future internet

MDPI

Article

IoT Nodes Authentication and ID Spoofing Detection Based on Joint Use of Physical Layer Security and Machine Learning

Dania Marabissi *,†,‡, Lorenzo Mucchi †,‡ and Andrea Stomaci †,‡

Department of Information Engineering, University of Florence, 50121 Firenze, Italy; lorenzo.mucchi@unifi.it (L.M.); andrea.stomaci@unifi.it (A.S.)

* Correspondence: dania.marabissi@unifi.it

† Current address: Department of Information Engineering, University of Florence, Via di Santa Marta 3, 50139 Florence, Italy.

‡ These authors contributed equally to this work.

Abstract: The wide variety of services and applications that shall be supported by future wireless systems will lead to a high amount of sensitive data exchanged via radio, thus introducing a significant challenge for security. Moreover, in new networking paradigms, such as the Internet of Things, traditional methods of security may be difficult to implement due to the radical change of requirements and constraints. In such contexts, physical layer security is a promising additional means to realize communication security with low complexity. In particular, this paper focuses on node authentication and spoofing detection in an actual wireless sensor network (WSN), where multiple nodes communicate with a sink node. Nodes are in fixed positions, but the communication channels varies due to the scatterers' movement. In the proposed security framework, the sink node is able to perform a continuous authentication of nodes during communication based on wireless fingerprinting. In particular, a machine learning approach is used for authorized nodes classification by means of the identification of specific attributes of their wireless channel. Then classification results are compared with the node ID in order to detect if the message has been generated by a node other than its claimed source. Finally, in order to increase the spoofing detection performance in small networks, the use of low-complexity sentinel nodes is proposed here. Results show the good performance of the proposed method that is suitable for actual implementation in a WSN.

Keywords: physical layer security; wireless fingerprinting; machine learning; intrusion detection

Citation: Marabissi, D.; Mucchi, L.; Stomaci, A. IoT Nodes Authentication and ID Spoofing Detection Based on Joint Use of Physical Layer Security and Machine Learning. *Future Internet* **2022**, *14*, 61. https://doi.org/10.3390/fi14020061

Academic Editor: Wei Yu (Leader)

Received: 21 January 2022
Accepted: 15 February 2022
Published: 17 February 2022

1. Introduction

The demand for mobile data capacity is continuously increasing, and future wireless systems are expected to support a wide variety of services that spans from low data rate machine-to-machine (M2M) type communications to enhanced broadband in extremely different application scenarios. Consequently, a high amount of data will be exchanged via radio signals, introducing a significant challenge for security since the broadcast nature of wireless channel makes the communications extremely vulnerable to several security threats, such as wiretapping, spoofing, message falsification, and jamming, which are, in general, dynamic and difficult to predict. Traditionally the communication security is managed by high layers and solved by means of a wide variety of ciphers and key management systems. The basic idea is that using complex calculations, the brute force attack is generally not affordable with a non-quantum computer. However, these approaches are usually computationally expensive and require protocols with high overhead. Moreover, the continuous growth in computational power makes vulnerable such ciphers initially considered unbreakable. Additionally the key distribution can be a problem especially in dynamic systems and introduces latencies that can be unacceptable for delay-constrained services [1–3]. Consequently, in new networking paradigms, such as the Internet of Things (IoT), traditional methods for security cannot achieve the desired performance due to the

radical change of requirements and constraints for establishing secure communication. In particular, cryptographic techniques can lead to excessive transmission overhead, communication latency and power consumption, which are not suitable for IoT devices with limited resources.

The application of IoT services and systems spans every aspect of our lives: health, education, industry, smart homes and cities, transportation, and utilities, just to name a few. The potential benefits offered by IoT are endless and will become more effective with the increase in the number of connected devices, but this requires to address the new security challenges and threats that arise. This is particularly critical for some applications, such as e-health or remotely controlled cars [4].

IoT systems will be made of a huge number of devices that very often will have limited computation, memory and energy resources. Hence, the use of complex asymmetric cryptography schemes will be impractical in many cases. Symmetric cryptography is more suitable for many IoT devices from a complexity and energy consumption point of view, but in this case, the distribution of the keys remains a challenge.

In this context, physical (PHY) layer security (PLS) is a promising additional means to realize communication security with low complexity [3] since it substantially operates independently on the higher layers and can be used to enhance the security of existing approaches. The basic idea is exploiting the randomness of the propagation channel, noise and interference to limit the information that can be wiretapped by an unauthorized user. In addition, PLS can be used to generate secure keys and to identify unauthorized users [5]. Indeed, PLS can be realized in different ways:

- *Secret communications without encryption*—with a suitable design of the transmitted waveform (coding, modulation, precoding schemes, etc.) together with the exploitation of the available channel state information, it is possible for the intended receiver to be enabled to successfully decode the data while the potential eavesdropper is not.
- *Secure key generation*—when the use of encryption is preferred, the randomness of the channel between two nodes can be exploited to generate keys to be used for symmetric encryption.
- *Node authentication/spoofing detection*—by means of the identification of specific distinguishing features of the wireless channel experienced by a node or of the transmitting device, the receiver can detect if the message has been illegitimately modified by a node other than its legitimate source.

The PLS is not thought to replace the traditional security, but it is an additional security layer which helps to enhance the security level, in particular, when low-resourced devices are used with a wireless connection [6]. Indeed, PLS (i) involves only the physical layer, (ii) lies on the variation and randomness of the wireless channel rather than on computational complexity of hard mathematical problems, (iii) uses the randomness of the wireless channel as a "secure key" avoiding key management burden, and (iv) can authenticate legitimate nodes quickly before demodulation and decoding, thus reducing the overall latency. The use of PLS has been proposed and adopted in the literature for several years [7], and the recent development of the IoT has given a great impetus to the research community to use PLS. Often in IoT networks, hard encryption procedures cannot been performed, at least with high frequency, and the effectiveness of encryption is related to the distribution and protection of a secret key that in IoT systems can fail [8]. Moreover, PLS approaches do not require modifications to the existing systems and hence, can be easily added in a very short time. Hence, the use of physical layer characteristics as a security tool can be seen as a method to help the higher layers to protect the system and, at the same time, implement security, even in low-resourced devices [9,10].

This paper focuses on PHY layer continuous authentication and spoofing detection. In particular, the paper proposes a *machine-learning (ML) wireless fingerprinting* method for a wireless sensor network (WSN), where multiple nodes communicate with a sink node that is in charge of their authentication. The idea is to exploit ML capabilities to verify if the characteristics of the propagation channel of current messages correspond to those

of previous transmissions of authorized users. ML allows to implement more efficient data protection, having the capability of analyzing multi-dimensional information, without the need of an analytical model, and in a continuous way, thus taking into account the time-varying effects [1].

As specified before, the physical layer authentication (PLA) method proposed in this paper does not want to eliminate or replace the traditional procedures, but simply adds a "first line of defense". Traditional security protocols, such as WPA2, protect access to the communication network by using encryption. The method proposed here could be helpful in contexts where great and frequent efforts cannot be posed to perform hard encryption. In addition to the general features of PLS explained before, in the proposed method, all the computation (the ML algorithm) is executed at the sink node, while the low-complexity sensors have nothing to do, except transmit their data. This is not possible using encryption algorithms that operate at both sides of each communication link. Moreover, existing Wi-Fi security access protocols weaknesses are known [11], and protecting the massive IoT requires new security features which are currently not included into the wireless standards [8].

1.1. Motivation and Related Literature

PHY layer authentication techniques are gaining a lot of interest. These can be distinguished between those that are key based or keyless, depending on whether a shared secrecy key is used by the transmitter and receiver or not [12]. Different key-based authentication approaches have been proposed. The main idea is superimposing an authentication or a noise-like signal to the message, or introducing a certain level of randomness in the signal [13–15]. This in general requires additional computational complexity to recover the signal through demodulation and decoding and to generate the keys. Differently, keyless PHY layer authentication methods exploit the properties of the wireless link (*wireless fingerprint*—WF) to identify the legitimate node: specific characteristics of the transmitter or of its communication channel are extracted from the received signal and compared with those of previous authenticated messages (this can be provided, for example, using an initial higher layer authentication procedure) to identify a claimed source. In this way, the receiver can continuously authenticate the transmitting node, and being that the channel properties are location dependent, these are difficult to be obtained by a malicious user if it is not very close to the legitimate one. Several PHY layer authentication methods have been proposed, under different assumptions of the available channel information and system models. These methods exploit channel state information and spatial domain measurements, such as channel gains, path delay profile, carrier frequency offset, power spectra density, and received signal strength as, for example, in refs. [16–19]. In general, such methods compare a specific PHY layer feature of the received signal with a test threshold that influences the authentication accuracy. However, choosing the appropriate threshold can be challenging due to the characteristics of the propagation environment and the unknown spoofing model. Additionally, in ref. [20], it is shown that under a low-SNR regime, the authentication based on a binary hypothesis testing cannot guarantee robust performance; thus an adaptive threshold is proposed. In ref. [21], a Q-learning method is used to select the optimal test threshold based on the received signal strength indicator (RSSI). Another problem of many of existing PHY layer authentication methods is that they use only one channel attribute that might be inaccurate and not enough to provide sufficient differentiation among transmitters. Indeed, channel attributes are time varying and must be estimated; hence, they can be affected by errors. Analyzing multiple PHY layer attributes improves the authentication robustness. In fact, it is more difficult for a malicious user to predict all attributes from the signal received on a different location: the legitimate user has high-dimensional protection. Toward this goal, ML has emerged as a viable solution for security and in particular for authentication [1,2]. It allows considering multiple parameters; moreover, it can provide a model-free authentication, also under dynamic network conditions. Differently, most of the existing approaches are model based, thus requiring a lot of data for achieving an

accurate model that in complex environments may be difficult to obtain with a consequent performance degradation. ML was applied to PHY layer authentication only very recently, in general not considering time-varying channels and in simple network models, where a single node communicates with an authorizing entity that has to distinguish between the authorized node and a potential malicious user. In ref. [22], an extreme learning machine is used, exploiting multi-dimensional attributes and using the training data generated from a spoofing model. Attributes must have the same statistical distribution (Gaussian), and the spoofing model is needed, so its applicability is limited. In ref. [23], authentication is based on a logistic regression model assisted by multiple landmarks at different locations that use multiple antennas to estimate the RSSI of the transmitter. In both previous papers, a static environment is considered; moreover, the required computation load cannot be affordable for low-complexity devices such as the one we consider. In ref. [24], a kernel least mean square scheme is presented, where the dimensionality of the multiple-attribute authentication is reduced by the kernel function that is able also to track the variations in time. Other sophisticated ML algorithms proposed for PLA based on Q-Learning and neural networks as in [25] cannot be suitable for resource-constrained scenarios, such as the one of interest in this work.

Approaches more suitable for an IoT context for their low computational complexity are based on one-classical ML classification schemes, as in [26–28]. Different PLA algorithms are presented and compared in [26], based on classical hypothesis testing and on ML, particularly nearest neighbor (NN) and support vector machine (SVM) algorithms. The proposed solutions exploit the characteristics of a set of parallel wireless channels (modeled as an OFDM transmission). These ML approaches are unsupervised, allowing a clustering algorithm to decide the nature of the received packets. Channel coefficients are fixed. In refs. [27,28], two ML approaches based on SVM and k-means clustering are investigated in two different contexts, such as in multiple input multiple output (MIMO) stationary systems [27] or in unmanned aerial vehicle (UAV) aided wireless systems [28]. Channel variations are not considered.

1.2. Paper Contribution

In this paper, we focus on a scenario where, first, legitimate devices are authenticated by means of a higher level procedure, and a unique ID is assigned to each of them. Successively, during communication, the sink node performs a continuous PLA (and spoofing detection), which uses multiple PHY layer attributes to verify the correspondence of the WF of each user with the assigned ID. In particular, the continuous authentication is performed by a two-step procedure:

1. Each transmitting node is identified by means of a ML supervised non-parametric classification algorithm where training data are labeled (i.e., it is associated to the corresponding user's ID). It means that an eventual malicious node is not detected at this step, but it is classified as belonging to one of the authorized nodes' class.
2. In order to detect a spoofing attack, a successive cross-check of the PHY layer classification results and the ID declared by the transmitting node is performed.

The cross check of the ID and PLA outcome gives a higher level of protection with respect to the exclusive use of an ID: the ID can be stolen, and while the PLA aims for supporting the legitimate devices by a reciprocal wireless link, the wireless channel features can be used as an additional unique security signature. The spoofing detection capability increases with the network dimension (i.e., the number of nodes), and this is important for future IoT systems where a massive machine access is foreseen. However, we propose also the introduction of *sentinel nodes*, which can significantly enhance the detection capability, especially in small networks.

Summarizing the contributions of the paper are the following:

- Proposal of a continuous authentication/spoofing detection system, suitable for an actual WSN, where multiple IoT nodes communicate with a sink node. Differently, previous works on PLA [16–28] were usually based on scenarios with a single au-

thorized node that must be distinguished by the unauthorized one, and hence, only a binary decision is needed (i.e., binary classification). However, binary classification is not suitable for large-scale IoT networks, and extending these methods to a multi-user scenario may not be straightforward. In any case, it means seeking an optimal threshold individually for each IoT legitimate node, which is expensive in terms of resources and signaling. Moreover, adopting a ML approach means adopting and training one machine for each node. Here, instead, the malicious user must be distinguished by multiple authorized nodes using their WFs. This is a more complex scenario because there is a higher variability of legitimate channels, and the probability that the spoofing attacker is close to one of them is higher. The proposed approach allows to simultaneously distinguish the malicious node from all the legitimate ones, using a single machine.

- We propose a threshold-free method, thanks to the integration of the classification of devices using their PHY-layer attributes and the associated device ID. Differently, in most of the approaches proposed in the literature, a threshold is needed to distinguish between legitimate and malicious user data [16–19,19–21,24,26]. However, the threshold must be optimized for each scenario with a consequent performance degradation, especially in time-varying scenarios.
- We propose a method that does not require any knowledge and specific statistical distribution, neither for the PHY-layer attributes nor for the spoofing model. This makes the proposed system more applicable in actual contexts. Conversely, many of the proposed approaches in the literature make assumptions difficult to obtain in a real environment, such as specific channel models and knowledge of data belonging to the attacker [16,18,22,26,28].
- We investigate a solution for a supervised classification of devices based on CART and random forest algorithms that were not previously investigated in this context. Random forest was adopted in ref. [29], using channel and hardware features to distinguish different nodes; however, the investigation is limited to node identification (i.e., no spoofing detection) and is very limited and related to a single static experimental setup.
- The proposed approach integrates multiple-attributes to provide a higher identification accuracy. Differently, most of the papers in the literature use only one channel attribute [17–19,25,28,30] or multiple observations of the same attribute, exploiting time or spatial diversity [22,23,26,27]. However, attributes can be estimated with a different level of reliability; thus, having different attributes allows to compensate for low-reliable attributes with high-reliable ones. Only a few papers have integrated different attributes, but often these are limited, such as in ref. [16,31] where only CSI and delay are considered. A wide range of attributes are considered only in [24,29]. Moreover, the angle of arrival (AoA) attribute is rarely considered, and in different scenarios, such as in ref. [32], where AoA is used for improving the security of channel training authentication, integrating the AoA with the pilot randomization, or in ref. [33], where AoA information is cross checked with the GPS information for improving security. Finally, here, the effects of different attributes are separately evaluated. To our knowledge, only [24] provides an analysis based on the availability of different attributes but in a different context.
- The proposal of the use of *sentinel nodes* to improve spoofing detection in small networks. Cooperative solutions for PLA have been rarely considered as in ref. [23], but here the goal of sentinel nodes is completely different. These nodes do not have to perform any operation except sending periodical beaconing signals.
- The evaluation of the system performance in an actual and general time-varying channel, also considering different environment conditions, while most of the papers in the literature consider fixed channel parameters and simple channel models. Moreover, the effects of different channel attributes in classification results are evaluated.

2. System Model

Why is security necessary in wireless sensor networks (WSNs)? Due to the broadcast nature of the transmission medium, wireless sensors are vulnerable. Another vulnerability is that nodes are often placed in a hostile or dangerous environment, and they are not physically safe. Most of the threats and attacks against security in WSNs are almost similar to their wired counterparts, while some are exacerbated with the inclusion of wireless connectivity. Attacks on WSNs can be classified as follows:

1. Attacks against security mechanisms;
2. Attacks against basic mechanisms (such as routing mechanisms).

In many applications, the data obtained by the sensing nodes need to be authentic. A false or malicious node could intercept private information in the absence of proper security or could send false messages to nodes in the network. In this paper, we considered a dense WSN, used as a smart environmental monitoring system, where N low-complexity sensing nodes are distributed on an area $\mathcal{A}$ and communicate with a sink node. The considered IoT network is based on a classical star-topology network, where the sink node coordinates the sensor devices distributed around it (Figure 1). Sensor nodes are supposed to be devices with low-resource (i.e., computation, memory and energy), performing simple tasks that monitor some physical parameters (e.g., humidity, gas, water level, vibration, pressure, etc.) and transmitting them to the coordinator. Hence, sensor devices are equipped with a low-power microcontroller with an integrated radio transceiver equipped with a single antenna and a sensor interface. Differently, the coordinator is a higher-powerful device with mode complex functionalities. Indeed, the sink node has in charge the management of the access and communication in the network (e.g., access and resource management, authentication, channel estimation, etc.) and could also perform processing of received data. The coordinator is supposed to have more computing and memory resources and always be connected to a power source. The transceiver is equipped with multiple antennas so that the spatial information can be exploited in the network.

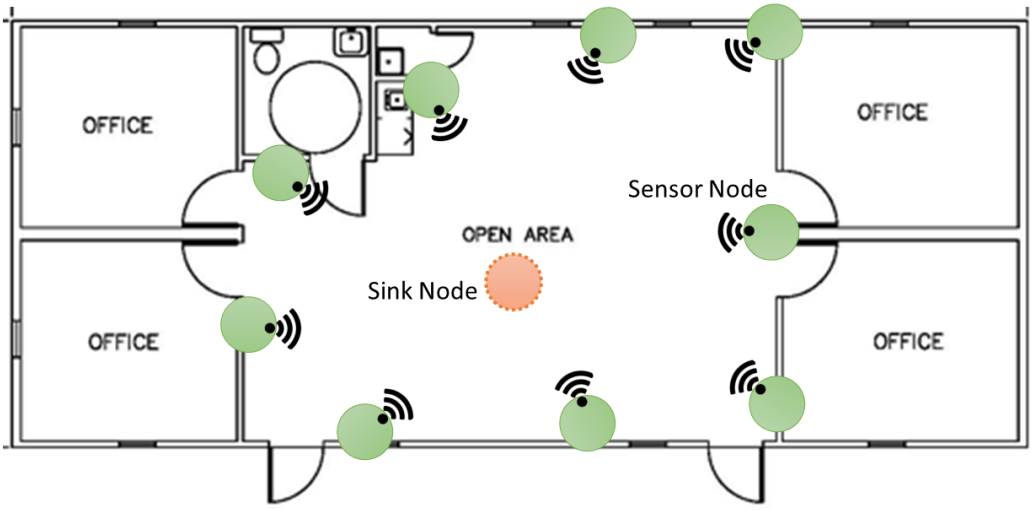

Figure 1. WSN with sink node responsible for the security of the system.

The proposed WF authentication method is based on PHY-channel features; hence, we have to resort to a suitable channel model. In particular, we consider the 802.11ac™ (TGac) multipath fading channel [34]. This is a system level model, which can describe an arbitrary number of propagation environment realizations for single or multiple radio links for all the defined scenarios, with one mathematical framework by different parameter sets. The TGac channel model follows a stochastic channel modeling approach, as the channel parameters are determined stochastically, based on statistical distributions extracted from channel measurements. This model is frequently used for indoor area wireless communication systems operating in a 5 GHz spectrum with a bandwidth up to 160 MHz. In this paper, we

selected the Model-D scenario [35] that represents the propagation conditions in a typical large indoor open environment, with mobility (0–5 km/h). More in detail, we assume that the transmitting/receiving nodes are in fixed positions, but we consider a certain time variability to take into account the scatterers' movement in the area.

The 802.11ac™ model represents a multiple input multiple output (MIMO) channel, with M transmitting and Q receiving antennas. However, we focus on the particular case with $M = 1$; hence, we focus on a SIMO (single input multiple output) system. Indeed, we consider low-complexity IoT sensor nodes, equipped with a single antenna. The multipath fading SIMO channel is modeled as a tapped delay line (TDL) with L taps (paths), and the channel matrix can be written as follows:

$$\mathbf{H}(t) = \sum_{l=1}^{L} \mathbf{H}_l(t)\delta(t - \tau_l) \tag{1}$$

where $\mathbf{H}_l(t)$ is the SIMO channel matrix of the l-th path, τ_l is the delay of the l-th path and $\delta(\dot{})$ is the delta function defined as $\delta(t) = 1\ if\ t = 0, \delta(t) = 0$, otherwise. Assuming that all paths are Rice distributed with mean power γ_l, the matrix $\mathbf{H}_l(t)$ can be separated into a fixed matrix $\mathbf{H}_l^F(t)$ representing the LOS (non variable) part, and a Rayleigh-distributed matrix $\mathbf{H}_l^V(t)$ which represents the NLOS (variable) part. The matrix $\mathbf{H}_l(t)$ can be thus written as follows:

$$\begin{aligned} \mathbf{H}_l(t) = \sqrt{\gamma_l}\left(\sqrt{\frac{\zeta}{\zeta+1}}\mathbf{H}_l^F(t) + \sqrt{\frac{1}{\zeta+1}}\mathbf{H}_l^V(t)\right) = \\ \sqrt{\gamma_l}\left(\sqrt{\frac{\zeta}{\zeta+1}}\begin{bmatrix} e^{j\phi_1(t)} \\ e^{j\phi_2(t)} \\ \vdots \\ e^{j\phi_Q(t)} \end{bmatrix} + \sqrt{\frac{1}{\zeta+1}}\begin{bmatrix} X_1(t) \\ X_2(t) \\ \vdots \\ X_Q(t) \end{bmatrix}\right) \end{aligned} \tag{2}$$

where

- $X_i(t)$ is the coefficient of the i-th receiving antenna in the NLOS condition. The X_i coefficients are correlated complex Gaussian random variables with zero mean and unitary variance;
- $\phi_i(t)$ is the phase difference between the transmitting and the i-th receiving antenna;
- ζ is the Ricean factor;
- γ_l is the mean power of the l-th path at the receiver.

Each tap $\mathbf{H}_l(t)$ is composed by a cluster of individual propagation rays so that the complex Gaussian assumption is valid.

The path loss model is a free space loss breakpoint model with two fixed slope values: a standard L_{FS} (slope of 2) up to the breakpoint distance and slope of 3.5 afterwards

$$\begin{cases} L(d) = L_{FS}(d), & \text{for} \quad d <= d_{BP} \\ L(d) = L_{FS}(d_{BP}) + 35\log_{10}(d/d_{BP}), & \text{for} \quad d > d_{BP} \end{cases} \tag{3}$$

where d is the distance [m] with $5 < d < 100$ and d_{BP} is the breakpoint distance [m].

In our proposed system, we are interested in several channel attributes, not only those related to the signal amplitude. Hence, we have integrated the TGa model with the WINNER II [36] model for what concerns the delays and the angle of arrival (AoA) information. In particular, since paths delays are fixed in the TGa model in every channel realization, we used the distribution proposed by WINNER II to model the paths delays. In the WINNER II model, each user has a delay profile randomly selected: the average delay of each path, τ_l^{avg}, is generated using an exponential distribution with parameter λ [36]. Moreover, to take into account the scatterers' movement in the surrounding environment as well as delays estimation errors, we introduced a certain variability of the delay values around their mean value, τ_l^{avg}. The delay of each path, τ_l, is derived from an uniform

distribution with mean τ_l^{avg} and variance $\sigma_\tau^2 = 1/\lambda$. For the same reasons and following a similar procedure also, AoA values are randomly distributed around their mean value. In particular, following the model in [36], AoA is normal distributed $\mathcal{N}(\mu, \sigma_{AoA}^2)$, where the mean value μ is chosen as the geometrical direction of the sink-node link and the variance is σ_{AoA}^2.

3. Proposed System

The proposed system is proposed as a means to enhance and integrate the higher level authentication, for identifying potential illegal nodes trying to transmit unauthorized data. The basic idea is that during the initial access procedure, each sensing node is authenticated by means of a high-level procedure, and a unique ID is assigned to each one. Consequently, the sink node has a list of N authorized nodes with their corresponding identification ID. Successively, a continuous PLA is performed during normal communication involving only the physical layer. In particular, the sink node verifies if the received message has been illegitimately modified/generated by a node other than its claimed source, exploiting the WF that provides an additional unique identifier of the radio link between two nodes. Therefore, even if the malicious node is able to intercept and use a valid ID, the WF identification allows to detect the intrusion, thanks to the spatial decorrelation of radio channels of the malicious and authorized node using the same ID.

The WF is obtained by the extracting some PHY attributes from the signal received by a specific device and, hence, by a specific propagation channel. In this paper, we considered the following PHY attributes:

- **AoA**: the direction of arrival of the signal at the sink node;
- **Maximum delay spread (MDS)**: the time interval needed to collect all paths of the signal;
- **Peak value**: the maximum value of the channel impulse response;
- **Energy**: the sum of the squared absolute value of the signal;
- **Received signal power (RSP): calculated as** the ratio between the **energy** and the **MDS**.

These attributes are used for the PLA of devices by means of a ML approach. In particular, we focus on a supervised learning multi-class classification approach:

- During the *training phase*, the ML algorithm is trained using N-labeled training sequences belonging to the N legitimate sensor devices. Each one is composed of X samples of the received signal. Hence, only data of the authorized nodes are used for training, since it is impractical to assume to know the dataset of the spoofing node.
- Then, during the *communication phase*, the received signal samples are classified as belonging to one of the N classes. However, in this way, even a malicious node is identified as a legitimate one, so an additional step is needed for its detection: the classification output is cross checked with the declared ID and if they match, the authentication is successful, while otherwise it fails. In the second case, the node communication is blocked and a new authentication at higher layers must be performed.

The proposed procedure is represented in Figure 2.

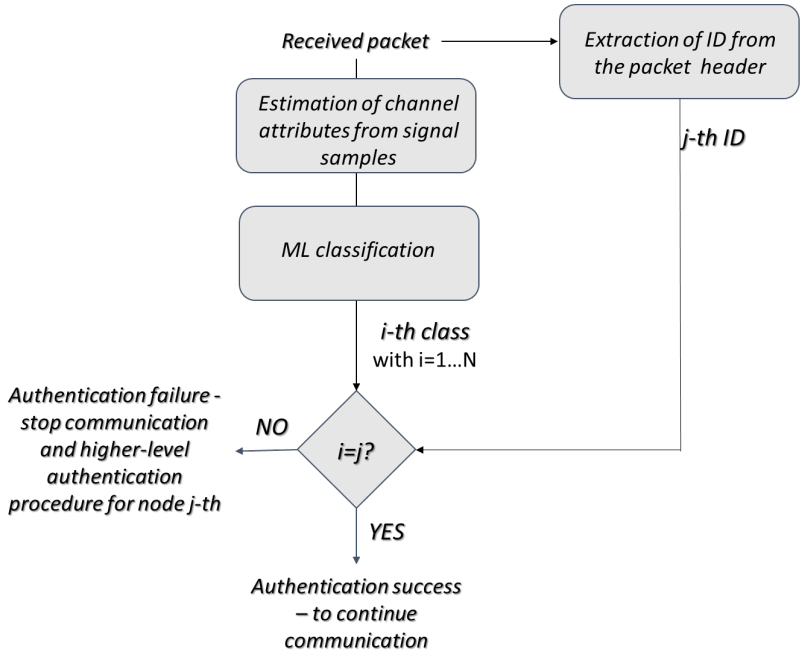

Figure 2. Flow diagram of the proposed approach.

More in detail, let us distinguish between two cases:

- *The spoofing node is not present*—the ML classification algorithm detects the class of the incoming authorized data and then cross checks the classification outcome with the declared ID: if the data belong to the node with claimed $ID = j$, the identification is successful if the ML classification result is j; otherwise, it fails and an alarm of spoofing is generated for the j-th node. Hence, the ML algorithm *accuracy* is defined as the probability of correctly identifying the class of an authorized user. At the opposite, if the ML classification fails, an authenticated user is erroneously blocked; hence, we define the *probability of blocking an authorized node* as $P_{ban} = 1 - Accuracy$.
- *The spoofing node is present*—if the transmission belongs to an authorized user, we fall into the previous case. If the transmission belongs to the spoofing node, the ML classification algorithm classifies it as an authorized node with $ID = i$ and $i = 1, \cdots, \mathcal{N}$. At this stage, the spoofing node cannot be detected; hence, the probability of detection of a spoofing node does not directly depend on the ML algorithm. The spoofing node can be detected only by cross checking its declared ID with the classification result since each class is labeled with a specific node ID. The probability that a unauthorized node is classified as authorized, named the *probability of miss spoofing detection*, P_{msd} is the probability that an unauthorized node claiming the i-th ID is classified as belonging to the i-th class.

The basic idea is that a spoofing node cannot know how the sink node will classify its signal; hence, even if it is able to steal a valid ID, likely this ID will not correspond to the classification output. This probability increases as the number of authorized nodes in the network increases.

We underline that P_{ban} directly derives from the ML algorithm. Indeed, denoting with $P(i,j)$ the probability that the predicted class is j when the true class is i (see the confusion matrix in Figure 3), we have that $P_{ban} = \sum_{i=1}^{N} \sum_{j=1 j \neq i}^{N} \frac{1}{N(N-1)} P(i,j) = 1 - \sum_{i=1}^{N} \frac{1}{N} P(i,i) = 1 - Accuracy$. Hence, the *accuracy* is the probability that a node is correctly classified within the class labeled with its ID.

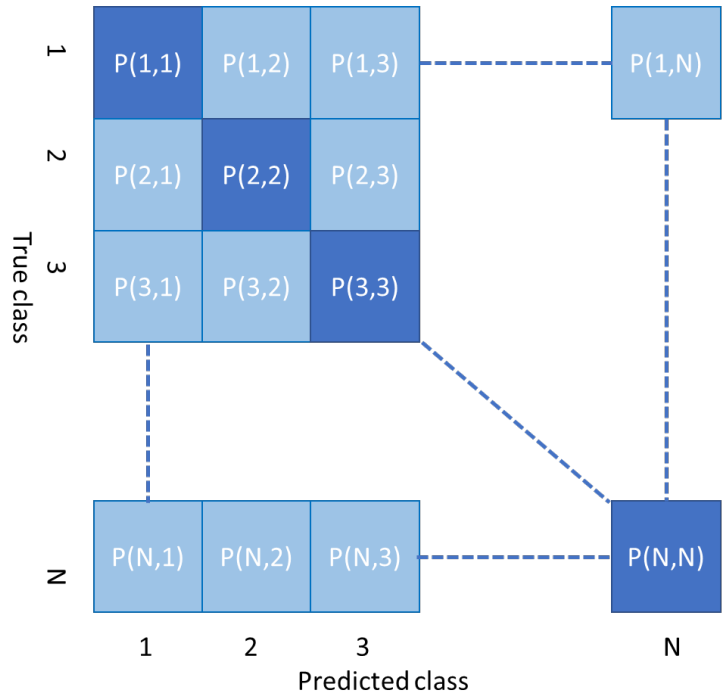

Figure 3. Confusion matrix.

Conversely, P_{msd} does not directly derive from the ML algorithm; indeed, it depends on the probability of selecting a given ID that decreases as N increases.

We want to stress that the proposed method represents an additional level of security (in addition to the first authentication step) especially for low-complexity nodes, where complex encryption algorithms cannot be executed. In particular, (i) high-level authentication can be only used to assign a unique ID to the node, then during communication the reliability of the received data is related to the outcome of the proposed method since encryption is not used, and (ii) high-level authentication provides both an unique ID and a secret key that can be used in successive encrypted communications. In this case, the proposed approach is an additional security level that avoids spoofing, even if the secret key is detected by the attacker, especially in the presence of low-robust encryption algorithms.

3.1. ML for Devices Classification

As stated before, sensor devices identification is performed by means of a ML approach that exploits multiple PHY-parameters of their unique propagation channel. In particular, we resort here to a non-parametric classification approach, so not depending on information from a certain sort of distribution that is difficult to achieve in dynamic environments. Moreover, this method is suitable for a low-cost/low-consumption WSN. In particular, two different algorithms are investigated. First, a CART algorithm is used [37]. This is a supervised ML algorithm that generates a decision tree in order to solve a classification or a regression problem. Because of their readability and simplicity, decision trees are among the most popular machine learning methods. In particular, the CART algorithm is well suited in the case of high-dimensional data; it contains the criteria for choosing the best attribute for the data splitting and assigning a class to the leaf. Then input data are classified based on their attributes through logical *"if–then"* statements.

More in detail, during the *training phase*, the CART algorithm builds the decision tree using a dataset containing samples of signals received by the N sensor nodes. Let us

assume that data are characterized by K attributes $\mathcal{A} = a_1, \cdots, a_K$, and let us consider the Shannon's entropy of a dataset $\mathcal{D}$, that is

$$\boldsymbol{H}(\mathcal{D}) = -\sum_{n=1}^{N} p_n \log_2 p_n \tag{4}$$

where $p_n = |\mathcal{D}(n)|/|\mathcal{D}|$ is the ratio between the number of elements of $\mathcal{D}$ belonging to the n-th class, $\mathcal{D}(n)$, and the total number of elements in $\mathcal{D}$ (i.e., the operator $|\cdot|$ represents the cardinality of the set). To build the decision tree, the CART algorithm at each step performs the split of a dataset $\mathcal{D}$ in two disjoint datasets $\mathcal{D}_{1/2}$, using the *information gain* as the metric to select the best attribute for the splitting. The *information gain* of the splitting of the dataset $\mathcal{D}$ based on the attribute a_i, $I_{gain}(\mathcal{D}, a_i)$ is defined as the difference between the entropy value of the original dataset, $H(\mathcal{D})$ and the sum of the entropy of the two subsets generated by performing the split based on the attribute a_i with $i = 1 \cdots, K$ as

$$I_{gain}(\mathcal{D}, a_i) = H(\mathcal{D}) - R(\mathcal{D}, a_i) \tag{5}$$

where $R(\mathcal{D}, a_i) = H(\mathcal{D}_1(a_i)) + H(\mathcal{D}_2(a_i))$ and $H(\mathcal{D}_{1/2}(a_i)$ is the entropy of the dataset $\mathcal{D}_{1/2}$ obtained using the attribute a_i. Hence, the best attribute $\hat{a}$ for performing the split is selected as

$$\hat{a} = \max_{a_1, \cdots, a_K} I_{gain}(\mathcal{D}, a_i) \tag{6}$$

The algorithm is iterative: initially, the whole training dataset is considered (tree root), and at the first step, this is split in two disjoint datasets (using the best attribute), then the two generated datasets are in turn split, each one into two datasets (using the best attribute for each split), and so on until one of the following conditions is reached:

1. The maximum number of split has been performed (it is set as a parameter);
2. One leaf is "pure", that is, all input data in the leaf belong to the same class;
3. One leaf contains only one input sample.

Fixing the maximum number of splits limits the dimension of the tree and, hence, the test complexity as detailed later. Moreover, having a tree with limited dimension avoids also overfitting problems that can arise, having leaves with a few sample data.

During the *classification* phase, the received signal samples are moved in the decision tree from the root down to the leaf that represents the most suitable class for those samples. In particular, input data are compared with the attribute selected at each node of the tree and moved in the corresponding branch.

The second algorithm that is considered is random forest [38,39], which is introduced to counteract the decision tree's overfitting tendency by reducing the data variance. This is an ensemble learning technique, which creates and aggregates multiple decision trees trained on different datasets, each one obtained from the initial dataset by random sampling it with the replacement (bootstrapping). The decision trees are created using the CART algorithm described before, but with a subset of the original attributes randomly selected. The dimension of the subset is the nearest integer of $\log_2(K+1)$ (where K is the total number of attributes) [38,39]. During the classification phase, the received signal samples are moved in the different decision trees, and the result is taken by the majority.

Algorithm Considerations

In this section, some issues on the applicability of the proposed method are discussed.

- **Suitable scenario**: The proposed approach is suitable for a scenario with a limited variability on the network topology, where nodes are distributed in an area on almost-fixed positions, for example, for monitoring purposes (e.g., surveillance, anti-intrusion, and environment monitoring). When a new node is added to the network, the set-up phase has to be run again, i.e., the learning must be performed again to add the new class. However, this urgency is not present if a node leaves the network (and its ID is

disabled). Indeed, in this case, the classification still works: if an attacker is classified as the disabled ID, it must be certainly blocked.

- **Complexity and Scalability** The complexity of the considered ML approaches must be evaluated separately for the two phases: training and test. During the training for each attribute (K), the information gain is calculated for the $M = NX$ elements of the dataset (with complexity $O(KM)$) and values are sorted to find the right splitting threshold. The complexity of the sorting operation is $O(KMlog_2M)$ that, asymptotically, is the complexity of the training phase. As the RF algorithm complexity must take into account the number of trees T, the complexity is $O(T\log_2(K+1)Mlog_2M)$. In our system, the number of attributes is $K = 5$, and, as shown in the numerical results section, both CART and RF need short training sequences, thus resulting in fast and limited complexity training. Obviously, the complexity increases as $Nlog_2(NX)$, as the number of nodes, N, increases. On the other side, the testing phase complexity is proportional to the tree depth P that depends on the number of splits that must be at least equal to N. In the numerical results section, we verified that selecting a number of splits slightly higher than N provides a slight improvement in the accuracy, but a further increase does not provide advantages. For simplicity, assuming that the number of split is N, in the best case (totally balanced-tree is $P = log_2N$) and in the worst case is $P = N$. Hence, in the classification (test) phase, the algorithm complexity in the worst case is linear with N, thus, scaling efficiently with N. Indeed, this aspect makes the decision tree algorithms very fast and resource efficient during the test stage, and hence, suitable even for real-time machine learning deployment and large scenarios.
  In terms of performance increasing the number of nodes in the area, we can expect two opposite behaviors, indeed, as explained before, the spoofing detection capability improves if N increases, but on the other side, the P_{ban} can increase due to a reduction of the accuracy of the classification since nodes are closer to each other and it is more difficult to discriminate them. However, in the numerical results section, we verified that the performance degradation is not significant within a certain value; we tested node density up to around 50,000 nodes/km^2. Obviously, the number of needed splits of the trees increases.

3.2. Sentinel Nodes

The classification algorithm allows to associate each received signal to one of the possible WF classes that are labeled with the authorized nodes' ID. When a malicious node wants to access the network, supposing it is able to steal the ID to one of the nodes, it sends its message with the associated ID. The sink node classifies the node as stated before and then cross checks the classification result and the claimed ID. Being that the malicious user is classified as one of the authorized users, the spoofing detection fails when the WF class and ID match. Assuming, for example, that the unauthorized user randomly selects one of the possible IDs, this occurs with probability $1/N$. This means that in dense WSNs (i.e, when N is large) the probability of selecting the ID of the class resulting from the classification algorithm is very low, but it increases in small networks. For this reason, we propose to use some simple cooperative nodes named *sentinel nodes* that allow to increase the classification space, thus increasing the detection. Sentinel nodes periodically send a beaconing signal, and thus are classified as an additional authorized source. In this way, the number of WF classes increases, and the previous probability is reduced as $1/(N + N_S)$, where N_S is the number of sentinel nodes. Using cooperative nodes is already proposed in the literature, such as, for example in ref. [23], where the additional nodes estimate the RSSI of the authorized communication link and forward this information to the sink node for enhanced detection. Here, cooperative nodes are simpler and do not perform any action. These simply periodically send a beaconing signal. This is more suitable for a large deployment and for low-cost and low-complexity WSNs.

4. Numerical Results

This section presents the numerical results of the proposed authentication/spoofing detection method derived by means of simulations using the Matlab environment. An area $\mathcal{A} = 30 \times 30$ m representing a large indoor hall with the sink node positioned in the center is considered. The number of connected nodes is $N = 15$ if not differently indicated. The channel attributes are characterized stochastically as described in Section 2, taking into account also their time variability due to the scatterers' movement. This allows to analyze different scenarios, as detailed later, and the capability of the proposed scheme to follow the attributes' variations. As specified in each scenario, nodes are randomly placed in the considered area with a uniform distribution or following a cluster distribution. Moreover, for what concerns the spoofing detection capability of the system, this is evaluated by averaging the value P_{msd} over different positions of the spoofing node in the area as specified later.

4.1. Probability of Blocking an Authorized Node

First of all, we are interested in evaluating the false spoofing detection capability of the system. It is related to the accuracy (i.e., the capability of the classification method of correctly classifying the authorized nodes) of the classification method as $P_{ban} = 1 - Accuracy$. Indeed, if the classification is not correct, an authorized node is erroneously associated to a different class, and the ID check fails. In the basic scenario, we refer to the model channel parameters described before: the scatters' speed is in the range [0–5] km/h, $\sigma_\tau = 1/\lambda$ with $\lambda = 1.664 \cdot 10^7$ and $\sigma_{AoA} = 1.5849$ [36]. However, in order to test the effectiveness of the classification also under more challenging conditions, we have also considered the following different scenarios:

- *Scenario A1*—nodes are randomly placed in $\mathcal{A}$ according to a bidimensional probability distribution. The Doppler spread is related to a scatterers' movement in the range [0–5] km/h;
- *Scenario A2*—nodes are randomly placed as in A1, but scatterers' speeds are increased in the range [0–15] km/h;
- *Scenario A3*—nodes and speeds are set as in A2, but also the angle and delay spread are increased, considering a variance that is three times the original one;
- *Scenario B1*—nodes are placed in clusters as shown in Figure 4, and the signals experience the Doppler effect under the same conditions as case A1;
- *Scenario B2*—clustered nodes are paired with the same environmental conditions of case A3.

Different datasets were created for each scenario to train the machine. In particular, for each node, 5000 impulse responses were sampled and of those, the first 100 were used as the training dataset, while the rest of them were used to evaluate the performance of the classifier. As shown in Figure 5 for the CART algorithm (similar results were derived also for the random forest algorithm), the value of 100 for the training sequence length was selected because an increase does not provide a noticeable performance improvement. Moreover, until the length of 80 (it is more evident with very short lengths, 5/10) we can note an overfitting effect due to the fact that with a few data, the training is too fitted on these; hence, there is a consequent significant loss of performance after training.

First of all, we evaluated the performance of the CART algorithm, varying the number of splits in the range [20–60]. We saw that in basic scenarios, there is not a high variance of the achieved values with the number of splits. Differently, when the Doppler, variance of angle and delay spread increase, a higher number of splits is beneficial. In general, 20 splits is a good trade off. Table 1 reports the maximum and minimum values of the *classification accuracy* for different scenarios and the number of splits for which these values are reached.

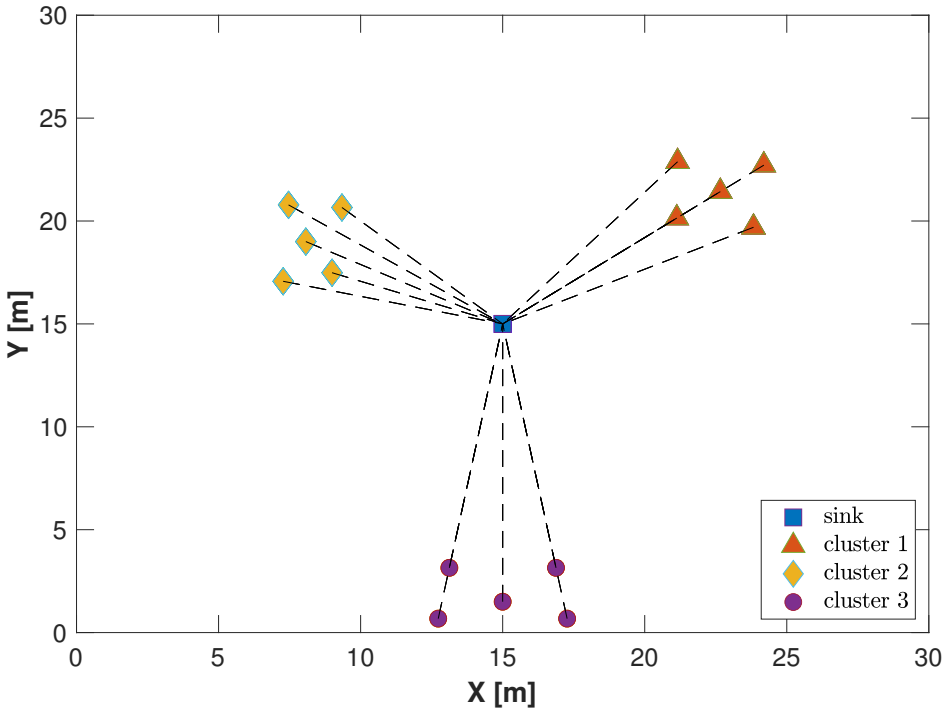

Figure 4. Example of the nodes position in a clustered scenario.

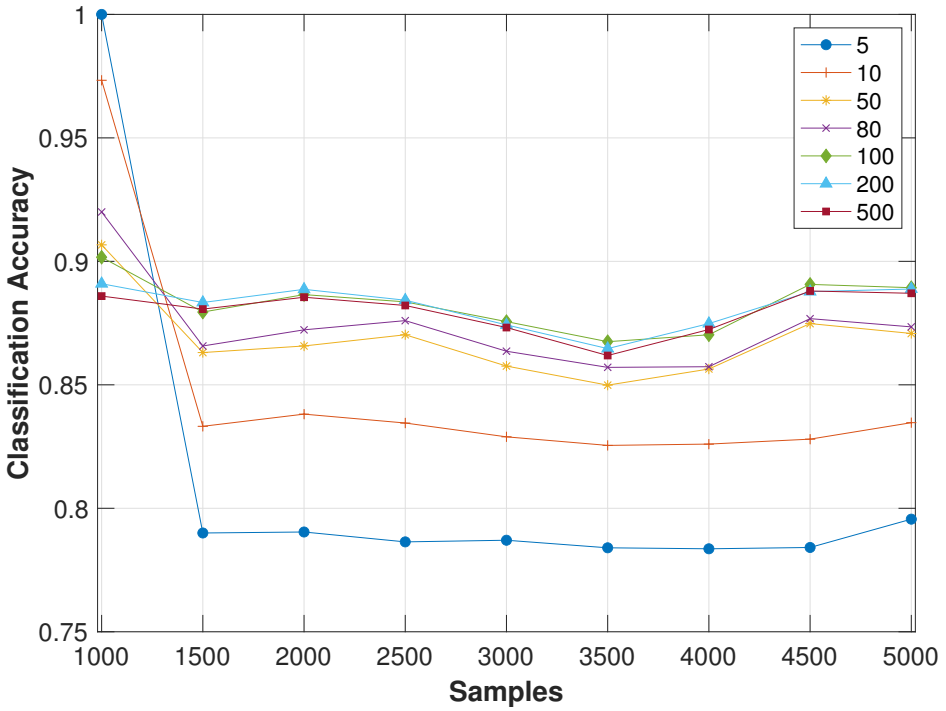

Figure 5. CART classification accuracy for different training sequence lengths in A1 scenario.

Table 1. Accuracy variation vs. number of CART splits.

Scenario	Min (n.Splits)	Max (n.Splits)
A1	95.28% (15)	95.84% (20)
A2	95.00% (15)	96.04% (20)
A3	91.02% (15)	96.07% (25)
B1	88.19% (15)	95.62% (25)
B2	86.04% (15)	89.89% (40)

The following results were derived, assuming a CART classifier with a maximum of 20 splits and a random forest classifier composed of five trees of the same size as the CART one. Figure 6 shows the accuracy of the two classifiers on the A1 dataset for the whole sample sequence of 5000 samples (graphs starts at the 100-th sample because the first 100 samples are used for training). Both classifiers show a good stability with little loss of accuracy over time, with an average percentage of correct classification around 95% (i.e., in average $P_{fsd} = 5\%$). There are not significant differences in the performance of CART and random forest algorithms; the gain of random forest is substantially negligible.

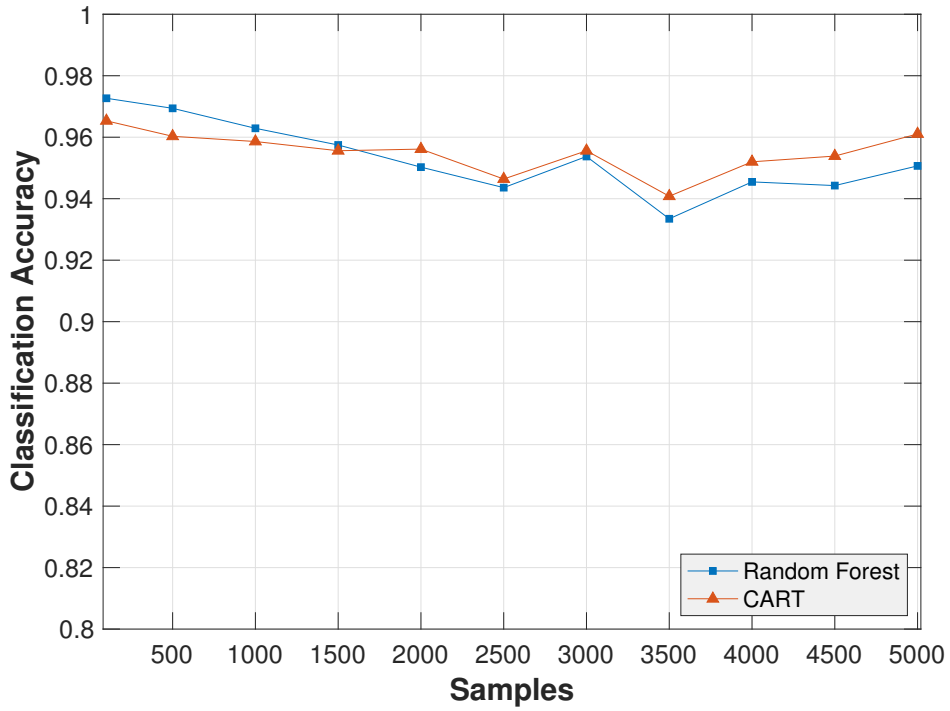

Figure 6. Classification accuracy for scenario A1.

The accuracy averaged on the whole dataset for all scenarios is reported in Table 2.

Table 2. Average classifiers accuracy for different scenarios.

Scenario	CART Accuracy	Random Forest Accuracy
A1	95.43%	95.49%
A2	95.95%	96.14%
A3	96.53%	96.66%
B1	93.91%	95.04%
B2	87.96%	89.74%

Results in Table 2 show that only in scenario B2 is there a noticeable reduction in the accuracy, which goes down around 88–89%. We evaluated also the effect of the number of nodes in the area. In scenarios A1–3, we varied the number of nodes in the range $[15, 30]$

and we saw that there is not a performance degradation in terms of accuracy, but obviously the number of splits must be increased. In particular, for up to 25 nodes, the sufficient number of splits is N increased by the 20%/25%. At 30 nodes instead, with 40 splits, the accuracy decreases down to 79%, and 65 splits are needed to reach the 95% value.

In order to further investigate this issue, we considered also different clusters' distributions. We noted that there is not a significant difference if the clusters' position varies, but the number of nodes/clusters is the same. Similarly, leaving unchanged the number of nodes per cluster and increasing the number of clusters up to 10/12 (which corresponds to more than 60,000/70,000 nodes/km^2), performance is not significantly affected because the number of nodes that can create confusion in the classification process (since AoA and delay attributes are very similar within a cluster) is the same. Obviously, with a higher density of clusters, it is more likely that cluster overlapping occurs (being that the clusters are randomly placed). Thus, a lower number of clusters with a higher number of nodes occurs, and the overall accuracy decreases. We investigated also the case of a higher number of clusters and nodes per cluster as well as the extreme case, where all nodes belong to the same cluster. In the first case, considering 6 clusters with 7 nodes per cluster, there is only a slight reduction in the accuracy, due to the presence of more nodes within the cluster that have similar attributes: the average value of the CART algorithm in scenario B1 is 91.5%, while in the extreme case of a single cluster with 15 nodes, the performance worsens and average accuracy of CART is 88%. In general, up to a certain nodes/clusters density, the reduction in accuracy is limited, but obviously when the density significantly increases, there is a reduction in the accuracy due to the high probability that different nodes have similar attributes.

Since the proposed method is based on multiple attributes, it is interesting to evaluate, as these impact the accuracy of the classification. For this reason, the classifiers were used with different sets of attributes, particularly the following:

- The whole set;
- The whole set without AoA attribute;
- The whole set without delay attribute;
- Only attributes related to the signal intensity (i.e., RSP, peak value and energy) without AoA and delay;
- Only AoA and delay attributes.

Table 3 reports the accuracy averaged over the whole dataset for different scenarios. The results show that the classifier using all the attributes outperforms others, using only a subset; in particular, the information provided by AoAs and delays improves drastically the prediction accuracy when compared to a classifier that relies only on "energy-based" attributes.

Table 3. Average classifiers accuracy for different scenarios.

Scenario	Full	No AoA	No Delay	"Only Energy"	AoA & Delay
CART					
A1	95.43%	90.64%	94.67%	55.88%	90.70%
A2	95.95%	89.92%	94.87%	55.94%	90.80%
A3	96.53%	83.78%	81.08%	52.11%	95.40%
B1	93.91%	75.32%	76.96%	51.43%	91.69%
B2	87.96%	68.76%	77.44%	51.82%	81.84%
Random Forest					
A1	95.49%	88.77%	93.27%	61.61%	92.56%
A2	96.14%	89.26%	94.43%	62.93%	92.88%
A3	96.66%	89.35%	80.70%	65.56%	95.45%
B1	95.04%	86.01%	76.61%	57.23%	92.60%
B2	89.74%	79.84%	72.22%	61.89%	83.39%

The P_{ban} depends on the ML classification algorithm; hence, we compared the results of CART and random forest with those of other two basic ML classification methods, SVM and k-NN, to show the effectiveness of the selected ones. In Table 4, the classification accuracy of the four methods is reported for scenario A1. We considered a linear kernel for the SVN and $k = 20$ for the k-NN (different values of k were tested).

Table 4. Comparison of classifiers accuracy.

CART	RF	SVM	k-NN
95.43%	95.49	95.59%	94.59%

The results show that in this scenario, accuracy is similar using different classification algorithms, thus supporting the effectiveness of the selected ones. Moreover, these present low complexity and less degrees of freedom, which can affect their performance. Indeed, k-NN is usually a low-complexity approach, but its performance requires a suitable selection of k that should be differently optimized for different scenarios; moreover, the computation load increases with k. SVM instead requires a large amount of time to process; hence, it is suitable only if the data size is small and provides poor performance with overlapped classes (it can happen with proximity nodes). Finally, performance strongly depends on the hyperparameter settings.

4.2. Probability of Missed Spoofing Detection

The second performance indicator is the mis-detection of an unauthorized user access, that is P_{msd}. We want to verify what happens when a spoofing node is present. This node can be in any position; hence, first of all, we want to verify if there is a relation between the spoofing node position and its classification. As an example, Figure 7 shows a scenario with $N = 8$ authorized nodes, whose positions are indicated with the red triangles, and each one is identified by a different color (i.e., each color corresponds to a different class). The sink node is considered in the center of the area, even if not represented. The area $\mathcal{A}$ is divided in 10×10 squares, and the malicious node classification is performed placing the malicious user in the center of each square not occupied by an authorized node. The color of the square indicates the output of the classification (i.e., the unauthorized node in each specific position is classified as the authorized node that has the same color). We can see that even if there is a certain spatial correlation, the classification of a malicious user in different positions is quite mixed in the area.

Since the attacker tries to embody another node by transmitting a packet to the sink with the label of the node whose identity it is trying to spoof, we consider two different cases.

First, we assume that the malicious node randomly selects one of the available node IDs in the network (with probability $1/N$). Hence, P_{msd} goes as $\frac{1}{N}$; indeed, given the classification results, the probability of selecting the ID that matches with the resulting class is $1/N$. This is shown in Figure 8. These results were derived averaging the P_{msd} over all the possible positions of the malicious user. Obviously if the number of nodes is low, $1/N$ is high, and hence, the P_{msd} is high. To overcome this problem in small networks, *sentinel nodes* can be introduced, each one with its assigned ID. For example, adding $N_S = 10$ sentinel nodes, the P_{mds} is significantly reduced as shown in Figure 8. Sentinel nodes are randomly placed in the area.

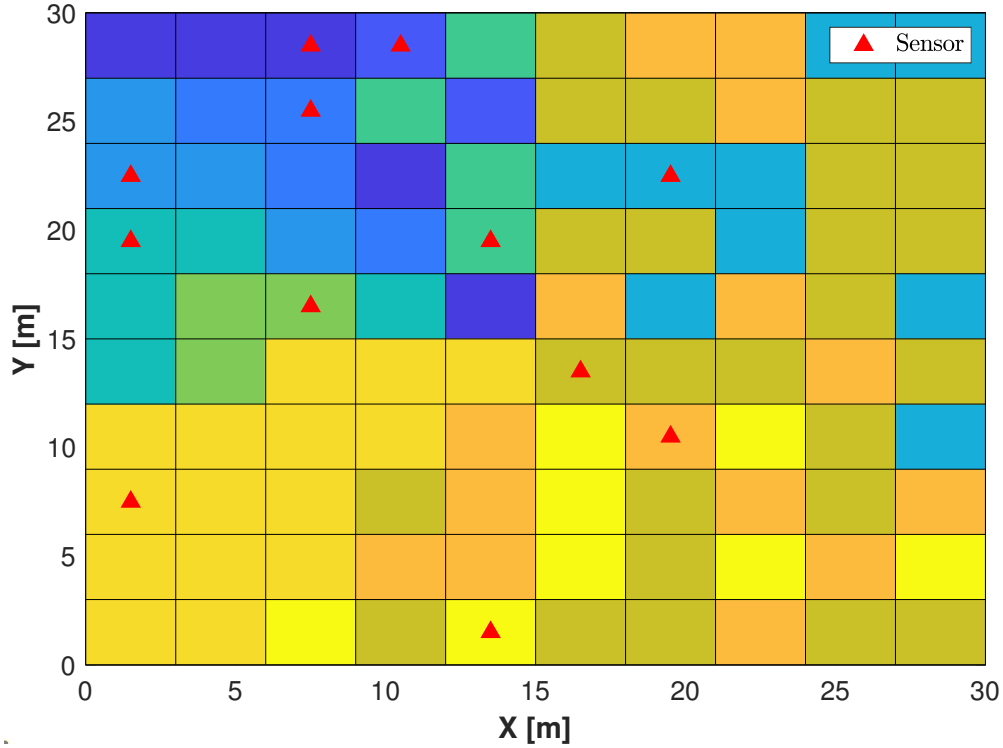

Figure 7. Example of unauthorized node classification depending on its position in the area.

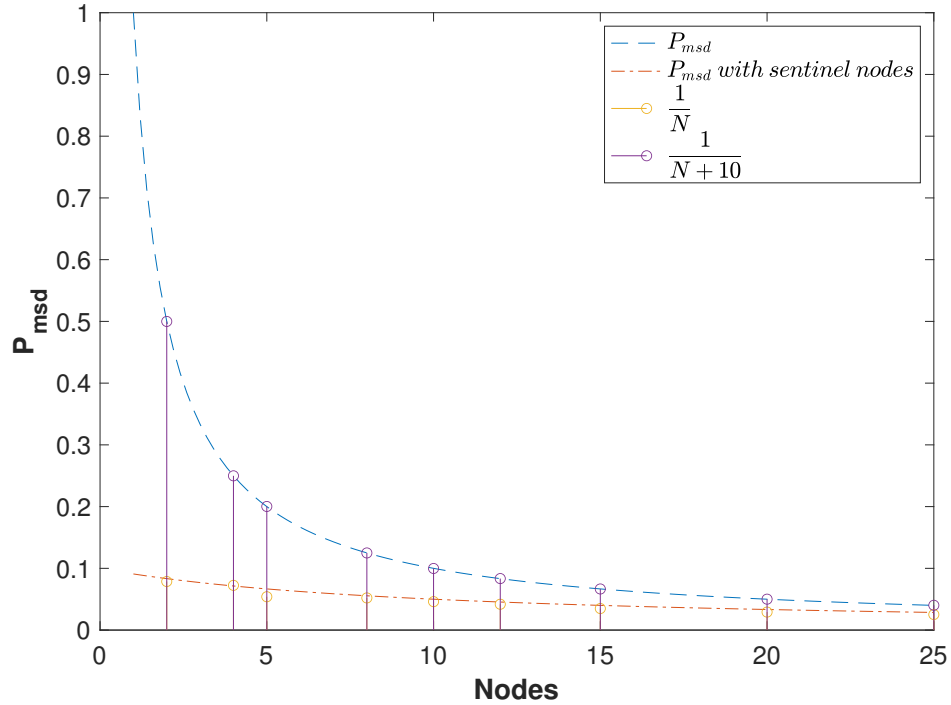

Figure 8. P_{msd} vs. number of authorized nodes in the area, with and without sentinel nodes ($N_S = 10$).

As a second scenario, we considered the worst case in which the malicious node tries to impersonate the nearest authorized node (i.e., it is able to intercept its ID). Results are shown in Figure 9 in the case without sentinel nodes. We can see that performance slightly worsens, due to the spatial correlation of the classification results shown in Figure 7, but it is still close to the $1/N$ curve because the spatial correlation is not so high.

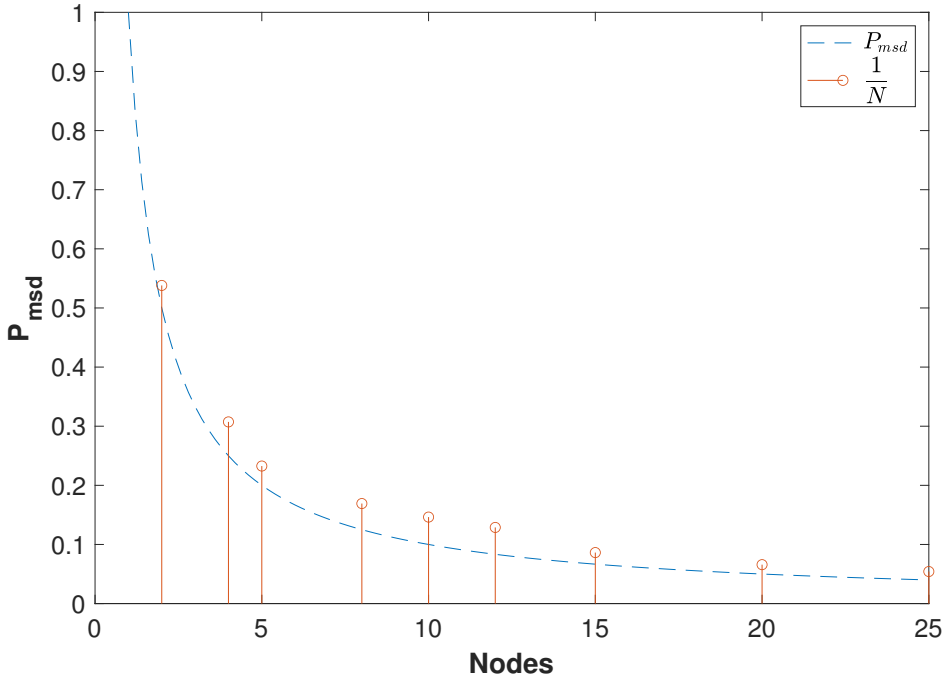

Figure 9. P_{msd} vs. number of authorized nodes in the area when the malicious user selects the ID of the nearest sensor.

4.3. Limits of the Proposed Solution and Future Works

The proposed PLA scheme can be used in IoT scenarios with low environment mobility to enhance the authorization/identification in a network, especially when nodes have low computational capabilities and are not able to perform complex encryption algorithms. It was proven that this approach is able to correctly classify and authorize nodes with a high accuracy, even in the presence of challenging channel attributes variability; however, in the considered scenario, the nodes' position is assumed to be fixed, and hence, the mean values of delay and AoA do not change, but in a high mobility scenario, this could be not possible, thus reducing the ML classification accuracy. Different approaches should be considered in this case for classification.

Moreover, the spoofing detection capability is achieved, thanks to the use of the node ID, and increases as the number of authorized nodes increases. This is suitable for a future scenario where a massive number of machines will require access to the network; however, in the case of small networks, the number of nodes is a limit that can be overcome with the introduction of sentinel nodes as we proposed. An alternative solution could be using different ML algorithms that, even if trained on N datasets, are able to detect $(N+1)$ classes, where the $(N+1)$-th class is the one of an unauthorized node. Toward this goal, algorithms must be suitably selected and modified in order to work in a multi-class environment. These solutions could not only make the spoofing detection probability independent on the number of nodes in the network, but also avoids the use of the node ID.

These aspects are currently under investigation for a future extension of this work.

5. Conclusions

This paper presented a PHY-layer continuous authentication and spoofing detection scheme based on wireless fingerprinting for an actual wireless sensor network, where several nodes communicate with a central sink node. The identity of authorized nodes is confirmed, verifying the correspondence of specific attributes of the wireless link with previous transmissions of the same nodes. A machine learning (ML) approach is used for classifying the authorized users, so that the capability of analyzing multi-dimensional information without the need of an analytical model is exploited. In particular, the framework

proposed in the paper is based on two ML approaches based on the decision tree. Moreover, the attack of a malicious node can be revealed by performing a cross-check of the classification result and the declared ID. Numerical results show that, even in challenging scenarios, the considered algorithms are able to reach high levels of accuracy in the classification that corresponds to a correct identification of an authorized user. Similarly, the system presents good performance in terms of spoofing detection, especially in large networks as foreseen by the future IoT application scenarios. However, even in small networks, good protection can be achieved by adding simple sentinel nodes that periodically send beaconing signals containing their ID.

Author Contributions: Conceptualization, D.M. and L.M.; methodology, D.M., L.M. and A.S.; software, A.S.; validation, D.M. and L.M.; formal analysis, D.M., L.M. and A.S.; investigation, A.S.; resources, D.M. and L.M.; data curation, A.S.; writing—original draft preparation, D.M. and A.S.; writing—review and editing, D.M. and L.M.; visualization, A.S.; supervision, D.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the European Telecommunication Standard Institute (ETSI) technical committee (TC) on Smart Body Area Network (SmartBAN), and in part by the European Union's Horizon 2020 Research and Innovation Program under Grant 872752.

Data Availability Statement: Not Applicable, the study does not report any data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fang, H.; Wang, X.; Tomasin, S. Machine Learning for Intelligent Authentication in 5G and Beyond Wireless Networks. *IEEE Wirel. Commun.* **2019**, *26*, 55–61. [CrossRef]
2. Xiao, L.; Wan, X.; Lu, X.; Zhang, Y.; Wu, D. IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security? *IEEE Signal Process. Mag.* **2018**, *35*, 41–49. [CrossRef]
3. Wang, N.; Wang, P.; Alipour-Fanid, A.; Jiao, L.; Zeng, K. Physical-Layer Security of 5G Wireless Networks for IoT: Challenges and Opportunities. *IEEE Internet Things J.* **2019**, *6*, 8169–8181. [CrossRef]
4. Restuccia, F.; D'Oro, S.; Melodia, T. Securing the Internet of Things in the Age of Machine Learning and Software-Defined Networking. *IEEE Internet Things J.* **2018**, *5*, 4829–4842. [CrossRef]
5. Liu, Y.; Chen, H.; Wang, L. Physical Layer Security for Next Generation Wireless Networks: Theories, Technologies, and Challenges. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 347–376. [CrossRef]
6. Mukherjee, A. Physical-Layer Security in the Internet of Things: Sensing and Communication Confidentiality Under Resource Constraints. *Proc. IEEE* **2015**, *103*, 1747–1761. [CrossRef]
7. Shiu, Y.S.; Chang, S.Y.; Wu, H.C.; Huang, S.C.H.; Chen, H.H. Physical layer security in wireless networks: A tutorial. *IEEE Wirel. Commun.* **2011**, *18*, 66–74. [CrossRef]
8. Ometov, A.; Petrov, V.; Bezzateev, S.; Andreev, S.; Koucheryavy, Y.; Gerla, M. Challenges of Multi-Factor Authentication for Securing Advanced IoT Applications. *IEEE Netw.* **2019**, *33*, 82–88. [CrossRef]
9. Trappe, W. The challenges facing physical layer security. *IEEE Commun. Mag.* **2015**, *53*, 16–20. [CrossRef]
10. Mucchi, L.; Nizzi, F.; Pecorella, T.; Fantacci, R.; Esposito, F. Benefits of Physical Layer Security to Cryptography: Tradeoff and Applications. In Proceedings of the 2019 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Sochi, Russia, 3–6 June 2019; pp. 1–3. [CrossRef]
11. Tsitroulis, A.; Lampoudis, D.; Tsekleves, E. Exposing WPA2 Security Protocol Vulnerabilities. *Int. J. Inf. Comput. Secur.* **2014**, *6*, 93–107. [CrossRef]
12. Tomasin, S. Analysis of Channel-Based User Authentication by Key-Less and Key-Based Approaches. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 5700–5712. [CrossRef]
13. Wu, X.; Yang, Z.; Ling, C.; Xia, X. Artificial-Noise-Aided Physical Layer Phase Challenge-Response Authentication for Practical OFDM Transmission. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 6611–6625. [CrossRef]
14. Taha, H.; Alsusa, E. Secret Key Exchange and Authentication via Randomized Spatial Modulation and Phase Shifting. *IEEE Trans. Veh. Technol.* **2018**, *67*, 2165–2177. [CrossRef]
15. Yu, P.L.; Sadler, B.M. MIMO Authentication via Deliberate Fingerprinting at the Physical Layer. *IEEE Trans. Inf. Forensics Secur.* **2011**, *6*, 606–615. [CrossRef]
16. Liu, J.; Wang, X. Physical Layer Authentication Enhancement Using Two-Dimensional Channel Quantization. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 4171–4182. [CrossRef]
17. Xiao, L.; Greenstein, L.J.; Mandayam, N.B.; Trappe, W. Channel-based spoofing detection in frequency-selective rayleigh channels. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 5948–5956. [CrossRef]

18. Xiao, L.; Greenstein, L.J.; Mandayam, N.B.; Trappe, W. Using the physical layer for wireless authentication in time-variant channels. *IEEE Trans. Wirel. Commun.* **2008**, *7*, 2571–2579. [CrossRef]
19. Hou, W.; Wang, X.; Chouinard, J.; Refaey, A. Physical Layer Authentication for Mobile Systems with Time-Varying Carrier Frequency Offsets. *IEEE Trans. Commun.* **2014**, *62*, 1658–1667. [CrossRef]
20. Liu, F.J.; Wang, X.; Tang, H. Robust physical layer authentication using inherent properties of channel impulse response. In Proceedings of the 2011-MILCOM 2011 Military Communications Conference, Baltimore, MD, USA, 7–10 November 2011; pp. 538–542. [CrossRef]
21. Xiao, L.; Li, Y.; Han, G.; Liu, G.; Zhuang, W. PHY-Layer Spoofing Detection With Reinforcement Learning in Wireless Networks. *IEEE Trans. Veh. Technol.* **2016**, *65*, 10037–10047. [CrossRef]
22. Wang, N.; Jiang, T.; Lv, S.; Xiao, L. Physical-Layer Authentication Based on Extreme Learning Machine. *IEEE Commun. Lett.* **2017**, *21*, 1557–1560. [CrossRef]
23. Xiao, L.; Wan, X.; Han, Z. PHY-Layer Authentication With Multiple Landmarks With Reduced Overhead. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 1676–1687. [CrossRef]
24. Fang, H.; Wang, X.; Hanzo, L. Learning-Aided Physical Layer Authentication as an Intelligent Process. *IEEE Trans. Commun.* **2019**, *67*, 2260–2273. [CrossRef]
25. Wang, Q.; Li, H.; Zhao, D.; Chen, Z.; Ye, S.; Cai, J. Deep Neural Networks for CSI-Based Authentication. *IEEE Access* **2019**, *7*, 123026–123034. [CrossRef]
26. Senigagliesi, L.; Baldi, M.; Gambi, E. Comparison of Statistical and Machine Learning Techniques for Physical Layer Authentication. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 1506–1521. [CrossRef]
27. Yoon, J.; Lee, Y.; Hwang, E. Machine Learning-based Physical Layer Authentication using Neighborhood Component Analysis in MIMO Wireless Communications. In Proceedings of the 2019 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 16–18 October 2019; pp. 63–65. [CrossRef]
28. Hoang, T.M.; Nguyen, N.M.; Duong, T.Q. Detection of Eavesdropping Attack in UAV-Aided Wireless Systems: Unsupervised Learning With One-Class SVM and K-Means Clustering. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 139–142. [CrossRef]
29. Adamsky, F.; Retunskaia, T.; Schiffner, S.; Köbel, C.; Engel, T. WLAN Device Fingerprinting using Channel State Information (CSI). In Proceedings of the WiSec '18: Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks, Stockholm, Sweden, 18–20 June 2018; pp. 277–278. [CrossRef]
30. Tugnait, J.K. Wireless User Authentication via Comparison of Power Spectral Densities. *IEEE J. Sel. Areas Commun.* **2013**, *31*, 1791–1802. [CrossRef]
31. Liu, F.J.; Wang, X.; Primak, S.L. A two dimensional quantization algorithm for CIR-based physical layer authentication. In Proceedings of the 2013 IEEE International Conference on Communications (ICC), Budapest, Hungary, 9–13 June 2013; pp. 4724–4728. [CrossRef]
32. Xu, D.; Ren, P.; Ritcey, J.A. Independence-Checking Coding for OFDM Channel Training Authentication: Protocol Design, Security, Stability, and Tradeoff Analysis. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 387–402. [CrossRef]
33. Abdelaziz, A.; Burton, R.; Barickman, F.; Martin, J.; Weston, J.; Koksal, C.E. Enhanced Authentication Based on Angle of Signal Arrivals. *IEEE Trans. Veh. Technol.* **2019**, *68*, 4602–4614. [CrossRef]
34. Kermoal, J.; Schumacher, L.; Pedersen, K.; Mogensen, P.; Frederiksen, F. A stochastic MIMO radio channel model with experimental validation. *IEEE J. Sel. Areas Commun.* **2002**, *20*, 1211–1226. [CrossRef]
35. Erceg, V.; Schumacher, L.; Kyritsi, P.; Molisch, A.; Baum, D.S.; Gorokhov, A.Y.; Oestges, C.; Li, Q.; Yu, K.; Tal, K.N.; et al. Wireless LANs Indoor MIMO WLANTGn Channel Models. 2004. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.465.9926&rep=rep1&type=pdf (accessed on 20 January 2022).
36. Kyösti, P.; Meinilä, J.; Hentila, L.; Zhao, X.; Jämsä, T.; Schneider, C.; Narandzic, M.; Milojević, M.; Hong, A.; Ylitalo, J.; et al. WINNER II Channel Models. *IST-4-027756 WINNER II D1.1.2 V1.2.* 2008. Available online: http://www.ero.dk/93F2FC5C-0C4B-4E44-8931-00A5B05A331B?frames=no& (accessed on 20 January 2022).
37. Breiman, L.; Friedman, J.; Stone, C.J.; Olshen, R.A. *Classification and Regression Trees*; CRC Press: Boca Raton, FL, USA, 1984.
38. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
39. Ho, T.K. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 14–16 August 1995; Volume 1, pp. 278–282. [CrossRef]