Wifi Device Identification through Neural Network based on Channel State Information

Li-Hsien Lin[1], Hsin-Hung Cho[2], Chi-Yuan Chen[3*]
[1,2,3]Dept. Computer Science and Information Engineering, National Ilan University
[1]R0843006@ms.niu.edu.tw, [2]hhcho@niu.edu.tw, [3]chiyuan.chen@ieee.org

Abstract

The tremendous growth of wireless networks has brought a great deal of convenient to our daily lives. But the wireless devices are vulnerable to several attacks such as unauthorized client access. Therefore we want to be able to identify the legitimate and malicious Access Point (AP) for security purpose. The malicious user may attempt to apply attacks such as ARP attack, a method to change the ARP table to get legitimate user's data, Evil twin attack or Rouge AP, the attacker imitates the SSID as the legitimate Wifi AP to misleading the user to connect to the malicious APs. It will be a challenge to recognize the difference between legitimate AP and malicious AP when they have exactly same SSID and MAC address. This paper proposed a method to identity the Wifi APs through physical characteristics. The experiment is performed as a supervised learning LSTM and RNN based on Channel State Information (CSI). Comparing between LSTM and RNN, the results show that LSTM architecture is able to perform a better classification than RNN.

Keywords: Channel State Information, Wifi AP Identification, Machine Learning, Neural Network

[*] Author (Corresponding author.)

1. Introduction

APs applied IEEE 802.11 are venerable to several attacks, one of them is unauthorized client access [3] [18]. The wireless devices identification is an important issue in the wireless security field [2] [4] [5] [10] [14] [15] [21]. The scenario is when a user trying to connect with the legitimate AP and the attacker trying to apply attacks such as Man-In-The-Middle, rouge access point, evil twin etc. We proposed a method using supervised learning neural network, LSTM and RNN, base on CSI to identify the APs among legitimate and malicious devices.

Some researches applying CSI for localization and movement detection [11] [12] [17] [19]. It also shows that, even if the wireless devices are from the same manufactory, there are still some imperfection between identical wireless devices [20]. And by collecting these signals and featuring the imperfection, we are able to apply machine learning to classify the APs. The neural network is able to distinguish the difference between off-the-shelf APs, even when they are from the same manufactory. The experiment will focus on identifying the APs based on CSI, the data are collected and extract from the 802.11n signal.

The structure of the paper is organized as follow. Section 2 presents the background and related works. Section 3 is about data collection / pre-processing and shows the progression of the experiment. Section 4 discusses the results Finally, conclusion is provided in Section 5.

2. Background

2.1 Channel State Information (CSI)

CSI is a metric contains the properties of a communication link in a certain frequency band and also represents the combined effect of the transmitter and receiver. The APs with IEEE 802.11n standards are using Multiple Input Multiple Output (MIMO) system. CSI can be measured and parsed from the PHY layer by Orthogonal Frequency Division Multiplexing (OFDM) technology. In the frequency domain, the wireless channel can be described as [24] [25]:

$$Y = H \times X + N. \tag{1}$$

X is the transmitted signal vectors and Y is for the received, H is the channel matrix presented in the format of CSI, and N is the additive white Gaussian noise vector. H can be express as follow [28]:

$$H(i) = |H(i)|e^{j\sin \angle H(i)}. \tag{2}$$

$$H = \begin{bmatrix} V_{1,1} & \cdots & V_{1,N_{Tx}} \\ \vdots & \ddots & \vdots \\ V_{N_{Rx},1} & \cdots & V_{N_{Rx},N_{Tx}} \end{bmatrix}. \quad (3)$$

The values of i^{th} subcarrier of CSI is represented as $H(i)$, which contains the amplitude and phase of CSI, $\angle H(i)$ and $|H(i)|$ are the phase and the amplitude of the i^{th} subcarrier, respectively. V is the value in H, the number of receiving antenna is noted as N_{Rx} and transmitting antenna is noted as N_{Tx} .

In this paper, we utilize Intel Wi-Fi Wireless Link 5300 on the Linux workstation to record CSI packets. The Linux 802.11n CSI Tool [26] allows us to collect CSI in a format which reports the channel matrices for 30 subcarrier groups, that is about 1 group for every 2 subcarriers at 20 MHz. CSI contain signal strength and phase information for OFDM subcarriers and between each pair of sender and receiver's antennas. The CSI contain n data streams from each data packet can be express as follow according to the MIMO system:

$$CSI = \begin{bmatrix} H_{1,1} & \cdots & H_{1,30} \\ \vdots & \ddots & \vdots \\ H_{n,1} & \cdots & H_{n,30} \end{bmatrix}. \quad (4)$$

2.2 Evil Twin Access Point

An Evil Twin AP is a malicious AP that have identical SSID or MAC address or even both which can cause the user accidentally connect to the malicious AP and conduct Man-In-The-Middle attack. An Evil Twin AP may change its SSID and MAC address to the same as the legitimate AP. If a user is connect to the Evil Twin AP, the user will not able to distinguish if it's from the legitimate AP by the MAC address and this will cause the problem of data leaking. Fig. 1 shows the scenario of Evil Twin. The rouge AP has similar effect to the network, both of them are unauthorized client access.

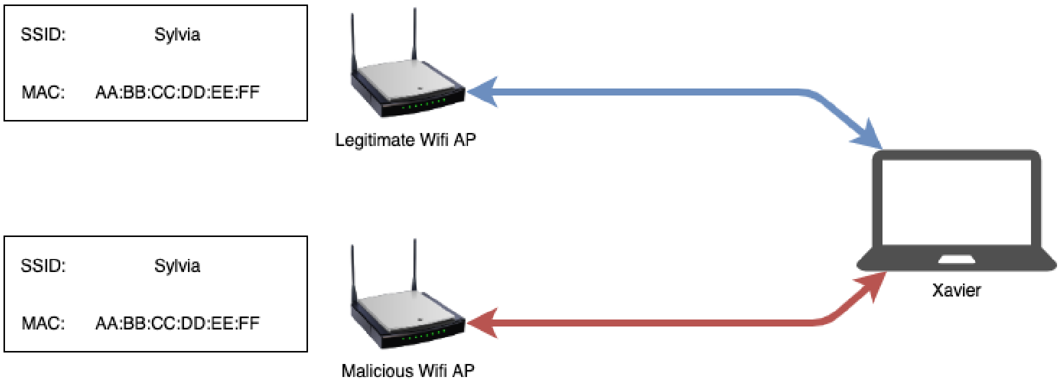


Figure 1: To illustrate the Evil Twin AP, the user Xavier wants to connect to the legitimate AP, Sylvia, but there are a malicious AP has the same SSID.

2.3 Recurrent Neural Network (RNN)

The concept of RNN was brought up by David E. Rumelhart et al. [27] on 1986. Basic RNN is a neural network consisted with multiple layers or stages. Each stage is represent as time *T*. The next stage time *T*+1 will put time T in to consideration as one of the signals. Thus, the output of time *T* will pass into time *T*+1. A RNN layer will work as an encoder as it process the input data sequence and return its own internal state. The problem of RNN is that when the contribution getting smaller and smaller as the timestamp moves the chain of gradients gets longer. This is addressed as vanishing gradient problem [16], [22]. The problem of vanishing gradient was discovered by Sepp Hochreiter Et al., Yoshua Bengio et al. on 1990s. Therefore, the basic RNN is not appropriate for the long-term dependencies problem.

2.4 Long-Short Term Memory (LSTM)

Since a vanilla RNN is not suitable for the long-term dependencies problem. This is when the LSTM comes in. The LSTM [16] is belong to one type of RNN. LSTM is attempt to model time or sequence dependent behavior. LSTM is built upon RNN, by adding a memory component, LSTM is able to help propagate the information learned at a time *T* to the future *T*+1, *T*+2, and so on. With the forget component, it is able to forget the irrelevant data from the previous state. The forget gate is a sigmoid layer, which will look at the previous output as input and outputs a number between 0 and 1 to determine if the data will be forget or not. By updating the state with memory gate and forget gate, LSTM is capable to remember important inputs. This allows them to handle both short-term and long-term dependencies data.

2.5 Classification based on Neural Network

By plotting out the features of the APs based on preprocessed CSI as the heatmap Fig. 2. We can observed the features are difficult to be spotted. A neural network is able to detect the features. The challenges of deciding what the number of hidden neurons should be, activation function to use, optimizer to apply and model to build. If the number of the neurons in the layer are too small, the model will encounter over-fitting. On the other hand, if the number is too large, the model will suffering from under-fitting. As a result, we have to decided what is the number of hidden neurons should be. The activation function will influence how the data propagate to next layer and the output. The optimizer will control how the model being optimized. And the model structure will be build based on the data properties.

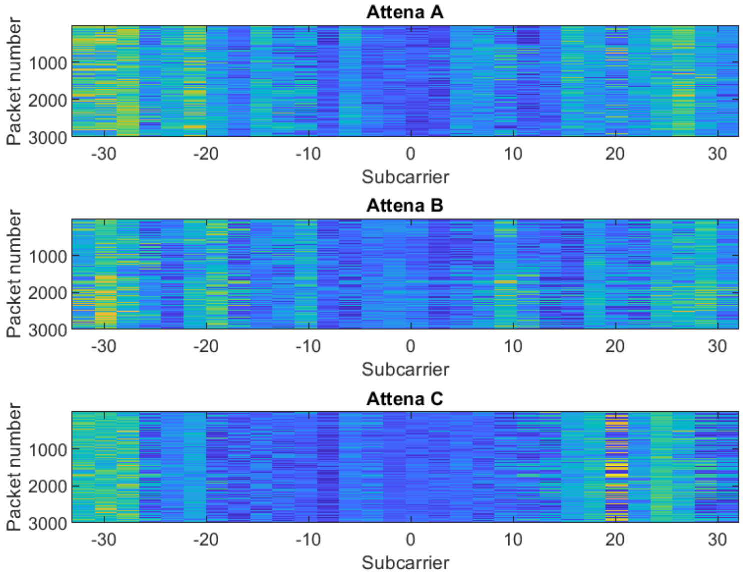


Fig. 2: The heatmap of 3000 packets from one of the AP.

2.6 Related works

Wireless device identification has become an important issue in contemporary wireless security to prevent unauthorized client. The approach for identifying unwelcome APs can be classified into 3 groups [18]: Client-side approach, Server-side approach, Hybrid approach. The Client-side approach is the client has to prevent itself from the unauthorized client. The Server-side approach is that the server has more memory, computational ability than the client or APs so that the client does not need to install any additional software or hardware. The Hybrid approach is efficient because it includes the inabilities of the Client-side approach and the Server-side approach. Both of the APs and client are actively involved, so even if the server failed, the APs or clients can till reduce the risk of connecting to an unauthorized client.

The imperfections between every individual device can be taken as wireless fingerprinting for identification. For Internet of Things (IoT) devices identification [1] [6] [7] [8] includes LoRa, ZigBee, WiFi, Bluetooth. These researches show the characteristic of information from the physical layer such as In-Phase and Quadrature (IQ) samples can implement on identifying devices. Vladimir Brik et al [9] demonstrated a SVM-based approach call PARADIS. It is capable to identify 130 identical 802.11 network interface cards with the accuracy of 99% based on PHY signals (modulation accuracy). The result shows that identical hardware would have imperfections and it can be considered as the fingerprint of the devices. On the other hand, the measured CSI phase can also be taken as a fingerprint [13]. CSI contains the properties of a communication link in Wifi and also represents the combined effect of the transmitter and receiver. Pengfei Liu et al. [14] presented a mechanism to identify 30 different wifi terminals based on CSI. However, CSI is sensitive to the change of environment. Before the identification, this approach took a step to create the fingerprint library which extract phase errors from CSI.

2.7 Contributions

We proposed a neural network model to perform machine learning to extract the feature of CSI and identify APs. CSI is more easy to be obtained from commodity devices. However, it is difficult to extract time-invariant feature from changing CSI. Different from other works to extract features from CSI directly, we adopt LSTM to model time or sequence dependent features. To decide the neural network hyper-parameters, we also took the structure of CSI in consideration. We demonstrate that our approach is capable of identifying AP practical and accurate.

3. Experiment

3.1 Experiment Environment

The Intel 5300 wireless card worked as a receiver. The packets are transmitted by the connected AP which are 3 TP-Link WR841N APs and 1 D-Link DIR868L AP in our experiment. The testbed was built with the Linux 802.11n CSI tool and the Intel 5300 are installed on a Linux Ubuntu 16.6ETS workstation. The workstation and the 4 different WiFi APs were set up in mixed IEEE 802.11 a/g/n on bandwidth 20MHz in the laboratory. The process shows as Fig. 3.

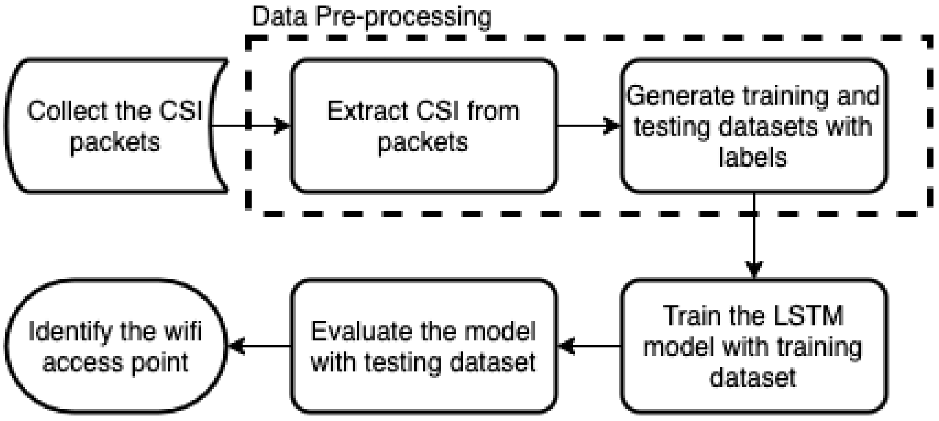

Data Pre-processing
Collect the CSI packets
Extract CSI from packets
Generate training and testing datasets with labels
Train the LSTM model with training dataset
Evaluate the model with testing dataset
Identify the wifi access point

Fig. 3. The flowchart of the classification process.

3.2 Data Collection and Pre-processing

We conduct data collection through Linux 802.11n CSI tool. The tool will allow us to record Wifi signal only 802.11n transmission using High Throughput rate. The entry is a complex number for each channel matrix. It specifies the gain and phase of the signal path between a single transmit-receive antenna pair.

The data was collected in a Faraday cage we built. Each AP was collected 3,000 packets, 12,000 in total. The collected CSI is a 3 dimension data Eq. (4), each packet is composed as $N_s \times N_{Rx} \times N_{Tx}$. The number of subcarrier is noted as N_s. Extracted the CSI packets from the data as shown in Fig. 4 and each packet is labeled corresponding to the wifi AP which is transmitted from. The labels are presented as one-hot-encoding. One-hot-encoding is a method to encode the categories based to unique value as shown in Table II. We only extracted the information from one of the receiving antenna and make sure the shape of each data is the same size of $N_s \times N_{Rx}$, which is a 30 by 3 matrix as Table I. Value of each antenna is noted as $V_{rx,tx}$, *rx*, *tx* is antenna *A*, *B*, *C*. The collected data are split in to 2 datasets, training dataset and testing dataset. 70% of the data, 8,400 packets, was for the training and the rest 30%, 3,600 packets, is for testing, and the validation data will be took from the training data during the training phase. Evaluating the performance by the confusion matrix of the testing dataset. The confusion matrix shows the percentage of the prediction from the trained model.

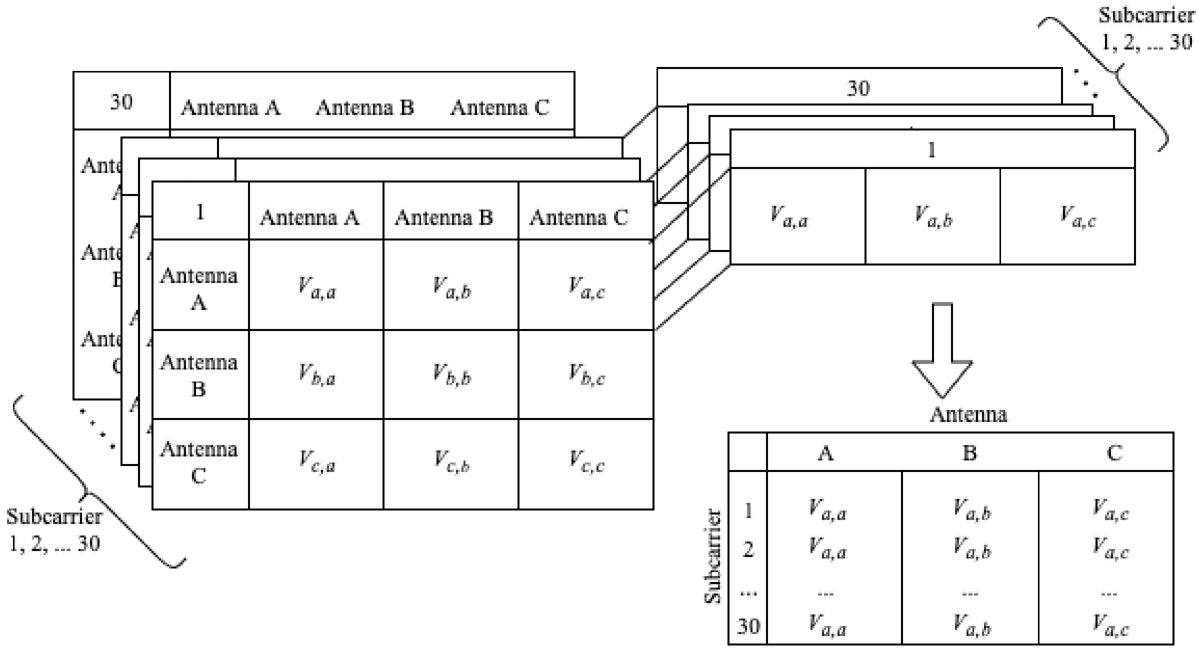


Fig. 4: Extracting the data from CSI packet and pre-process the data for train.

Table 1: The pre-processed packet.

Subcarrier	Antenna		
	A	*B*	*C*
1	$V_{a,a}$	$V_{a,b}$	$V_{a,c}$
2	$V_{a,a}$	$V_{a,b}$	$V_{a,c}$
⋮	⋮	⋮	⋮
30	$V_{a,a}$	$V_{a,b}$	$V_{a,c}$

Table 2: The one-hot-encoding.

Device	*Class*	*One-hot-encoding*
AP 1	0	[1, 0, 0, 0]
AP 2	1	[0, 1, 0, 0]
AP 3	2	[0, 0, 1, 0]
AP 4	3	[0, 0, 0, 1]

3.3 Neural Networks

We assume the identification problem as a classification task. The input is the pre-processed CSI data, and the output will be the identity of the AP. A vanilla RNN is the option for sequential problem. But LSTM is more suitable to meet our requirement. The CSI data are time-serial data and LSTM has the ability to perform a better result on both short-term and long-term dependencies. We still put basic RNN in experiment to verify the assumption.

The hyper-parameters were decided based on the structure and rules of thumb [23]. The hidden neurons Eq. (5), knows as units number N_h was calculated based on the input size N_{input} and output size N_{output}. The input size of the model is composed as [timesteps, feature], which is assigned to be $[N_s,\ N_{Rx}]$. The time-steps will be the subcarrier and the feature will be the 3 antennas. The batch size was decided based on the training data size. 70% of the training data, 5880 can be divided into 40×147. The batch size set to 40 means there will be 147 iteration in one epoch and each iteration has 40 packets. The layers was tested by experiments through trial and error.

As a multi-classification problem, we apply softmax as the activation function for the output layer, and choose categorical cross entropy as loss function, this allow the output that has the highest value to be considered as 1 and the rest will be set to 0. The loss function Eq. (6) will effect on how to model update the weight. Softmax will give a probability between 0 and 1 for each class in the outputs. The activation function softmax Eq. (7) represent as $\sigma(z)$, K is the dimension of the output z, for $j = 1, 2, \dots, K$. For the loss function, y is the output, is the i-th scalar value in the model output. And the OutputSize will be the scaler of the output. Preventing the model from overfitting by adding a dropout layer set to 20% after a LSTM layer.

The dropout layer will stop some neurons from activating. The result shows that the accuracy can achieve over 90% with 3 layers model. Even though RNN is not able to identify the Wifi APs as highly, we are still able to observe the improvement when changing the layers from 1 to 3 for the RNN architecture.

To examine how different neural network influence the performance, we operated the experiments with same setting of hidden neurons number, activation function and loss function. The only different between models will be only the layers. The All of the models trained under the same batch size, 50 epochs, dropout rate 20% and validation on 30% of training data.

$$N_h = N_s \times N_{Rx} \tag{5}$$

$$loss = -\sum_{i=1}^{OutputSize} y_i \log \hat{y}_i \tag{6}$$

$$\sigma(\mathrm{z})_j = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, j = 1, \ldots, K \tag{7}$$

4. Result and Discussion

We explored the classification methods on different type of neural network architectures, 1 layer RNN, 3 layers RNN, 1 layer LSTM and 3 layers LSTM which shown as Fig. 5. According to our outcomes in Fig. 6, the LSTM model with 3 layers has the ability to achieve the accuracy of 92.81% with the validation dataset Table III. The LSTM took longer time to train than RNN, but the performance is better than RNN. Especially for the 3 layer LSTM is 1% over 1 layer LSTM. The trade-off is on the time of training. The structure of 3 layers LSTM is more complex than 1 layer LSTM, thus the model acquire more time to train. The confusion matrix Fig. 7 is composed with evaluation on validation dataset. The values have been normalized. The average of evaluation Fig. 7(a), Fig. 7(b), Fig. 7(c) and Fig. 7(d) are 88.7%, 91.3%, 91.9% and 92.8% correspondingly. The LSTM architectures can exceed 90% on evaluation.

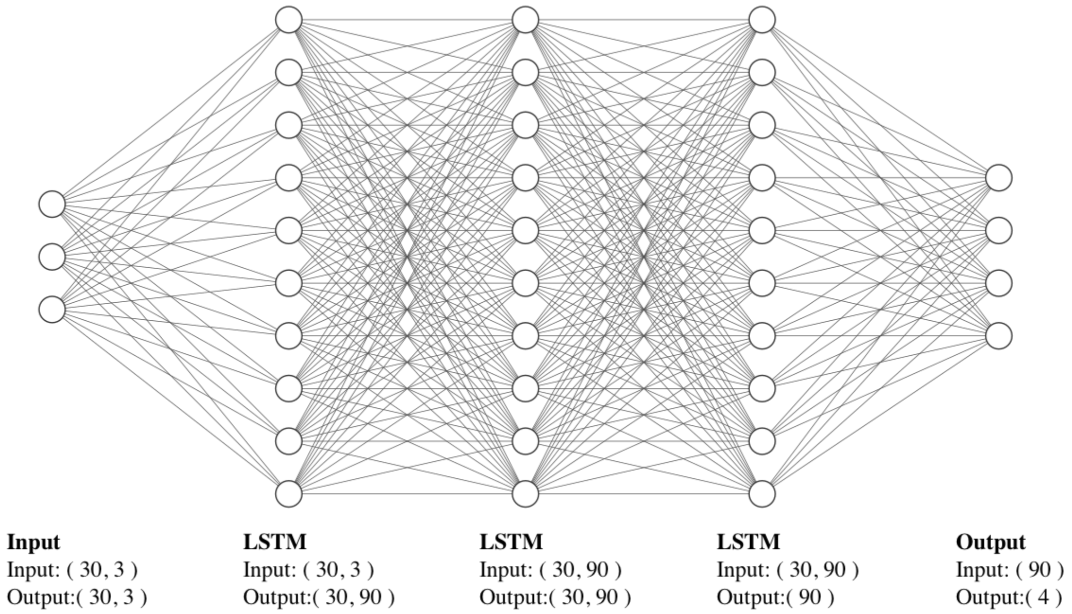


Fig. 5: Illustrating the structure of 3 LSTM layers and 1 Dense layer for output.

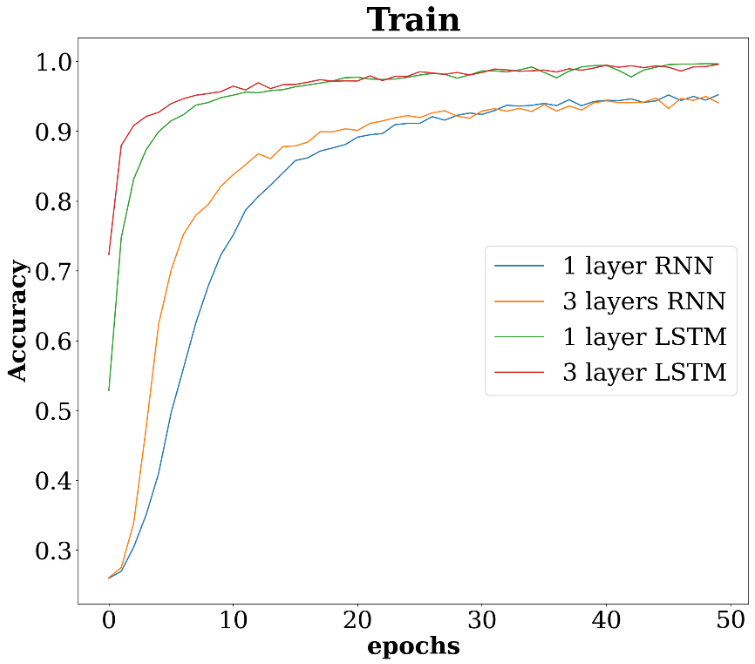
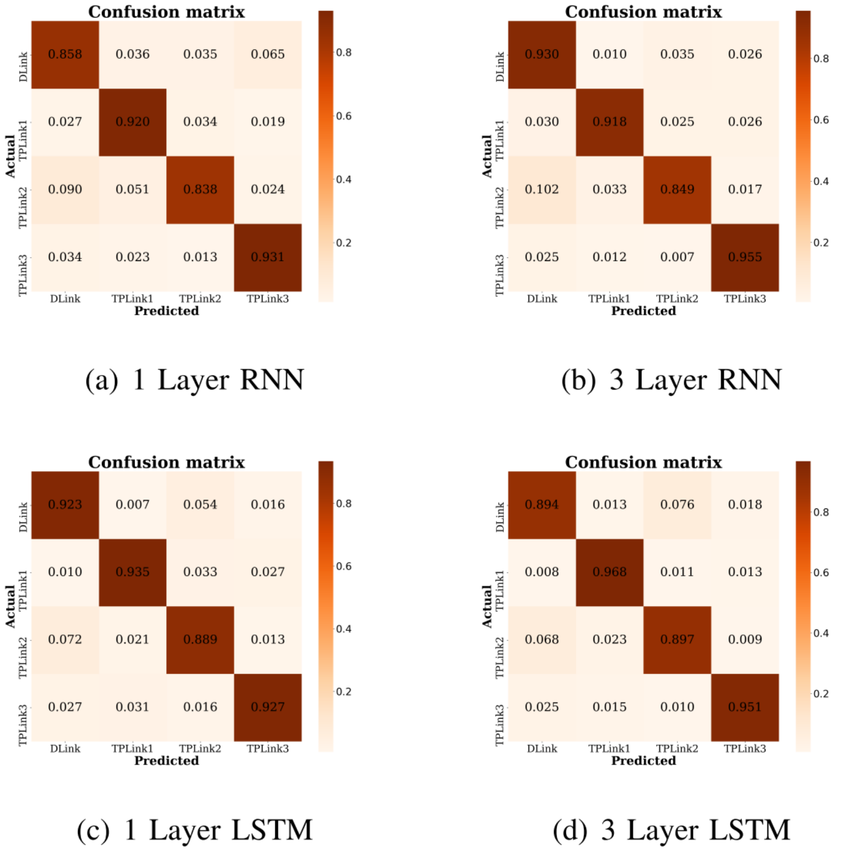


Fig. 6. Accuracy of identifying the 4 APs. In the graph show the performance of different Neural Network model



(a) 1 Layer RNN

(b) 3 Layer RNN

(c) 1 Layer LSTM

(d) 3 Layer LSTM

Fig. 7: The confusion matrix of the validation data.

5. Conclusion

We explored the classification methods on the different type of neural network architectures. Extracting the CSI features with neural network for identification can perform promising capability among 4 off-the-shelf Wifi APs with LSTM and RNN. Evaluated and demonstrated the CSI-based fingerprinting can be implemented in identify an unauthorized client. The hyper-parameters for the neural networks was decided by the structure and the characteristic of the CSI data. We are looking forward to extending the study on different protocols and larger scale.

Acknowledgment

This research was partly funded by Ministry of Science and Technology (MOST) in Taiwan under grants 107-2221-E-197-005-MY3 and 108-2221-E-197-012-MY3.

References

[1] R. Das, A. Gadre, S. Zhang, S. Kumar and J. M. F. Moura, ”A Deep Learning Approach to IoT Authentication,” 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, 2018, pp. 1-6, doi: 10.1109/ICC.2018.8422832.

[2] S. Riyaz, K. Sankhe, S. Ioannidis and K. Chowdhury, ”Deep Learning Convolutional Neural Networks for Radio Identification,” in IEEE Communications Magazine, vol. 56, no. 9, pp. 146-152, Sept. 2018, doi: 10.1109/MCOM.2018.1800153.

[3] V. L. L. Thing, ”IEEE 802.11 Network Anomaly Detection and Attack Classification: A Deep Learning Approach,” 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, 2017, pp. 1-6, doi: 10.1109/WCNC.2017.7925567.

[4] K. Merchant, S. Revay, G. Stantchev and B. Nousain, ”Deep Learning for RF Device Fingerprinting in Cognitive Communication Networks,” in IEEE Journal of Selected Topics in Signal Processing, vol. 12, no. 1, pp. 160-167, Feb. 2018, doi: 10.1109/JSTSP.2018.2796446.

[5] H. Weng, X. Dong, X. Hu, D. G. Beetner, T. Hubing and D. Wunsch, ”Neural network detection and identification of electronic devices based on their unintended emissions,” 2005 International Symposium on Electromagnetic Compatibility, 2005. EMC 2005., Chicago, IL, 2005, pp. 245-249 Vol. 1, doi: 10.1109/ISEMC.2005.1513508.

[6] O’Shea, Timothy J., Corgan, Johnathan and Clancy, T. Charles, ”Convolutional Radio Modulation Recognition Networks,” Engineering Applications of Neural Networks, Cham, 2016, pp. 213-226, isbn: 978-3-319-44188-7.

[7] N. Bitar, S. Muhammad and H. H. Refai, ”Wireless technology identification using deep Convolutional Neural Networks,” 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, 2017, pp. 1-6, doi:10.1109/PIMRC.2017.8292183.

[8] H. Jafari, O. Omotere, D. Adesina, H. Wu and L. Qian, ”IoT Devices Fingerprinting Using Deep Learning,” MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, 2018, pp. 1-9, doi: 10.1109/MILCOM.2018.8599826.

[9] V. Brik, S. Banerjee, M. Gruteser and S. Oh, ”Wireless Device Identification with Radiometric Signatures,” Proceedings of the 14th ACM International Conference on Mobile Computing and Networking, New York, NY, USA, 2008, pp. 116-127, doi: 10.1145/1409944.1409959.

[10] L. Cheng and J. Wang, ”Walls Have No Ears: A Non-Intrusive WiFi-Based User

Identification System for Mobile Devices," in IEEE/ACM Transactions on Networking, vol. 27, no. 1, pp. 245-257, Feb. 2019, doi:10.1109/TNET.2018.2886411.

[11] G. Wu and P. Tseng, "A Deep Neural Network-Based Indoor Positioning Method using Channel State Information," 2018 International Conference on Computing, Networking and Communications (ICNC), Maui, HI, 2018, pp. 290-294, doi: 10.1109/ICCNC.2018.8390298.

[12] C. Hsieh, J. Chen and B. Nien, "Deep Learning-Based Indoor Localization Using Received Signal Strength and Channel State Information," in IEEE Access, vol. 7, pp. 33256-33267, 2019, doi: 10.1109/ACCESS.2019.2903487.

[13] Y. Zhuo, H. Zhu, H. Xue and S. Chang, "Perceiving accurate CSI phases with commodity WiFi devices," IEEE INFOCOM 2017 – IEEE Conference on Computer Communications, Atlanta, GA, 2017, pp. 1-9, doi: 10.1109/INFOCOM.2017.8056964.

[14] P. Liu, P. Yang, W. Song, Y. Yan and X. Li, "Real-time Identification of Rogue WiFi Connections Using Environment-Independent Physical Features," IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, Paris, France, 2019, pp. 190-198, doi: 10.1109/INFOCOM.2019.8737455.

[15] Y. Chapre, A. Ignjatovic, A. Seneviratne and S. Jha, "CSI-MIMO: Indoor Wi-Fi fingerprinting system," 39th Annual IEEE Conference on Local Computer Networks, Edmonton, AB, 2014, pp. 202-209, doi:10.1109/LCN.2014.6925773.

[16] S. Hochreiter and J. Schmidhuber, "Long Short Term Memory," Neural Comput.,Cambridge, MA, USA,1997 doi:10.1162/neco.1997.9.8.1735

[17] Y. Wang, C. Xiu, X. Zhang, and D. Yang, "WiFi Indoor Localization with CSI Fingerprinting-Based Random Forest," Sensors, vol. 18, no. 9, p. 2869, Aug. 2018.

[18] S. Anmulwar, S. Srivastava, S. P. Mahajan, A. K. Gupta and V. Kumar, " Rogue access point detection methods: A review," International Conference on Information Communication and Embedded Systems (ICICES2014), Chennai, 2014, pp. 1-6, doi: 10.1109/ICICES.2014.7034106.

[19] T. Z. Chowdhury, "Using Wi-Fi channel state information (CSI) for human activity recognition and fall detection," T, University of British Columbia, 2018.

[20] B. Danev, D. Zanetti and S. Capkun, "On Physical Layer Identification of Wireless Devices, "ACM Comput. Surv. Association for Computing Machinery, New York, NY, USA,2012, doi:10.1145/2379776.2379782

[21] S. Shetty, M. Song and L. Ma, "Rogue Access Point Detection by Analyzing Network Traffic Characteristics," MILCOM 2007 – IEEE Military Communications Conference, Orlando, FL, USA, 2007, pp. 1-7, doi: 10.1109/MILCOM.2007.4455018.

[22] Y. Bengio, P. Simard and P. Frasconi, "Learning long-term dependencies with gradient

descent is difficult," in IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 157-166, March 1994, doi: 10.1109/72.279181.

[23] J. Heaton, "Introduction to Neural Networks for Java, 2nd Edition," Heaton Research, Inc., 2008, pp.127-129, isbn:1604390085.

[24] S. Yousefi, H. Narui, S. Dayal, S. Ermon and S. Valaee, "A Survey on Behavior Recognition Using WiFi Channel State Information," in IEEE Communications Magazine, vol. 55, no. 10, pp. 98-104, Oct. 2017, doi:10.1109/MCOM.2017.1700082.

[25] Z. Wang, B. Guo, Z. Yu and X. Zhou, "Wi-Fi CSI-Based Behavior Recognition: From Signals and Actions to Activities," in IEEE Communications Magazine, vol. 56, no. 5, pp. 109-115, May 2018, doi:10.1109/MCOM.2018.1700144.

[26] D. Halperin, W. Hu and A. Sheth and D. Wetherall, "Tool Release: Gathering 802.11n Traces with Channel State Information," SIGCOMM Comput. Commun. Rev., New York, NY, USA, Jan 2011, vol. 41, no. 1, pp. 53, doi:10.1145/1925861.1925870.

[27] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors," Nature 323, 533–536 (1986). https://doi.org/10.1038/323533a0.

[28] E. Perahia and R. Stacey, "Next generation wireless LANs: Throughput, robustness, and reliability in 802.11n," pp. 323-326, 2008, doi:10.1017/CBO9780511541032.