IoT Devices Fingerprinting using Deep Learning

Hossein Jafari, Oluwaseyi Omotere, Damilola Adesina, Hsiang-Huang Wu and Lijun Qian
CREDIT Center
Prairie View A&M University, Texas A&M University System
Prairie View, TX 77446, USA
Email: hjafari@student.pvamu.edu, {omotere, damiadesina87, virtuoso.wu}@gmail.com, liqian@pvamu.edu

***Abstract*—Radio Frequency (RF) fingerprinting as a physical layer authentication method could be used to distinguish legitimate wireless devices from adversarial ones. In this paper, we present a wireless device identification platform to improve Internet of things (IoT) security using deep learning techniques. Deep learning is a promising method for obtaining the characteristics of the different RF devices through learning from their RF data. Specifically, three different deep learning models, namely Deep Neural Network (DNN), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN) are considered here to identify wireless devices and distinguish among wireless devices from the same manufacture. As a case study, large data sets of RF traces from six "identical" ZigBee devices are collected using a USRP based test bed. We captured RF data across a wide range of Signal-to-Noise Ratio (SNR) levels to guarantee the resilience of our proposed models to variety of wireless channel conditions in practical scenarios. Experimental results demonstrate high accuracy of deep learning methods for wireless device identification that potentially could enhance IoT security.**

***Index Terms*—RF fingerprinting, Deep Learning, ZigBee, Internet of Things.**

I. Introduction

The present wireless ecosystem is getting saturated due to the high volume of devices and applications being deployed. Emerging Technologies such as IoT and 5G networks will add huge number of devices in addition to the existing ones, where multiple wireless technologies will be deployed. Although these wireless devices provide ubiquitous internet access to people and enable device to device communications, the inherent broadcast nature of wireless transmissions introduces security and privacy challenges where malicious users can carry out many types of attacks [1]. IoT security and physical layer authentication remains a key requirement to support the dense deployment of heterogeneous networks expected in future wireless networks to improve user experience [2].

In many practical situations, commercial-of-the-shelf (COTS) IoT devices have been deployed extensively. Furthermore, a lot of these devices may be identical devices from the same manufacture. For example, WiFi devices are widely deployed in many places such as in buildings. If these devices must be located and tracked to ensure security and privacy, traditional software based identifiers such as MAC/IP addresses may not be enough since they can be easily modified. Similarly, IMSI/IMEI in phones and many types of electronic serial numbers (ESN) used to distinguish physical devises can be modified and compromised as well. Furthermore, in order to prevent adversary impersonating legitimate users using identical devices from the same manufacture, unique "signatures" must be obtained for each individual device in order to uniquely identify each device.

In order to address these challenges, powerful and efficient algorithms are needed to prevent device impersonation. These techniques must be stable when environment change or device moves. Among the seven layers of OSI protocol stack, physical layer is the best candidate to enhance the security. Physical layer based features are device specific and more difficult to impersonate if not impossible. One of the promising methods at the physical layer is Radio Frequency (RF) fingerprinting [3].

Similarly as in other electronic components, there are imperfections in wireless transmitter devices. Even with current improvement of semiconductor manufacturing process at 14 nm scale, tiny differences exist in the electronic devices used in front end power amplifier near antenna of wireless devices [4]. Although these imperfections are so small that can satisfy communication standard to pass certificate and also function well, they can be sufficiently large to distinguish devices of the same model and production fab.

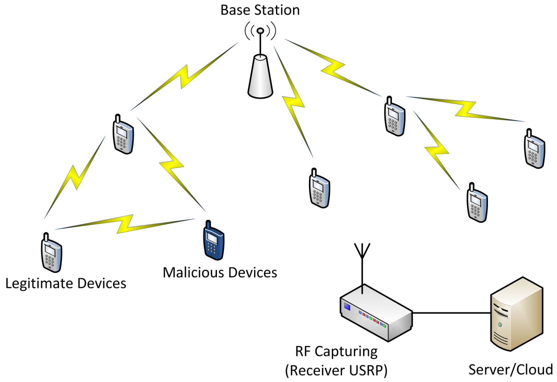

Fig. 1. RF device identification for IOT Security

The intelligent capabilities required by future radios can be achieved by taking advantage of wireless big data [5] and machine learning [6], [7]. For example, in [8], CNN is applied to identify interference in wireless communications. Recently in [9], authors introduced application of deep learning to the

978-1-5386-7185-6/18/$31.00 ©2018 IEEE

physical layer. The authors in [4] surveyed comprehensively RF fingerprinting methods in passive/active modes based on information collected through different layers of protocol stack from PHY to Application layer. Although current methods are applied mostly to WLAN networks including Wi-Fi (802.11x/xx) devices, emerging IoT and 5G technologies utilize non-WiFi devices as well. In this paper, we focus on one of the popular IoT devices, ZigBee (IEEE802.15.4) radios. They are used extensively in wireless sensor networks (WSN) and ZigBee is the bases for several other wireless technologies such as LoRa, 6LowPAN, Z-Wave, and etc.

Figure 1 shows our setup. ZigBee devices can use several network topologies including star, tree, and mesh to forward data from source to base station or to other peer node. For training, RF signals (IQ) from multiple ZigBee devices are collected by USRP device. Then our proposed method including several deep learning algorithms is used to train and learn to distinguish among ZigBee devices. Proposed model is supervised learning based technique where we assume labeled data are available for pre-registered ZigBee devices that we call white-list. After training, model can be used to distinguish among devices in the field. Our proposed model is passive and transparent to RF devices. It does not require installing any extra hardware/software in wireless devices. The classification result can be used for intrusion detection and alarm could be issued for security breach by unregistered adversary.

The structure of the paper is as follows: Section II provides previous researches related to device identification. Section III give a description of the methodology details used to collect the RF signals from ZigBee devices. Section IV provides the experimental results and the related analysis where a comparison of different deep learning algorithms is performed. Section V shows performance results from the deep learning models. Finally Section VI concludes the paper.

II. Related Research

While higher layers of OSI protocol stack may impersonate by adversary, the use of efficient physical layer RF fingerprinting techniques is vital to improve the IoT security and mitigating attacks. Statistical and data driven modern machine learning/deep learning techniques such as PARADIS [10] elaborate the need for improved device identification algorithms allowing us to distinguish among the trusted legitimate devices and detect the adversary trying to impersonate. None of these methods as it is can deal with a growing number of IoT devices with several wireless transmitter technologies.

In an IoT network with dense deployment, wireless device identification using machine learning can help improve authentication, security, and accessibility to service [4]. The authors in [11] applied entropy based statistical features of permutation and dispersion to IoT devices. The overall accuracy of machine learning applied including KNN, SVM, and decision tree is about 82%. Both theoretical modeling and experiment validation are given in [12] for physical layer identification under several constraints based on the spectrum features. PARADIS technique in [10] used machine learning algorithms to classify among 130 Wi-Fi network interface card (NIC) with accuracy near perfect 99%. The radiometric extracted features including frequency error, SYNC correlation, I/Q offset, magnitude/phase error are used to feed as input to KNN/SVM models. The authors in [13] introduced user capacity of physical layer identification as challenge. They applied information theoretic modeling to the features of RF fingerprinting. To characterize the user capacity of physical layer identification system they used mutual information between RF fingerprinting and user identity.

Supervised learning based RF fingerprinting needs RF data from legitimate devices. However, pre-registration of these fingerprints prior to put system in service is impractical in some applications. Entering new devices, guest devices, and upgrading hardware can be challenging. Thus when generating white-list database is impractical, several methods have been developed to use clustering [4]. Although these unsupervised methods are not able to distinguish between legitimate and malicious devices, they can use information from other layers such as the MAC layer to detect several types of attacks including masquerade and Sybil attacks [4]. The main challenge in unsupervised method is that the number of devices is not known. In [14], infinite Gaussian mixture model (IGMM) technique is applied to identify the number of live RF devices. The fingerprint space of multiple physical devices (of the same or different device IDs) is modeled as an infinite Gaussian mixture. They used frequency and phase shift difference as location independent features. Then a non-parametric Bayesian approach applied to unsupervised clustering with an unbounded number of mixtures could be developed.

Although these methods tried to solve RF device identification by extracting hand engineered features and feeding them to statistical or machine learning techniques, none of these works consider multiple varying levels of Signal to Noise Ratio (SNR). As a result, these machine learning models trained by RF data with fixed SNR could not handle channel variations. In this study, we developed USRP based platform to capture RF IQ signals from ZigBee devices across multiple SNR levels. Then we train deep learning models to extract features automatically. It is shown that the obtained models are robust to changing channel conditions.

III. Generation of Dataset

The dataset used in this work, generated from ZigBee(802.15.4) devices, MICAz-MPR2400. MICAz is one of the Motes from Crossbow Technology. It uses CC2420 chip which is IEEE 802.15.4 compliant, ZigBee ready RF transceiver and works from 2400MHz to 2483.5 MHz (MPR2400 model). MTS310 sensor board attached to it is including temperature, light, accelerators, magnetometer, and microphone sensors. MICAz nodes configured in mesh mode to forward sensors measurement data to the base station node.

ZigBee developed for low power and low data rate communication in sensor networks. Recently, its been used in smart grid, smart home, automation and IoT. The goal of ZigBee

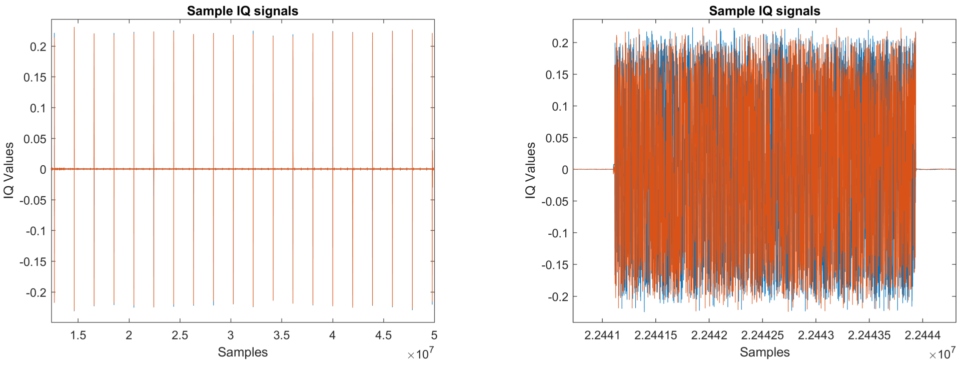

Fig. 2. RF IQ Captured Data from ZigBee (a) 20 Frames (2 Frame/Second) (b) One sample Frame

device is long life with transmitting sensors data couple time of seconds or minutes. Practically, their duty cycle is so small and most of time they are in sleep mode. As shown in Figure 2, our ZigBee device (MICAz) sensors nodes configured by MoteView platform to transmit sensor data two times a second. We filtered out ZigBee data and fed it to deep learning models.

A deep learning model can make accurate classification of ZigBee devices after training with historical RF trace dataset. This is possible because the dataset ground truth is known, so we are able to label the dataset appropriately for deep learning training. The classification of ZigBee devices based on their RF signal can be treated as an M-class decision problem where input to the deep learning model is a complex baseband time series representation of the received ZigBee signals represented by:

$$r(t) = s_1(t) + n_1(t) \tag{1}$$

where $n(t)$ is the noise.

We considered six different ZigBee devices (MICAz) and their Radio Frequency (RF) IQ trace were captured by NI USRP 293x in receiver mode and LabVIEW software. The USRP-293x is a Software Defined Radio Device, a tunable RF transceiver with a high-speed analog-to-digital converter and digital-to-analog converter for streaming baseband I and Q signals to a host PC over 1/10 Gigabit Ethernet saved in TDMS file format. The USRP-293x offer frequency ranges up to 4.4 GHz with up to 20 MHz of instantaneous bandwidth. Communications applications such as white space, broadcast FM, public safety (land-mobile, low-power unlicensed devices on industrial, scientific, and medical (ISM) bands), sensor networks, cell phone, amateur radio, or GPS can be implemented using NI USRP 293x.

In our setup, 2 MHz bandwidth configured for receiver USRP that is needed for ZigBee protocol. Channel 26, 2480 MHz, configured as carrier frequency to have less interference with other device in 2.4GHz ISM band. Datasets for six different devices were collected from the testbed. This 6-classes classification problem includes classes 1 to 6 related to ZigBee devices number 1 to 6, respectively. Each of the six ZigBee devices configured to transmit at different SNR levels of 0, -1, -5, -10, and -15 dBm and related RF IQ data captured by NI USRP 293x and saved accordingly. The dataset attributes are the In-phase (I) and Quadrature (Q) of the collected RF traces. Figure 2 shows an example plot of I and Q traces at 0 dBm. The collected dataset size for each class at every SNR is approximately 10 GB for 5 minutes. Because we captured among five different SNR level, so for every class the total size of collected RF data is 50 GB (And for 6 devices totally 300 GB RF data collected). This big RF data captured from wireless devices are suitable for training deep learning models.

IV. Proposed Deep Learning Models

The complicated nature of the RF device fingerprinting using identical devices from same manufacture that configured to send similar data at the same channel in our setup makes identification of the device a challenge. Traditional hand-engineered expert systems will not be able to do classification to distinguish among devices. Deep learning techniques have advanced the fields of image and speech recognition solving complicated multi-class classification problems.

To evaluate suitable machine learning algorithms for IoT device identification, we considered three popular deep learning models, which are Deep Neural Network (DNN), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). Deep learning are machine learning algorithms that model functions of increasing complexity by adding more layers and more non-linear processing neurons within a layer, it has ability to learn higher level representations of input data [15]. Deep learning has produced huge improvement in applications such as automatic machine translation, object detection etc. Therefore, deep learning is a suitable model for this problem because they can learn from data from different RF signals compare to hand-engineered system selected based on the belief that they best describe RF signals pertinent to a specific RF task. The DNN, CNN and LSTM models training and prediction were implemented in Keras deep learning library [16] running on top of TensorFlow [17] on Nvidia Tesla P100-PCIE-16 GB GPU. These deep learning models perform automatic feature extraction from the RF data as input. The inputs to these models are historical In-phase and Quadrature

(I and Q) data from the six ZigBee devices transmitting at 0, -1, -5, -10, and -15 dBm used for training and prediction, this ensure the deep learning models can make reasonable predictions when there is variation in system's SNR. These models where trained, and tested with samples across the five SNR values.

Training is conducted using a categorical cross entropy loss function and an Adam solver with over several learning rates 0.001, 0.01, 0.1 were chosen to optimize these models. We used different window sizes of 16, 32, 64, 128, and 256 which represent number of I and Q input sequence into the models. All other hyper-parameters of the DNN, CNN and LSTM models considered in these work were not tuned to find the hyper-parameters that give optimum performance. For every DNN, CNN, and LSTM model each training setup in the range of specific learning rate and window size. Furthermore, every training setup implemented with similar parameters is repeated ten times with different initialization values. Then mean and variance values of performance metrics for each model is calculated. As we will explain in next section, all the figures and table results are averaged over ten different run shown with error bar in the related figures.

A. Deep Neural Network

A typical DNN consist of more than one hidden layer that are fully connected, these hidden layers are between the input and output layers [18], [19]. Each hidden layer has several nodes called hidden units. The DNN model her contains 4 dense layers of size 256, 256, 128 and 6-classes neurons at the output. The first 3 layers use ReLU activation functions except for a Softmax activation on the output layer. The I and Q samples after filter is reshaped into $N_{examples} \times Dimensions$ 2 dimensional tensor suitable for Keras dense layer, where dimensions here are I and Q reshaped into window size to form required input shape into the DNN.

B. Convolutional Neural Network

A CNN is typically used for processing grid like data such as image and video [19]. It consists of the convolutional layers and pooling layers. The CNN model is a 4-layer network consisting of two convolutional layers and two dense fully connected layers. These layers use ReLU activation functions except for the Softmax activation on the output layer. The convolutional layers follow with two dimensional zero padding. The zero padding layer adds rows and columns of zeros at the top, bottom, left and right side of an input tensor, used to preserve the spatial size of the input volume so the input and output width and height are the same. The CNN model contains 256 1x3 filters in layer 1, 128 1x3 filters in layer 2, 256 neurons in layer 3 and 6 neurons at layer 4 the output. The I and Q samples shape $N_{examples} \times Dimension$ are reshaped into 4 dimensional tensor suitable for Keras convolutional layer, this takes the form $N_{examples} \times N_{channels} \times Dimension_1 \times Dimension_2$. We have $N_{examples}$ examples from I and Q samples after reshape, each consisting of window size time series samples, $N_{channels}$ = 1 similar to RGB values in imagery, Dim_1 = 2 holding our I and Q channels, and Dim_1 = window size.

C. Long Short Term Memory

RNNs are used for tasks that require sequential inputs, such as speech and language. The RNN gets input sequence one element at a time, this is held in their hidden units. The hidden unit holds information relating to the history of all the past input of the sequence. Long Short Term Memory (LSTM) a variant of RNN is used in this work. Its ability to learn input data with long-term dependencies makes it attractive because traditional RNN performance deteriorate when the distance between the relevant information and the point where it is needed becomes very large [20]. The proposed LSTM model is a 4-layer network consist of 3 stacked LSTM layers and a dense layer as layer 4. This model offers the benefit of learning and remembering over long sequence. The LSTM layer uses $tanh$ activation functions except for Softmax activation on the output layer. The LSTM model contain 32 memory blocks in each of the 3 layers, this is an important parameter in defining an LSTM layer, layer 4 the output has 6 neurons. The I and Q samples shape $N_{examples} \times Dimension$ are reshaped to fit LSTM layer expected input of 3 dimensional tensor suitable for Keras LSTM layer. The 3D tensor is given as $N_{examples} \times N_{timesteps} \times Features$. $N_{examples}$ is the examples from I and Q samples after reshape. $N_{timesteps}$ is the number of time steps set to window size which define the number of input variable used to predict the next time step. $Features$ are separate measures observed at the time of observation which are the I and Q samples.

TABLE I
COMPLEXITY AND OVERALL TEST PERFORMANCE

Window size	Models	Complexity	Performance at 50 epoch	Training time per epoch(S)	Test time(S)
32	DNN CNN LSTM	198278 280774 21318	96.3 % 94.7 % 76 %	6.2 11.17 173.92	0.3 0.44 4.87
64	DNN CNN LSTM	231046 305350 21318	96.2 % 94.2 % 75 %	3.32 5.29 169.04	0.16 0.24 4.43
128	DNN CNN LSTM	296582 354502 21318	93.9 % 94.0 % 73 %	1.87 2.95 162.74	0.08 0.12 4.28

V. RESULTS

The performance of DNN, CNN and LSTM models are evaluated and compared using test dataset to show deep learning models suitability for wireless ZigBee device identification. We trained on the 80% of total I and Q samples from our data set collected from six ZigBee devices (MICAz) over five different SNR levels. We validated and tested data set each one on 10% of the I and Q samples. For each model we repeat the training process over ten different initialization and finally averaged test accuracy with related variance calculate and plotted as error bar in the related figures. Furthermore, for each model we repeated the learning process over several learning rates and window sizes to compare their effect on the performance of test accuracy. We calculated complexity of the models as the number of parameters/weights needed to learn in Table I. For all the mentioned models the computation time spent on training and testing phases are measured and shown in Table I.

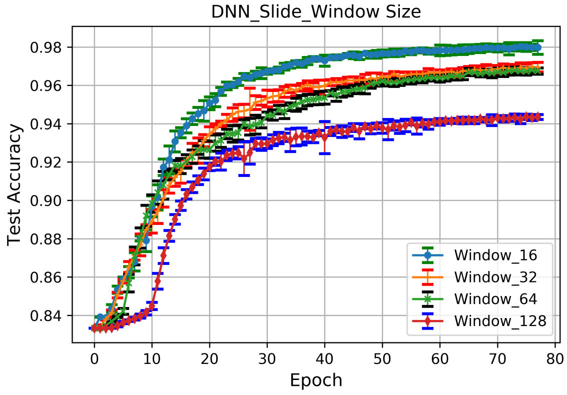

Fig. 3. Test accuracy for DNN with window size: 32, 64 and 128

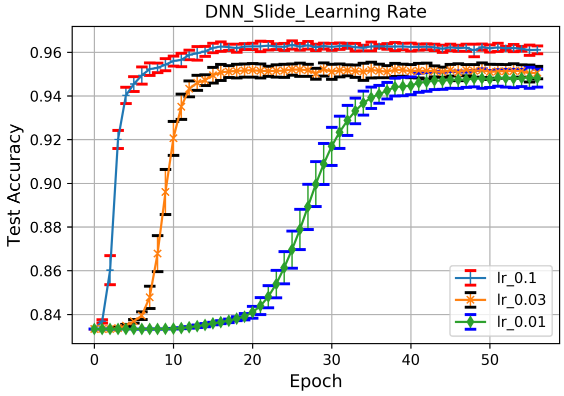

Fig. 4. Test accuracy for DNN with learning rate: 0.01, 0.03, and 0.1

Figure 3 shows the performance of test accuracy for DNN model with window size of 16, 32, 64 and 128. It is clear that increasing window size decrease the speed of convergence and test accuracy. It also increase the variance in learning. It can be explained based on the number of samples captured for each symbol. Test accuracy for DNN model with learning rate of 0.01, 0.03, and 0.1 is shown in Figure 4, it can be seen that decreasing learning rate make training process slower. For CNN model, Figure 5 shows that increasing window size does not change performance a lot, except for large window such as 128 where performance of learning decreases and need very long time to train. While increasing learning rate for CNN based on Figure 6 shows that the performance of learning decrease drastically, slower the learning, and increase the variance. Based on Figures 4 and 6 DNN starts to learn faster and converge and stops to learn early. While CNN model learn gradually and continue to learn and converge slowly. Figure 7 shows the performance of test accuracy for RNN-LSTM model with window size of 16, 32, 64, and 128. Increasing window size increase the variance of learning and degrade the speed of convergence. Although for large window size of 128 in the first 60 epochs there is oscillating behavior, but it starts to improve performance suddenly. Compare to other models, RNN is slower and converge gradually. Also based on Table I its training time per epoch is longer. Although increasing window size slows the training to converge but as shown in Table I doubling window size decrease the training time and test time by half.

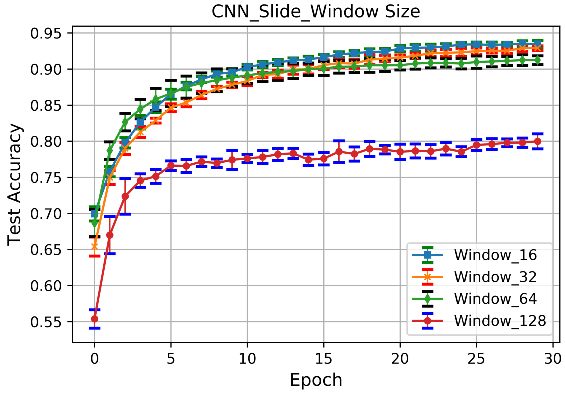

Fig. 5. Test accuracy for CNN with window size: 64, 128, and 256

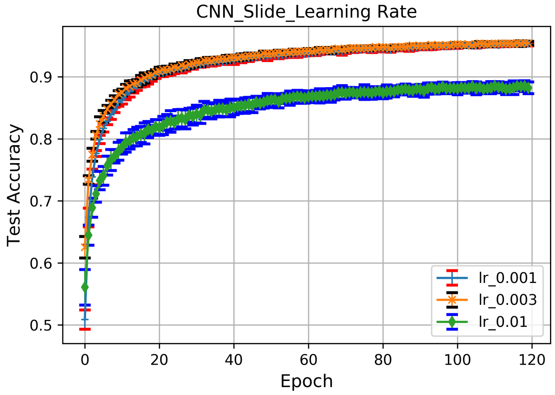

Fig. 6. Test accuracy for CNN with learning rate: 0.001, 0.003, and 0.01

For distinguishing identical RF devices from same manufacture based on their RF signal over SNR levels, deep learning models such as DNN, CNN and LSTM models shows high prediction accuracy making it a suitable option for predicting ZigBee RF device identification.

RF fingerprinting to distinguish between legitimate devices and adversary one can be seen as intrusion detection system (IDS) where impersonate attack can be detected by the model. By detecting malicious device who tried to masquerade itself as a legitimate one, our proposed model can declare alarm for security breach. It enhance the wireless network security and improve the overall service accessibility, authentication and integrity.

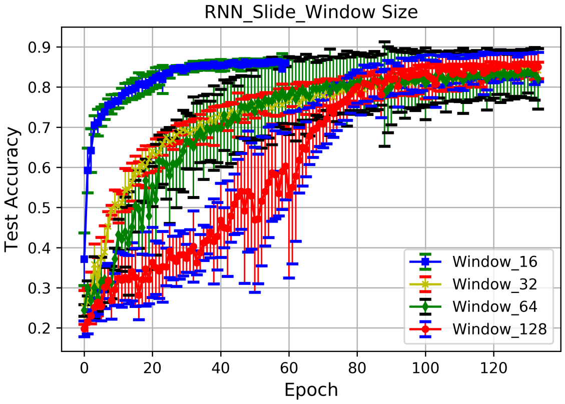

Fig. 7. Test accuracy for RNN with window size: 32, 64, and 128

VI. Conclusions

In this work, we presented a physical layer authentication scheme to identify RF devices using deep learning to improve IoT security. We compare the ability of DNN, CNN and LSTM to model the RF traces collected from ZigBee devices over several SNR levels to guarantee the resilience of the proposed models to various wireless channel conditions. The availability of wireless big data and state-of-the-art deep learning techniques make this a suitable option for RF fingerprinting to distinguish between authorized RF devices and adversaries equipped with similar devices. Experiments using real RF traces collected in a USRP and ZigBee devices based test bed demonstrate that deep learning models are suitable for analyzing the sequence dataset of the RF traces to distinguish RF devices.

Acknowledgment

This research work is supported in part by the U.S. Dept. of Navy under agreement number N00014-17-1-3062 and the U.S. Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E)) under agreement number FA8750-15-2-0119. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Dept. of Navy or the Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E)) or the U.S. Government.

References

[1] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, “A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications,” *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, Oct 2017.

[2] M. Maskery, V. Krishnamurthy, and Q. Zhao, “Decentralized dynamic spectrum access for cognitive radios: cooperative design of a non-cooperative game,” *IEEE Transactions on Communications*, vol. 57, no. 2, pp. 459–469, February 2009.

[3] S. U. Rehman, K. Sowerby, and C. Coghill, “Analysis of receiver front end on the performance of rf fingerprinting,” in *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*, Sept 2012, pp. 2494–2499.

[4] 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, Firstquarter 2016.

[5] L. Qian, J. Zhu, and S. Zhang, “Survey of wireless big data,” *Journal of Communications and Information Networks*, vol. 2, no. 1, pp. 1–18, 2017. [Online]. Available: http://dx.doi.org/10.1007/s41650-017-0001-2

[6] T. J. O’Shea and J. Corgan, “Convolutional radio modulation recognition networks,” *CoRR*, vol. abs/1602.04105, 2016. [Online]. Available: http://arxiv.org/abs/1602.04105

[7] T. J. O’Shea, N. West, M. Vondal, and T. C. Clancy, “Semi-supervised radio signal identification,” *CoRR*, vol. abs/1611.00303, 2016. [Online]. Available: http://arxiv.org/abs/1611.00303

[8] M. Schmidt, D. Block, and U. Meier, “Wireless interference identification with convolutional neural networks,” *CoRR*, vol. abs/1703.00737, 2017. [Online]. Available: http://arxiv.org/abs/1703.00737

[9] T. OShea and J. Hoydis, “An introduction to deep learning for the physical layer,” *IEEE Transactions on Cognitive Communications and Networking*, vol. PP, no. 99, pp. 1–1, 2017.

[10] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, “Wireless device identification with radiometric signatures,” in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, ser. MobiCom ’08. New York, NY, USA: ACM, 2008, pp. 116–127. [Online]. Available: http://doi.acm.org/10.1145/1409944.1409959

[11] G. Baldini, R. Giuliani, G. Steri, and R. Neisse, “Physical layer authentication of internet of things wireless devices through permutation and dispersion entropy,” in *2017 Global Internet of Things Summit (GIoTS)*, June 2017, pp. 1–6.

[12] W. Wang, Z. Sun, S. Piao, B. Zhu, and K. Ren, “Wireless physical-layer identification: Modeling and validation,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 2091–2106, Sept 2016.

[13] W. Wang, Z. Sun, K. Ren, and B. Zhu, “User capacity of wireless physical-layer identification,” *IEEE Access*, vol. 5, pp. 3353–3368, 2017.

[14] N. T. Nguyen, G. Zheng, Z. Han, and R. Zheng, “Device fingerprinting to enhance wireless security using nonparametric bayesian method,” in *2011 Proceedings IEEE INFOCOM*, April 2011, pp. 1404–1412.

[15] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14539

[16] F. Chollet *et al.*, “Keras,” https://github.com/fchollet/keras, 2015.

[17] J. Allaire, D. Eddelbuettel, N. Golding, and Y. Tang, *tensorflow: R Interface to TensorFlow*, 2016. [Online]. Available: https://github.com/rstudio/tensorflow

[18] Y. Bengio, “Learning deep architectures for ai,” *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009. [Online]. Available: http://dx.doi.org/10.1561/2200000006

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[20] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735