexNet and ResNet architectures are adopted to fingerprint 7 DJI M100 drones. The generalization test performed here involves training and testing on different bursts of the emission collected during the same time frame. A ZigBee emitter fingerprinting with Differential Constellation Trace Figure (DCTF) using a LeNet-5 CNN model was proposed in [32]. The ZigBee emissions are collected, trained, and tested on the same time frame, location, and testbed setup. However, these works do not perform the generalization test where the classifiers are trained with data obtained from *a certain time frame, location, and testbed setup* and tested on another unseen time frame, location, and testbed setup, quoted in our article as TTD scenario. As discussed previously, this will be key to developing reliable real-world RF fingerprinting algorithms.

B. Studies Considering Generalization Test

The authors of [21] conduct CNN-based WiFi fingerprinting on a custom as well as large-scale DARPA dataset using a channel equalization approach. Here, the authors perform a generalization test where only the time frame of the testing dataset is different from the training, reported in their article as Train on One Test on Another (TDTA) scenario. The authors report an accuracy of only 23.2% with their Baseline CNN model even with WiFi channel equalization. A LoRa emitter fingerprinting using a spectrogram-based CNN is presented in [33]. The authors focus on the carrier frequency offset (CFO) of the LoRa emitters in their work. The evaluations to validate model generalization is carried out for different day training and test for a wired setup where the emitters are cabled over an attenuator to the receiving Universal Serial Radio Peripheral (USRP) radio. For the analysis conducted in the wireless setting, only same day training and testing setup is conducted. In [34], the authors employ a triplet loss based CNN model to fingerprint base stations transmitting either of 5G New Radio, LTE, or WiFi waveforms. However, these base stations are software-defined radio (leverages USRP B210) based rather than real-world base stations and emit synthetically generated waveforms with MATLAB's LTE, WLAN, and 5G toolboxes. In this work, the authors perform a generalization test by training and testing on different days. However, the multipath effect, fading, and orientation experienced by the emissions under our challenging TTD scenario is closer to the deployment setting faced in the real-world setting. Note that the proposed Mbed-ATN framework attains a 46.5% accuracy in fingerprinting the challenging frequency hopping BT emitters under the TTD setting.

C. Spatio-temporal model

In [35], the authors study the fingerprinting efficacy of recurrent neural networks (RNN) and a hybrid of RNN and CNN on Quadrature Phase Shift Keying (QPSK) waveforms from 8 emitters. However, the study utilized synthetic QPSK waveform generated in software and emitted using USRP radios. In contrast, in order to advance the practical application and demonstrate the usability of fingerprinting algorithm, we adopted 10 COTS IoT emitters that generate commercial standard Bluetooth 5.0 signals. We also demonstrate the implementation of a scalable architecture that benefits from the spatio-temporal feature extraction capability of the attentional module as well as conduct the evaluation in a generalized setting under different propagation conditions. Further, we critically dissect, evaluate, and infer the memory, computational, and fingerprinting performance of the proposed architecture alongside other benchmark models.

While the vast RF fingerprinting literature delves into waveforms such as ADS-B, WiFi, LoRa, and Zigbee [2], a robust DL based approach to fingerprint BT devices capable of handling unseen configuration is still lacking [20]. The core challenge stems from the rapid frequency hopping nature of the BT. In this work, for the first time, we introduce a unique embedding-assisted attentional framework (Mbed-ATN) for fingerprinting BT emitters and evaluate it in depth. We further demonstrate (with visualization) the challenging nature of the BT waveform in conjunction with realistic deployment conditions under different location, testbed setup, and time frame settings than those the model have been trained with.

D. Key Contributions

Unlike existing literature, we comprehensively evaluate the model's complexity, prediction capability, and generalization merit. We measure the generalization power of the proposed DL model by evaluating with unseen data obtained from a different time frame, location, and testbed setup compared to the training data. Our contributions are summarized below,

- **Architecture:** We propose for the first time, an embedding-assisted attentional framework for fingerprinting BT devices that provide on average 90.5% fingerprinting accuracy and demonstrate the lightweight and scalable nature of the proposed DL model to validate its practical deployment capability.
- **Dataset:** We collect real-world BT emissions from actual IoT devices under two indoor laboratory scenarios in rich multipath propagation, noise, and scattering (due to obstacles) settings. The datasets utilized in this article have also been published in IEEE dataport [36] to foster deployment-friendly research in the fingerprinting realm. Such a challenging dataset that facilitates model generalization validation for BT emitters has never been collected and made available to the public, hence it is considered extremely pivotal in RF fingerprinting research.
- **Experimental evaluation:** We present the evaluation results of the proposed DL model in contrast to the benchmark with RF data collected under a different time frame, location, and experimental setup than the training data in addition to analyzing the effects of input tensor length and anti-aliasing filtering.
- **Model Analysis:** We further shed light on how to critically evaluate neural network architecture on the basis of their memory consumption and other aspects of

model complexity that are essential to building models for practical deployment. The proposed model consumes $65.2\times$ lesser memory, $16.9\times$ fewer FLOPs, and $7.5\times$ fewer trainable parameters in contrast to the state-of-the-art architecture.

III. EMBEDDING-ASSISTED RF FINGERPRINT EXTRACTOR

In this section, we elaborate on the design of the proposed embedding-assisted RF fingerprint extractor (Mbed-ATN) enabling it to classify BT emitters in an unseen challenging environment. The proposed Mbed-ATN is a deep learning framework that adopts a convolutional neural network (CNN)-based embedding module (Mbed) that serves as the feature extractor and dimensionality reduction module. The Mbed module maps the high dimensional BT signal input tensor ($3 \times M, M =$ [10k, 100k, 1M]) to a one dimensional (1D) 1024×1 vector which feeds into a CNN and gated recurrent unit (GRU)-based attentional (ATN) classifier. The ATN module extracts the spatial and sequential patterns in the input vector allowing it to efficiently isolate the fingerprint from other confounding factors. This unique Mbed-ATN framework that combines the advantages of CNN and GRU in extracting the unique emitter characteristics is shown in Fig. 1. Emphasizing the significance of deployment of the Mbed-ATN in real-world operational scenarios, we enforce a lightweight and scalable architecture that can generalize well to the real-world environment.

Input Data Preprocessing: We denote the time domain BT signal of length N samples captured by the receiver as $y(t)|_{i=1}^N$. In our previous work [20] and as an ongoing study, we have empirically determined that BT emitter fingerprinting requires larger input sample lengths and additional features in the input tensor for acceptable classification accuracy. The capture length in this study is intentionally kept large enough ($N = 40$ MS) to experiment with data segmentation and other signal processing required to determine the input format that yields an acceptable fingerprinting accuracy. We subject the captured BT signal to the following operations to generate a $3 \times M$ tensor.

$$\mathbf{Y}^{3 \times M} = \mathfrak{F}\big[\hat{y}(t)_{t=1}^M\big] \tag{1}$$

$$= \begin{bmatrix} |\hat{y}(t)_{t=1}^M| \\ \angle \hat{y}(t)_{i=1}^M \\ PSD\big(\hat{y}(t)_{t=1}^M\big) \end{bmatrix} \tag{2}$$

where $\hat{y}(t)_{t=1}^M$ is the downsampled version of $y(t)_{t=1}^N$, the first two rows contain the magnitude and phase of the decimated signal $\hat{y}(t)$ and the third row is the power spectral density (PSD) of the decimated signal.

Embedding Module: We resort to the powerful feature extraction capability of CNNs to process the input tensor $\mathbf{Y}$. The Mbed module acts as a dimensionality reduction step in mapping the large 3D input tensor to a condensed 1D feature-embedded vector. It treats the 3D input tensor as a 3-channel input and adopts 1D convolutional kernels to encode the dependencies between the adjacent samples in each input channel. The architectural detail of the Mbed module is shown in Table I. We resort to using the parametric ReLU (PReLU)

activation function [37] in the convolutional layers as it has shown considerable improvement when the negative values are not zeroed out. The PReLU performs non-linear mapping of an input x as in equation 3.

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ ax, & \text{if } x \leq 0 \end{cases} \tag{3}$$

Here, a is the trainable parameter and hence, the name PReLU. The dense layer utilizes ReLU activation. Unlike PReLU, the ReLU maps all negative values to 0, or in other words, when the $a = 0$ in equation 3, the function is equivalent to ReLU.

TABLE I: Architectural detail of Mbed module.

Input $3 \times M$
Conv (100,1,10) - Stride (1,10) $\forall M < 1e6$ - Stride (1,20) $\forall M \geq 1e6$
Conv (50,1,6) - Stride (1,3) $\forall M < 1e6$ - Stride (1,6) $\forall M \geq 1e6$
Maxpool (1,8) Dropout 0.5
Conv (40,1,10) - Stride (1,10) $\forall M < 1e6$ - Stride (1,5) $\forall M \geq 1e6$
Maxpool (1,5) - active for $M \geq 1e6$ Dropout 0.5
Dense 1024
Activation: Conv Layers - PReLU, Dense Layer - ReLU

Attentional module: We resort to adopting an attentional mechanism to extract the inter-dependencies in the samples and *focus only on the relevant portions of the samples* with fewer layers. Our past experience and experiments show that the adoption of the attentional module can outperform deep network architectures and preempt the need for denser networks. The ATN module is a hybrid model that combines the benefits of CNN and GRU. While the CNN performs 1D convolutions on the embedding vector $\mathbf{f}$ to capture the timing relationship of the samples, the GRU extracts the before-after timing dependencies of the samples. We consider this as a pivotal step in characterizing and comprehensively extracting the fingerprint features, especially owing to the hopping nature of the BT waveform as it traverses the multipath propagation channel. GRU is an efficient form of long short term memory

TABLE II: Architectural detail of ATN module.

Input 1024×1
Branch-1: Conv (15,1,7) - Stride 1 - Padding 1 - PReLU - Dropout 0.1 Conv (32,1,7) - Stride 1 - PReLU - MaxPool (1,2) - Dropout 0.5
Branch-2: Conv (15,1,3) - Stride 1 - Padding 1 - PReLU - Dropout 0.1 Conv (32,1,3) - Stride 1 - PReLU - MaxPool (1,2) - Dropout 0.5
Branch-3: GRU hidden size = 80, #layers = 3, Dropout = 0.5
Dense 1024 - PReLU - Dropout 0.2
Dense 64 - PReLU - Dropout 0.2
Dense 10 - Softmax

(LSTM) since it uses only two gates - *Update* and *Reset* - instead of three gates as in LSTM. Further, GRU does not possess an internal memory or an output gate. Therefore, GRU uses fewer training parameters and memory and hence trains faster than LSTM. The update ($\mathbf{u}_t$) gate controls the amount of past information that needs to be carried over to the next state. The reset ($\mathbf{r}_t$) gate determines the amount of previous
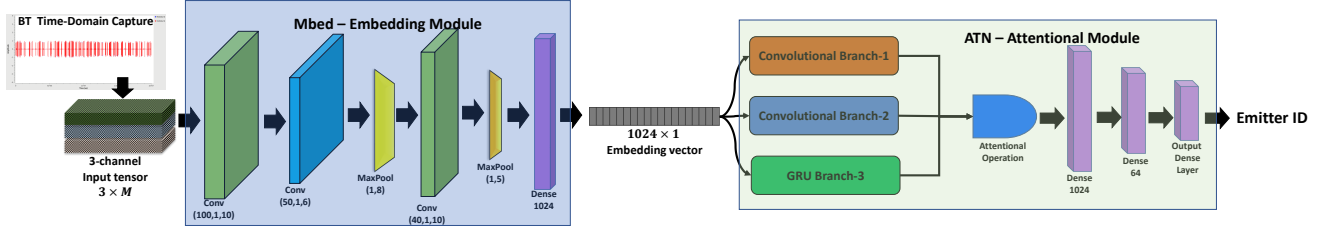
Fig. 1: Proposed Scalable Mbed-ATN framework

history that needs to be forgotten. The GRU units are defined by the following set of equations,

$$\mathbf{u}_t = \sigma\big(\mathbf{W}_u\mathbf{x}_t + \mathbf{R}_u\mathbf{h}_{t-1} + \mathbf{b}_u\big) \tag{4}$$

$$\mathbf{r}_t = \sigma\big(\mathbf{W}_r\mathbf{x}_t + \mathbf{R}_r\mathbf{h}_{t-1} + \mathbf{b}_r\big) \tag{5}$$

$$\hat{\mathbf{h}}_t = \tanh\big(\mathbf{W}_h\mathbf{x}_t + \mathbf{R}_h\big(\mathbf{r}_t \odot \mathbf{h}_{t-1}\big) + \mathbf{b}_h\big) \tag{6}$$

$$\mathbf{h}_t = (1 - \mathbf{u}_t) \odot \mathbf{h}_{t-1} + \mathbf{u}_t \odot \hat{\mathbf{h}}_t \tag{7}$$

where $\mathbf{x}_t$ is the input vector, $\mathbf{W}_i$ and $\mathbf{R}_i$ are the weight matrices, $\mathbf{b}_i$ the bias vector, $\mathbf{h}_t$ indicates candidate hidden state, $\tanh(\circ)$ is the hyperbolic tangential activation function, and $\sigma(\circ)$ is the sigmoid activation function.

As in the architectural diagram in Fig.1, the input ($\mathbf{f}$) to the ATN module feeds into two convolutional branches, and a GRU branch. The notations $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{G}$ denote the operations of the first convolutional branch, second convolutional branch, and the GRU branch respectively. The layer details of the branches are presented in Table II. The output from the convolutional branches is vectorized (flattened) form of their respective feature maps. The GRU branch is a many-to-1 type of GRU whose output is also a vector. The operations of the ATN module are governed by the following set of equations,

$$\mathbf{o}_1 = \mathcal{C}_1\big(\mathbf{f}\big) \qquad\qquad \mathbf{s} = \mathrm{SiLU}\big(\mathbf{o}_3\big) \tag{8}$$

$$\mathbf{o}_2 = \mathcal{C}_2\big(\mathbf{f}\big) \qquad\qquad \mathbf{a} = \big[\mathbf{o}_1; \mathbf{o}_2; \mathbf{s}\big] \tag{9}$$

$$\mathbf{o}_3 = \mathcal{G}\big(\mathbf{f}\big) \tag{10}$$

where $\mathbf{s}$ is the scoring vector function approximation obtained by applying Sigmoid Linear Unit (SiLU) activation to the output from the GRU branch. The SiLU activation [38] multiplies the input (x) by its sigmoid activation ($\sigma(x)$). The operator ; indicates vector concatenation. The final attentional vector $\mathbf{a}$ is generated by concatenating the outputs from the convolutional branches with the scoring vector $\mathbf{s}$ as in equation (9). This scoring vector is fed into the subsequent Dense layers for the final softmax emitter classification.

Training Mbed-ATN framework: We train the end-to-end Mbed-ATN framework as in Algorithm 1. The Mbed module is initially trained to classify the emitters by adding an output softmax Dense layer to the architecture in Table I. This layer is dropped after training and the 1024×1 feature vector $\mathbf{f}$ is fed to the ATN module. The ATN module is then trained independently while keeping the weights of the Mbed module unchanged. The modules are trained for maximum epochs of 2000 with Adam optimizer at a learning rate of 0.0001. The

network convergence is monitored during the training process and the parameters are frozen at the best point of convergence.

Algorithm 1 Backpropagation to train Mbed-ATN framework

Train Mbed module:
Initialize network weights Θ_{Mbed}.
for epoch = 1 to MAX_EPOCHS **do**
    for steps = 1 to STEPS **do**
        Input batch $\mathbf{x}$ and **Compute** loss
        $\ell_{Mbed}(\Theta_{Mbed})$ [standard forward pass]
        Compute gradients $\nabla\ell_{Mbed}(\Theta_{Mbed})$
        Update weights
        $\Theta^*_{Mbed} \longleftarrow \Theta_{Mbed}$ [standard backward pass]
    end for
    Stop training once model stops learning (starts to diverge)
end for
Freeze the Mbed module with learned weights Θ^*_{Mbed}
Eliminate the output softmax Dense layer of Mbed module and feed the 1024×1 feature vector $\mathbf{f}$ to ATN module.
Train Mbed-ATN module:
Initialize network weights $\Theta^*_{Mbed}, \Theta_{ATN}$.
for epoch = 1 to MAX_EPOCHS **do**
    for steps = 1 to STEPS **do**
        Input batch $\mathbf{x}$ and **Compute** loss
        $\ell_{Mbed-ATN}(\Theta^*_{Mbed}, \Theta_{ATN})$ [standard forward pass]
        Compute gradients $\nabla\ell_{Mbed-ATN}(\Theta^*_{Mbed}, \Theta_{ATN})$
        Update weights of ATN module
        $\Theta^*_{ATN} \longleftarrow \Theta_{ATN}$ [standard backward pass]
    end for
    Stop training once model stops learning (starts to diverge)
end for

IV. EXPERIMENTAL EVALUATION

Real-world IoT Datasets: We consider a testbed with real-world commercial IoT devices for the practical application and evaluation of the proposed RF fingerprinting framework. We collect the BT emissions from 10 IoT emitters in two challenging settings in an indoor multipath environment with other unavoidable interferences and obstacles rendering a rich multipath propagation scenario. A passive listener USRP X300 tuned into a 2 MHz bandwidth of a 2.414 GHz center frequency is streaming samples at the rate of 2 MS/s. The USRP X300 is outfitted with a UBX160 daughterboard and a VERT2450 antenna affixed to the RX2 antenna port. The respective emitters are positioned to transmit the BT bursts throughout the capture while the receiving radio records 40 MS in one capture. For the benefit of research in the domain, the datasets utilized in this article have also been published in IEEE dataport [36].

From the RF fingerprinting-specific feature extraction perspective, the BT waveform is challenging in itself owing to the frequency hopping nature which hops at the rate of 1600 hops/second over the 2.4 GHz ISM band. This implies the signal will be periodically visiting the tuned in BT channel making it a harder waveform to capture and fingerprint. The shorter input sample lengths therefore cannot comprehensively capture the emitter characteristics and will therefore need larger sample lengths [20].

Setup 1: Here the emitters and the receiving radio are positioned in line-of-sight (LoS) settings. The separation between the emitter and receiver is varied from 1.6 ft to 9.8 ft in steps of 0.8 ft.

Setup 2: This setup is considered a challenging setting given the rich multipath propagation settings between the emitters and the receiver. Here the emitter is placed at the four corners of the indoor laboratory while the receiver is placed at the center of the laboratory space. In this setup, the maximum separation between one of the corners and the receiver amounts to approximately 24.2 ft. Figure 2 shows the indoor laboratory setup with emitter locations and receiver placement. The multipath nature of the laboratory contributed by the walls, furniture, and lab equipment are clearly portrayed here (to keep the figure less crowded some obstacles are not shown).
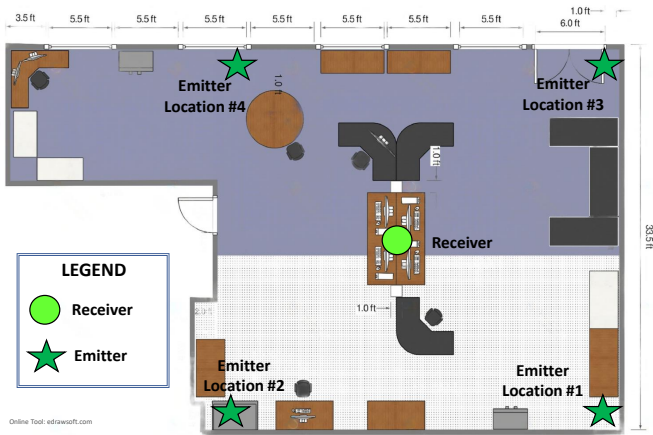

Fig. 2: Testbed layout for Setup 2 demonstrating the emitter in one of the corners and receiver at the center of the laboratory.

Generalization Evaluation: We consider two categories of experimental evaluations to quantify the generalization capability of the model to suit real-world deployment.

1) Train Test Same time frame, location, and testbed setup (TTS) - Here, the samples for the training and testing set are drawn from the data captured in the same time frame, location, and testbed setup. For the TTS scenario, we use the samples from the Setup 1 for training and testing.
2) Train Test Different time frame, location, and testbed setup (TTD) - In this scenario, the model is trained with samples collected from Setup 1 and later tested on captures from Setup 2.

We validate the argument that the training and testing data under the TTD have different distributions by utilizing the well-known t-Distributed Stochastic Neighbor Embedding (t-SNE) visualization tool. The training and testing data distribution under the TTS and TTD setting are shown in Fig.3a and Fig.3b respectively. Figure 3a indicates same distributions for TTS unlike TTD (Fig.3b) where the samples possess different features/distributions.
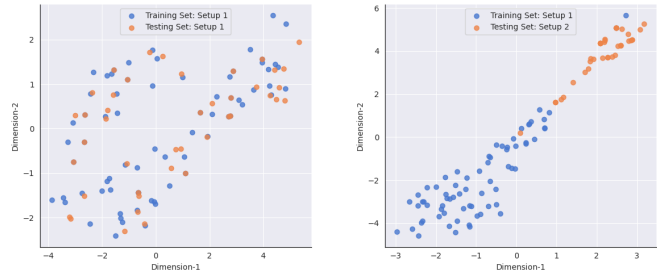
(a) TTS Distribution   (b) TTD Distribution

Fig. 3: t-SNE visualization of training and testing data distribution under the TTS and TTD evaluation settings.

A. Key Performance Indicators

In this section, we specify the performance metrics that are used to evaluate the models. The predictions made by any DL model or in other words, the confusion matrix can be categorized into true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

1) *True positive rate (TPR) or Recall*: quantifies the positive predictions made by the model with respect to total positive predictions. For a multi-class classification, it is $TPR = \frac{\sum_i TP_i}{\sum_i (TP_i + FN_i)}$, where i denotes the class i.
2) *False positive rate (FPR)*: measures the false predictions of the model in proportion to the total false predictions. Its computed as $FPR = \sum_{i=1}^{L} \frac{FP_i/(FP_i + TN_i)}{L}$.
3) T*op-1 accuracy* (or balanced accuracy): is the arithmetic mean of the recall for each class.
4) *FLOPs*: accounts for the total number of floating point operations in the model.
5) *Model parameters*: measures the total number of trainable parameters in the model.
6) *Supported sample lengths*: the maximum measured input tensor lengths supported by the model without causing any out-of-memory (OOM) GPU errors.

B. Complexity Analysis

Model complexity is an often overlooked factor by developers while designing and training deep learning (DL) models. According to a recent empirical study on 4960 failed DL jobs in Microsoft, 8.8% of the job failures were caused due to the depletion of GPU memory accounting for the largest category in all DL specific failures [39]. The frequency hopping nature of the BT waveform requires a scalable architecture that can process larger input sample lengths. This makes the model architecture challenging since it must be *large enough to process larger input samples but at the same time be lightweight for supporting commercial-off-the-shelf (COTS) deployment*

platforms. In this section, we perform a systematic review of the memory footprint of the proposed Mbed-ATN model and benchmark it against Oracle [22] and the GRU-based network proposed in [35].

Having a firm grasp of the memory usage of a model is imperative in designing efficient and lightweight DL models. In order to answer this critical practical usage question, we elucidate the maximum memory consumption of a model and demonstrate it on the proposed Mbed-ATN and benchmark models. The training of a DL model can be segmented into roughly five stages:

1) **Model Loading:** This stage involves moving the model parameters to the GPU memory. Here the current memory usage is the model memory.
2) **Forward pass:** Here the input batch is passed through the model and the intermediate activations are stored in memory for use by backpropagation. Here the current memory consumption is contributed by the model and the activations.
3) **Backward pass:** The gradients are computed from the end of the network to the beginning while discarding the saved activations during the traversal. The memory usage in this step is by the model and the gradients.
4) **Optimizer parameters:** The optimizer parameters are updated during the backpropagation. The parameters would vary depending on the type of learning algorithm such as Adam, RMSProp, etc. For example, Adam would estimate the first and second moments of the gradients. Here the memory is depleted by the model, gradients, and gradient moments.
5) **Training iterations:** Once the first iteration has passed and the optimizer has taken a step, the gradient and gradient moments are updated and stored in memory. So the maximum memory consumption in the subsequent training iterations will be in parts by the model, activations, gradients, and gradient moments.

We used the PyTorch framework and a Quadro RTX 6000 GPU in implementing and evaluating the models. Figure 4 demonstrates the GPU memory usage by Mbed-ATN model and benchmark models under the same evaluation settings on the challenging frequency hopping BT emissions. In Fig.4a, we analyze the GPU memory usage when the input tensor lengths are configured to $M = 10$ kS and $M = 100$ kS. It can be seen that the memory usage rapidly scales up to trigger OOM with the Oracle model while the proposed Mbed-ATN maintains manageable and very low memory usage. While the GRU network has a lesser memory usage than Oracle it is still higher and computationally much slower than the proposed Mbed-ATN rendering it infeasible for practical BT fingerprinting applications. To give the reader a quantifiable comparison, at the sample length of $M = 100$ kS, the GRU network has an inference time of ~ 6.7s. Such slow computation is inherent to RNN which processes samples within each example sequentially (output at each step depends on the previous) unlike CNN that can perform parallel computation. This corresponds to a $7.3\times$ and $65.2\times$ lesser memory usage with Mbed-ATN architecture in contrast to

Oracle at sample lengths of $M = 10$ kS and $M = 100$ kS respectively. Similarly, Mbed-ATN scores a $1.15\times$ and $9.17\times$ lesser memory usage at sample lengths of $M = 10$ kS and $M = 100$ kS respectively when compared to GRU network. This evaluation was carried out with a batch size of 2. A higher batch size of 70 and an input sample length $M = 1$ MS were not feasible with the benchmark models. However, to provide more insight to the readers, we characterize the proposed Mbed-ATN at different batch sizes and sample lengths in Fig.4b. These analyses demonstrate the GPU memory usage with the proposed Mbed-ATN well under the GPU memory capacity.
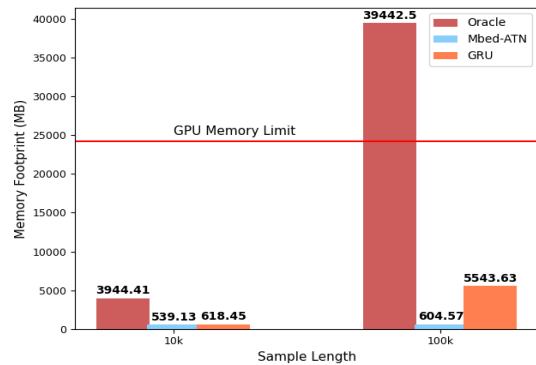
(a) GPU Memory Consumption of training Oracle and Proposed Mbed-ATN models with different sample lengths and Batch size=2.
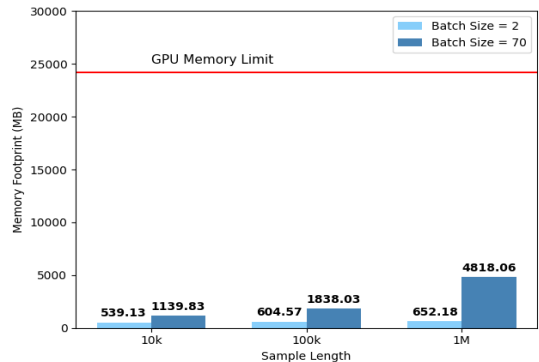
(b) GPU Memory Consumption of training the proposed Mbed-ATN and benchmark models with different sample lengths and batch sizes.

Fig. 4: GPU Memory Consumption of training the proposed Mbed-ATN and benchmark models. The red line indicates the memory capacity of the Quadro RTX 6000 GPU.

To shed more light on the model complexity from a deployment standpoint, we also evaluate the floating point operations (FLOPs), the number of trainable parameters, and inference time of the proposed Mbed-ATN and benchmark models for an input tensor length of $M = 10$ kS (Table III). The proposed Mbed-ATN showcases a $16.9\times$ fewer FLOPs and $7.5\times$ lesser trainable parameters when compared to Oracle. Although the GRU network possesses fewer trainable parameters and FLOPs in contrast to both Mbed-ATN and Oracle, the fully recurrent nature of the network renders it computationally much slower as showcased by an inference time which is $21.21\times$ and

TABLE III: FLOPs analysis with the benchmark.

Model	FLOPs	#Parameters	Supported Sample length	Inference Time
Mbed-ATN	2.181G	33.951M	1M	1.4ms
Oracle [22]	36.87G	256M	10k	0.95ms
GRU [35]	0.01G	3.627M	10k	29.7ms

$31.3\times$ that of the Mbed-ATN and Oracle models respectively. We clarify here that the slightly higher inference time of the proposed Mbed-ATN in contrast to Oracle despite the significantly lesser computational and memory requirement is attributed to the GRU unit employed in the ATN module of the architecture that processes samples present in each input sequentially. Recall here that the FLOPs and inference time are evaluated under the same settings for all models and at a sample length of 10 kS. These experiments demonstrate the superior lightweight nature of the proposed Mbed-ATN model in terms of memory footprint and computational requirements while being capable of supporting a larger input sample length of 1 MS. Further, it also showcases the scalability limitation of the benchmark models.

C. Effect of Sample length

In this section, we critically evaluate how the length of the input tensor affects the performance of the fingerprinting framework. As mentioned previously, the capture length is $N = 40$ MS which is subsequently decimated to different sample sizes (M) such as 10 kS and 1 MS. In these evaluations, we also characterize the Mbed module's fingerprinting performance separately to showcase the need for the ATN unit.

Since it is only feasible to support (for the given hardware) a sample length of $M = 10$ kS with the benchmark models, we also present its fingerprinting performance in Table IV. We measure the performance of the models when they are trained and validated with the dataset collected using *Setup 1* and tested with a portion of the test set obtained from unseen data collected from the same scenario, i.e., TTS. Under the TTS evaluation, the proposed Mbed model outperforms the Mbed-ATN as well as benchmark models in terms of the TPR, FPR, and Top-1 accuracy. The performance of Mbed and Mbed-ATN was measured for a sample length of 1 MS Table V. Comparing the KPIs of these evaluations under the TTS scenario, demonstrate an increase in TPR and Top-1 accuracy while lower FPR at higher sample lengths. These evaluations further portray that the significance of the ATN module comes into play at a larger sample length ($M = 1$ MS). This is intuitive as the GRU branch of the proposed Mbed-ATN can decipher the time series relation better with longer sequence lengths. Table V shows a 2.8% higher TPR, 23% fewer false alarms, and 0.5% greater top-1 accuracy with Mbed-ATN in contrast to the Mbed unit alone under the TTS condition.

D. Generalization on different location, testbed, and time frame setting.

An important aspect of training and deploying a DL model for real-world applications is its generalization capability. The fingerprinting literature has often resorted to evaluating this in

TABLE IV: Fingerprinting performance at $M = 10$ kS.

Scenario	Model	TPR	FPR	Top-1 Acc.
TTS	Mbed	**0.762**	**0.027**	**0.775**
	Mbed-ATN	0.738	0.029	0.742
	Oracle [22]	0.738	0.029	0.742
	GRU [35]	0.19	0.089	0.211
TTD	Mbed	**0.158**	**0.094**	**0.145**
	Mbed-ATN	0.079	0.103	0.075
	Oracle [22]	0.079	0.103	0.069
	GRU [35]	0.157	0.095	0.135

TABLE V: Fingerprinting performance at $M = 1$ MS.

Scenario	Model	TPR	FPR	Top-1 Acc.
TTS	Mbed	0.881	0.013	0.885
	Mbed-ATN	**0.905**	**0.01**	**0.91**
TTD	Mbed	0.105	0.095	0.175
	Mbed-ATN	**0.211**	**0.086**	**0.275**

terms of train one day and test another (TDTA) scenario where the DL model is trained and validated with one dataset while evaluating it with a test set collected on a different day (time frame) [21]. However, unlike the past works, we make it even more challenging by *testing on data collected from not just a different time frame but also under a different location and testbed setup*, i.e., TTD. Under the TTD setting, the models are trained and validated with data collected from *Setup 1* and tested on data captured from *Setup 2*.

The TTD evaluation in Table IV shows marginally higher performance of the proposed Mbed unit in contrast to the benchmark models. The significance of the ATN module is depicted in Table V where it achieves 2.6% higher top-1 accuracy in contrast to the Mbed unit.

E. Effect of applying anti-aliasing decimation

In this study, we evaluate the effect of anti-aliasing decimation as opposed to straightforward downsampling. For this, we decimate the 40 MS capture to 1 MS with an anti-aliasing order 8 Chebyshev type I filter. Here, the waveform is subject to anti-aliasing filtering prior to downsampling. The effects of anti-aliasing decimation is shown in Table VI. Here, we can evidently see the performance increase of the models with reference to the downsampling without anti-aliasing filtering in Table V. This implies that the plain decimation was causing aliasing thereby distorting the emitter signatures affecting their distinguishability. We show that Mbed-ATN achieves 23.1% higher TPR, 11.1% lesser false alarms, and 17.7% higher accuracy compared to the Mbed module under the TTD setting. To truly understand, the effect of anti-aliasing, we contrast the Mbed-ATN framework's performance under the TTD setting in Table V and Table VI. Note the 99.5%

TABLE VI: Fingerprinting performance with anti-aliasing decimation (sample length = 1M).

Scenario	Model	TPR	FPR	Top-1 Acc.
TTS	Mbed	0.905	0.011	0.905
	Mbed-ATN	0.905	0.011	0.905
TTD	Mbed	0.342	0.072	0.395
	Mbed-ATN	**0.421**	**0.064**	**0.465**

(a) GRU network [35] performance with the maximum supported sample length

(b) Oracle [22] performance with the maximum supported sample length

(c) Mbed-ATN architecture performance with maximum supported sample length
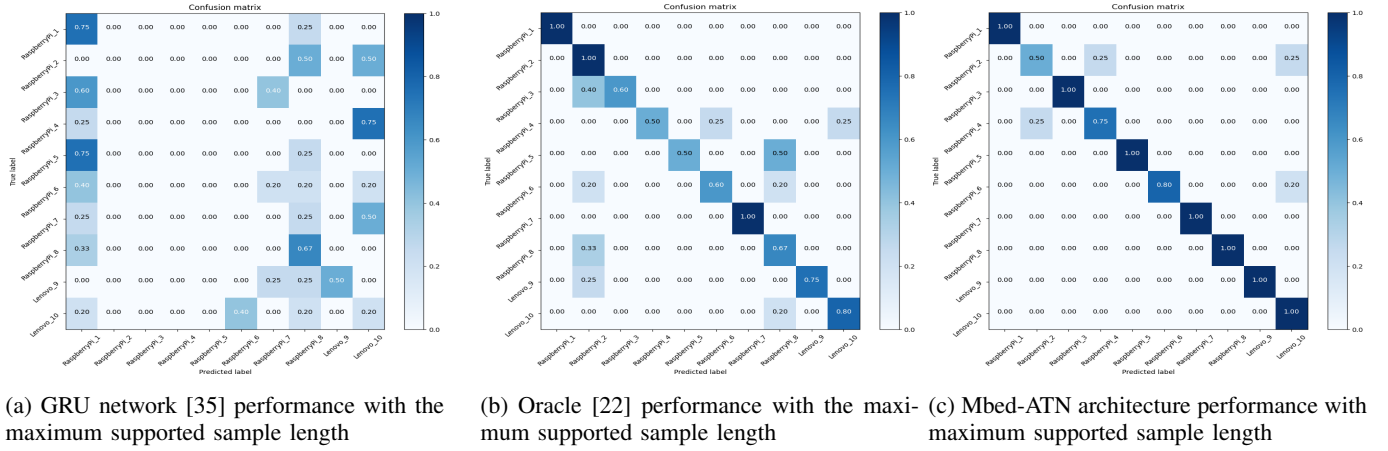
Fig. 5: Achievable BT fingerprinting performance on TTS (same day same location testbed setup)

(a) GRU network [35] performance with the maximum supported sample length

(b) Oracle [22] performance with the maximum supported sample length

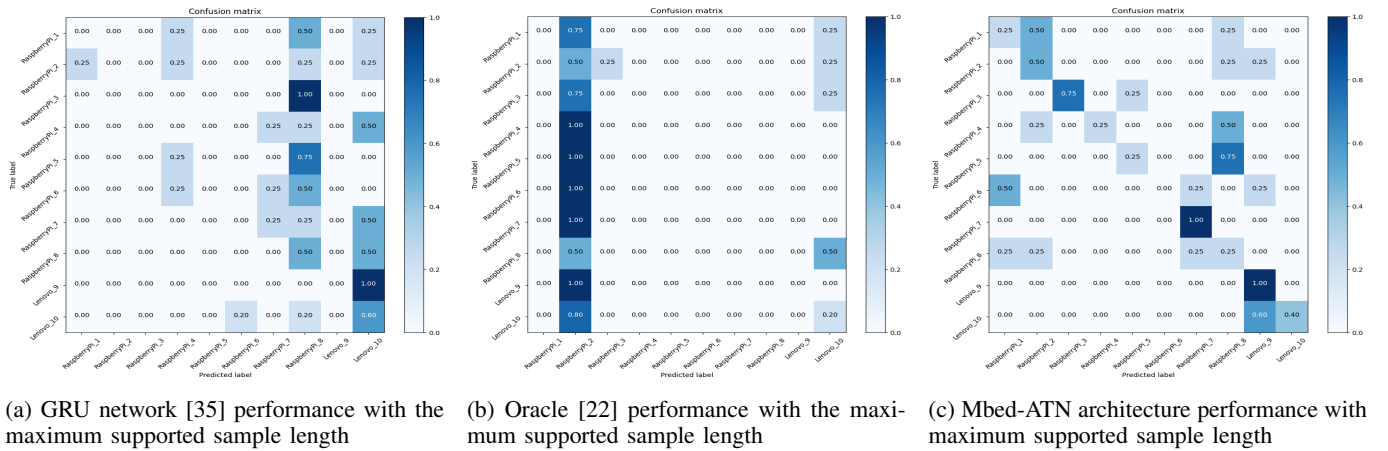(c) Mbed-ATN architecture performance with maximum supported sample length

Fig. 6: Achievable BT fingerprinting performance on TTD (different day different location testbed setup)

increase in TPR, 25.6% drop in false alarms, and 69.1% spike in the top-1 accuracy of the Mbed-ATN with the anti-aliasing decimated samples. With respect to the TTD case in Table IV, the Mbed-ATN demonstrates a $5.32\times$ higher TPR, 37.9% fewer false positives, and $6.74\times$ higher accuracy with the increased sample length subject to anti-aliasing filtering. To shed more light on this visually, we depict this increase in accuracy in Fig.7. Here, the label 1M_AA denotes a sample length of 1 MS with anti-aliasing decimation and we also show the accuracy with a sample length of 100 kS. The improved generalization capability with longer sample length and anti-aliasing decimation under the challenging TTD setting with the adoption of the ATN module can be clearly seen in Fig.7b. This study shows the combined effect of higher sample length and anti-aliasing on the performance of the proposed Mbed-ATN fingerprinting framework. Figures 5 and 6 show the confusion matrices of the benchmark models and proposed Mbed-ATN at their maximum supported input tensor lengths under the TTS and TTD experimental settings. The superiority of the proposed Mbed-ATN model in terms of the true positives, true negatives, false positives, and false negatives are evident in both the challenging TTD scenario and the TTS setup. These evaluations validate the improved generalization

and lightweight nature of the proposed Mbed-ATN framework taking it one step closer to practical deployment. While this is a significant improvement in the TTD setting for BT compared to existing literature, we are currently working on other refinements to make the TTD metrics close to the TTS. Furthermore, we also share the dataset so that the larger research community can continue to work on improving the generalization capability of RF fingerprint approaches using real-world data [36].

F. Why not raw IQ samples ?

We also validate the necessity to use the extracted features as in the expression 2 in section III as input rather than raw IQ samples. To achieve this, we used raw IQ samples arranged as a tensor of length 1 MS with two rows containing I and Q samples separately in each row as input to the Mbed-ATN architecture under the TTS setup. The confusion matrix of the architecture with raw IQ samples is shown in Fig.8. The false predictions and the inability of the model to distinguish between the emitters can be clearly seen from the confusion matrix. The need for the appropriate input tensor is obvious in comparison to the confusion matrix in Fig.5c which has only negligible false predictions. We also tabulate the performance
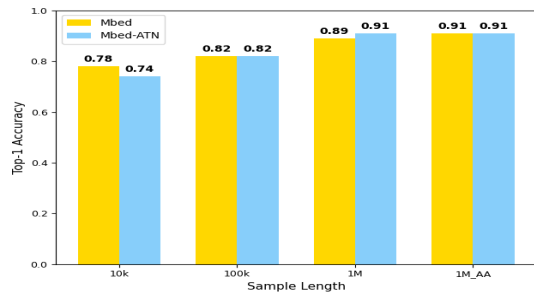
(a) TTS Scenario
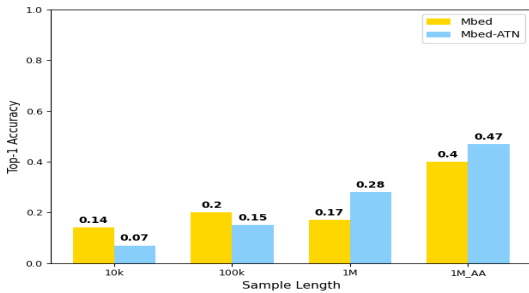
(b) TTD Scenario

Fig. 7: Demonstrating effect of sample length and anti-aliasing decimation on Mbed-ATN fingerprinting performance.

TABLE VII: Fingerprinting performance under various input formats (sample length = 1M).

Model	TPR	FPR	Top-1 Acc.
Mbed-ATN (IQ)	0.166	0.089	0.195
Mbed-ATN (Tensor 2)	**0.905**	**0.011**	**0.905**

metrics and provide a direct comparison with the Mbed-ATN using the appropriate input tensor as in expression 2 in Table VII. Adopting the appropriate input format resulted in a top-1 accuracy of 91% and FPR of 1.1%.
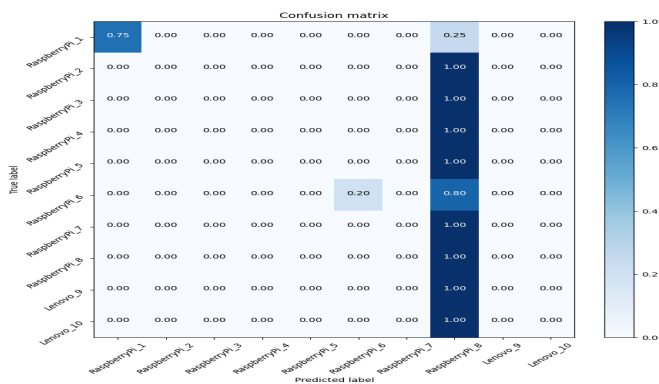

Fig. 8: Performance using raw IQ samples under a TTS setup.

V. CONCLUSION

We proposed and presented a detailed analysis of Mbed-ATN, an embedding-assisted attentional framework for enhancing the generalization capability of the fingerprinting architecture. The proposed model is scalable for supporting large

input tensor lengths of 1 MS while using significantly less GPU memory. The proposed Mbed-ATN utilizes $65.2\times$ lesser memory in contrast to the state-of-the-art Oracle architecture for an input length of $M = 100$ kS. Further, for a 10 kS sample length, the Mbed-ATN utilizes $16.9\times$ fewer FLOPs and $7.5\times$ fewer trainable parameters with respect to Oracle. We showed that the inference time of the proposed Mbed-ATN is $21.21\times$ lesser than that of the benchmark GRU model and attains a $1.15\times$ and $9.17\times$ lesser memory usage at sample lengths of $M = 10$ kS and $M = 100$ kS respectively when compared to GRU network. A detailed empirical study on the effect of higher sample length and anti-aliasing decimation was demonstrated for the proposed Mbed-ATN framework in showcasing the improved generalization capability of the model with the introduction of attentional learning. Unlike the existing literature, we resorted to the challenging different time frame, location, and experimental setup (TTD) scenarios along with demonstrating the GPU efficiency of the model in validating the real-world deployment merit of the proposed framework.

REFERENCES

[1] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device fingerprinting in wireless networks: Challenges and opportunities," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 94–104, 2016.

[2] A. Jagannath, J. Jagannath, and P. S. P. V. Kumar, "A Comprehensive Survey on Radio Frequency (RF) Fingerprinting: Traditional Approaches, Deep Learning, and Open Challenges," *Computer Networks*, vol. 219, p. 109455, 2022.

[3] N. Alikhani, S. Amirinanloo, V. Moghtadaiee, and S. A. Ghorashi, "Fast fingerprinting based indoor localization by wi-fi signals," in *Proc. of 7th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2017.

[4] S. He and S.-H. G. Chan, "Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 466–490, 2016.

[5] V. Moghtadaiee and A. G. Dempster, "Indoor location fingerprinting using fm radio signals," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 336–346, 2014.

[6] H. Givehchian, N. Bhaskar, E. R. Herrera, H. R. L. Soto, C. Dameff, D. Bharadia, and A. Schulman, "Evaluating physical-layer ble location tracking attacks on mobile devices," in *Proc. of IEEE Symposium on Security and Privacy (SP)*, 2022.

[7] T. Issoufaly and P. U. Tournoux, "Bleb: Bluetooth low energy botnet for large scale individual tracking," in *Proc. of 1st International Conference on Next Generation Computing Applications (NextComp)*, 2017.

[8] J. K. Becker, D. Li, and D. Starobinski, "Tracking anonymized bluetooth devices," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, pp. 50 – 65, 2019.

[9] K. Bonne Rasmussen and S. Capkun, "Implications of radio fingerprinting on the security of sensor networks," in *Proc. of Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007*, 2007.

[10] Y. Lin, J. Jia, S. Wang, B. Ge, and S. Mao, "Wireless device identification based on radio frequency fingerprint features," in *Proc. of IEEE International Conference on Communications (ICC)*, 2020.

[11] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia, "Neural Networks for Signal Intelligence: Theory and Practice,," in *Machine Learning for Future Wireless Communications*, ser. Wiley - IEEE Series, F. Luo, Ed.　John Wiley & Sons, Limited, 2020.

[12] T. Huynh-The, Q.-V. Pham, T.-V. Nguyen, T. T. Nguyen, R. Ruby, M. Zeng, and D.-S. Kim, "Automatic modulation classification: A deep architecture survey," *IEEE Access*, vol. 9, pp. 142 950–142 971, 2021.

[13] J. Jagannath, N. Polosky, D. O. Connor, L. Theagarajan, B. Sheaffer, S. Foulke, and P. Varshney, "Artificial Neural Network based Automatic Modulation Classifier for Software Defined Radios," in *Proc. of IEEE Intl. Conf. on Communications (ICC)*, Kansas City, USA, May 2018.

[14] J. C. Clement, N. Indira, P. Vijayakumar, and R. Nandakumar, "Deep learning based modulation classification for 5g and beyond wireless systems," *Peer-to-Peer Networking and Applications*, vol. 14, 2021.

[15] X. Fu, G. Gui, Y. Wang, T. Ohtsuki, B. Adebisi, H. Gacanin, and F. Adachi, "Lightweight automatic modulation classification based on decentralized learning," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 1, pp. 57–70, 2022.

[16] A. Jagannath and J. Jagannath, "Multi-task Learning Approach for Modulation and Wireless Signal Classification for 5G and Beyond: Edge Deployment via Model Compression," *Physical Communications (Elsevier)*, vol. 54, p. 101793, 2022.

[17] P. Qi, X. Zhou, S. Zheng, and Z. Li, "Automatic modulation classification based on deep residual networks with multimodal information," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 21–33, 2021.

[18] T. Huynh-The, C.-H. Hua, V.-S. Doan, Q.-V. Pham, and D.-S. Kim, "Accurate deep cnn-based waveform recognition for intelligent radar systems," *IEEE Communications Letters*, vol. 25, pp. 2938–2942, 2021.

[19] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia, "Machine learning for wireless communicationsin the internet of things: A comprehensive survey," *Ad Hoc Networks (Elseier)*, vol. 93, p. 101913, 2019.

[20] A. Jagannath, Z. Kane, and J. Jagannath, "RF Fingerprinting Needs Attention: Multi-task Approach for Real-World WiFi and Bluetooth," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, Rio de Janeiro, Brazil, December 2022.

[21] A. Al-Shawabka, F. Restuccia, S. D'Oro, T. Jian, B. C. Rendon, N. Soltani, J. Dy, K. Chowdhury, S. Ioannidis, and T. Melodia, "Exposing the Fingerprint: Dissecting the Impact of the Wireless Channel on Radio Fingerprinting," *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, 2020.

[22] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "Oracle: Optimized radio classification through convolutional neural networks," in *Proc. of IEEE INFOCOM - IEEE Conference on Computer Communications*, 2019.

[23] N. Chen, A. Hu, and H. Fu, "Lora radio frequency fingerprint identification based on frequency offset characteristics and optimized lorawan access technology," in *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5, 2021.

[24] I. Bisio, C. Garibotto, F. Lavagetto, A. Sciarrone, and S. Zappatore, "Unauthorized amateur uav detection based on wifi statistical fingerprint analysis," *IEEE Comm. Magazine*, vol. 56, no. 4, pp. 106–111, 2018.

[25] R. W. Klein, M. A. Temple, and M. J. Mendenhall, "Application of wavelet-based rf fingerprinting to enhance wireless network security," *Journal of Communications and Networks*, vol. 11, no. 6, pp. 544–555, 2009.

[26] F. Zhuo, Y. Huang, and J. chen, "Radio frequency fingerprint extraction of radio emitter based on i/q imbalance," *Advances in Information and Communication Technology Proceedings of 7th International Congress of Information and Communication Technology (ICICT)*, vol. 107, 2017.

[27] Y. Yuan, Z. Huang, H. Wu, and X. Wang, "Specific emitter identification based on hilbert-huang transform-based time-frequency-energy distribution features," *IET Communications*, vol. 8, no. 13, pp. 2404–2412, 2014.

[28] Y. Yuan, X. Liu, Z. Liu, and Z. Xu, "Mfmcf: A novel indoor location method combining multiple fingerprints and multiple classifiers," in *2019 3rd International Symposium on Autonomous Systems*, 2019.

[29] S. Deng, Z. Huang, X. Wang, and G. Huang, "Radio frequency fingerprint extraction based on multidimension permutation entropy," *International Journal of Antennas and Propagation*, vol. 2017, 2017.

[30] G. Baldini, R. Giuliani, G. Steri, and R. Neisse, "Physical layer authentication of internet of things wireless devices through permutation and dispersion entropy," in *Global Internet of Things Summit*, 2017.

[31] N. Soltani, G. Reus-Muns, B. Salehi, J. Dy, S. Ioannidis, and K. Chowdhury, "Rf fingerprinting unmanned aerial vehicles with non-standard transmitter waveforms," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15 518–15 531, 2020.

[32] L. Peng, J. Zhang, M. Liu, and A. Hu, "Deep learning based rf fingerprint identification using differential constellation trace figure," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 1091–1095, 2020.

[33] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, "Radio frequency fingerprint identification for lora using deep learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2604–2616, 2021.

[34] G. Reus-Muns, D. Jaisinghani, K. Sankhe, and K. Chowdhury, "Trust in 5g open rans through machine learning: Rf fingerprinting on the powder pawr platform," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, 2020.

[35] D. Roy, T. Mukherjee, M. Chatterjee, and E. Pasiliao, "Rf transmitter fingerprinting exploiting spatio-temporal properties in raw signal data," in *Proc. of 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019.

[36] A. Jagannath and J. Jagannath, "Rf-fingerprint-bt-iot: Real-world frequency hopping bluetooth dataset from iot devices for rf fingerprinting," 2022. [Online]. Available: https://dx.doi.org/10.21227/364j-6j73

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. of IEEE International Conference on Computer Vision*, 2015.

[38] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks- Special issue on deep reinforcement learning*, vol. 107, pp. 3–11, 2018.

[39] Y. Gao, Y. Liu, H. Zhang, Z. Li, Y. Zhu, H. Lin, and M. Yang, "Estimating gpu memory consumption of deep learning models," in *Proc. of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2020.

Anu Jagannath is the Founding Associate Director of Marconi-Rosenblatt AI/ML Innovation Lab at ANDRO Computational Solutions, LLC. She received her MS degree from State University of New York at Buffalo in Electrical Engineering. She is also a part-time Ph.D. candidate at Northeastern University, USA. Her research focuses on deep earning, reinforcement learning, adaptive signal processing, software defined radios, and signal intelligence. She has rendered her reviewing service for several leading IEEE conferences and Journals. She is an IEEE Senior Member. She is the co-Principal Investigator (co-PI) and Technical Lead in multiple Rapid Innovation Fund (RIF) and SBIR/STTR efforts involving applied AI/ML for wireless communications. She is also the co-inventor of 11 US Patents (granted and pending).

Jithin Jagannath is the Chief Scientist of Technology and Founding Director of the Marconi-Rosenblatt AI/ML Innovation Lab at ANDRO Computational Solutions. He is also the Adjunct Assistant Professor in the Department of Electrical Engineering at the University at Buffalo. He received his Ph.D. degree in Electrical Engineering from Northeastern University. He has been the Principal Investigator in research efforts for several customers including the U.S. Army, U.S. Navy, U.S. SOCOM, DHS, and AFSOR in the field of Beyond 5G, signal processing, RF signal intelligence, cognitive radio, cross-layer ad-hoc networks, Internet-of-Things, AI-enabled wireless, and machine learning.

Dr. Jagannath is an IEEE Senior member and has been invited as Keynote Speaker and Panelist discussing his vision at leading IEEE Conferences. He serves as a full member of the IEEE SPS Applied Signal Processing Systems Technical Committee and several other Technical Program Committees (TPCs). Additionally, he serves on the editorial board of Computer Networks (Elsevier). Dr. Jagannath's recent research has led to over 40 peer-reviewed journal and conference publications. He is the co-inventor of 16 U.S. Patents (granted and pending). Dr. Jagannath was the recipient of 2021 IEEE Region 1 Technological Innovation Award with the citation. He is also the recipient of AFCEA International Meritorious Rising Star Award for achievement in engineering and AFCEA 40 Under 40 award.