Ethernet Device Authentication via Physical Layer Fingerprinting

William Suski*, Christopher Card†, Brian Few‡
Applied Engineering Concepts, Incorporated, Eldersburg, MD, USA
Email: *wsuski@ae-concepts.com, †ccard@ae-concepts.com, ‡bfew@ae-concepts.com

***Abstract*—Device authentication is an important element of any strong defense-in-depth strategy for securing cyber-physical, industrial control, and other critical infrastructure systems. However, the current stable of solutions available to perform device authentication are not suitable for deployment on many operational technology networks due to the power and processing limitations of legacy and state-of-the-art Internet-of-Things devices. In this paper, we propose a machine learning method for passively and non-invasively authenticating or fingerprinting Ethernet devices, at the physical layer (PHY), using their transmitted signals. This technique exploits the unique, intrinsic physical features of a device that are created by the operational characteristics of its discrete physical components. These characteristics are imprinted on a device's communication signal, can be externally monitored to authenticate/authorize registered devices, and can quickly detect the introduction of unknown and/or unauthorized devices. We assess the performance of our proposed technique based on the discrimination power in a device classification scenario.**

***Index Terms*—authentication, intrusion detection, network access control**

I. Introduction

Device authentication is recognized as a critical element of any strong defense-in-depth strategy for cyber-physical, industrial control, and other critical infrastructure systems by multiple standards organizations. This includes the United States' National Institute for Standards and Technology (NIST) in its "Guidelines for Smart Grid Cybersecurity" (NISTIR 7628) [1], the International Electrotechnical Commission (IEC) in its 62443 standard [2], and NIST 800-53 "Security and Provacy Controls for Information Systems and Organizations" [3], which is part of the well-known Risk Management Framework (RMF). However, these documents do not propose any specific and/or universal means to authenticate devices which has led to the development of multiple methods.

Most device authentication solutions rely on assigned credentials such as digital certificates or username/password combinations, e.g. the Institute of Electrical and Electronics Engineers (IEEE) 802.1X protocol. This method is effective for today's enterprise information technology (IT) networks that contain a relatively homogeneous set of computationally capable devices. However, it is inadequate for diverse operational technology (OT) networks due to the proliferation of legacy and low power industrial control system (ICS) and/or Internet of Things (IoT)/Industrial Internet of Things (IIoT) devices that have limited computational resources available to execute sophisticated encryption algorithms and protocols [4]. This is in addition to vulnerabilities that arise from default, insecure, or poorly managed passwords [5].

More advanced device authentication techniques build profiles of devices, often referred to as fingerprints, based on high-level characteristics such as manufacturer, model, medium access control (MAC) address, operating system, web browser, etc. These solutions build device profiles by collecting data using active or passive means. In OT networks, active probing is often not permitted because it can cause disruptions to device operation and passive data collection methods are less effective [4]. Furthermore, this type of device profile is relatively easily spoofed because the attributes are high-level [6]. Other techniques are more invasive and require some type of software to be run on the target host. For example, many state-of-the-art network access control (NAC) solutions utilize agents to collect in-depth data about a device such as operating system (OS) patch level, installed software, etc.

In this paper, we propose a machine learning (ML) method for passively and non-invasively authenticating Ethernet devices, at the physical layer (PHY), using their transmitted signals. It exploits the unique, intrinsic physical features of a device that are created by the operational characteristics of its discrete physical components. These characteristics are imprinted on a device's communication signal, can be externally monitored to authenticate/authorize registered devices, and can quickly detect the introduction of unknown and/or unauthorized devices.

Despite being initially defined in 1990, the 10Base-T and its main successor, 100Base-TX, protocols are still used heavily in ICS and OT networks. This interest in older networking technology is driven by the need for industrial-grade network infrastructure equipment that can withstand extended temperature ranges as well as the legacy devices deployed within these networks. These protocols are part of a collection of PHY standards for wired communications referred to as IEEE 802.3 or Ethernet [7]. The collection includes a number of standards that define the physical and data link layers of wired Ethernet. One of the earliest of these is 802.3i which defines the 10Base-T protocol for communication over twisted pair cabling and provides a throughput of 10 Mbits/s. As a means

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Cybersecurity, Energy Security, and Emergency Response (CESER), under Award Number DE-SC0019922.

of determining technical feasibility, the work presented in this paper is focused on PHY device authentication of devices using the 10Base-T protocol. Additional protocols, including 100/1000Base-TX, have been studied but are omitted due to length restrictions.

At its core, PHY device authentication is the application of pattern recognition techniques to the classification of devices. Until recently, most published research in this field has been focused on traditional machine learning approaches, which rely on highly engineered features and well-known statistical pattern recognition algorithms, to perform device fingerprinting. Much of this work has been focused on radio frequency (RF) devices. One such application, referred to as specific emitter identification (SEI), has existed for many years and is focused on uniquely identifying RADAR systems by their RF transmissions [8]. Similar techniques, when applied to wireless communications devices, have been referred to as RF fingerprinting [9], [10], [11]. Other work has considered fingerprinting wired communications devices. For example, Lopez, et al. utilized a set of engineered features with multiple discriminant analysis (MDA) and random forest classifiers [12] to authenticate wired devices using the Highway Addressable Remote Transducer (HART) protocol.

The recent explosion in applications of deep learning (DL) techniques to classification tasks, especially in image processing [13], has led to similar applications in wireless device fingerprinting. The latest techniques avoid reliance on engineered features by training deep neural networks, using large labeled data sets, to classify wireless devices [14], [15], [16]. However, the authors believe that this work is the first application of DL techniques to Ethernet device fingerprinting.

Our main contributions are:

- We provide a potential architecture for the deployment of future device authentication systems based on fingerprinting techniques.
- We demonstrate that unique, unintended perturbations transmitted as part of a device's Ethernet signal can be used to identify a target device within a larger set.
- We quantify the discrimination power of the proposed method as a means of comparison to existing and future techniques.

The remainder of this paper is organized as follows. Section II provides additional context via a description of the PHY device authentication use case and threat model. Section III describes the type of data collected, collection method, and produced data set. Section IV discusses some previously published approaches as well as our proposed DL-based approach and outlines the classification process. Results are described in Section VI. Finally, conclusions are provided in Section VII.

II. Use Case and Threat Model

A high-level architecture for the deployment of PHY device authentication is depicted in Figure 1. It shows a wired Client device attempting to communicate with a Target Host over a protected Ethernet network. Packet Data is encoded into electrical signals, transmitted from Client to the connected Switch, and concurrently monitored by the PHY Sensor. Periodically, these signals are sampled by the PHY Sensor, preprocessed, packaged, and transported to the Authenticator. The Authenticator executes the proposed authentication algorithm and determines if the device that generated the sampled packet is authorized or not. The results are then passed along to any number of third party utilities, e.g. port security, a software-defined network (SDN) control plane, NAC system, or an intrusion detection system (IDS). If the device is authorized, everything proceeds as normal. If it is not, the appropriate action can be taken to include event logging, administrator notification, and/or port blocking.

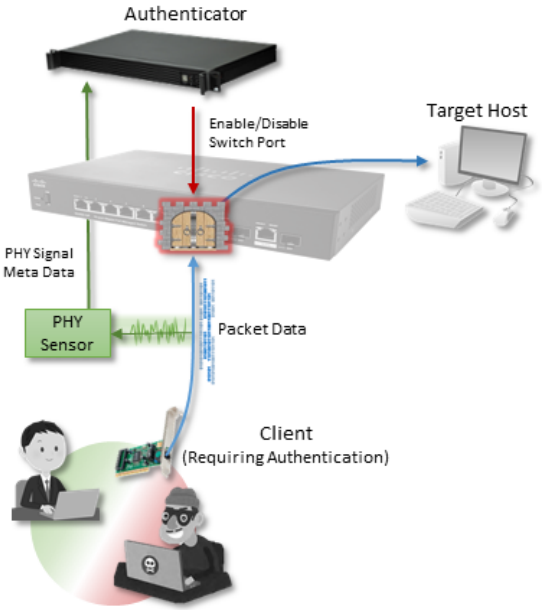

Fig. 1: High-level architecture of deployed PHY authentication system.

The proposed PHY device fingerprinting/authentication method is intended to mitigate the threat posed by the introduction of unauthorized devices into a wired Ethernet network. This requires that an attacker gain access to a network's physical infrastructure and circumvent any port security measures through MAC address spoofing, etc. To mitigate this threat, this work relies on a number of assumptions:

Appropriate Sensor Placement. As shown in Figure 1, the PHY sensor must be placed between the device-of-interest and the first piece of network infrastructure equipment. This is a necessary condition to be able to monitor the unique signals emanated by the connected device. Any other location within a network would be observing packet data that has been retransmitted by other equipment, e.g. a switch or router.

Secure Communication. It is assumed that a secure connection can be established between the PHY sensor and the Authenticator device, i.e. an authentication server, to securely transport fingerprint data between the two. This connection can be made via any appropriate secure communication pro-

tocol. Other hardware-based fingerprinting techniques assume similar connections [17].

Port Control. The proposed PHY device authentication system operates passively and does not interfere with network operations in any way. Therefore, it is assumed that a third party solution is deployed that can decide the appropriate action to take based on authentication results. This solution may be switch port security, a NAC system, or an SDN control plane.

III. Data Set Creation

Like all machine learning techniques, PHY device authentication requires a large data set made up of samples collected from as many devices as possible. Often, publicly available data sets can be used to quickly test theories and compare performance, as is the case with image processing problems. However, the digitally sampled electrical signals required to perform device authentication at the PHY are not readily available. Therefore, a data set has been generated specifically for this work. The hardware, software, and data collection setup used to build this data set are described further in the following sections.

A. Hardware & Setup

Our approach relies on digitally sampling voltage(s) from the transmit (TX) pin(s) of the device-of-interest and storing them for off-line analysis. Previous work has utilized a network-controllable digital oscilloscope to collect the required data [12]. In this work, the Siglent SDS2000X-E series Super Phosphor Oscilloscope (Model SDS2352X-E) [18] was used. This two-channel oscilloscope has a 350 MHz bandwidth with real-time sampling capability up to 2 GS/s and enough storage for 28M samples, e.g. 14 ms at full sampling rate. Figure 2 shows the hardware configuration used to collect device data.

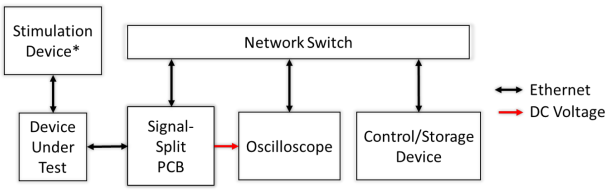

Fig. 2: Hardware setup for data collection.

As mentioned previously, these devices have been forced into 10Base-T mode for this initial study. Figure 2 shows the hardware configuration used to collect device data. The Control/Storage Device stimulates the Device-Under-Test (DUT) (via the Network Switch) to generate a response packet (using the ping utility). This packet is transmitted over the Ethernet cable as a direct current (DC) voltage. The Signal-Split printed circuit board (PCB) provides in-line access to all eight of the signal lines coming from the DUT. Data is sampled from one of the TX pins (red line). This data is digitally sampled, truncated to contain samples from a single packet preamble, and forwarded by the Oscilloscope to the Control-Storage Device. This process is repeated 10,000 times for each device. If the DUT is a network switch port, the Control/Storage device "pings" the Stimulation Device to elicit a packet that is re-transmitted by the switch. This re-transmitted packet is then sampled and stored by the oscilloscope.

An automated collection tool suite was developed to efficiently collect data with minimal required operator oversight. It configures the oscilloscope, provides device stimulation, retrieves the sampled data from the oscilloscope, and packages it for storage. Each packet preamble is 5.7 μs long and results in approximately 11,400 samples. These samples are packaged into Hierarchical Data Format 5 (HDF5) and stored for off-line algorithm development and evaluation. HDF5 is commonly used in Python software and is quickly/easily accessed by the downstream machine learning software.

B. Data

The data collected for the 10Base-T protocol consists of digitally sampled voltages from one of the DUT TX pins during transmission of the packet preamble. The Ethernet preamble, as defined in the IEEE 802.3i standard, consists of 8 bytes of Manchester encoded data used to synchronize the receiver's clock. The first 7 bytes have a value of 10101010 and the last byte has a value of 10101011. Figure 3 is a time domain plot of a single packet preamble.

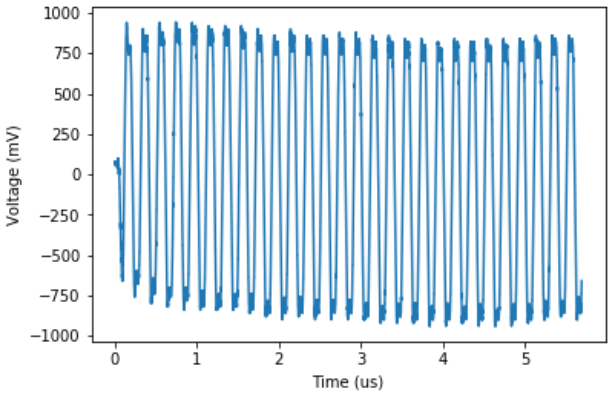

Fig. 3: Example of sampled 10Base-T packet preamble data.

The 10Base-T data set includes samples of 10,000 packet preambles collected from 20 different wired Ethernet devices: three switches (manufactured by D-Link, LinkSys, and TPLink), two Raspberry Pi single-board computers, and a USB-to-Ethernet adapter.

IV. PHY Device Fingerprinting

This section begins by describing some previous approaches to device fingerprinting which utilize engineered features and traditional pattern recognition techniques. Later, we describe the proposed approach to PHY device fingerprinting which relies on training deep neural networks to identify the salient signal features, generate fingerprints, and classify devices.

A. Previous Approaches

Traditional machine learning approaches to device fingerprinting and authentication generate their fingerprints from raw signal data by calculating statistics over a set of engineered discriminating features as shown in Figure 4. The most commonly used features are instantaneous amplitude ($a[k]$), phase ($\phi[k]$), and frequency ($f[k]$). Their use comes from the field of automated modulation recognition, which attempts to classify received communication signals by their data modulation type [19]. Previous research has also used these features to perform device classification of 802.11a [9], ZigBee [20], and WiMax [21] wireless devices.

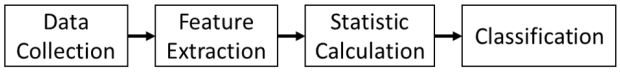

Fig. 4: Existing device fingerprinting process.

B. Our Approach: Neural Network-based Fingerprinting

In 1989, artificial neural networks were first shown to be universal function approximators [22]. Since then, they have become an important tool for a wide range of tasks including classification. In this work, we propose the use of deep artificial neural networks to generate fingerprints from Ethernet communication signals that are capable of uniquely identifying the transmitting devices. A key difference in our approach is the use of deep artificial neural networks to fingerprint and classify devices using raw signal input data as opposed to previous approaches which utilize statistics of engineered features.

Our approach applies deep neural networks in two ways. The first application trains a deep neural network to directly classify the input signal data as one of a subset of known devices without explicit generation of a device fingerprint. These networks use a rectified linear unit (ReLU) activation function on the output layer and typically select the maximum output value as the predicted class. This application is the typical use of deep neural networks in classification problems and is used to demonstrate the discrimination power of these networks as applied to the PHY device authentication problem. This section describes two types of neural network architectures, the convolutional neural network (CNN) and the residual neural network (RESNET), that can be used to authenticate devices.

The second application is focused on using deep neural networks to generate stand-alone device fingerprints. These fingerprints can be stored and classified/authenticated later using any method, including non-DL methods. This application is relevant in circumstances where classes must be added or removed frequently and thus cannot wait for full deep neural network training, due to the amount of time required. This is similar to content-based image retrieval techniques which generate fingerprints and then compare them in various ways to determine similarity [23]. Our method modifies the activation function on the output layer of a neural network previously trained for device classification. We change the activation function from ReLU to linear and use the output vector as a device fingerprint. In the case of a neural network trained to classify 20 devices, the output fingerprint is a vector of 20 floating point numbers. This second application is used to demonstrate the homogeneous discrimination power and stability of the proposed approach to device fingerprinting which is described further in Section V-B.

1) Convolutional Neural Networks: This architecture was chosen for initial experiments based on the results reported in [24] for performing modulation recognition using raw signal inputs. The CNN experiments were focused on exploring the hyperparameter space of networks similar to those initially designed by the Visual Geometry Group (VGG) at Oxford University for the ImageNet challenge in 2014 [25]. The hyperparameters considered in this study are listed in Table I.

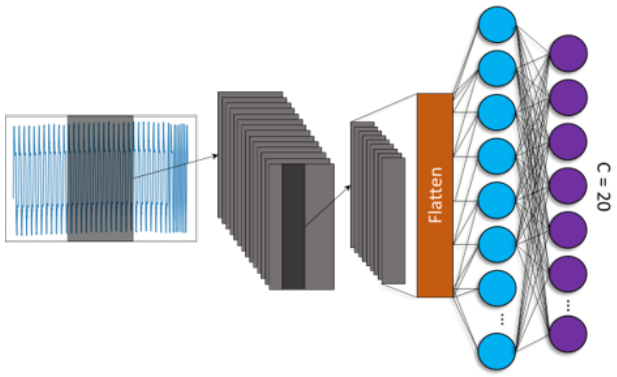

Fig. 5: Generic CNN architecture.

Figure 5 is a graphical representation of a notional CNN architecture for Ethernet device fingerprinting. On the far left, it shows a random selection of M contiguous samples from the full Ethernet packet preamble. This signal subset is then convolved with a number of kernels in each succeeding convolutional layer. Finally, the outputs of the last convolutional layer are flattened and passed through a series of dense layers that act as a classifier. The output layer consists of C outputs, one for each device or class in the model. This results in a single device being chosen as the prediction. During training, this prediction is compared to the labeled input data and then backpropagation is used to adjust the weights of the preceding layers via stochastic gradient descent. As mentioned previously, a partition of the original labeled data is set aside for validation and testing of the model on previously unseen data. A dropout rate of 0.1 is used throughout the network for the purposes of regularization.

PARAMETER	VALUES
Input Length	128, 256, 512, 1024, 2048, 4096, 8192
Hidden Layers	3, 4
Layer Width	8, 16, 32, 64, 128, 256, 512, 1024

TABLE I: Hyperparameters considered during CNN model experimentation.

Using a Python-based library called Hyperas [26], a hyperparameter optimization (HPO) test was run across the full

range of parameters listed in Table I. Hyperas combines the machine learning library Keras with an optimization library called Hyperopt [27] and allows for quick prototyping of deep learning HPO experiments. It begins with a random selection of hyperparameters, trains a model, and stores the resulting classification accuracy. Then, using a Tree-structure Parzen Estimator, it estimates the performance of the objective function, in this case model training, on the given set of parameters [28]. It then selects the next set of parameters to be tested with the goal of pushing towards the optimal set of parameters in an intelligent fashion.

2) Residual Neural Networks: In recent years, it has been shown [29] that bigger is not always better in the world of CNNs. More specifically, deeper CNNs have been shown to be less accurate than shallower implementations due to the vanishing gradient problem [30]. This realization led to the development of the residual block and RESNET. Residual blocks introduce the concept of skip connections which allow deeper neural networks to be trained, without the issues created by vanishing gradients. Experiments were performed with RESNET to determine the classification accuracy achievable by this network for PHY device authentication. The RESNET-50 implementation is used throughout this work and the discrimination power of various input data lengths is explored.

V. Fingerprint Assessment Methods

As described in [31], there is a need to measure the quality associated with a fingerprinting method to aid in comparing different techniques. Sanchez-Rola, et al. propose six fingerprint characteristics that should be evaluated: discrimination power, stability, homogeneous discrimination, efficiency, resilience to evasion, and resilience to change. We address three of these characteristics below and leave efficiency, resilience to evasion, and resilience to change for future work.

A. Discrimination

The discrimination power of a fingerprinting method is the most commonly discussed measure of quality. A high-quality fingerprinting method is capable of uniquely identifying a specific target from a set of possible candidate targets with as few errors as possible. This is often referred to as classification accuracy and these phrases are used interchangeably here. In this work, classification accuracy is reported through the use of confusion matrices, which demonstrate classification accuracy for each device in comparison to all others, as well as through overall accuracy percentage.

B. Homogeneous Discrimination

The ability to discriminate between classes that are expected to be quite similar is referred to as homogeneous discrimination, e.g. two devices of the same make and model. The data set described in Section III contains samples from multiple ports residing on the same network switch. This results in a highly homogeneous classification problem because, not only are the devices similar, they share much of the same underlying circuitry. In this work, we estimate the Kullback-Leibler (KL) divergence [32] between sets of samples from different devices to demonstrate the similarity of fingerprints from ports on the same device as well as the uniqueness of fingerprints from completely different devices. The use of KL divergence is intended to be similar to the entropy-based method used in [33] to quantify the amount of identifying information present in the fingerprint data.

The KL divergence, also referred to as relative entropy, is a measure that quantifies how close a probability distribution is to that of a target distribution. A value of 0 indicates that two distributions are identical [34] and greater values indicate more difference between the two distributions.

The true probability distribution of the RESNET-generated device fingerprints is unknown. However, the KL divergence can be estimated based on fingerprint observations. As described previously, neural networks trained to classify devices are used to generate device fingerprints by modifying the output activation function. The KL divergence for a set of observed fingerprints can be estimated using the k-nearest-neighbor method described in [35]. In this work, we use a Python implementation of this method, by Hartland [36], that is freely available and utilizes the scikit-learn [37] implementation of NearestNeighbors.

C. Stability

The stability of a fingerprinting method is its ability to repeatedly reproduce the same or similar fingerprints from multiple measurements. In this work, fingerprint stability is demonstrated by examining the KL divergence between two sets of observations from the same class. A stable fingerprinting method will have small KL divergence values for observations from the same class, i.e. identical distribution. Therefore, the observation data from a single class is split into two sets and the KL divergence values are compared to verify stability. A similar test can be performed to demonstrate resilience to change using fingerprint data collected over a longer time period such as weeks or months. In future work, our data set will be expanded to explore this characteristic.

VI. Results & Discussion

This section describes the results of this work as they relate to the fingerprint assessment methods discussed in Section V.

A. Discrimination: DL

This section describes the discrimination power of VGG-like CNNs and RESNET-50 as applied to PHY device fingerprinting. Classification performance is again presented using overall classification accuracy and confusion matrices. As mentioned previously, raw digitally sampled signals are used as input, eschewing the need for engineered features. Therefore, model discrimination performance is compared across a range of input data lengths.

1) CNN: Discrimination power for CNNs was explored for a range of hyperparameters. One of the main hyperparameters of interest for this study is input data length. Figure 6 shows the training loss and accuracy curves for the training process

of a single model architecture ($N \times 32 \times 64 \times 32 \times 20$) across seven different lengths of input data, N. As described in Section III, the data set used for this effort contains 10,000 packet preambles from each of 20 devices. The minimum input length considered for this experiment was 128 samples and the maximum was 8192. It can be seen that longer input data lengths result in higher classification accuracy.

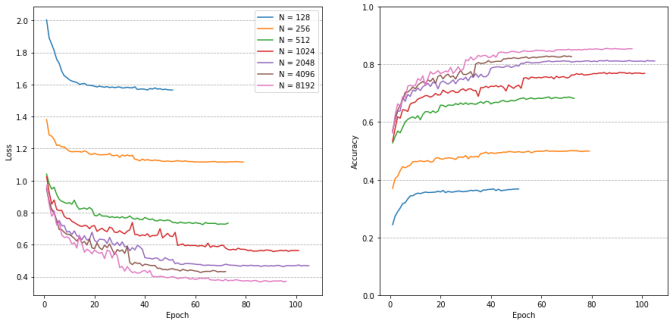

Fig. 6: CNN loss and accuracy for training steps across seven input lengths.

Shorter input sample lengths were considered to gain an idea of the overall classification capacity of these networks. In circumstances where the CNN size is overly large for the complexity of the input data, a phenomenon referred to as overfitting occurs. Figure 7 shows the output of the training process for a large CNN trained with input data samples of length 128. Overfitting occurs when the validation loss curve (blue line on the left) begins to trend upward after about 6 epochs. This also results in continued training classification accuracy improvement (orange line on the right), but a flattening out of validation classification accuracy of the model (blue line on the right). Therefore, this model is clearly too large for only 128 input samples. The maximum overall classification accuracy achieved for this configuration was only 34%.

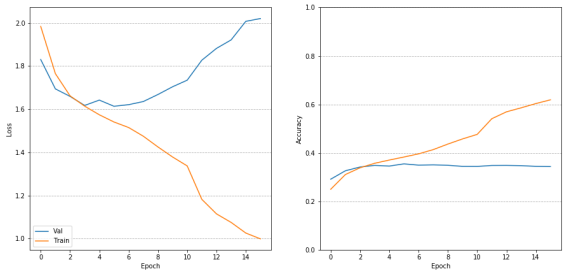

Fig. 7: CNN loss and accuracy for training and validation epochs of a $128 \times 512 \times 1024 \times 1024 \times 20$ node network.

The same model architecture with the input data length expanded to 2048 points is shown in Figure 8. This model clearly shows no overfitting through over 50 training epochs and results in an overall classification accuracy of approximately 82%.

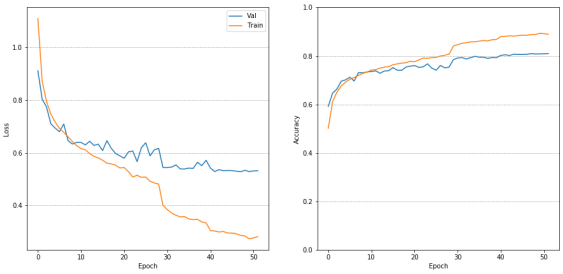

Fig. 8: CNN loss and accuracy for training and validation epochs of a $2048 \times 512 \times 1024 \times 1024 \times 20$ node network.

2) RESNET: This section describes experiments performed with the RESNET architecture to determine achievable discrimination power for the PHY device authentication application. In addition to overall classification accuracy, this experiment was again focused on the input sample length. A slightly modified RESNET-50 architecture is used throughout this experiment and is a fixed size as shown in Table II.

LAYER	BLOCKS	
conv1	1×3, 64, stride 3 1×3 max pool, stride 3	
conv2	$\begin{bmatrix} 1 \times 1, 64 \\ 1 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix}$	$\times 3$
conv3	$\begin{bmatrix} 1 \times 1, 128 \\ 1 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix}$	$\times 4$
conv4	$\begin{bmatrix} 1 \times 1, 256 \\ 1 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix}$	$\times 6$
conv5	$\begin{bmatrix} 1 \times 1, 512 \\ 1 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix}$	$\times 3$
	average pool, softmax	

TABLE II: RESNET-50 architecture from [29].

Models were trained for input sample lengths of $N = \{128, 256, 512, 1024, 2048, 4096, 8192\}$ with the RESNET-50 architecture. Each data element consists of a contiguous block of samples of length N randomly selected from a packet preamble containing 11,400 points. This is the same as described for the CNN experiments. The training process is depicted for each of these models in Figure 9 through validation loss and accuracy plots. It can again be seen that as input sample length increases, so does the discrimination power of the model. However, there is limited improvement in classification accuracy for input sample lengths of longer than 2048.

Figure 10 shows the classification results for a RESNET-50 implementation with length 2048 input in confusion matrix form using 2000 samples from each device that were not used for training or validation. The overall classification accuracy is 96.4%. This model shows high classification accuracy throughout, but still struggles with some of the ports on the same switch, e.g. TPLink ports 7 and 8 (bottom right).

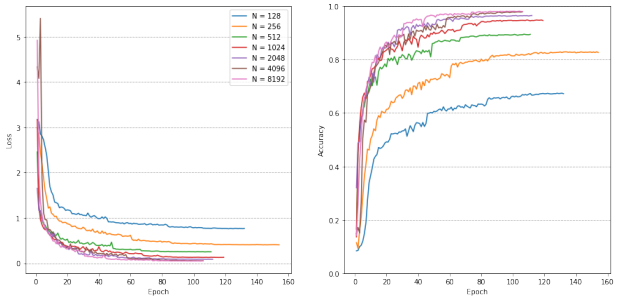

Fig. 9: RESNET-50 loss and accuracy for training and validation steps across all tested input lengths.

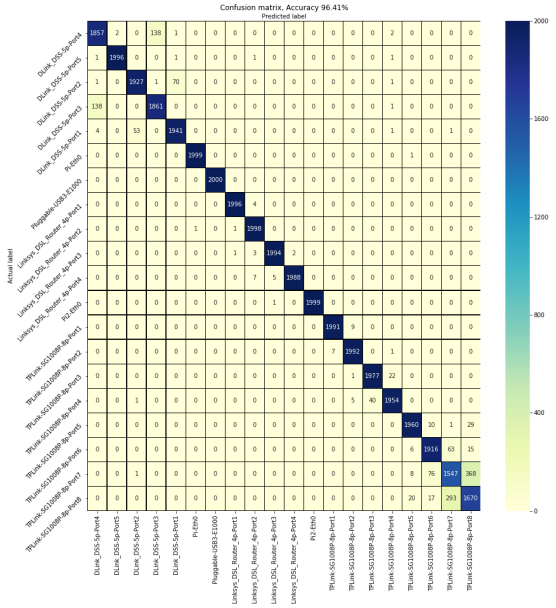

Fig. 10: RESNET-50 results for length 2048 input represented as a confusion matrix.

3) Discrimination Summary: Table III summarizes the classification performance. The RESNET-50 architecture achieves the highest classification performance of 96.4%.

METHOD	ACCURACY
CNN	83.5%
RESNET	96.4%

TABLE III: Summary of classification performance.

B. Homogeneous Discrimination & Stability

The KL divergence was estimated between fingerprints for each of the 20 classes of the PHY device fingerprinting data set, as described in Section V-B. Table IV shows the values of KL divergence calculated for DLink ports 0 through 4 when compared with each other as well as with the average of the KL divergence for all 15 non-DLink devices. It can be seen that the divergence values within the DLink ports is less than 7 and the average when compared to all non-DLink devices is greater than 30. This shows that fingerprints from ports on the same device are highly different from others. There is also at least an order of magnitude difference between the KL divergence values of ports from the same device. This demonstrates the homogeneous discrimination power of the DL-based fingerprints. Not only does the fingerprint provide discrimination power between discrete devices, but also across highly similar devices existing within a single physical platform and likely sharing circuitry.

PORT	0	1	2	3	4	Others
0	0.28	1.71	2.88	1.81	3.09	31.29
1	1.69	0.00	5.65	3.81	5.93	33.35
2	3.28	6.83	0.24	1.97	2.04	30.05
3	1.74	4.46	2.05	0.38	4.24	32.37
4	2.62	5.67	1.59	3.86	0.06	28.23
Others	38.93	40.74	38.80	40.36	38.04	0.67

TABLE IV: KL Divergence estimates for samples from a 5-port D-Link switch and means of all others.

The KL divergence values in Table IV also demonstrate fingerprint stability. The values along the diagonal of the matrix were calculated by splitting the data set for each class in half. The estimated KL divergence values are less than 1 for each class when compared to itself indicating stability across fingerprint collections.

VII. Conclusions

Device authentication for network security applications has been on-going for a number of years, predominantly in the wireless domain. However, there is significant need for passive, non-invasive device authentication solutions that are difficult to spoof as a means of securing critical infrastructure from the introduction of unauthorized devices. Current guidelines for ICS and OT networks include device authentication as a best practice but it is difficult to deploy this level of security in highly heterogeneous networks.

In this paper, we propose a method for performing device authentication based on Ethernet fingerprinting using deep machine learning techniques. We describe a potential architecture for the deployment of future device authentication systems based on techniques and also demonstrate that unique, unintended perturbations transmitted as part of a device's Ethernet signal can be used to identify a target device within a larger set. More specifically, we quantify the discrimination power and find that greater than 90% classification accuracy can be achieved using the RESNET-50 architecture with greater than 2048 input sample points. Finally, we quantify the homogeneous discrimination capability, and stability of the proposed method by estimating the KL divergence between classes. This finds similarity between fingerprints generated from ports that are on the same physical devices and orders of magnitude difference between homogeneous devices and others.

References

[1] V. Y. Pillitteri and T. L. Brewer, "Guidelines for Smart Grid Cybersecurity," Tech. Rep. NIST.IR.7628r1, Sep. 2014.

[2] I. T. AG, "Achieving Strong Industrial Security: How to Implement IEC 62443," https://www.infineon.com/cms/en/product/promopages/iec62443/, 2020.

[3] "Security and privacy controls for information systems and organizations," https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final, 2020.

[4] D. Formby, P. Srinivasan, A. Leonard, J. Rogers, and R. A. Beyah, "Who's in Control of Your Control System? Device Fingerprinting for Cyber-Physical Systems." in *NDSS*, 2016.

[5] National Cyber Awareness System, "Risks of Default Passwords on the Internet," https://www.us-cert.gov/ncas/alerts/TA13-175A, Jun. 2013.

[6] K. Lee, H. Yeuk, K. Yim, and S. Kim, "Analysis on Manipulation of the MAC Address and Consequent Security Threats," in *Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats*, ser. MIST '16. Vienna, Austria: Association for Computing Machinery, Oct. 2016, pp. 113–117.

[7] "IEEE Standard for Ethernet," *IEEE Std 802.3-2018 (Revision of IEEE Std 802.3-2015)*, 2018.

[8] L. E. Langley, "Specific emitter identification (SEI) and classical parameter fusion technology," in *Proceedings of WESCON '93*, Sep. 1993, pp. 377–381.

[9] W. C. Suski, II, M. A. Temple, M. J. Mendenhall, and R. F. Mills, "Radio Frequency Fingerprinting Commercial Communication Devices to Enhance Electronic Security," *International Journal of Electronic Security and Digital Forensics*, vol. 1, no. 3, pp. 301–322, Oct. 2008.

[10] D. R. Reising, M. A. Temple, and M. Mendenhall, "Improved wireless security for GMSK-based devices using RF fingerprinting," *International Journal of Electronic Security and Digital Forensics*, vol. 3, Mar. 2010.

[11] S. Banerjee and V. Brik, "Wireless Device Fingerprinting," in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA: Springer US, 2011, pp. 1388–1390.

[12] J. Lopez, N. C. Liefer, C. R. Busho, and M. A. Temple, "Enhancing critical infrastructure and key resources (CIKR) level-0 physical process security using field device distinct native attribute features," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1215–1229, 2018.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[14] T. J. O'Shea, N. West, M. Vondal, and T. C. Clancy, "Semi-Supervised Radio Signal Identification," *arXiv:1611.00303 [cs, math, stat]*, Nov. 2016.

[15] T. Jian, B. C. Rendon, A. Gritsenko, J. Dy, K. Chowdhury, and S. Ioannidis, "MAC ID Spoofing-Resistant Radio Fingerprinting," in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. Ottawa, ON, Canada: IEEE, Nov. 2019, pp. 1–5.

[16] K. Sankhe, M. Belgiovine, F. Zhou, L. Angioloni, F. Restuccia, S. DrOro, T. Melodia, S. Ioannidis, and K. Chowdhury, "No Radio Left Behind: Radio Fingerprinting Through Deep Learning of Physical-Layer Hardware Impairments," *IEEE Trans. Cogn. Commun. Netw.*, pp. 1–1, 2019.

[17] Y. Cheng, X. Ji, J. Zhang, W. Xu, and Y.-C. Chen, "DeMiCPU: Device Fingerprinting with Magnetic Signals Radiated by CPU," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '19. London, United Kingdom: Association for Computing Machinery, Nov. 2019, pp. 1149–1170.

[18] "SDS2352X-E (350 MHz)," https://siglentna.com/product/sds2352x-e-350-mhz/, 2019.

[19] E. E. Azzouz and A. K. Nandi, *Automatic Modulation Recognition of Communication Signals*. Norwell, MA, USA: Kluwer Academic Publishers, 1996.

[20] C. M. Rondeau, J. A. Betances, and M. A. Temple, "Securing ZigBee Commercial Communications Using Constellation Based Distinct Native Attribute Fingerprinting," in *International Conference on Computing, Networking and Communications (ICNC)*, Jul. 2018.

[21] M. D. Williams, S. A. Munns, M. A. Temple, and M. J. Mendenhall, "RF-DNA Fingerprinting for Airport WiMax Communications Security," in *2010 Fourth International Conference on Network and System Security*, Sep. 2010, pp. 32–39.

[22] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, Jan. 1989.

[23] H. Wang, Y. Cai, Y. Zhang, H. Pan, W. Lv, and H. Han, "Deep Learning for Image Retrieval: What Works and What Doesn't," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. Atlantic City, NJ, USA: IEEE, Nov. 2015, pp. 1576–1583.

[24] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International Conference on Engineering Applications of Neural Networks*. Springer, 2016, pp. 213–226.

[25] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *International Conference on Learning Representations*, 2015.

[26] M. Pumperla, "Maxpumperla/hyperas," https://github.com/maxpumperla/hyperas, May 2020.

[27] "Hyperopt Documentation," https://hyperopt.github.io/hyperopt/, 2020.

[28] J. Czakon, "Hyperparameter Optimization in Python. Part 2: Hyperopt." https://towardsdatascience.com/hyperparameter-optimization-in-python-part-2-hyperopt-5f661db91324, Jan. 2020.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv:1512.03385 [cs]*, Dec. 2015.

[30] V. Fung, "An Overview of ResNet and its Variants," https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035, Jul. 2017.

[31] I. Sanchez-Rola, I. Santos, and D. Balzarotti, "Clock Around the Clock: Time-Based Device Fingerprinting," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. Toronto, Canada: Association for Computing Machinery, Jan. 2018, pp. 1502–1514.

[32] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, Mar. 1951.

[33] P. Laperdrix, W. Rudametkin, and B. Baudry, "Beauty and the Beast: Diverting Modern Web Browsers to Build Unique Browser Fingerprints," in *2016 IEEE Symposium on Security and Privacy (SP)*. San Jose, CA: IEEE, May 2016, pp. 878–894.

[34] J. Shlens, "Notes on Kullback-Leibler Divergence and Likelihood," *arXiv:1404.2000 [cs, math]*, Apr. 2014.

[35] Q. Wang, S. R. Kulkarni, and S. Verdu, "Divergence Estimation for Multidimensional Densities Via k-Nearest-Neighbor Distances," *IEEE Trans. Inform. Theory*, vol. 55, no. 5, pp. 2392–2405, May 2009.

[36] N. Hartland, "Nhartland/KL-divergence-estimators," https://github.com/nhartland/KL-divergence-estimators, 2018.

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.