PLAIDS: Physical Layer–Assisted Intrusion Detection System in 5G-IoT Networks

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

28-01-2022 / 01-02-2022

CITATION

Sood, Keshav (2022): PLAIDS: Physical Layer–Assisted Intrusion Detection System in 5G-IoT Networks. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.19083404.v1>

DOI

[10.36227/techrxiv.19083404.v1](https://doi.org/10.36227/techrxiv.19083404.v1)

PLAIDS: Physical Layer–Assisted Intrusion Detection System in 5G-IoT Networks

Keshav Sood, Mohammad Reza Nosouhi, *Member, IEEE*, Frank Jiang, *Senior Member, IEEE*, Morshed Chowdhury, and Robin Doss, *Senior Member, IEEE*

Abstract—Network security is a critical issue since smart Internet of Things (IoT) applications are vulnerable to cyber threats. The introduction of 5G has further expanded the surface for attackers. Although Intrusion Detection Systems (IDS) are an effective measure to distinguish the legitimate and non-legitimate IoT nodes. However, in large scale 5G-enabled heterogeneous and resource-constrained IoT networks the conventional solutions are not fully efficient. We present Physical Layer–Assisted IDS (PLAIDS), a learning–based intrusion detection system for 5G-IoT heterogeneous networks. PLAIDS learns the unique physical layer features (inherent radio frequency properties of transmitter modules) of IoT nodes. Further, we have developed a learning-based anomaly detection module using Deep Autoencoders (DAE). Our proposal is 99.5% accurate and effective to detect the legitimate nodes among the non-legitimate ones, is effective in different channel conditions such as signal to noise ratio (SNR) and shifts the computation/communication overhead completely on IoT gateways. Moreover, it addresses the heterogeneity issue of 5G–based IoT systems since it is compatible with any wireless protocol at the physical layer and independent to upper layers protocols. Also, our experiments (security analysis) confirms that the proposal is secure from replay attacks.

Index Terms—5G Security; Deep Learning; Internet of Things (IoT); Intrusion Detection; Physical Layer Security.

I. INTRODUCTION

THERE will be around 14.7 billion IoT connections by 2023 which is nearly equal to the half of the global connected devices [1], [2]. It is unfortunate that the significant benefits of developing advanced IoT applications are getting at the price of compromising new security vulnerabilities. Indeed, developing a large–scale IoT network by using cheap and insecure devices (which are easy to temper), could have a catastrophic result. For example, in 2016, a massive Distributed Denial of Service (DDoS) attack had occurred on many popular websites, including Netflix, Twitter, CNN, and Reddit in Europe and the US [3] which had knocked down the services of these websites. The attack was conducted against Dyn (a large DNS provider with a big share in controlling the internet’s DNS infrastructure) to disrupt DNS service in Europe and North America. It cleverly had took advantage of poorly secured/insecure IoT devices (such as IoT–enabled CCTVs) to conduct a massive attack [4].

Therefore, to protect networks, Intrusion Detection Systems (IDSs) have been recognized as a fundamental and effective

tool for intruder/anomaly detection [4], [5]. These IDSs are working based on the detection of suspicious network activities (network–based IDSs). However, network–based IDSs or approaches are not efficient for anomaly detection in large scale IoT networks [4], [1]. In addition, researchers in [6] and [4] has validated that the high level of heterogeneity (in IoT devices, technologies, and protocols) adversely impact the security solutions, means the heterogeneity in IoT stands in the way of achieving good security. To address this issue, a number of research efforts have been made to develop field–based IDS solutions [4], [2]. Unfortunately, limited computation, storage, and energy resources of wireless IoT devices prevents them from performing complex security protocols (e.g., EPS–AKA) that impose high computation/communication overheads [4], [5], [2], [7]. In view of all these points, we reiterate that, adopting the network–based IDS approach may not result in an efficient detection of intrusions in large scale 5G enabled heterogeneous IoT networks with a huge number of IoT nodes [4], [1].

Motivated from this, we have proposed *Physical Layer–Assisted IDS (PLAIDS)*, learning–based IDS for 5G–enabled IoT heterogeneous networks. We have utilized the unique physical layer features, actually caused by the inconsistencies in Transmitter (Tx) manufacturing processes, of legitimate IoT wireless devices to recognize an intruder device. This feature uniqueness is due to the nonideality of Radio Frequency (RF) circuits in the Tx module of IoT wireless devices. It is actually caused by the inconsistencies in Tx manufacturing processes which is almost unavoidable (or at least very expensive to be avoided) [8]. For example, considering IEEE 802.11ac standard, a maximum frequency offset of ± 116 kHz (± 20 ppm) is allowed for every Tx module [9]. Thus, RF features such as Carrier Frequency Offset (CFO), Phase Offset in the in–phase and quadrature components of the Tx signal (PO–I and PO–Q, respectively), etc. can be considered as inherent RF properties of Tx modules. These are known as *RF signature* of a Tx module in the literature [10]. However, this has no negative impact on the performance of a Tx module as long as it complies with the specifications of relevant wireless standard and will be compensated at the receiver circuit (Rx).

We embrace these nonideality of Tx modules and develop a Deep Autoencoder (DAE) model [11] to learn the RF signature of legitimate IoT devices in the network. In PLAIDS, a DAE model serves a single 5G network slice resides at the core segment of the 5G network. This is because in multi–tenant 5G networks, there is separate Management and Orchestration (MANO) unit for every network slice through which each

Authors are with Centre of Cyber Security Research Innovation (CSRI), School of Information Technology, Deakin University, Geelong, Australia. E-mail: keshav.ood@deakin.edu.au; m.nosouhi@deakin.edu.au; frank.jiang@deakin.edu.au; morshed.chowdhury@deakin.edu.au; robin.doss@deakin.edu.au

tenant can manage the security of its own slice [6]. This is done with the collaboration of IoT gateway/edge computing devices that have direct radio communication links to the IoT wireless devices and can extract their RF signature.

The contributions of this paper are as follows.

- We propose PLAIDS, a highly accurate and lesser time computational approach leverages physical layer features of wireless IoT devices in 5G-IoT networks to authenticate wireless IoT nodes.
- We propose a high level framework to shown the integration of PLAIDS into the standard 5G ETSI architecture. The approach can be used either as a single stand-alone authentication approach or it could be a part of the multi-factor authentication process in scenarios where high trust is required.
- To justify that the PLAIDS is secure from attacks, a security model is presented with a solution that makes the malicious replay efforts unsuccessful. In fact, the proposed solution makes it infeasible for a malicious device to reuses the RF signals of legitimate devices (obtained through eavesdropping attempts) to get access to the network.

Benefits. The key benefits of PLAIDS are: (1) Interestingly, PLAIDS does not require any additional hardware at the Tx side. Thus, it results in no (zero) computation/communication overhead on the resource-constrained IoT nodes. In fact, the computation load required to perform the intrusion detection procedure is completely transferred to the IoT gateway/edge computing devices and deep learning server. (2) PLAIDS works with any Tx module/wireless protocol at the physical layer. It is also independent of the upper layers protocols. Thus, the heterogeneity issue of 5G-based IoT systems does not degrade its performance. (3) It is well suited for slice-isolated 5G-based IoT networks and can be effectively integrated into the standard architecture of 5G networks, later shown in Section IV. (4) PLAIDS enables network providers to offer the authentication service to IoT applications and cloud-based services (e.g., in the form of a multi-factor authentication mechanism). For more benefits PLAIDS offers in 5G, please refer Section IV.

Novelty. In contrast to the existing works, to the best of our knowledge, this is the first research work that uses RF signature for intrusion detection in 5G-based IoT systems. The proposal is fully compatible with 5G enabled IoT networks as shown in this paper. We also analyzed the security of PLAIDS itself to show that it is a secure approach for intrusion detection in IoT networks. In our experiments, we reached 99.5% accuracy in the detection of intruder devices which confirms the effectiveness of PLAIDS.

II. RELATED WORK

We have divided the existing IDS solutions (for IoT networks) into four groups which are based on the adopted detection approaches.

1. Signature-based IDSs approaches. There are some works given in [12], [13] in which a signature is defined for every known attack. When a suspicious network or system behaviour is detected (for network-based and host-based IDSs,

respectively), an alert is triggered if the suspicious behaviour is matched with the signature of an attack. This technique is usually an accurate and effective approach for detecting known attacks. However, it is ineffective to detect new attacks since no signature is available for them.

2. Anomaly-based approaches. In [14] a profile of normal network activities is defined as a reference and any deviation from this normal behaviour generates the alert. Although this approach is effective in the detection of new attacks, it may result in high false positive rates [5]. This is a result of the high level of complexity in defining a profile of normal network activities. Thus, in case of selecting a tight threshold, any network activity that does not completely match the normal profile is regarded an intrusion [4], [5].

3. Specification-based approaches. In this approach [15], [16], predefined specifications (sets of rules/profiles that determine the normal network behaviour) are used by the system to detect any anomaly. However, unlike the second approach, the rules and profiles in the specifications are manually defined by a human expert. This results in better performance (lower false positive rates). However, this approach may not be effective in highly dynamic network environments and is prone to human faults and errors. In addition, it can be a time-consuming process to manually generate the specifications [5], [15].

4. Hybrid approaches. Finally, there are some hybrid approaches utilize positive features of the signature-based, anomaly-based, and specification-based methods to increase their effectiveness and limit the weaknesses [17], [18]. However, the solutions based on this approach are mostly too complicated with additional computation overheads [5], [18]. Moreover, in some cases, they result in low accurate intrusion detection or high false positive rates [19], [17].

Regarding the *field-based IDSs*, a novel detection mechanism has been recently proposed in [20] that uses one-class classifiers to detect botnets in IoT devices. To train the model, the classifiers do not need a labelled dataset of malicious activities. They have achieved a high level of accuracy in the botnets detection. However, it only works on IoT devices with a specific level of storage/processing abilities [20]. Authors in [4] proposed that the multi-level information on the monitored devices is collected using some tracing techniques. The collected data is then sent to an analysis engine which can be located either on a device inside the network system or on a dedicated machine on the cloud. The analysis engine uses machine learning algorithms to detect anomalies on the device behaviour. However, it only works with the IoT devices that support the TCP/IP protocol stack (i.e., it suffers from the heterogeneity issue in terms of supported protocols). Moreover, its performance heavily depends on the learning process and the training data set (supervised learning models have been employed) [4].

In [10], a learning-based authentication mechanism has been proposed based on the uniqueness of RF features. The authors have used a supervising machine learning model to uniquely identify 10,000 different wireless devices and achieved the high accuracy of 99% in their experiments. However, the deployed learning model has not been clearly discussed in the paper. This is very important since supervised

Fig. 1: A high level architecture of the proposed approach.

machine learning models usually have poor classification performance in the scenarios with a large number of classes. In addition, they have not proposed a defence mechanism against replay attacks to show the effectiveness of the proposal against attacks.

III. PLAIDS: THE PROPOSED SOLUTION

In Fig. 1 we present our proposed IDS approach for 5G-based IoT networks. We consider that the unclonable RF signals with inherent unique properties, (Tx signatures), are received by the feature extraction module resides at RAN segment of 5G network. This module performs the real-time features extraction (select key RF signals) of wireless IoT devices which are then used to train a Deep Learning based anomaly detection module (model) for it to learn these unique RF features of the legitimate devices in the network. Eventually to distinguish legitimate and non-legitimate IoT node.

A). Feature Extraction (FE) Module: In the field of RF fingerprinting, two different techniques for RF feature extraction have been used. These are (1) transient signal analysis and (2) steady-state feature extraction, in which the RF features are extracted from the transient and steady-state parts of the received signal, respectively. The transient segment of a signal starts immediately after the transmitter is powered on and lasts for a very short period of time (e.g., in the range of μsec). This is the period during which the output of hardware components of the transmitter (e.g., modulator, power amplifier, etc.) make the transition to their steady-state level. After this period, the transmitter sends the data which is the start of steady-state segment of the RF signal. Although feature extraction from the transient part of the signal results in high identification accuracy (due to the high level of uniqueness in the extracted features) as well a high level of security, it needs the received signal to be sampled at very high rates [10]. Moreover, the beginning and end points of the transient segment should be accurately identified. These requirements make the required hardware for feature extraction complicated and expensive.

The steady-state feature extraction approach does not suffer from these issues. However, it is dependent on the data being sent. In fact, in different data transmissions (with different data) the extracted features may slightly change. This degrades the overall performance in terms of accuracy. To address this issue, the wireless device needs to send (predefined) preamble signals along with the data to enable the detection module at the receiver side to identify the device. In fact, in this method, the features are extracted and analyzed based on the knowledge of the preamble signal sent by the device. In other words, the

Fig. 2: The feature extraction procedure.

extracted features are compared with a precalculated set of features that are expected to be extracted from the signal of a specific device.

However, this technique (sending preamble signals) causes two issues. First, it may result in security vulnerabilities since a wireless device might be impersonated through conducting a replay attack. In addition, it needs some changes in the Tx module to perform the preamble transmission at specific times. Moreover, each standard wireless transmission protocol has its own form of preamble signal which results in implementations of the feature extraction module. This prevents the authentication solution to be independent of the transmission protocol at physical layer. As a result, the preamble-based solutions are inefficient for large scale IoT networks with heterogeneous resource-constraint IoT devices. To address this issue, we can remove the need of preamble transmission by deploying a learning-based detection model (instead of a deterministic model) and train it with a sufficient number of feature sets resulted from sending different bit streams. In fact, the detection model learns the possible sets of features (as many as required to achieve an acceptable detection accuracy) based on different data streams transmitted by the wireless device. This makes the detection performance independent of the data that is being transmitted.

In conclusion, we have adopted a steady-state feature extraction approach in which the need of preamble transmission has been eliminated (by deploying a Deep Learning-based anomaly detection method). To perceive the feature extraction procedure, we have considered an IoT device (equipped with a digital radio transmission system) that is transmitting data to a wireless IoT gateway, as shown in Fig. 2. At the Tx side (considering a QAM modulation/demodulation system), the flow of input data is first applied to a serial to parallel converter (SPC). Then, two separate Digital to Analog Converters (DACs) receive the two parallel bit streams to form the in-phase ($I(t)$) and quadrature ($Q(t)$) components of the modulating signal. These components modulate the two relevant carrier signals that have same frequency but out of phase with each other by 90° (i.e., $\sin 2\pi f_c t$ and $\cos 2\pi f_c t$). The modulated signals are then added together to create $x(t)$ which is filtered, amplified, and transmitted. The orthogonality of the two carrier signals enables the demodulator circuit (at

the receiver (Rx) side) to easily separate them and obtain the I and Q components. So far, the wireless IoT device just performs its normal (routine) procedures in sending the data to the IoT gateway, i.e., it does not need to collaborate with the gateway's Rx module for the fulfilment of feature extraction process (e.g., transmitting a preamble signal).

At the Rx side, DC Offset is the first feature that can be extracted by means of the capacitor coupled Low Noise Amplifier (LNA). Then, using the local carrier synchronization unit, the Carrier Frequency Offset (CFO) feature of the Tx local oscillator can be measured [21]. This unit adjusts the frequency of local oscillator (applied to the mixers) based on the frequency of the received signal. This is done to compensate the effect of CFO in the receiver circuit. Note that in this case, we need to have a very accurate reference clock to be able to measure CFO with the required resolution.

In the next stage, using the I-Q imbalance compensator unit, the phase offset and amplitude mismatch in the in-phase and quadrature components of the base-band signal can be obtained [22]. These features are unique for every transmitter and can be effectively used in device identification [23]. Power Amplifier Back-off (PAB) is another feature that can be extracted at this stage. It is caused when the power amplifier in the Tx module is driven with a voltage that is high enough to saturate the output (i.e., the amplifier enters the non-linear region of its operation). This feature can be obtained through I-Q baseband imbalance measurements. Finally, the measured features (in vector form) are applied to the DAE-based AD module for the real-time detection of anomaly.

B). Anomaly Detection (AD) Module: The aim of this module is to find out that (with high probability) the extracted RF features are associated with either a legitimate IoT device or an intruder. This is done through developing a Deep Learning model which is trained to learn the unique RF features of the legitimate devices in the network. We consider that each network slice has its own security manager (PLAIDS), so an AD module serves a single network slice this is also to address the scalability issues. It learns the features of all the 5G-IoT devices that are taking 5G service from the relevant tenant as later shown in Fig. 3. This makes PLAIDS compatible to the standard architecture of 5G networks.

We have deployed a Deep Autoencoder (DAE) as the core of our learning model. DAEs are a type of Artificial Neural Networks (ANN) that are known as an effective approach to perform anomaly detection in an unsupervised manner. The other benefit of deploying a learning model is that it eliminates the need of preambles (i.e., predefined fixed bit streams) in the received RF signals. As discussed before, in the steady-state RF fingerprinting mechanisms, the wireless device needs to send a preamble before sending the data. This enabled the detection module (at the receiver side) to compare the RF features (after extraction from the received signal) with the expected (pre-calculated) values. This is indeed a significant advantage of PLAIDS because the wireless IoT devices do not need to collaborate with the intrusion detection module, i.e., they just follow their own routine procedures. Thus, no hardware/software change or update is required at the IoT devices. In addition, deploying a learning-based detection

module makes the system resilient against the changes in RF features of individual devices. These changes may occur due to several reasons such as change of environmental parameters (e.g., temperature), channel conditions, battery voltage, etc. All these changes can be considered during the model training phase to increase the detection accuracy.

C). Mathematical Analysis: Assume $\mathbf{F} \in \psi$ is the set of training vectors $\mathbf{f}^i = \{f_1, f_2, \dots, f_n\}$ where ψ is an n -dimensional feature spaces. If $g_\alpha : R^n \rightarrow R^m$ and $g'_\beta : R^m \rightarrow R^n$ are the encoding and decoding functions, respectively, we have

$$\mathbf{e}^i = g_\alpha(\mathbf{f}^i) \quad (1)$$

$$\hat{\mathbf{f}}^i = g'_\beta(\mathbf{e}^i) = g'_\beta \circ g_\alpha(\mathbf{f}^i) \quad , \quad (2)$$

where $\mathbf{e}^i = \{e_1, e_2, \dots, e_m\}$ and $\hat{\mathbf{f}}^i = \{\hat{f}_1, \hat{f}_2, \dots, \hat{f}_n\}$ are the representation (encoded) and reconstructed (decoded) vectors, respectively, associated with the input vector $\mathbf{f}^i$. Note that $\mathbf{E} \in \theta$ shows the set of encoded vectors where θ is an m -dimensional feature spaces ($m < n$). α and β are parameters of the encoder and decoder functions, respectively.

Now, assume there is an unknown probability distribution ρ defined over ψ . Given Δ as a dissimilarity (error) function (such as Mean-Absolute Error (MAE), Mean-Squared Error (MSE), Euclidean Distance, etc.), the relevant *autoencoder problem* is to find α and β such that the expected value of the dissimilarity function Δ is minimized, i.e.

$$\min_{(\alpha, \beta)} E_{(\mathbf{f}^i, \hat{\mathbf{f}}^i) \sim \rho}(\Delta(\mathbf{f}^i, \hat{\mathbf{f}}^i)) = \min_{(\alpha, \beta)} E(\Delta(\mathbf{f}_i, g'_\beta \circ g_\alpha(\mathbf{f}_i))) \quad (3)$$

As the probability distribution ρ is unknown, it is not feasible to obtain the expected value of the dissimilarity function, so we limit the autoencoder problem to the space of the training vectors, i.e.

$$\min_{(\alpha, \beta)} \Delta(\mathbf{f}^i, \hat{\mathbf{f}}^i) = \min_{(\alpha, \beta)} \sum_{j=1}^n \Delta(f_j, g'_\beta \circ g_\alpha(f_j)) \quad (4)$$

The above problem is solved for every $\mathbf{f}^i$ and $\hat{\mathbf{f}}^i$ in $\mathbf{F}$ and $\hat{\mathbf{F}}$ (respectively), i.e., all the vectors in the training dataset are learnt. Different types of autoencoders can be derived from this general model depending on the choice of functions f_α , f'_β , and the dissimilarity function Δ . Moreover, applying additional constraints such as regularization can change the type of autoencoder, e.g., if MSE is selected as the dissimilarity function, for the autoencoder problem we have

$$\min_{(\alpha, \beta)} \frac{1}{n} \sum_{j=1}^n (f_j - \hat{f}_j)^2 = \min_{(\alpha, \beta)} \frac{1}{n} \sum_{j=1}^n (f_j - g'_\beta \circ g_\alpha(f_j))^2 \quad (5)$$

To solve the above autoencoder problem, gradient-based optimization approach is a popular and effective method to choose. There exists several versions of gradient-based optimization algorithms. For example, in Batch Gradient Descent (BGD), the gradients of all samples are calculated at first. Then, based on the obtained gradients, the neural network parameters are updated. However, it is used in offline training

Algorithm 1: PLAIDS mechanism

Inputs: d_i (address of the i th legitimate IoT device)
 n (number of features)
 e_{thr} (error threshold)
Output: $R \in \{\text{"Authorized Access"}, \text{"Unauthorized Access"}\}$

```

1:  $\mathbf{f}(d_i, t) = FE\_VNF(d_i)$ 
2:  $\hat{\mathbf{f}}^i = AD\_VNF(\mathbf{f}(d_i, t))$ 
3:  $error = \Delta(\mathbf{f}(d_i, t), \hat{\mathbf{f}}^i) = \frac{1}{n} \sum_{j=1}^n (f_j(d_i, t) - \hat{f}_j^i)^2$ 
4: if  $error \leq e_{thr}$ :
    return "Authorized Access"
else:
    return "Unauthorized Access"
  
```

applications in which the whole set of the training dataset is available. However, in online (real-time) applications, training samples may become available after the model is employed. On the other hand, Stochastic Gradient Descent (SGD) can be used in online training applications. Each time, it updates the parameters using an instant training sample. In other words, in BGD, all the training samples must be learnt before a single update is done on the network parameters. However, in SGD, one or a subset of the training samples can be learnt in order to update the network parameters. This makes SGD an efficient optimization algorithm. Specifically, in high-dimensional optimization problems, SGD performs very efficient in terms of speed and computational overhead.

Since we build the autoencoder for an online (real-time) application with a huge number of data points, we solve the autoencoder problem using the SGD approach. Therefore, we have

$$\alpha^{(k+1)} = \alpha^{(k)} - \epsilon^{(k)} \nabla_{\alpha} \Delta_i(\alpha^{(k)}) \quad (6)$$

$$\beta^{(k+1)} = \beta^{(k)} - \epsilon^{(k)} \nabla_{\beta} \Delta_i(\alpha^{(k)}) \quad (7)$$

where $\nabla_{\alpha} \Delta_i(\alpha^{(k)})$ and $\nabla_{\beta} \Delta_i(\alpha^{(k)})$ are the gradients taken using α and β , respectively (considering a training sample $\mathbf{f}^i$). ϵ is the learning rate that is used to adjust the speed of convergence. It determines the size of steps that are taken to reach the optimum parameters. Using larger values for ϵ results in faster training but at the risk of missing the optimum values (loss in accuracy). On the other hand, a smaller ϵ makes the convergence of algorithm slower. When the optimization problem is solved, parameters α and β are obtained. This means that the autoencoder model has been built and vectors $\hat{\mathbf{f}}^i = g'_{\theta} \circ g_{\phi}(\mathbf{f}_j)$ can be obtained as the reconstructed vectors.

IV. PLAIDS INTEGRATION WITH 5G NETWORKS

In 5G, virtually isolated slices of the network are allocated to Mobile Virtual Network Operators (MVNOs). We assume that every slice is managed by its own orchestration module, this is in line to the existing research work shown in [6]. Regarding security, the slice's orchestration module coordinates the security functions via a security manager module as we have shown in Fig. 3. Since PLAIDS is a security-relevant mechanism, it can be performed by the security manager of each slice. Or say, each network size has its own orchestration module, and PLAIDS is a sub-module of each orchestration module. In this approach, each part/module of the PLAIDS

Fig. 3: A High Level Framework of PLAIDS integration with 5G networks.

mechanism is performed through invoking one or multiple Virtual Network Functions (VNFs). These VNFs are managed and customised by the slice's security manager. Using this approach, PLAIDS not only receives (automatically) the benefits of the standard ETSI framework/architecture of 5G networks, but also it becomes a practical solution due to the following reasons.

- 1). Network slices are virtually isolated and might be managed by different entities. Thus, considering the differences in security policies of different tenants, it is wise decision to perform any field-based IDS mechanism using a per-slice method.
- 2). The tenant of a slice is the appropriate entity that can make necessary interactions with end users to collect and store the hardware features of users' IoT devices (when they join the network for the first time) to train and update the learning model in the AD module.
- 3). PLAIDS enables the network provider to offer the authentication service to IoT applications and cloud-based services (e.g., in the form of a multi factor authentication mechanism). Moreover, it is much easier to manage and handle the device authentication requests (received from IoT applications with different security requirements) using a slice-based approach.
- 4). In terms of scalability, this approach offers a high level of efficiency because the task of authenticating a huge number of IoT devices is shared between individual learning models run by different slices. As a result, the learning model used in the AD module does not need to learn the features of all the IoT devices in the network. This is indeed a significant advantage for an IDS solution in large-scale 5G-based IoT networks.

The slice's security manger performs the PLAIDS mechanism by invoking the relevant VNFs. It first invokes $FE_VNF()$ that extracts and returns the real-time RF features of the device, as seen in Algorithm 1. This is done through the associated IoT gateway/edge device that has established a direct wireless link to the device. Upon receiving the real-time extracted features, the security manger invokes the $AD_VNF()$ which performs the anomaly detection procedure explained earlier. In addition to $FE_VNF()$ and $AD_VNF()$, a number of other VNFs should be defined to handle the other procedures of PLAIDS. For example, $FE_ND_VNF()$ is invoked when a new device is joining to the network. Note that the feature extraction process done in such cases is more comprehensive

TABLE I: RF features used in our experiments.

Feature	Mean	Standard Deviation
Carrier Frequency Offset (CFO)	2.4 GHz	48 kHz (20 ppm)
Phase Offset (In-Phase) (PO-I)	0°	10°
Phase Offset (Quadrature) (PO-Q)	0°	10°
Amplitude Mismatch (In-Phase)(AM-I)	0 dB	3 dB
Amplitude Mismatch (Quadrature) (AM-Q)	0 dB	3 dB

than what is done by *FE_VNF()*. In fact, the RF features of a new device should be extracted and learned in different ways, e.g., using different bit streams, levels of transmission power, etc. This is done to make the AD module (1) a preamble-less detection method and (2) resilient to unavoidable situations regarding the possible changes in the condition of communication channel, environment, level of transmission power, etc.

V. PERFORMANCE EVALUATION

We have used the Wireless Waveform Generator toolbox of MATLAB to generate RF signals based on random bit streams of data to create a dataset in which the RF features of 100 wireless devices have been collected. For every device, we have slightly changed the frequency, phase, and DC parameters to model the nonideality of RF circuits in the Tx modules. We have changed the signal to noise ratio (SNR) parameter to consider the effects of channel situation in our dataset. Finally, we have created a file of size 1 MB consisted of pseudo-random bits (generated by the *random* module of Python) and applied the file to the wireless waveform generator module to generate the RF signals for each device. This makes the AD module independent of the input bit stream, i.e., it learns the features of a device from a variety of RF signals generated based on a sufficient number of random input bit streams. It also eliminates the need of preambles in the input bit streams which is an advantage of PLAIDS. Moreover, changing the SNR parameter enables the AD module to learn the features under different channel circumstances.

We train the AD module as much as possible and in a variety of situations in terms of the input bit stream, channel circumstance, environmental situation, etc. We have performed the experiments at 2.4GHz and based on the QAM-16 modulation. Finally, using the generated signals, we have obtained five RF features (Table I) for each device as CFO, PO-I, PO-Q, AM-I, and AM-Q in different iterations based on the applied pseudo-random bit streams. In the second part of the experiments, we have used the obtained dataset to develop a DAE model in *TensorFlow* 2.0 and *keras* libraries of Python. We have used 70% of the dataset for training, and 10% and 20%, respectively for the validation and test procedures.

The number of devices are increased from 20 to 100 to see the effect on the PLAIDS's detection performance. We have expected the detection accuracy to be considerably degraded in the scenarios with a larger number of devices because in such cases, (intuitively) the feature vectors of distinct devices may become closer to each other. This makes sense in traditional supervised machine learning models (e.g., SVM) in which classification performance is notably decreased in high dimensional classification scenarios. However, in our case, the

amount of reduction in the detection accuracy was not significant. This is indeed a unique attribute of deep autoencoders that has made them as an effective learning model for the fulfillment of anomaly detection tasks. As seen in Fig. 4(a), increasing the number of devices from 20 to 100 results in a 1% reduction in the detection accuracy (approximately) for each level of SNR. We have noticed that a higher level of SNR improves the detection performance because an RF signal is less distorted in a less noisy channel than a very noisy channel which results in only small changes in the RF features extracted from the received noisy signal. Consequently, the extracted feature vector is more similar to the vectors that the AD module has learned during the training phase. However, this behaviour is seen in low levels of SNR only, i.e., if SNR is high enough (say greater than 20 dB), any increase in the SNR does not result in a significant improvement of accuracy.

We have examined different architectures of the DAE model. As Fig. 4(b) indicates, by increasing the number of hidden layers the detection accuracy increases as well because each layer provides a deeper level of knowledge for the model. However, we did not record any significant difference in the accuracy for the scenarios in which three and four hidden layers have been implemented. Since deploying more hidden layers results in a longer training time for the model (Fig. 5(c)), the optimum selection for the number of hidden layers in our experiments is three. Regarding the effect of number of epochs on the detection accuracy, the experiments confirmed that the DAE model performs better if a higher number of epochs is used during the training phase as seen in Fig. 4(c). This is due to the fact that by applying a higher number of epochs may result in overfitting of the model. To avoid overfitting, we have enabled the *EarlyStopping()* feature of the *keras* library in Python so that the training procedure will be automatically stopped as soon as the model becomes over-fitted.

We have also used three different activation functions in our implementations, i.e., *Sigmoid*, *Tanh*, and *ReLU*. The best results were obtained using the *Tanh* activation function as shown in Fig. 5(a). In Fig. 5(b), we have noticed the effect of encoding dimension on the detection accuracy. The accuracy reduces for lower encoding dimensions because in these cases, too much compression is done on the input vectors which makes the decoding procedure more difficult, eventually less accurate. Moreover, for a fixed number of encoding dimension, (as discussed before) increasing the number of hidden layers results in a more accurate detection. We have also trained the DAE model using different batch sizes on the training data. As we have expected, the training time is significantly affected by a change in the batch size. This is because the DAE model learns all the data available in a batch (and updates the parameters of neurons) before the next batch is learned. Thus, a small batch size increases the number of times that model learns and updates its parameters which results in much longer training phase. Fig. 5(c) shows the result. Moreover, employing a higher number of hidden layers has the disadvantage of having longer training time.

Fig. 4: Average detection accuracy of the AD module for different number of wireless devices and different values of (a) SNR, (b) number of hidden layers, and (c) number of epochs.

Figure 10 consists of three subplots (a), (b), and (c) showing the performance of the proposed model under different configurations.

(a) Average Detection Accuracy (%) vs. Number of Tx Modules. The plot shows the performance of three activation functions: Sigmoid (blue line with circles), ReLU2 (red line with squares), and Tanh (black line with circles). The accuracy generally decreases as the number of Tx modules increases from 20 to 100.

Number of Tx Modules	Sigmoid (%)	ReLU2 (%)	Tanh (%)
20	99.1	99.3	99.5
40	98.8	99.1	99.4
60	98.6	98.8	99.1
80	98.3	98.6	99.0
100	97.9	98.2	98.7

(b) Average Detection Accuracy (%) vs. Encoding Dimension. The plot shows the performance of the model with 2, 3, and 4 hidden layers across different encoding dimensions (1, 2, 3, 4). The accuracy is relatively stable, ranging from approximately 98.7% to 99.6%.

Encoding Dimension	2 Hidden Layers (%)	3 Hidden Layers (%)	4 Hidden Layers (%)
1	98.7	98.8	98.9
2	98.8	99.0	99.1
3	99.1	99.2	99.2
4	99.4	99.5	99.6

(c) Training Time (ms) vs. Batch Size. The plot shows the training time for 2, 3, and 6 hidden layers across different batch sizes (32, 64, 128, 256, 512). The training time decreases as the batch size increases.

Batch Size	2 Hidden Layers (ms)	3 Hidden Layers (ms)	6 Hidden Layers (ms)
32	920	910	1150
64	700	810	980
128	620	650	780
256	540	580	710
512	510	520	620

VI. SECURITY ANALYSIS

$FE_VNF()$ function to perform the feature extraction procedure at the IoT gateway/edge computing device. At the same time, it invokes a timer initialized with a short and predefined period of time T . However, when the server receives the NAR message of D_M , it does not reply to it until the timer is up. Upon receiving the AR message, D_L transmits the reply signal which is processed by the FE module to extract its features. This signal is targeted by D_M to be intercepted and re-transmitted to the server. However, D_M has not received any AR message from the network yet, thus, it can not re-transmit the signal immediately.

In other words, the eavesdropped signal must be delayed by D_M until the relevant AR message is received. This is an infeasible task for D_M to perform if T is large enough (e.g., 1 msec). The reason is that delaying the transmission of an RF signal without making changes on the amplitude, frequency, and phase features of the signal is infeasible. The current solutions for the implementation of passive delay lines work mostly based on either coaxial/optical fibre cables [24] (note that active delay lines do not work for the attacker in this attack scenario since they definitely change the RF features of the signal). However, these solutions are not practical when the required delay should be in the millisecond (or larger) ranges. For example, considering the speed of light in a coaxial/optical fibre cable, 200 metres of the cable is required to delay the signal by 1 microsecond.

Thus, adopting T in the millisecond range prevents the attacker to deploy an appropriate passive delay line. On the other hand, selecting a large value for T may result in performance degradation (latency) of the IoT application. For example, considering $T = 1$ msec, the server can authenticate 1000 devices per second (regardless of other delays caused by signal propagation, processing, software running, etc.) and the created delay is much smaller than the latency requirements of most of the latency critical IoT applications [25].

VII. SUMMARY AND FUTURE WORK

An IoT node authentication approach to detect malicious IoT nodes using the unique RF features of the wireless legitimate IoT devices is presented. The approach is effective and accurate under different channel conditions. We also proposed a framework for the integration of PLAIDS into the standard 5G architecture. We have validated that the proposal in itself is secure from certain attacks, i.e, replay attacks. This we have demonstrated by mounting an attack and by proposing a novel delay-based solution that makes the malicious replay efforts unsuccessful. In future, PLAIDS's proper integration and the performance evaluation into the existing cloud (virtual) management technologies to capture the difficulty of trying to integrate the proposed IDS into a real-life network slicing architecture, will be conducted. Another interesting research direction is to utilize the unique RF features for the development of a secure data provenance mechanism to maximize trustworthiness in 5G-base IoT networks.

REFERENCES

- [1] L. Cui, Y. Qu, G. Xie, D. Zeng, R. Li, S. Shen, and S. Yu, "Security and privacy-enhanced federated learning for anomaly detection in iot infrastructures," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2021.
- [2] P. Gope, B. Sikdar, and O. Millwood, "A scalable protocol level approach to prevent machine learning attacks on puf-based authentication mechanisms for internet-of-medical-things," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2021.
- [3] G. Kambourakis, C. Kolias, and A. Stavrou, "The mirai botnet and the iot zombie armies," in *MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)*, 2017, pp. 267–272.
- [4] R. Gassais, N. Ezzati-Jivan, J. M. Fernandez, D. Aloise, and M. R. Dagenais, "Multi-level host-based intrusion detection system for internet of things," *Journal of Cloud Computing*, vol. 9, no. 1, pp. 1–16, 2020.
- [5] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in internet of things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.
- [6] K. Sood, K. K. Karmakar, V. Varadarajan, N. Kumar, Y. Xiang, and S. Yu, "Plug-in over plug-in (pop) evaluation in heterogeneous 5g enabled networks and beyond," *IEEE Network*, pp. 1–7, 2021.
- [7] S. Behrad, E. Bertin, S. Tuffin, and N. Crespi, "A new scalable authentication and access control mechanism for 5g-based iot," *Future Generation Computer Systems*, vol. 108, pp. 46–61, 2020.
- [8] S. Sen, V. Natarajan, S. Devarakond, and A. Chatterjee, "Process-variation tolerant channel-adaptive virtually zero-margin low-power wireless receiver systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 12, pp. 1764–1777, 2014.
- [9] R. P. F. Hoefel, "Ieee 802.11ax: A study on techniques to mitigate the frequency offset in the uplink multi-user mimo," in *2016 8th IEEE Latin American Conference on Communications (LATINCOM)*, 2016, pp. 1–6.
- [10] B. Chatterjee, D. Das, S. Maity, and S. Sen, "Rf-puf: Enhancing iot security through authentication of wireless nodes using in-situ machine learning," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 388–398, 2019.
- [11] G. Zhang, Y. Liu, and X. Jin, "A survey of autoencoder-based recommender systems," *Frontiers of Computer Science*, vol. 14, no. 2, pp. 430–450, 2020.
- [12] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, "Denial-of-service detection in 6lowpan based internet of things," in *2013 IEEE 9th international conference on wireless and mobile computing, networking and communications (WiMob)*. IEEE, 2013, pp. 600–607.
- [13] P. Kasinathan, G. Costamagna, H. Khaleel, C. Pastrone, and M. A. Spirito, "An ids framework for internet of things empowered by 6lowpan," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 1337–1340.
- [14] G. Abdelmoumin, D. B. Rawat, and A. Rahman, "On the performance of machine learning models for anomaly-based intelligent intrusion detection systems for the internet of things," *IEEE Internet of Things Journal*, pp. 1–1, 2021.
- [15] J. P. Amaral, L. M. Oliveira, J. J. P. C. Rodrigues, G. Han, and L. Shu, "Policy and network-based intrusion detection system for ipv6-enabled wireless sensor networks," in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 1796–1801.
- [16] S. Misra, P. V. Krishna, H. Agarwal, A. Saxena, and M. S. Obaidat, "A learning automata based solution for preventing distributed denial of service in internet of things," in *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, 2011, pp. 114–122.
- [17] J. Krimmling and S. Peter, "Integration and evaluation of intrusion detection for coap in smart city applications," in *2014 IEEE Conference on Communications and Network Security*. IEEE, 2014, pp. 73–78.
- [18] C. Cervantes, D. Poplade, M. Nogueira, and A. Santos, "Detection of sinkhole attacks for supporting secure routing on 6lowpan for internet of things," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2015, pp. 606–611.
- [19] S. Raza, L. Wallgren, and T. Voigt, "Svelte: Real-time intrusion detection in the internet of things," *Ad hoc networks*, vol. 11, no. 8, pp. 2661–2674, 2013.
- [20] V. H. Bezerra, V. G. T. da Costa, S. Barbon Junior, R. S. Miani, and B. B. Zarpelão, "Iotds: A one-class classification approach to detect botnets in internet of things devices," *Sensors*, vol. 19, no. 14, p. 3188, 2019.
- [21] Y. H. You and H. K. Song, "Efficient sequential detection of carrier frequency offset and primary synchronization signal for 5g nr systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 9212–9216, 2020.
- [22] H. Cheng, Y. Xia, Y. Huang, L. Yang, and D. P. Mandic, "A normalized complex lms based blind i/q imbalance compensator for gfdm receivers

and its full second-order performance analysis,” *IEEE Transactions on Signal Processing*, vol. 66, no. 17, pp. 4701–4712, 2018.

- [23] K. Merchant, S. Revay, G. Stantchev, and B. Noursain, “Deep learning for rf device fingerprinting in cognitive communication networks,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 160–167, 2018.

- [24] T. Manzanque, R. Lu, Y. Yang, and S. Gong, “Low-loss and wideband acoustic delay lines,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 67, no. 4, pp. 1379–1391, 2019.

- [25] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, I. Riedel *et al.*, “Latency critical iot applications in 5g: Perspective on the design of radio interface and network architecture,” *IEEE Communications Magazine*, vol. 55, no. 2, pp. 70–78, 2017.