Anomaly Attack Detection in Wireless Networks Using DCNN

Van-Lan Dao * and Björn Leander *†
*Mälardalen University,Västerås, Sweden, van.lan.dao@mdu.se
†ABB AB, Process Control Platform,Västerås, Sweden, bjorn.leander@se.abb.com

***Abstract*—The use of wireless devices in industrial sectors has increased due to its various advantages related to cost and flexibility. However, legitimate wireless communication systems are vulnerable to cybersecurity attacks, due to its inherent open nature. Detection of rogue devices therefore plays a crucial role in critical wireless applications.**

In this paper we design a deep convolutional neural network (DCNN) to classify legitimate and rogue devices using raw IQ samples as input data. An algorithm is presented to find the optimal number of convolutional layers and number of filters for each layer under an accuracy constraint, in order to enable fast prediction time. Furthermore, we investigate how wireless channel models affect the accuracy and prediction time of the designed DCNN model. Our obtained results are benchmarked against previous DCNN models. Moreover, we discuss how the systems should react to a detected rogue device, considering the IEC 62443 standard.

***Index Terms*—deep learning, rogue device detection, fingerprinting, IEC 62443**

I. Introduction

Wireless network connectivity is becoming an increasingly important part of industrial systems. With the technological evolution related to Industry 4.0 and the Industrial Internet of Things (IIoT), the equipment used for industrial scenarios are increasingly heterogeneous, and includes, e.g., wireless sensors. Therefore, industrial standards for wireless communication are being developed and implemented [1], e.g., WirelessHART [2], ZigBee [3] and WIA-PA [4].

Sensors in industrial systems are used to indicate different aspects of the state of a physical environment to the overlaying systems, in order to supervise and control the process. The integrity and authenticity of the signal values received from sensors are therefore of great importance - as malformed, malicious or misleading data may lead to incorrect decisions on control readings. Several techniques exist for establishing trust and authenticity for sensory devices, usually embedded as part of the signal payload - symmetric signatures, secure encrypted data-streams based on shared secrets, etc. These techniques may however not be applicable for all applications, e.g., in the case of low-resource devices unable to perform advanced cryptographic operations. Systems containing such devices are vulnerable to spoofing attacks [5], jamming attacks [6], eavesdropping attacks [7], etc., where *rogue devices* emits network data with the intent of fooling the system that it originated from a legitimate sensor. Timely and effective detection of rogue devices' therefore plays an important role for the security of wireless communication systems.

There exist a wide range of solutions for anomaly device detection, however, this paper only focuses on methods using machine learning techniques [8]–[10]. In the literature, machine learning techniques are adopted for the detection and identification of devices using collected traffic traces and wireless signals [8]. Based on the unique received signature, e.g. the in-phase (I) and quadrature phase (Q) features at the physical layer, different deep learning models are proposed to detect whether the given transmitter is a legitimate device or not [11], [12]. However, hyperparameter searching is still an open problem for the machine learning models [8], [13]. Optimal hyperparamters can be found under different constraints such as maximum accuracy. Fast detection of rogue devices is important to minimize their interference and ensure safe operations of the legitimate system. For fast prediction time, graphics processing unit (GPU) are often used, but most industrial platforms are not equipped with GPUs, instead they may include good central processing units (CPUs) [14]. The complexity of the machine learning model can significantly affect the prediction time using CPU, which is a problem not investigated by previous papers. Aiming at fast rogue device detection, the following research questions (RQs) are formulated:

- RQ1: How to find optimal hyperparameters for a designed deep convolutional neural network (DCNN) model under a constraint of accuracy threshold?
- RQ2: How do wireless channel models affect the accuracy and prediction time of the designed DCNN model?

To address RQ1, we find the accuracy threshold for the designed DCNN first without optimal hyperparameters in terms of number of convolutional layers and their number of filters. Secondly, we propose an algorithm finding the optimal parameters. The designed DCNN model is evaluated based on an empirical dataset. In [12], the DCNN model in can obtain 100% accuracy with a few training epochs, while a more complex DCNN model in [11] can not reach 100% accuracy. Therefore, to tackle RQ2, we consider different ideal wireless channel models such as Rayleigh, Rician-k, and an empirical indoor environment Model B [15]. We benchmark the obtained results against previous papers, e.g. [11], [12].

Furthermore, different strategies for handling a detected rogue device on systems level is discussed and contrasted considering IEC 62443 [16], [17], a commonly used standard for cybersecurity in industrial automation and control systems.

The remainder of this paper is organized as follows. In section II an illustrative set of industrial use cases are presented. In section III the system model and method for signature measurement is introduced, and section IV describes the neural network design used for fingerprinting, and the performance evaluation. The numerical results of the evaluation are presented and discussed in Section V. In Section VI impact and strategies for handling rogue device detection on a system level is discussed. Section VII concludes the paper and outlines future plans.

II. Use Cases

There are several scenarios in which wireless sensors are being used in industrial systems. In this section a few of these use cases are described, including the potential system impact of a rogue device transmitting.

In manufacturing and process industries, the use of wireless sensors are commonly not as part of closed automatic control loops, the use is rather for collecting quality and health data for production equipment and produced materials. There are however scenarios in which wireless sensors are more practical and secure than wired. In such scenarios the placement of the sensors are on mechanical parts of machine that moves in a way that makes wiring unfeasible or impossible.

The described use-cases indicate the increasing adaption of wireless technologies within industrial settings and underlines the need of detecting rogue wireless devices due to the potentially adverse impact on Health, Safety and Environment (HSE) of a successful attack.

A. Wireless sensors in a paper mill

Ahln *et al.* [18] writes about transitioning toward wireless connections between I/O and controllers, using a section of a paper mill as an example. The authors show that for a starch cooker, which is an important part of a paper mill, the control performance is acceptable for that specific process.

In case of a successful attack against sensors or actuators in this example, can clearly lead to HSE incidents, with overflowing tanks, dry cooking, with a potential worst case outcome inflicting damage on plant personnel or the environment.

Another example from a paper mill where wireless sensors may be preferred, is when measuring different aspects of the produced paper. To measures color, fiber orientation, gloss, etc. a frame is mounted over the paper-trail in which a measurement cage is mounted. The cage, containing a set of wireless sensors, moves back and fourth over the produced paper sheet, see Fig. 1. The measurements are used to continuously control the paper quality, for detecting defects and potentially changing production parameters accordingly.

An attack on any of these sensors could lead to impaired quality of the produced product, in case of attacks leading to lost readings which would otherwise have led to corrective actions, or conversely, costly corrective actions or even stopping of the plant could be the cause of injected data indicating process parameters out of range. Even though none of these consequences have implication on HSE, they can have serious economic consequences for the plant owner.

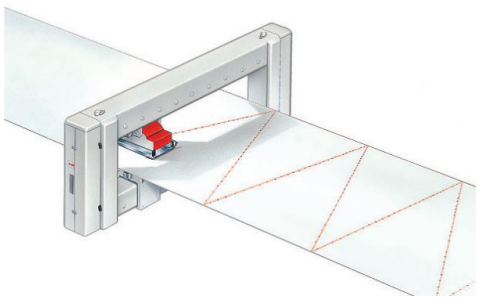

Fig. 1. Sensor mounting for paper mill use case. Image source: ABB

B. Waste-water handling - water quality measurements

Waste-water handling systems are typically divided into five main steps for decontamination [19] - pretreatment in the form of mechanical methods e.g., sedimentation, primary treatment using chemical methods such as coagulation, secondary treatment in the form of bio-degradation or filtration, tertiary treatment in the form of chemical methods such as oxidation and finally sludge treatment, which may be in the form of supervised tipping or incineration.

Within waste-water handling, there processes are typically quite slow and iterative, and therefore no fast control loops are generally needed for supervision of water quality. The systems are also geographically spread out over big areas. These two characteristics makes wireless sensors a suitable for this system type, such as e.g., described in [20].

The impact of a successful attack on a waste-water handling which ultimately produces clean drinking water, can have a severe effect on the society level, with far reaching HSE consequences, as illustrated by the Maroochy water service incident [21] in which over a million liters of untreated sewage was released into local waterways.

III. System Model and Signature Measurement

We consider a system consisting of n source nodes $S_i\,(1 \leq i \leq n)$ communicating with an access point (AP) in uplink as shown in Fig. 2. We assume that all source nodes and AP work in half-duplex mode with a single antenna using Time Division Multiple Access (TDMA) to avoid collisions among legitimate transmissions. These assumptions are reasonable in industrial settings. However, there exist a malicious actor who tries to attack the legitimate communication system for gaining different purposes, e.g. access to the system. While all legitimate nodes are protected inside the border by fences or walls, the attacker is only allowed to appear outside the border. Hence the legitimate nodes operating close to the border may be attacked by the attacker. It is more difficult for the attacker to reach the nodes located far away from the border, but it can potentially attack the whole system once any of the nodes close to the border is compromised. In contrast to the legitimate nodes, the attacker is modeled to be computationally powerful, communicate using full-duplex mode, and highly motivated to reach its objectives. The attacker works by means of signal jamming, spoofing or eavesdropping.

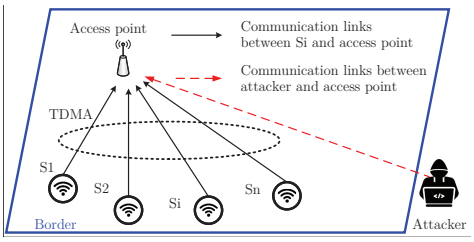

Fig. 2. System model.

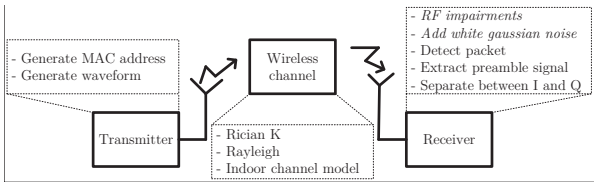

Fig. 3. The processing chain extracting RF fingerprinting.

Among the different available techniques for protection of the wireless communication system the detection of an anomaly node in run-time plays a crucial role to guarantee smooth operation of the system. This means that the legitimate system must deal with an anomaly node detection with different strategies during run-time.

To evaluate the system performance for the proposed DCNN, the IQ samples are extracted at the receiver based on the IEEE 802.11n standard, Fig. 3. This standard is used because it can provide higher throughput compared to the low energy and low throughput wireless communication standards based on IEEE 802.15 [22], [23]. At the transmitter side, a Medium Access Control (MAC) address is generated before creating a waveform for the given beacon frame bits. Then, the signal is transmitted via different types of wireless channel models such as Rician-k, Rayleigh, indoor environment channel model, etc. In practice, due to the imperfection of the hardware, IQ imbalance, phase noise, nonlinear distortion, and noise floor, we must add RF impairments and white Gaussian noise at the receiver side. This is also why the IQ samples as RF fingerprinting are unique for each pair of transmitter-receiver [11]. After that, packet detection and preamble signal extraction are conducted before separating I and Q parts. This is because the designed DCNN only accept real numbers as input data All aforementioned steps are implemented by using the *MATLAB*, *WLAN Toolbox*, and *Communications Toolbox* [24].

IV. Deep Convolutional Neural network Design

The raw time-series IQ samples are used as input to a neural network, a DCNN is thus adopted [13]. A DCNN can be used for many applications such as image classification, object detection, etc. In the literature, there are several typical DCNN configurations such as Alexnet [25], VGG-16 [26], ResNet-50 [27], and Efficient-B0 [28]. In line with these approaches, the DCNN as shown in Fig. 4 is proposed. Particularly, the proposed DCNN includes L convolutional

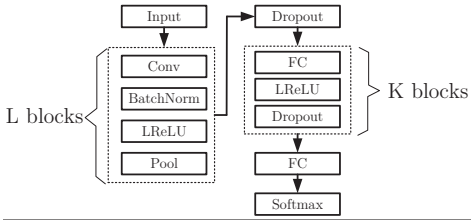

Fig. 4. Deep convolutional neural network architecture.

blocks, K fully connected layers blocks, dropout, last fully connected layer, and softmax layer. Each convolutional block is based on convolutional layer (Conv), batch normalization layer (BatchNorm), leaky rectified linear unit (LReLU), and pooling layer (Pool), while each fully connected layers block consists of fully connected layer (FC), LReLU, and dropout.

Extracting features from the raw input data is conducted by a Conv with a kernel, high level features can be obtained by multiple Convs. The size of features after each Conv is decreased and then redundant information is removed by a Pool to reduce the computational load. Moreover, BatchNorm and LReLU layers are adopted to speed up the training process, while dropout layer is deployed to cope with the overfitting problem. In addition, K blocks of fully connected layers also contribute to the extraction of higher level non-linear combinations of the features attained from the previous blocks.

To generate a dataset for the designed DCNN, we consider multiple pairs between the AP and wireless devices including possible attackers using different types of wireless channel models as described in section III. Then, the obtained dataset is divided into three parts consisting of 80% training set, 10% validation set, and 10% test set. Both training and validation sets are adopted during training phase, while test set is employed for performance evaluation phase. The performance metric used for evaluating the designed DCNN is its' accuracy.

By experimentation, we realized that K blocks of fully connected layers do not contribute significantly to the achievable accuracy of the designed DCNN. Moreover, an increase of complexity of the DCNN in terms of number of convolutional blocks L and fully connected layers blocks K lead to a significant growth of execution time for prediction.

To optimize the performance of neural networks, a number of methods have proposed in the literature such as manual hyperparameter tuning, automatic hyperparameter optimization, grid search, and so on [13]. **Algorithm 1** propose a method to find the optimal number of filters for each Conv of K convolutional blocks under a constraint of accuracy threshold. An increase of number L convolutional blocks as well as number of filters for each Conv layer can improve the accuracy performance. However, a more complex DCNN model has to deal with the overfitting problem, while the obtained accuracy does not improve significantly with complexity. The number of Conv layers as well as number of filters for each Conv layer and other parameters are found, first using a manual method

to define the accuracy threshold before running the proposed algorithm to find the optimal DCNN model. While lines 6-18 are responsible for finding number of filters for each Conv layer with a specific number of Conv layers l, the loop at line 5 is to find the optimal number of Conv layers. The proposed algorithm is not optimized for speed, but for reaching the best parameters, as it is executed offline.

Algorithm 1 Finding optimal number of convolutional blocks L and number of the filters for each Conv layer. Algorithm in pseudo-code.

1: **Input:** Maximum number of filters for each Conv layer Q, accuracy_threshold A_0
2: **Output:** L, number of filters for each Conv layer (N_l)
3: **function** main
4: Init maximum number of convolutional blocks L_0;
5: **for** $l = 1 : L_0$ **do**
6: **for** $j_1 = 1 : Q$ **do**
7: $N_1 = j_1$;
8: **for** $j_2 = 1 : Q$ **do**
9: $N_2 = j_2$;
10: **for** $j_l = 1 : Q$ **do**
11: $N_l = j_l$;
12: Evaluate the accuracy of the DCNN model, obtained_accuracy A.
13: **if** $A \geq A_0$ **then**
14: L = l; **Return** L, N_l;
15: **end if**
16: **end for**
17: **end for**
18: **end for**
19: **end for**
20: **end function**

V. Results and Discussions

The dataset is generated for 8 legitimate wireless devices and 10 unknown wireless devices with signal-to-noise ratio (SNR) = 20 dB using Rayleigh, Rician k = 2.8, and indoor environment Model B [15]. Each pair generates 5000 IQ samples [29]. All simulations, including the evaluation of the proposed algorithm, have been carried out on a 64-bit Windows 10 Pro machine with Intel Core I7-10700KF CPU @ 3.8 GHz, 16 GB memory and GPU NVIDIA GeForce RTX 3080 using the *Deep Learning Toolbox* [24]. l_2 regularization 0.0001, learning rate 0.0001, mini batch size 256, and Adam optimizer are configured.

First, we consider the proposed DCNN in [11] with L = 2 including 50 filters for each convolutional layer with size of [7 1] and [7 2], respectively, and stride of [1 1]; Max pooling with pool size [2 1] and stride [2 1]; K = 2 fully connected layers blocks with 256 and 80 neurons, respectively; dropout layer with dropout probability 0.5; nine neurons for classification layer. The accuracy during the training phase for different wireless channel models is shown in Fig. 5.

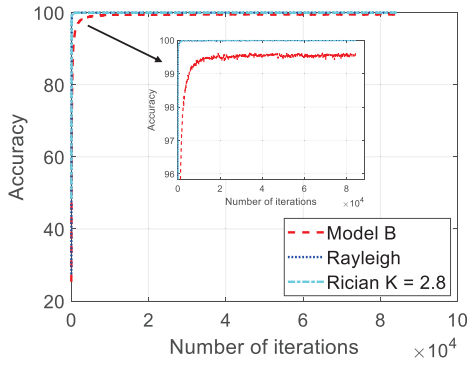

Fig. 5. Training progress for different wireless channels.

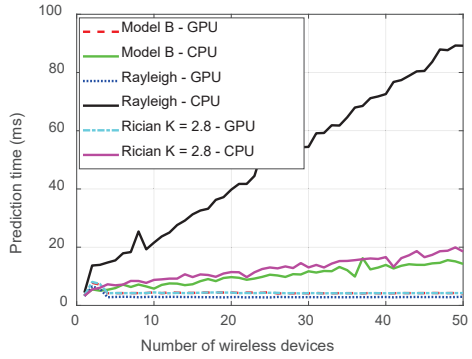

Fig. 6. Prediction time for different wireless channels.

It can be seen from the figure that the accuracy can reach 100% during a small number of iterations for both Rayleigh and Rician channel models as presented in [12], while it takes a much larger number of iterations to obtain 99.6% for the indoor environment Model B. This is due to ideal conditions of both Rayleigh and Rician channel models, while the indoor environment Model B is generated based on experiment data [15].

In Fig. 6, the prediction time following number of wireless devices is described when the prediction phase is executed on both CPU and GPU. It can be seen from the figure that the prediction time on GPU is very small, around 4.3 ms for both indoor Model B and Rician channel models and 2.8 ms for the Rayleigh channel model, and these values do not change much for different number of wireless devices. However, the prediction time for the Rayleigh channel model running on CPU increases significantly and higher than both indoor Model B and Rician channel model. Here, the prediction time for the indoor environment Model B is the smallest when the prediction phase runs on CPU.

Based on parameters for the DCNN model in [11], the optimal parameters are found using Algorithm.1, with an equivalent accuracy to the DCNN model in [11], [12], 99.61%. Resulting in 22 filters for the first Conv layer and 48 filters for the second Conv layer instead of 50 filters for each Conv layer in [11]. Fig. 7 illustrates the accuracy of training phase for different DCNN models.

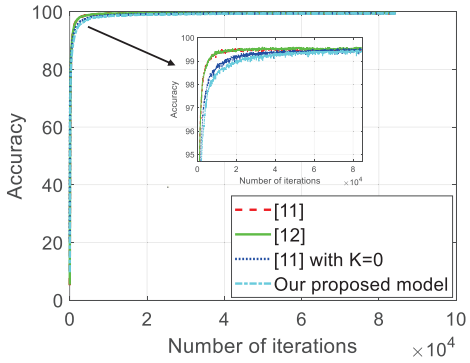

Fig. 7. Training progress for different DCNN models.

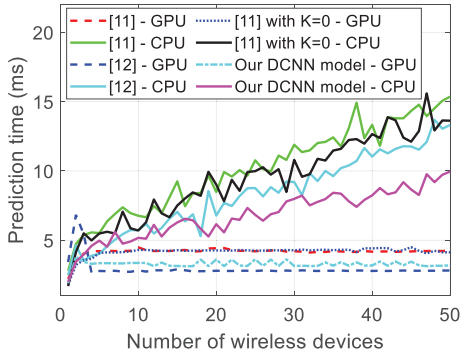

Fig. 8. Prediction time for different sets of hyperparameters of the DCNN.

We can see that both models in [11], [12] can reach the stable accuracy value sooner than others including the model with optimal parameters and the model in [11] with K = 0. However, after a large number of iterations, all models can obtain the same accuracy level. As stated previously, the training phase is implemented offline. Therefore, the optimal DCNN model is much less complex than others, leading to a reduction of prediction time. In Fig. 8, we can see how the prediction time changes versus number of wireless devices. Here, it is clear that the optimal DCNN model can offer a smaller prediction time compared to other DCNN models when executing on CPU.

VI. System Reaction

In previous sections, we describe a method which with high accuracy can identify wireless traffic from rogue devices using fingerprinting. Assuming such traffic is detected, the system must react in a meaningful way. The method for detection is by its nature distributed to each access point, but an effective response may require system-level coordination. The realization that a rogue device is sending traffic in the network may be an indication of an on-going attack and there could be good reason to increase the overall security posture of the system. It is noted that a classification of rogue devices is necessary to propose the most effective solution for each case [30]. This is out of scope of this paper. The legitimate communication systems can still deploy a wide range of solutions dealing with specific anomaly device such as resources allocation [31], [32], friendly jammer [33], etc. However, several cases, e.g., a jammer covering the full legitimate communication frequency band with very high transmit power, the aforementioned solutions can not work, and a system reaction is therefore needed.

Following are examples of strategies which could be applied on detection of rogue communication:

- Deny or close the connection for anomalous device, which in principle is equal to using fingerprinting for authentication. No system-wide response.
- Raising an alarm, e.g., to local system administrators or operators. This is required by the IEC 62443 standard [16] for security level 3 or higher[1].
- Raising an alarm after a threshold is reached, e.g., to a Security Information and Event Management (SIEM) system [34].
- Changing state of system, may include:
  - Revocation / closure of open sessions, requires clients to re-open session with fresh access tokens [35].
  - Graceful degradation - i.e. increase / enable security checks and similar which are normally disabled.
  - Switching to island mode, i.e., prevent communication through control system boundary[2].
- Any combination of above strategies.

As can be seen, several different strategies can be applied. It is important that the response is proportionate to the potential attack, otherwise the system reaction can be the actual goal of the attacker, as a means to force the system into a degraded state, or cause economic harm.

VII. Conclusions

In this paper, typical use cases are analyzed to highlight the rogue device detection problem. A method for rogue device detection using raw IQ signal data is presented, using DCNN. An algorithm is proposed for finding optimal number of convolutional layers and their number of filters is provided for the designed DCNN under a constraint of accuracy threshold to decrease the prediction time. The performed simulations indicates that the selected parameters significantly affect the prediction time when using a standard CPU, as compared to a GPU. Moreover, the effects of wireless channel models on the accuracy and prediction time of the designed DCNN model are investigated. Finally, we discuss system-wide reactions of the legitimate system when rogue devices are detected, using the the IEC 62443 standard.

As future work, we envision implementing and evaluating the rogue device detection algorithm in a realistic system, along with system-wide reactions, to investigate its feasibility. Evaluations of different heuristics to decrease the training time is another possible avenue of future work.

[1]IEC 62443-3-3 System Requirement 2.2, Requirement enhancement 1 - Identify and report unauthorized wireless devices.

[2]IEC 62443-3-3 System Requirement 5.2, Requirement enhancement 2 - Island Mode

ACKNOWLEDGEMENTS

This work is supported by the Swedish Foundation for Strategic Research through the Serendipity Project; ABB AB; the industrial postgraduate school Automation Region Research Academy (ARRAY), funded by The Knowledge Foundation; and the European Unions Horizon 2020 ECSEL JU project InSecTT under grant agreement No 876038[3]. The authors would like to acknowledge Tomas Lindström for his valuable discussion.

REFERENCES

[1] S. Vitturi, C. Zunino, and T. Sauter, "Industrial communication systems and their future challenges: Next-generation Ethernet, IIoT, and 5G," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 944–961, 2019.

[2] "IEC 62591 Industrial Networks - Wireless Communication Network and Communication profiles - WirelessHART," International Electrotechnical Commission, Geneva, CH, Standard, 2016.

[3] S. Khanji, F. Iqbal, and P. Hung, "ZigBee Security Vulnerabilities: Exploration and Evaluating," in *International Conference on Information and Communication Systems, ICICS 2019*. IEEE, 2019, pp. 52–57.

[4] "IEC 62601 industrial networks - wireless communication network and communication profiles - WIA-PA," International Electrotechnical Commission, Geneva, CH, Standard, 2021.

[5] M. H. Ylmaz and H. Arslan, "A survey: Spoofing attacks in physical layer security," in *IEEE Local Computer Networks Conference Workshops (LCN Workshops)*, Clearwater Beach, FL, USA, 2015, pp. 812–817.

[6] H. Pirayesh and H. Zeng, "Jamming attacks and anti-jamming strategies in wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 767–809, 2022.

[7] P. Angueira, I. Val, J. Montalban, O. Seijo, E. Iradier, P. S. Fontaneda, L. Fanari, and A. Arriola, "A survey of physical layer techniques for secure wireless communications in industry," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 810–838, 2022.

[8] Y. Liu, J. Wang, J. Li, S. Niu, and H. Song, "Machine learning for the detection and identification of internet of things devices: A survey," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 298–320, 2022.

[9] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (iot) security," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020.

[10] T. Jian, B. C. Rendon, E. Ojuba, N. Soltani, Z. Wang, K. Sankhe, A. Gritsenko, J. Dy, K. Chowdhury, and S. Ioannidis, "Deep learning for RF fingerprinting: A massive experimental study," *IEEE Internet of Things Magazine*, vol. 3, no. 1, pp. 50–57, 2020.

[11] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "ORACLE: Optimized radio classification through convolutional neural networks," in *IEEE Conference on Computer Communications*, Paris, France, 2019, pp. 370–378.

[12] N. S. Aminuddin, M. H. Habaebi, S. H. Yusoff, and M. R. Islam, "Securing wireless communication using RF fingerprinting," in *International Conference on Computer and Communication Engineering (ICCCE)*, Kuala Lumpur, Malaysia, 2021, pp. 63–67.

[13] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, http://www.deeplearningbook.org.

[14] TTTech, Inc., "Nerve edge intelligence for industrial automation," https://www.tttech.com/wp-content/uploads/TTTech_Fog-Computing_White-paper.pdf, accessed: 2022-06-8.

[15] V. Erceg, "Tgn channel models," *atftp://ieee: wireless@ ftp. 802wireless-world. com/11/03/11-03-0940-04-000n-tgn-channel-models. doc*, 2004.

[16] "IEC 62443 security for industrial automation and control systems," Internation Electrotechnical Commission, Geneva, CH, Standard, 2009-2018.

[17] B. Leander, A. Čaušević, and H. Hansson, "Applicability of the IEC 62443 Standard in Industry 4.0 / IIoT," in *Proceedings of the International Conference on Availability, Reliability and Security*, ser. ARES '19. New York, NY, USA: ACM, 2019.

[3]The document reflects only the authors view and the Commission is not responsible for any use that may be made of the information it contains.

[18] A. Ahlen, J. Akerberg, M. Eriksson, A. J. Isaksson, T. Iwaki, K. H. Johansson, S. Knorn, T. Lindh, and H. Sandberg, "Toward Wireless Control in Industrial Process Automation: A Case Study at a Paper Mill," *IEEE Control Systems*, vol. 39, no. 5, pp. 36–57, 2019.

[19] G. Crini and E. Lichtfouse, "Advantages and disadvantages of techniques used for wastewater treatment," *Environmental Chemistry Letters*, vol. 17, no. 1, pp. 145–155, 2019.

[20] J. Lloret, L. Garcia, J. M. Jimenez, S. Sendra, and P. Lorenz, "Cluster-Based Communication Protocol and Architecture for a Wastewater Purification System Intended for Irrigation," *IEEE Access*, vol. 9, pp. 142 374–142 389, 2021.

[21] J. Slay and M. Miller, "Lessons learned from the maroochy water breach," in *International conference on critical infrastructure protection*. Springer, 2007, pp. 73–82.

[22] A. Varghese, D. Tandur, and A. Ray, "Suitability of wifi based communication devices in low power industrial applications," in *IEEE International Conference on Industrial Technology (ICIT)*, Toronto, ON, Canada, 2017, pp. 1307–1312.

[23] Z. Fernndez, O. Seijo, M. Mendicute, and I. Val, "Analysis and evaluation of a wired/wireless hybrid architecture for distributed control systems with mobility requirements," *IEEE Access*, vol. 7, pp. 95 915–95 931, 2019.

[24] The MathWorks, Inc., "Matlab," https://se.mathworks.com/, accessed: 2022-06-8.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations, San Diego, CA, USA, May 7-9, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30*. IEEE Computer Society, 2016, pp. 770–778.

[28] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6105–6114.

[29] G. Reus-Muns and K. R. Chowdhury, "Classifying UAVs with proprietary waveforms via preamble feature extraction and federated learning," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 7, pp. 6279–6290, 2021.

[30] J. M. Hamamreh, H. M. Furqan, and H. Arslan, "Classifications and applications of physical layer security techniques for confidentiality: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1773–1828, 2019.

[31] V. N. Vo, H. Tran, V.-L. Dao, C. So-In, D.-D. Tran, and E. Uhlemann, "On communication performance in energy harvesting WSNs under a cooperative jamming attack," *IEEE Systems Journal*, vol. 14, no. 4, pp. 4955–4966, 2020.

[32] V.-L. Dao, L.-N. Hoang, S. Girs, and E. Uhlemann, "Defeating jamming using outage performance aware joint power allocation and access point placement in uplink pairwise NOMA," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 1957–1979, 2021.

[33] V. N. Vo, C. So-In, H. Tran, D.-D. Tran, and T. P. Huu, "Performance analysis of an energy-harvesting IoT system using a UAV friendly jammer and NOMA under cooperative attack," *IEEE Access*, vol. 8, pp. 221 986–222 000, 2020.

[34] S. Bhatt, P. K. Manadhata, and L. Zomlot, "The operational role of security information and event management systems," *IEEE Security and Privacy*, vol. 12, no. 5, pp. 35–41, 2014.

[35] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, May 2015.