UNIVERSITÀ
DEGLI STUDI
FIRENZE

Università degli Studi di Firenze
Dipartimento di Ingegneria dell'Informazione (DINFO)
Corso di Dottorato in Ingegneria dell'Informazione
Curriculum: Telecommunications and Telematics

Machine Learning based approaches for safe and secure communications at the Physical Layer

Candidate
Andrea Stomaci

Supervisors
Prof. Dania Marabissi

Prof. Lorenzo Mucchi

PhD Coordinator
Prof. Fabio Schoen

ciclo xxxvi, 2020-2023

Università degli Studi di Firenze, Dipartimento di Ingegneria
dell'Informazione (DINFO).

Thesis submitted in partial fulfillment of the requirements for the
degree of Doctor of Philosophy in Information Engineering.

Alla mia famiglia

Acknowledgments

I would like to acknowledge the efforts and input of my supervisors, Prof.Dania Marabissi and Prof. Lorenzo Mucchi, and all my colleagues of the Data Communication Networks System Lab (DaCoNetS) who were of great help during my research. I would like to thank also Prof. Hideki Ochiai and all the members of Ochiai Lab for their kind support and hospitality during my stay at Yokohama National University.

Contents

Part I

Physical Layer Authentication based on Machine Learning

Chapter 1

Introduction

Our society and the environment where we live are increasingly digitalized and data-driven. The demand for mobile data capacity is continuously increasing and future wireless systems are expected to support a wide variety of services that span from low data rate machine-to-machine machine-to-machine (M2M)-type communications to enhanced broadband, in extremely different application scenarios. In particular, the advent of the Internet of Things (IoT) will enable the interaction and interconnection of smart objects providing a wide range of emerging services and applications that span every aspect of our lives from health, to utilities, transportation, smart cities, and others [86]. 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 addressing the new security challenges and threats that arise. This is particularly critical for some applications such as e-health or remotely controlled cars [67]. Unauthorized access or control over these systems can have dire consequences, making authentication a vital component for ensuring the safety and reliability of these applications. Indeed, with the diffusion of IoT systems, a massive amount of confidential and sensitive data is transmitted in the wireless channel 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, jamming, that are in general dynamic and difficult to predict. Consequently, an efficient *spoofing detection* system able to distinguish between legiti-

mate and rogue devices is needed. Authentication mechanisms provide a robust layer of protection against unauthorized data access or tampering, safeguarding both privacy and data integrity. Moreover, smart and safe IoT networks require efficient IoT-device unique identification methods, since this greatly impacts security, reliability, robustness, and network real-time management. Many network tasks, such as providing personalized services, setting quality of service (QoS) parameters, and determining suitable resource allocation policies, depend on the type of connected device. Employing *IoT-device identification* is possible to take appropriate actions to manage it. Traditionally communication security is managed by higher layers and solved through a wide variety of ciphers and key management systems. The basic idea is that by 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 of the computational power makes vulnerable ciphers initially considered unbreakable. Also, the key distribution can be a problem, especially in dynamic systems, and introduces latency that can be unacceptable for delay-constrained services [16, 78, 86]. In new networking paradigms, such as the Internet of Things (IoT), traditional security methods cannot achieve the desired performance due to the radical change of requirements and constraints for establishing secure communication. The lack of resources (i.e., computation, memory, and energy), reduces the effectiveness of traditional security schemes [16, 17, 67, 78, 90]. Moreover, cryptographic techniques can lead to excessive transmission overhead, communication latency, and power consumption, hence, often in IoT networks hard encryption procedures cannot be performed, at least with high frequency [56]. In particular, the use of complex asymmetric cryptography schemes is impractical in many cases and key distribution operations can be difficult and can introduce significant latency and overhead in case of massive IoT-device access and dynamic/unplanned networks, [16, 19, 67, 90]. 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 the IoT context, and in general, in resource-constrained systems, physical Physical (PHY)-layer security physical layer security (PLS) is a promising additional method

to achieve communication security with low complexity [78]. Since it substantially operates independently on the higher layers and can be used to enhance the security of existing approaches. The use of PLS has been proposed and adopted in the literature for several years [72], and the recent development of the IoT has given a great impulse to the research community to use PLS. Using the physical-layer characteristics as a security tool can be seen as a method to help the higher layers protect the system and, at the same time, to implement security even in low-resourced devices [75], [54]. 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 [49]. 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 to enable the intended receiver 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* – through 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.

In this thesis, the focus is on Physical-Layer Authentication (PLA). Device identification and spoofing detection at the physical layer are considered promising security mechanisms [78, 90]. Indeed, PLS *(i)* involves only the physical layer; *(ii)* lies on the variation and randomness of the wireless channel rather than on the computational complexity of hard mathematical problems; *(iii)* uses the randomness of the wireless channel as a "secure key" avoiding key management burden; *(iv)* can authenticate legitimate nodes quickly before demodulation & decoding, thus reducing the overall latency. Complexity and overhead are reduced

since upper-layer processing is not required, thus a device can be authenticated quickly before demodulation and decoding. Differently from cryptography algorithms, the computational load can be almost all on the access point (AP), 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, PLA approaches do not require modifications to existing systems and, hence, can be easily added in a very short time. PLA is not designed to replace the upper-layer authentication but to enhance and supplement conventional cryptography-based methods to protect the system even in the presence of low-resourced devices [55]. PLA could be used for example to build a two-step authentication process with an upper-layer authentication mechanism used to identify the legitimate user while the PLA is used to authenticate the device used by the legitimate user. The PLS is not thought to replace traditional security, but it is an additional security layer that helps to enhance the security level, in particular when low-resourced devices are used with a wireless connection [55]. PLA simply adds a "first line of defense".

During this Ph.D. study, different physical layer solutions for PHY-layer continuous authentication and spoofing detection based on machine learning have been proposed analyzed, and compared. In particular, we propose different *Machine Learning (ML) wireless fingerprinting* solutions 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 for an analytical model and in a continuous way, thus taking into account time-varying effects [16].

1.1 State of the Art

PLA is emerging as an efficient approach to provide low-complexity security exploiting the physical layer's unique features and the communication channel's randomness [90]. In general, PLA methods can be classified as *active* or *passive* [74,90]. In the former case, the main idea is em-

bedding in the message a tag based on a secret key, through the superimposition of an authentication signal to the message or introducing a certain level of randomness to the signal for example in [43,73,81,89,95,97]. These methods usually require additional computational complexity to recover the signal through demodulation and decoding, and to generate keys. Differently, passive (i.e. keyless) PLA methods identify the device exploiting specific characteristics of the physical signal (*physical fingerprinting*). Specific characteristics of the transmitter or its communication channel are extracted from the received signal and compared with those of previous authenticated messages to identify a claimed source. In this way, the receiver can continuously authenticate the transmitting node. Passive PLA methods can be further divided into two classes *(i)* the *radio frequency* (RF) fingerprinting exploits HW imperfections for achieving a unique signal waveform, *(ii)* the *wireless fingerprinting* (WF) reflects the features of the channel experienced by the device. The first class is known as *radio frequency* (RF) fingerprinting, which exploits hardware (HW) imperfections for achieving a unique signal waveform, such as [12,58,59,61]. Often these kinds of schemes are data-dependent and/or do not take into account channel effects that can reduce their efficiency. Indeed, different devices usually have slightly different RF features that are difficult to distinguish if the signal is corrupted by noise and interference. Consequently, high-precision RF feature estimation circuits are needed implying high-cost and overhead. The second class is known as *wireless fingerprinting* (WF), which is based on the extractions of features of the propagation channel between the transmitting and the receiving devices, that cause unique and recognizable distortions to the received signal. Specifically, the channel state is location-dependent, and can significantly change if the transmitter moves more than a wavelength away from the original location [87]. Consequently, channel features of two different transmitters can be regarded as uncorrelated and a node can be identified by extracting the characteristics of the communication channel from the received signal and verifying the correspondence to those of previous transmissions of a claimed source. A malicious device can hardly emulate the channel properties of a trusted device.

In this PhD thesis, we focus on WF. In this context, the literature presents different approaches for legitimate nodes and rogue device

identification. The basic approach is using statistical hypothesis testing to determine if the transmission is done by a legitimate node or not. This is done by comparison of a specific channel feature (e.g. channel state information - Channel State Information (CSI), channel impulse response - CIR, received signal strength - RSS, power spectral density, etc.) with a test threshold as in [26, 48, 76, 83, 84]. Such schemes suffer from errors in channel estimation, and RSS fluctuations due to multipath and shadowing effects, and two users in different positions can have similar RSS. Moreover, CIR-based methods require the extraction of CIR which is not easy in real and time-varying systems. In addition, choosing the appropriate threshold can be challenging due to the characteristics of the propagation environment and the unknown spoofing model. In [45] it has been shown that under a low-SNR regime, the authentication based on a binary hypothesis testing cannot guarantee robust performance. For this reason [88] propose Q-learning-based approaches to obtain the optimal test threshold in spoofing attack detection. Recently, ML approaches have gained great interest in PLA [16,86] due to their prowess in complex pattern recognition and adaptability to the dynamic nature of wireless communication environments. By continuously analyzing intricate signal characteristics, ML algorithms can extract subtle patterns that aid in distinguishing legitimate transmitters from malicious entities, ultimately reducing false positives and enhancing the accuracy of authentication decisions. Additionally, machine learning excels in anomaly detection, proactively identifying irregularities in physical layer properties that may signify security threats, thereby allowing for early threat mitigation. With its capacity to process multiple features from the physical layer, scalability to accommodate a growing number of devices and resource-efficient nature, ML adds a robust layer of security, making it a valuable tool in fortifying authentication within modern wireless communication systems. ML techniques include parametric/non-parametric as well as supervised, unsupervised, and reinforcement learning approaches. In general, parametric models could be more accurate and simpler than non-parametric ones but require apriori knowledge of statistical properties of the attributes and the training functions. This implies computational resources and time to obtain such information, moreover, it is challenging to obtain in a complex and dynamic environment such as the wireless one. Differ-

ently, non-parametric methods, do not require any a priori knowledge and learn dynamically from data, thus are more flexible. The difference between supervised and unsupervised learning is the use of labeled data or not. Unsupervised learning does not require labeled data and aims at clustering data in different groups based on their similarity. In the PLA context, non-parametric and supervised approaches are mainly considered. Indeed, parametric approaches require the knowledge of models, as most of the existing approaches, that in complex environments may be difficult to obtain with consequent performance degradation. Nonparametric are model-free. Unsupervised approaches have a complexity that grows exponentially and usually require the knowledge of some information that limits their applicability. Deep Learning Deep Learning (DL)-based approaches have been proposed for improving the accuracy of CSI-based methods since they are capable of adapting to time-varying channels thus improving the identification of legitimate nodes for example in [44, 63, 79]. However, DL to be efficient requires that the behaviors of neural network (NN) is interpretable (i.e., to understand how a neural network associates an input with a corresponding label.). Moreover, often DL methods are useful to identify the devices but fail in detecting anomalies [50].

Finally, complex NN cannot be suitable for low-complexity IoT devices. These methods need numerous labeled data for training. However, collecting numerous labeled data is arduous and time-consuming, which cannot be scaled to the environment with more IoT devices. Consequently, lower-complexity ML classification approaches have been investigated. One-Class support vector machines One-Class Support Vector Machine (OC-SVM) and k-means clustering algorithms are considered in [25] for the detection of eavesdropping attacks in an Unmanned Aerial Vehicles (UAV) context. The method is based on the creation of artificial training data (ATD) based on the knowledge of the CSI of the legitimate node. ATD is used for training and labeling the OC-SVM model while it is used for labeling clusters of k-means. Using only one channel attribute whose estimation can be affected by errors, can be not enough to provide a sufficient differentiation among transmitters. Some papers exploit the system diversity to have multiple observations of the channel attribute that can be suitably combined for enhancing detection accuracy. ML and classical hypothesis testing solutions are compared

in [69] exploiting multiple CSI observations given by a set of parallel wireless channels (i.e., an OFDM system). In particular, k-nearest neighbour (k-NN) and Support Vector Machine (SVM) algorithms are considered. Similarly, [3, 4, 94] propose a SVM for device authentication exploiting multiple-observation of the considered attribute generated by spatial diversity of a Multiple Input Multiple Output (MIMO) system. In [94] CIR is exploited. The paper proposes to use a neighborhood component analysis (NCA)-based feature selection and then classification is performed by meansSVM. While in [4] the magnitude and real and imaginary parts of the received signals are used as features. Instead of using multiple observations of the same attribute, some approaches propose to use different characteristics of a given attribute. Euclidean distance and correlation coefficient of the channel estimates are used in [77, 80] to feed a linear classification in [80], and an extreme learning machine in [77]. Here, attributes are assumed to be Gaussian distributed and the spoofing model is required to improve the accuracy of authentication. A few papers propose using actual multi-attribute PLA schemes that are more robust since it is more difficult for a rogue device to predict many attributes of a signal received from a different location. The legitimate device has multi-dimensional protection. For example [44, 48, 98] define a classical hypothesis testing solution using channel and phase noise, CSI and path delay, and CSI and carrier frequency offset, respectively. A multi-attribute system is proposed in [15] using a kernel least mean square authentication scheme able to track time variations for a three-device scenario. Multi-attribute is mapped onto the one-dimensional subspace. In [52] a multi-device multi-attribute devices' identification approach exploiting the decision-tree classification is proposed.

1.2 Research goal

As stated before, WF-PLA solutions based on ML approaches have recently attracted a lot of interest for their potentialities, especially for low-complexity IoT nodes. However, the proposed solutions present some drawbacks and limits that must be addressed. In particular,

- almost all PLA systems performing spoofing detection focus on a three devices scenario (i.e., the legitimate transmitter, the legiti-

mate receiver, and the malicious device) [15,25,26,45,48,69,77,79, 83–85, 88, 94]. This is not suitable for future large-scale IoT networks where a huge number of IoT devices will be interconnected and need to be identified. The authenticating node must be able to distinguish not only the legitimate node from the malicious one but also to distinguish the legitimate nodes among others. Not all approaches proposed for the three-device scenario can be easily extended to the multi-user case, especially those based on a binary hypothesis test. Moreover, some multi-user solutions are designed for legitimate node identification but fail in spoofing detection. Moreover, a multi-device context is more complex since there is a higher variability of legitimate channels and the probability that the spoofing attacker is close to one of them is higher. Multi-user approaches have been proposed in [41, 44, 52]. Two multi-device classification algorithms based on decision tree are proposed in [52], introducing the capability of spoofing detection with a high-layer cross-check identification. In [41] a Convolutional NN (CNN) is used for authentication/spoofing detection. The system is based on the definition CSI profiles for legitimate and malicious users, that cannot be actually available in many scenarios. Also in [44] a CNN using CSI feature is considered. However, the effects of multiple users are not clearly investigated. An hypothesis testing solution is also proposed, for a scenario where each legitimate node is impersonated by a malicious user, hence, the detection is always one-to-one. Deep NN and data augmentation method have been combined in [43] to speed up the training phase of multi-device identification but the spoofing detection has not been considered.

- Most of the PLA methods performing spoofing detection are based on a thresholding method. This means that the threshold must be optimized for each scenario with consequent performance degradation, especially in a time-varying environment. Moreover, in a multi-user context, an optimal threshold value for each IoT node should be set, consuming plenty of network resources and causing signaling congestion in massive IoT systems;
- Several papers propose model-based authentication methods that need to obtain an accurate model, and it can be difficult in complex

environments, thus degrading the performance and can require a lot of data. Moreover, knowledge about the attacker is unrealistic in many practical scenarios;

- Most of the solutions proposed in the literature are based on the observation of a single channel attribute, multiple observations of the same attribute, or different characteristics of the same attribute. Only a few papers consider multiple attributes, that are usually limited to two. Various attributes are considered in [15] for a three-device scenario. Moreover, spatial information, in particular Angle of Arrival (AoA), is a rarely considered attribute. AoA is exploited in [92] to validate the claimed GPS location information in a vehicle-to-roadside communication using a two-side hypothesis testing problem and in [33] to authenticate a device through the comparison of measured AoA against the AoA stored in a database in an underwater environment. In [33] AoA is used as a decision metric for the hypothesis test.

In this PhD thesis, continuous authentication/spoofing detection systems, suitable for an actual WSN, where multiple IoT nodes communicate with a sink node are proposed and compared. The basic idea is that first legitimate devices are authenticated through a higher level procedure and a unique identification code (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. As detailed in the following chapters we consider two main approaches:

- *classification-based* - that resort to ML classification algorithms that for definition are multi-class, and hence, can be used in a multi-device scenario where each class corresponds to a node of the network. Indeed, these ML algorithms aim at assigning each element to be tested to one of the known classes based on patterns extracted from data used for training. The classifier analyzes patterns and relationships between channel attributes and the node ID and builds a model that can be used to predict the class of new unseen data. Since assuming the knowledge of malicious users' data can be unrealistic, only authorized users' data are used for

training and creating classes. Consequently, this kind of algorithm is not able to directly detect a malicious node, that would be in any case classified as belonging to one of the legitimate node's classes. An additional step is needed to detect malicious users. In particular, 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 compared to the exclusive use of an ID: the ID can be stolen, while the PLA aims to support 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 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.

- *anomaly detection-based* use ML algorithms that identify data that do not fit a previously known pattern so that it is possible to identify spoofing nodes. However, this class of algorithms is designed for a single-class scenario, since it differentiates between data of an authorized node and other data. They need to be adapted to the specific scenario where multiple legitimate nodes are involved.

For both solutions, to have an exhaustive comparison, we have considered different ML algorithm types: kernel-based, nearest neighbors, clustering, and binary tree. The optimization and the comparison of multi-device classification solutions with multiple one-to-one anomaly detection methods, which to the best of our knowledge has never been investigated before, especially in a multi-device scenario. Even if other works present some comparisons among different strategies as in [25,69], these are usually limited to a three-device scenario and a single type of ML approach. The proposed ML approaches exploit a multitude of attributes, including AoA. In [15, 52] it has been shown the relevance of having multiple-attributes. Particularly, in [52] the authors showed that the AoA and delay attributes are those most relevant since the signal strength fluctuates and, in the presence of many legitimate users, fingerprints of different users can overlap. Finally, we have extended our study to a mobility scenario. PLA becomes considerably more challeng-

ing in the presence of mobile IoT nodes due to several key factors [82]. First and foremost, mobility introduces dynamic variations in wireless channel characteristics, such as signal strength, phase, and multipath effects. These fluctuations can lead to inconsistencies in the physical layer features used for authentication, making it harder to establish and maintain a reliable baseline for node verification. Additionally, as IoT devices move, they may encounter different access points or network segments, each with its unique channel properties, further complicating the authentication process. The need to continuously adapt authentication criteria in real-time to accommodate node mobility adds complexity, as it requires efficient algorithms capable of rapid decision-making. Furthermore, the potential for frequent handovers or network reconfigurations in mobile IoT environments necessitates robust mechanisms to ensure uninterrupted authentication, enhancing the risk of false positives or negatives. In essence, the mobility of IoT nodes introduces a dynamic and challenging environment where physical layer-based authentication must contend with evolving channel conditions, network transitions, and real-time adaptation, making it a formidable task to ensure the security and reliability of IoT communications.

Chapter 2

Classification-based PLA approaches

In this chapter, we describe a system for WSN node authentication and spoofing detection based on the Physical Layer Security approach called wireless fingerprinting and ML classification algorithms. We focus on an actual wireless WSN, where multiple nodes communicate with a sink node. Nodes are in fixed positions but the communication channel varies due to the scatterers' movement. In the proposed security framework the sink node performs a continuous authentication of nodes during communication based on wireless fingerprinting. In particular, an ML approach is used for authorized nodes classification by means of identification through specific attributes of their wireless channel. The classification results are compared with the node ID to detect if the message has been generated by a node other than its claimed source. Finally, to increase the spoofing detection performance in small networks, the use of low-complexity sentinel nodes is proposed. Results show the good performance of the proposed method that is suitable for actual implementation in a WSN.

2.1 Introduction

As the first step of the research, we have started investigating a PLA method based on ML classification algorithms. In particular, we investigate a solution for a supervised classification of devices based on CART and Random Forrest algorithms that have not been previously investigated in this context. Random Forrest has been adopted in [5] 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 scheme detailed later, allows the identification of legitimate nodes and the spoofing detection using a cross-check with a higher layer used identification code (ID). The used authentication is based on a WF with multiple attributes, and the effects of different attributes are separately evaluated. To our knowledge only [15] provides an analysis based on the availability of different attributes, but in a different context. Moreover, to improve the spoofing detection capabilities, we propose the use of *sentinel nodes* in small networks. Cooperative solutions for PLA have been rarely considered as in [85] but here the goal of sentinel nodes is completely different and these nodes have not to perform any operation except sending periodical beaconing signals. The performance of the system is evaluated in an actual and general time-varying channel, also considering different environmental conditions, while most of the papers in the literature consider fixed channel parameters and simple channel models.

2.2 System model

Why is security necessary in 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

1. attacks against security mechanisms, and
2. attacks against basic mechanisms (like 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 research, we have 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 2.1). Sensor nodes are supposed to be devices with low-resource (i.e., computation, memory, and energy), performing simple tasks that are monitoring 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 more powerful device having more complex functionalities. Indeed, the sink node is in charge of 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 to be always connected to a power source. The transceiver is equipped with multiple antennas so that the spatial information can be exploited in the network.

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) multi-path fading channel [32]. 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 to describe indoor area wireless communication systems operating in the 5 GHz spectrum with a bandwidth of up to 160 MHz. In this research activity, we selected the Model-D scenario [14] that represents the propagation conditions in a typical large indoor open environment,

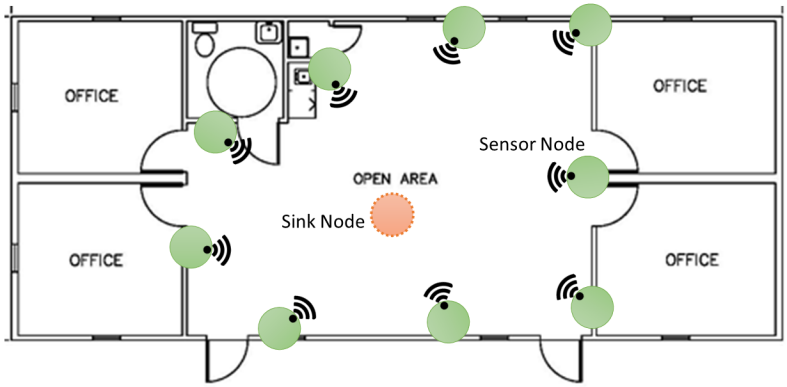


Figure 2.1: WSN with sink node responsible for the security of the system.

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 scatterers' movement in the area.

The 802.11ac™ model represents a MIMO channel, with M transmitting and Q receiving antennas. However, we focus on the particular case with $M = 1$, hence we focus on a Single Input Multiple Output (SIMO) 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

$$\mathbf{H}(t) = \sum_{l=1}^{L} \mathbf{H}_l(t)\delta(t - \tau_l) \tag{2.1}$$

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

$$\delta(t) = \begin{cases} 1, & \text{if } t = 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

. 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 (nonvariable) 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

$$\mathbf{H}_l(t) = \sqrt{\gamma_l}\left(\sqrt{\frac{\zeta}{\zeta+1}}\mathbf{H}_l^F(t) + \sqrt{\frac{1}{\zeta+1}}\mathbf{H}_l^V(t)\right) = \sqrt{\gamma_l}\left(\sqrt{\frac{\zeta}{\zeta+1}}\begin{bmatrix} e^{j\phi_1(t)} \\ e^{j\phi_2(t)} \\ \vdots \\ e^{j\phi_Q(t)} \end{bmatrix} + \sqrt{\frac{1}{\zeta+1}}\begin{bmatrix} X_1(t) \\ X_2(t) \\ \vdots \\ X_Q(t) \end{bmatrix}\right) \tag{2.3}$$

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 of 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 afterward

$$L(d) = \begin{cases} L_{FS}(d), & \text{for} \quad d \leq d_{BP} \\ L_{FS}(d_{BP}) + 35\log_{10}(d/d_{BP}), & \text{for} \quad d > d_{BP} \end{cases} \tag{2.4}$$

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 [39] model for what concerns the delays and the Angel of Arrival (AoA) information. In

particular, since path delays are fixed in the TGa model in every channel realization, we have used the distribution proposed in WINNERII to model the path 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 [39]. Moreover, to take into account the scatterers' movement in the surrounding environment as well as delay estimation errors, we have 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 [39], AoA is normally 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.

2.3 Proposed approach

This 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 using 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 can 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 extracting some PHY-attributes from the signal received by a specific device and, hence, by a specific propagation channel. In this work, we have 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 utilizing an ML approach. In particular, we focus on a supervised-learning multi-class classification approach, hence:

- During the *training phase*, as shown in Fig. 2.2 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 fingerprint of the attacker.
- Then, during *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: if they match the authentication is successful otherwise it fails. In the second case, the node communication is blocked and a new authentication at higher-layer must be performed.

The communication phase procedure is represented in Fig. 2.3.
In details, let's 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 belongs 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. On the opposite, if

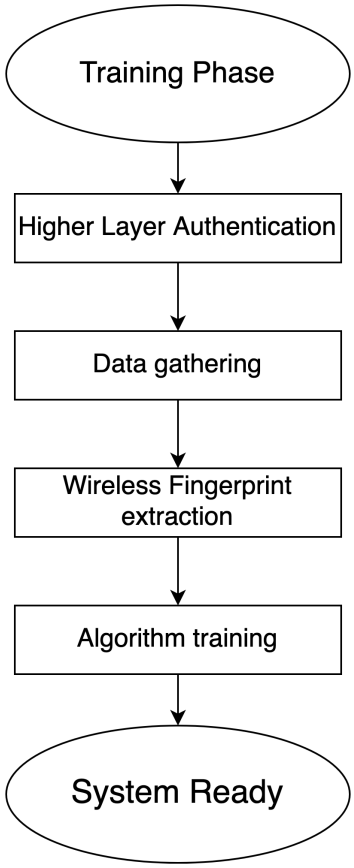


Figure 2.2: Flow diagram of the training phase.

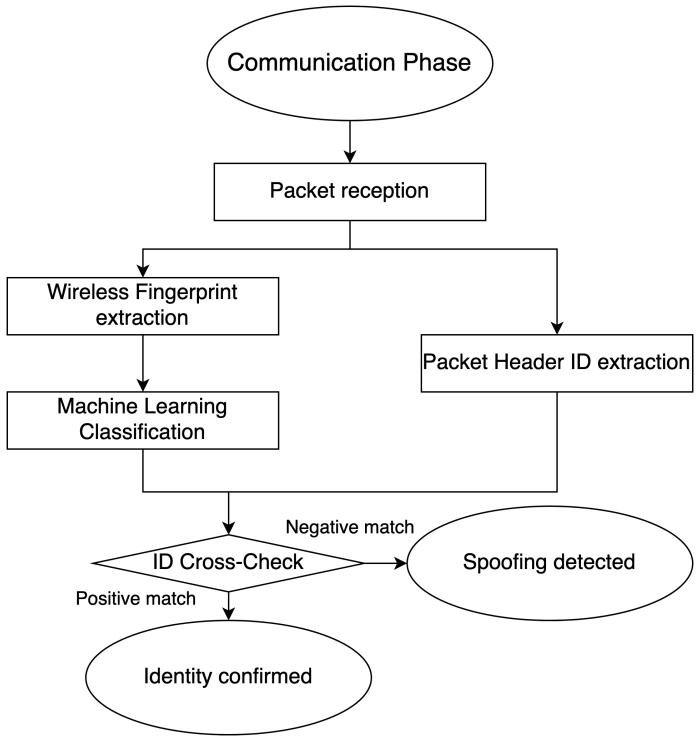


Figure 2.3: Flow diagram of the communication phase.

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 an unauthorized node is classified as authorized, named *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 can 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 2.4), 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.

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 received data is related to the outcome of the proposed method since encryption is not used, *(ii)* high-level authentication provides both an unique ID and a secret key that can be used in successive

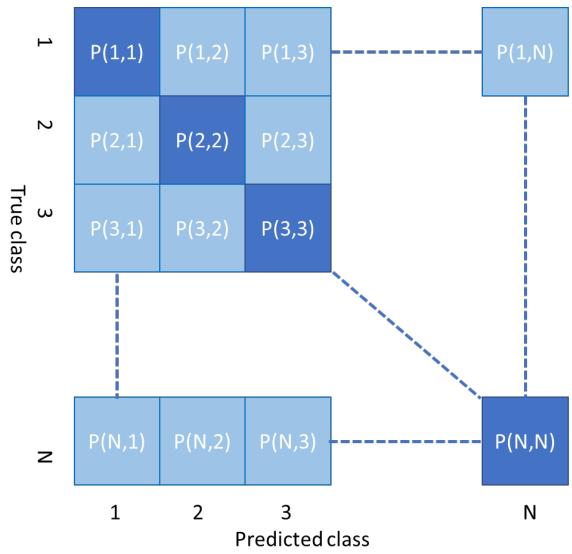


Figure 2.4: Confusion Matrix.

encrypted communications. In this case, the proposed approach is an additional security level that avoids spoofing even if the secret key has been detected by the attacker, especially in the presence of low-robust encryption algorithms.

2.3.1 ML for devices classification

As stated before, sensor device identification is performed using a ML approach that exploits multiple PHY-layer parameters of their unique propagation channel. In particular, we resort here to a non-parametric classification approach, so as not to depend on information from a certain sort of distribution difficult to achieve in dynamic environments. Moreover, this method is suitable for a low-cost/low-consumption WSN. In particular, two different algorithms have been investigated. First, a CART algorithm has been used [11]. This is a supervised ML algorithm that generates a decision tree 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. Lets assume that data are characterized by K attributes $\mathcal{A} = a_1, \cdots, a_K$, and lets consider the Shannon's entropy of a dataset $\mathcal{D}$ that is

$$\boldsymbol{H}(\mathcal{D}) = -\sum_{n=1}^{N} p_n \log_2 p_n \tag{2.5}$$

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

$$I_{gain}(\mathcal{D}, a_i) = H(\mathcal{D}) - R(\mathcal{D}, a_i) \tag{2.6}$$

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

$$\hat{a} = \max_{a_1, \cdots, a_K} I_{gain}(\mathcal{D}, a_i) \tag{2.7}$$

The algorithm is iterative: initially, the whole training dataset is considered (tree root), and at the first step this is split into 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 splits has been performed (it is set as a parameter);

2. one leaf is "pure", that is all input data in the leaf belongs 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 dimensions avoids also overfitting problems that can arise by 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 to the corresponding branch.

The second algorithm that has been considered is Random Forest [10, 24], which has been 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 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) [10, 24]. During the classification phase, received signal samples are moved in the different decision trees and the results are taken by evaluating 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, environment monitoring, etc.). 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 testing. 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 $O(KMlog_2M)$, which, asymptotically, is the complexity of the training phase. For the RF algorithm complexity must take into account the number of trees T, hence 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 have verified that selecting a number of splits slightly higher than N provides a slight improvement in accuracy, but a further increase does not provide advantages. For simplicity, assuming that the number of splits 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 (testing) 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, 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 have verified that the performance degradation is not significant within a certain value, we have tested the nodes' density up to around 50000 $nodes/km^2$. Obviously, the number of needed splits of the trees increases.

2.3.2 Sentinel nodes

The classification algorithm allows one to associate each received signal to one of the possible WF classes that are labeled with the authorized node ID. When a malicious node wants to access the network, supposing it attempts to copy 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 the malicious user classified as one of the authorized users, the spoofing detection fails when the wireless fingerprinting (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 for reduction of the classification space, thus increasing the detection. Sentinel nodes periodically send a beaconing signal, and thus are classified as an additional authorized source. 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, for example in [85], where the additional nodes, estimate the RSSI of the authorized communication link and forward this information to the sink node for an 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.

2.4 Numerical results

This section presents the numerical results of the proposed authentication/spoofing detection method derived through simulations using Mat-

lab software. An area $\mathcal{A} = 30 \times 30\,\text{m}$ representing a large indoor hall with the sink node positioned in the center has been considered. The number of connected nodes is $N = 15$ if not differently indicated. The channel attributes have been characterized stochastically as described in Sec. 2.2 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 attributes variations. As specified in each scenario nodes have been 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 has been evaluated by averaging the value P_{msd} over different positions of the spoofing node in the area as specified later.

2.4.1 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 to correctly classify 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: scatters' speed is in the range [0-5] km/h, $\sigma_\tau = 1/\lambda$ with $\lambda = 1.664 \cdot 10^7$ and $\sigma_{AoA} = 1.5849$ [39]. However, to test the effectiveness of the classification under more challenging conditions we have also considered different scenarios that are:

- *Scenario A1*- Nodes are randomly placed in $\mathcal{A}$ according to a bidimensional probability distribution. The Doppler spread is related to a scatters' 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 ad are set as in A2, but also angle and delay spread are increased considering a variance that is three times the original one;

Table 2.1: 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)

- *Scenario B1* - Nodes were placed in clusters as shown in Figure 2.6, and the signals are affected by 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 have been created for each scenario to train the machine. In particular, for each node 5000 impulse responses have been sampled and of those, the first 100 have been used as training dataset while the rest of them were used to evaluate the performance of the classifier. As shown in Figure 2.5 for the CART algorithm[1], the value of 100 for the training sequence length has been 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 algorithm is too fitted on these and, hence, there is a consequent significant loss of performance after training.

First of all, we have evaluated the performance of the CART algorithm varying the number of splits in the range [20-60]. We have seen that in basic scenarios there is not a high variance of the achieved values with the number of splits. Differently when Doppler and variance of angle and delay spread increase a higher number of splits is beneficial. In general, 20 splits is a good trade-off. Table 2.1 reports the maximum and minimum values of the *classification accuracy* for different scenarios and the number of splits for which these values are reached.

The following results have been derived assuming a CART classifier

[1]Similar results have been derived also for Random Forest algorithm.

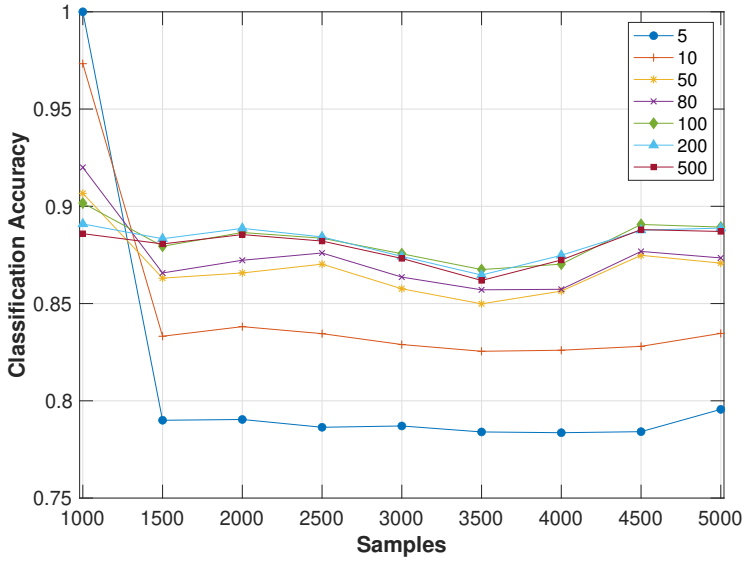


Figure 2.5: CART classification accuracy for different training sequence lengths in A1 scenario.

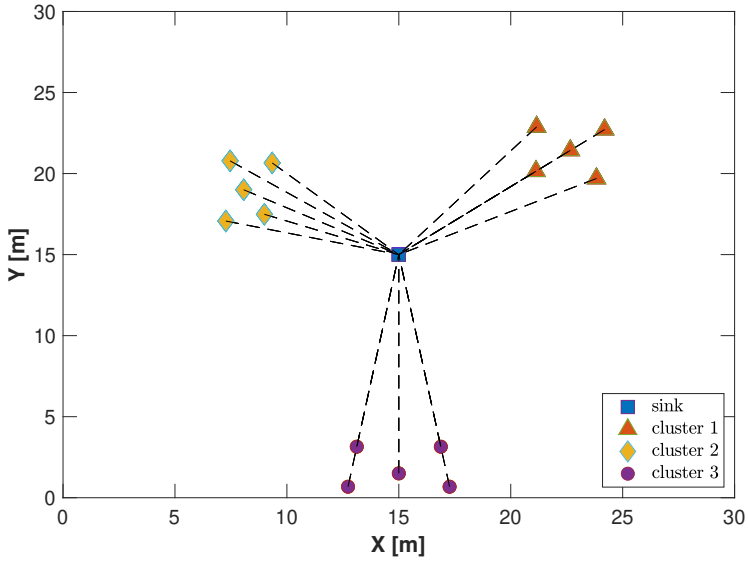


Figure 2.6: Example of the nodes' position in a clustered scenario.

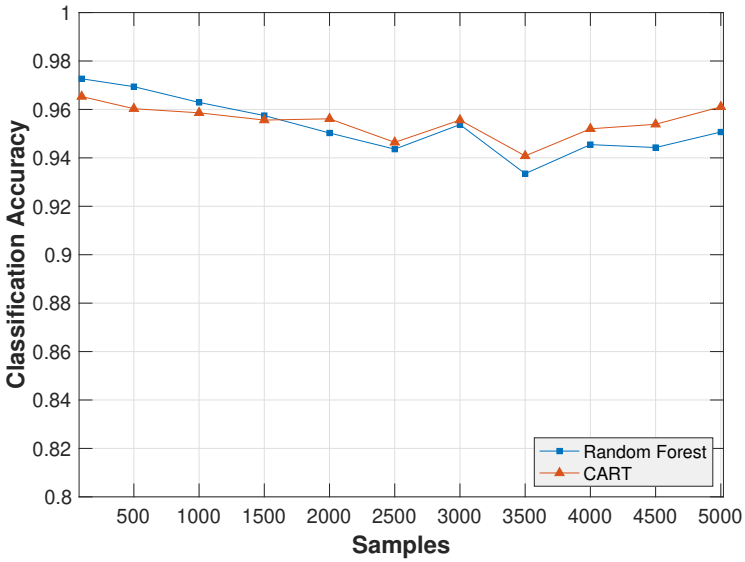


Figure 2.7: Classification accuracy for the scenario A1.

with a maximum of 20 splits and a Random Forest classifier composed of 5 trees of the same size as the CART one. Figure 2.7 shows the accuracy of the two classifiers on the A1 dataset for the whole sample sequence of 5000 samples (graphs start at 100-th sample because the first 100 samples are used for training). Both classifiers show good stability with little loss of accuracy over time and, with an average percentage of correct classification around 95% (i.e., in average $P_{fsd} = 5\%$). There are no significant differences in the performance of CART and Random Forest algorithms, the gain of Random Forest is substantially negligible. The accuracy averaged on the whole dataset for all scenarios is reported in Table 2.2.

Results in Table 2.2 show that only in the scenario B2 there is a noticeable reduction of the accuracy that goes down around 88-89%. We have evaluated also the effect of the number of nodes in the area. In scenarios A1,2,3 we have varied the number of nodes in the range $[15, 30]$ and we have seen that there is no performance degradation in terms of accuracy, but obviously the number of splits must be increased. In particular, up to 25 nodes the sufficient number of splits is N increased by

Table 2.2: 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%

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.

To further investigate this issue, we have considered also different clusters' distributions. We noted that there is no 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, more likely clusters overlapping occurs (being clusters randomly placed). Thus, a lower number of clusters with a higher number of nodes occurs and overall accuracy decreases. We have 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 of 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%. In the extreme case of a single cluster with 15 nodes, the performance worsens and the average accuracy of CART is 88%. In general, up to a certain node/cluster density the reduction of accuracy is limited, but obviously when the density significantly increases there is a reduction of 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 how these impact on the accuracy of the classifi-

Table 2.3: 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%

cation. For this reason, the classifiers have been used with different sets of attributes, in particular:

- 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 2.3 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.

The P_{ban} depends on the ML classification algorithm, hence, we have compared the results of CART and Random Forest with those of other two basic ML classification methods: SVM and k-NN, to show the

Table 2.4: Comparison of classifiers accuracy.

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

effectiveness of the selected ones. In Tab. 2.4 the classification accuracy of the four methods is reported for scenario A1. We have considered a linear kernel for the SVN and $k = 20$ for the k-NN[2].

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 fewer degrees of freedom that 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 hyper-parameters setting.

2.4.2 Probability of missed spoofing detection

The second performance indicator is the miss detection of 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 2.8 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 into 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 has

[2]Different values of k have been tested

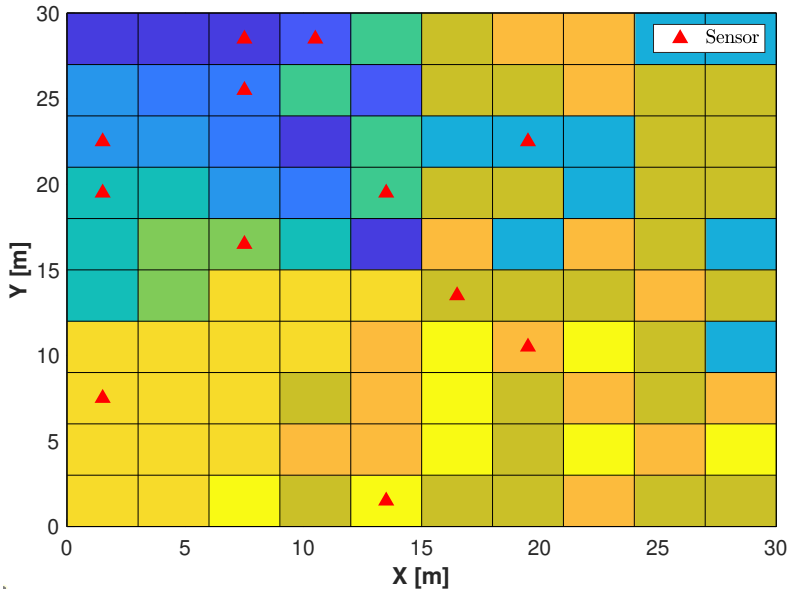


Figure 2.8: Example of unauthorized node classification depending on its position in the area.

been 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's 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 2.9. These results have been derived by 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, 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 2.9.

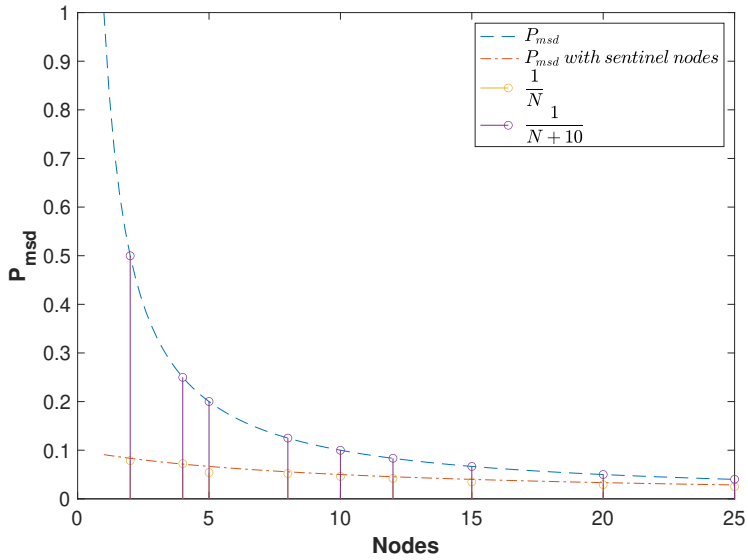


Figure 2.9: P_{msd} vs number of authorized nodes in the area, with and without sentinel nodes ($N_S = 10$).

Sentinel nodes are randomly placed in the area.

As a second scenario, we have considered the worst case in which the malicious node tries to impersonate the nearest authorized node (i.e., it can intercept its ID). Results are shown in Figure 2.10 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 2.8, but is still close to the $1/N$ curve, because the spatial correlation is not so high.

2.4.3 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 has been proven that this approach can correctly classify and authorize nodes with high accuracy even in the presence of challenging channel attributes variability, however, in the considered scenario, nodes' position is assumed to

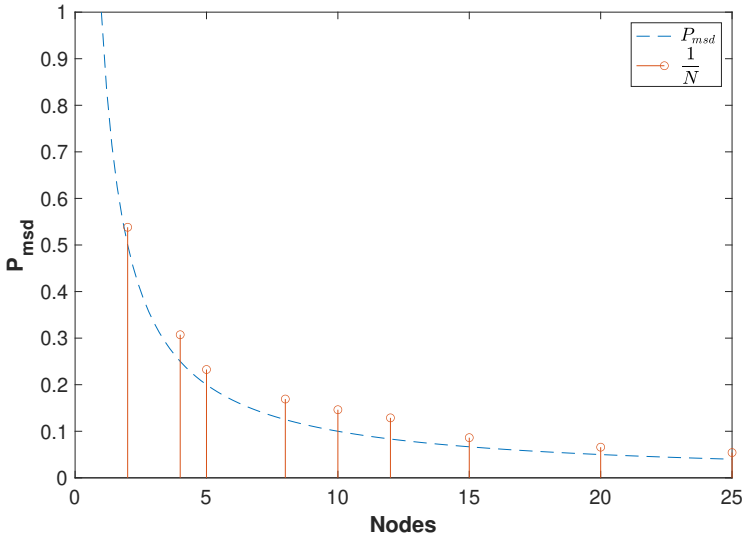


Figure 2.10: P_{msd} vs number of authorized nodes in the area, when the malicious user selects the ID of the nearest sensor.

be fixed, hence, the mean values of delay and AoA do not change, while 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 propose. 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 to work in a multi-class environment. These solutions could not only make the spoofing detection probability independent of the number of nodes in the network but also avoid the use of the node ID.

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

2.5 Conclusions

This chapter 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 approach is used for classifying the authorized users so that the capability of analyzing multi-dimensional information without the need for an analytical model is exploited. In particular, the framework proposed is based on two ML approaches based on 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 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.

Chapter 3

Anomaly Detection-based PLA approaches

Differently from the previous chapter, here, we investigate the effectiveness of Physical Layer Authentication (PLA) where the legitimated node is distinguished from potential attackers by exploiting the unique wireless channel features using four different anomaly detection ML strategies in their one class version: decision-tree, kernel-based, clustering and nearest neighbors. Our study highlights the advantages and disadvantages of each method, considering parameters optimization, training requirements, and time complexity. Results show that the use of multiple-attributes allows to achieve accurate detection performance. In particular, our results reveal that the kernel-based solution is the one that achieves the best results in terms of accuracy, but the nearest neighbors solution has very similar performance with a significant advantage in terms of complexity and no need for training, making it more suitable for time-varying contexts, and a promising choice for securing IoT nodes through PLA based on wireless fingerprinting. The other two alternatives have somewhat lower performance but low complexity. This research contributes valuable insights into enhancing IoT security through PLA techniques.

3.1 Proposed authentication/spoofing detection framework

In this chapter, we propose the use of different ML approaches to authenticate legitimate nodes and detect rogue devices trying to access the network claiming a false identity. In particular, while in the previous chapter we focused on classification-based solutions, here we consider anomaly detection solutions. The aim is to compare different different solutions based on different ML approaches.

For what concerns the system model we consider here a three-node scenario where the IoT network is composed of a transmitting node (Alice) and one sink node (Bob) which performs node authentication. Bob is the network coordinator and he is in charge of performing the authentication and the other security issues. Eve is the malicious user attempting to spoof Alice's identity. For what concerns the channel model we refer to the previous chapter 2.2.

The authentication/detection framework proposed here works in two phases:

- *Phase I*: Bob identifies Alice using a traditional authentication protocol, and collects a set with size n of data received from Alice to extract her WF and train the ML algorithm. This phase is identical to the training phase described in 2.3, except for the ML algorithm employed.

- *Phase II*: Bob receives a message without assurance that it comes from Alice, hence, he tries to verify its authenticity by extracting the WF attributes from the signal and it feeds them to the ML anomaly detection algorithm. A *positive* result of the ML algorithm means that a match of the WF extracted from the message against the WF acquired during the Phase I is found, so the sender of the message is considered legitimate. Conversely, a *negative* result implies a message rejection and consequent countermeasures, e.g., a new Phase I authentication of the sender. It is important to stress that phase II allows a continuous authentication of the IoT nodes, without any resource burden for the IoT nodes since all operations are performed by the sink node. Therefore this approach is suitable for resource-constrained IoT nodes. This phase

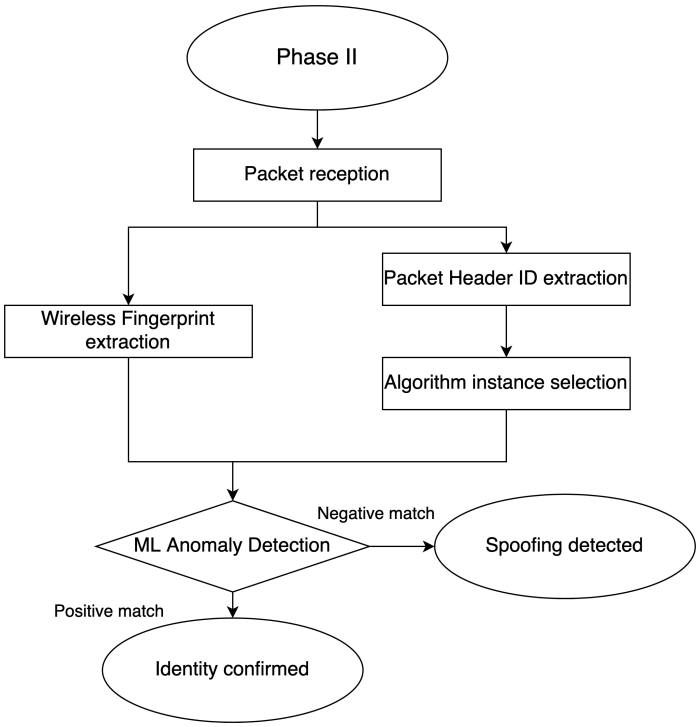


Figure 3.1: Flow diagram for Phase II.

is illustrated in Fig. 3.1, except for the ML algorithm employed.

The WF method operates by extracting multiple attributes from the channel between sender and receiver: (1) the Received Signal Strength (RSS), (2) the AoA of the main path, (3) the maximum path delay, and (4) the signal energy.

3.1.1 Machine learning-based anomaly detection algorithms

In this section, we briefly describe the ML-based authentication/anomaly detection schemes that have been considered here. In particular, their *one class* version has been considered: they distinguish only one class

(Alice) and everything else is considered an anomaly (Eve). These schemes do not require any knowledge about Eve, they can operate in the absence of negative class training samples (i.e., without collecting samples from Eve). In general, these kinds of schemes operate by defining a decision test around the positive class (Alice), for separating and identifying new fingerprints as legitimate or not. The classification is based on two parameters: the *distance* between the test element (i.e., the element to be classified) and the dataset characterizing the legitimate node (i.e., data used for training during Phase I), and a threshold. A sample is positively classified if the distance is lower than the *threshold*, otherwise, it is classified as negative (i.e., an anomaly). Different algorithms define these parameters differently.

The performance of the algorithms has been evaluated in terms of

- *True negative rate* is the ratio between the number of anomalous samples correctly detected and the total number of anomalous samples. It represents the probability of correctly detecting an anomaly (Eve);
- *False negative rate* is the ratio between the number of legitimate samples mistaken for anomalies and the total number of legitimate samples. It represents the probability of mistaking an authorized node (and blocking it), i.e., Alice is identified as a malicious node;
- *Balanced Accuracy BA* is the average between the true positive rate and the true negative rate.

A brief description of the ML algorithms is provided in the following.

OC-$k(j)$NN is an authentication/anomaly detection algorithm derived from the k-NN classification algorithm [34] that selects the class of a sample to be tested as the most frequent among its k nearest neighbors. OC-$k(j)$NN algorithm is adapted to a single class problem: first the k nearest neighbors, $\{y_1, \cdots, y_k\}$, of the test element x, are found in the dataset, then the j nearest neighbors, $\{z_{i1}, \cdots, z_{ij}\}$, for each of the first k neighbors (i.e., $i = 1, \cdots, k$) are found. The average Euclidean distances, $\overline{D}_{xy}$ between x and its k nearest neighbors, and $\overline{D}_{yz}$between those k neighbors and their own j closest neighbors are calculated.

The element to be test, x is considered an anomaly if $\frac{\overline{D}_{xy}}{\overline{D}_{yz}} > 1$.

In terms of complexity, OC-$k(j)$NN does not require training. During the test phase, assuming a number of extracted features, p, and a dataset size n, the algorithm calculates k times the distance of an element to be tested to every point in the dataset extracting every time that at minimum distance ($\mathcal{O}(nkp)$. Then for the k selected neighbours, it calculates j times distances to other points in the dataset extracting every time the element at minimum distance ($\mathcal{O}(kjnp)$). Then the algorithm calculates the ratio between the average distances. Neglecting the complexity for calculating the distances the complexity is $\mathcal{O}(kjnp)$.

OC-SVM is based on SVM algorithms that use a non-linear function (kernel) to map input data into a space with higher dimensions named the *feature space*, and then find decision boundaries to separate classes. OC-SVM has only one class, the boundary is decided using the available dataset, and any new data that lies outside that boundary is classified as an anomaly. We consider the solution that uses a hyperplane (a plane in m-dimensions) for the decision boundary [68]. During the training phase, elements of the dataset are projected in the feature space using a Gaussian kernel and then they are separated from the origin using a hyperplane minimizing the distance of the hyperplane from the origin. A parameter $\nu \in [0, 1]$ is used as the upper bound for the fraction of elements of the kernel transformed dataset that lies outside the hyperplane so that a low value of ν means that a few outliers are allowed, and the hyperplane is closer to the origin. Moreover, ν represents the lower bound for the number of support vectors that are critical elements of the dataset that define the decision boundary and are used to calculate the distances. SVM is a convex quadratic programming problem with linear constraints. The training complexity of non-linear SVM is generally between $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$ depending on the implementation. The test complexity depends on the used kernel function and the number of support vectors, s, since the kernel function must be computed for each support vector. Hence, the complexity is $\mathcal{O}(spf)$ where f is the complexity of the kernel function.

iForest is an anomaly detection algorithm belonging to decision tree algorithms [46], it is based on an ensemble of random binary trees,

called *isolation trees* (iTree). During the training, T different iTrees are built by splitting the dataset into sub-sets until each partition has only one element or a multiple of that same element.

When the dataset size, n, is big a sub-set of the whole dataset is used with dimension $\psi = min(n, 256)$. iTrees are created by successively splitting the resulting sub-set at each step, randomly selecting an attribute and a value in its range so that the split generates two complementary sub-sets. The path length, $h(x)$, is defined as the number of nodes traversed through the iTree to reach the leaf containing x. The anomaly score is then calculated as $s(x, n) = 2^{-\frac{E[h(x)]}{c(n)}}$ using the path length averaged on all iTrees, $E[h(x)]$, normalized to the average path length $c(n)$ of an unsuccessful search in a binary search tree built over a dataset of n elements [46].

It is expected that features of an anomalous element differ significantly from dataset elements, in particular, an anomaly should have a path length shorter than the average. The anomaly score is compared with a threshold $\rho \in [0, 1]$: values over the threshold are classified as anomalies.

During the training stage, T iTrees are built by recursively splitting the dataset of size ψ. The complexity of the training is $O(T\psi \log \psi)$. The anomaly detection complexity for a single element is $O(T \log \psi)$.

OC-$k$means is a modified version of the k-means clustering [51] algorithm that aims at dividing a given dataset into k clusters where each element is closer to the center of its cluster than to the center of other clusters. This is achieved through an iterative technique whereby the clustering operation is performed several times, using the resulting centers from each previous iteration as a starting point for the next one. We use here a modified version of the k-means algorithm to achieve anomaly detection [47]. This is done by dividing elements of the dataset into two clusters (i.e., $k = 2$). Then the average distance $\bar{D}$ between the two clusters' centers is calculated and used as threshold for the following decision test. During testing operations, received samples and dataset are clustered in two clusters and the resulting distance V between the

two clusters is compared with $\alpha\bar{D}$, where α is a scaling factor. If $V \leq \alpha\bar{D}$ a positive result is assumed, i.e., the received samples belong to Alice, otherwise an anomaly is detected.

K-means shows complexity $\mathcal{O}(npk)$ for each iteration: for each element of the dataset the distance from the k centroids is calculated using a vector of dimension p. In the anomaly detection implementation described before, both training and test phases are based on a dataset clustering with $k = 2$, then distances among clusters are evaluated and compared. Hence, the complexity for both phases is $\mathcal{O}(2npI)$ where I is the number of iterations of the algorithm.

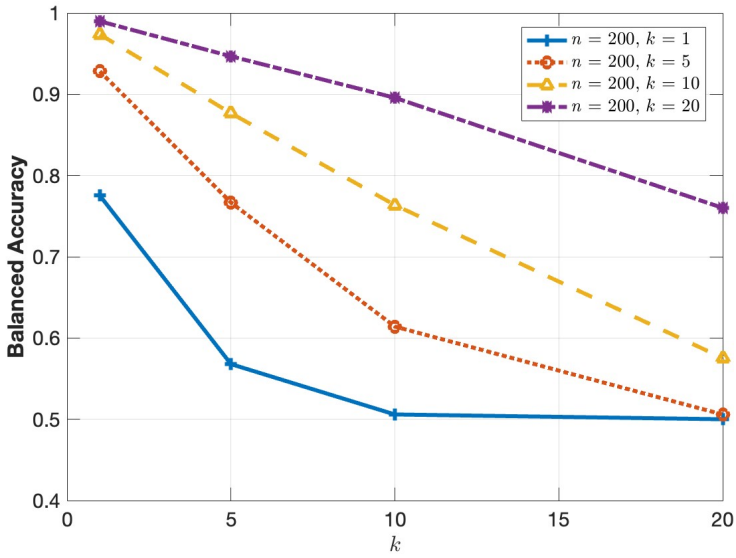
3.2 Numerical Results

The performance of the previously described ML-based anomaly detection methods are presented and compared. Numerical results have been derived through simulations. To have results not depending on a single specific dataset (i.e., a specific position distribution of nodes in the area), results from multiple datasets have been averaged. For each dataset, Eve and Alice are randomly placed with a uniform distribution in a square area $\mathcal{A} = 20 \times 20\,m$, with the sink node in the center. It is assumed that Alice's signal is received with a Signal-to-Noise Ratio (SNR) of 10 dB, while the SNR of Eve is consequently calculated considering its position in the area. The channel model and the probability distribution of the channel attributes have been described in Sec. 2.2, taking into account also their time-variability due to the scatterers' movement up to 5 km/h. The carrier frequency is 5.25 GHz with a bandwidth of 80 MHz.

First of all, we have evaluated the impact of different parameters settings on the detection performance of the algorithms for selecting the optimum ones. Moreover, the impact of the dataset size is evaluated

The OC-$k(j)$NN algorithm depends on two parameters k and j. Fig.3.2 shows the balanced accuracy (BA) vs the parameter k for different values of j. Dataset size is fixed at $n = 200$ samples. The algorithm behaves better when there is a higher unbalance between the two parameters with $k < j$. Indeed, performance increases as $\frac{k}{j}$ decreases up to a certain point, then benefits tend to disappear. With an equal k/j ratio, using a lower value of k gives almost the same performance but

Figure 3.2: OC-k(j)NN. BA vs parameter k varying j.



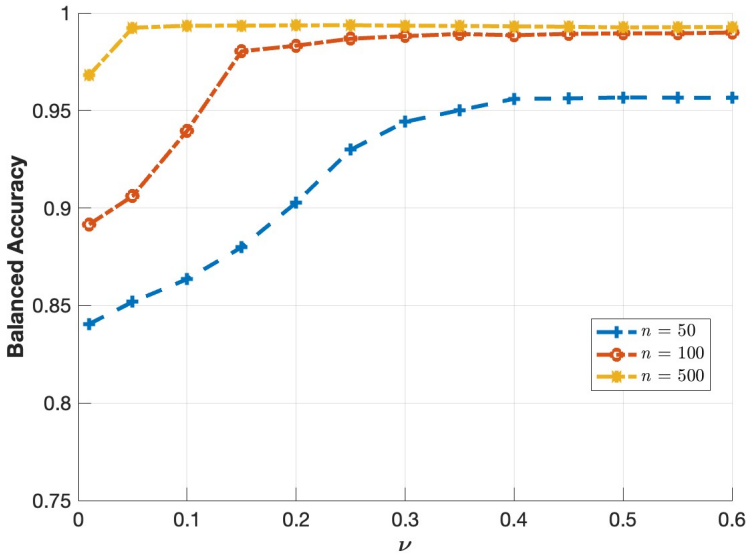
with lower complexity. These results are confirmed by Table 3.1 where the BA is presented for different values of the dataset size, n, for values of k and j in their best ranges. We can see that better performance is achieved with a dataset with limited size ($n \in [50 - 100]$).

Table 3.1: OC-$k(j)$NN: BA vs data-set size n

k		*1*		*5*	
j		*10*	*20*	*10*	*20*
n	30	0.7931	0.9286	0.9829	0.5665
	50	0,9772	0,9903	0,8910	0,9610
	100	0,9749	0,9911	0,8783	0,9534
	200	0,9736	0,9897	0,8766	0,9466
	300	0,9746	0,9898	0,8778	0,9442

OC-SVM requires setting the parameter ν that represents an upper bound for the fraction of outliers of the dataset and a lower bound for the number of supporting vectors. Fig. 3.3 shows BA vs ν for different values of the dataset size, n. We can see that performance significantly

Figure 3.3: OC-SVM. BA vs parameter ν varying n.



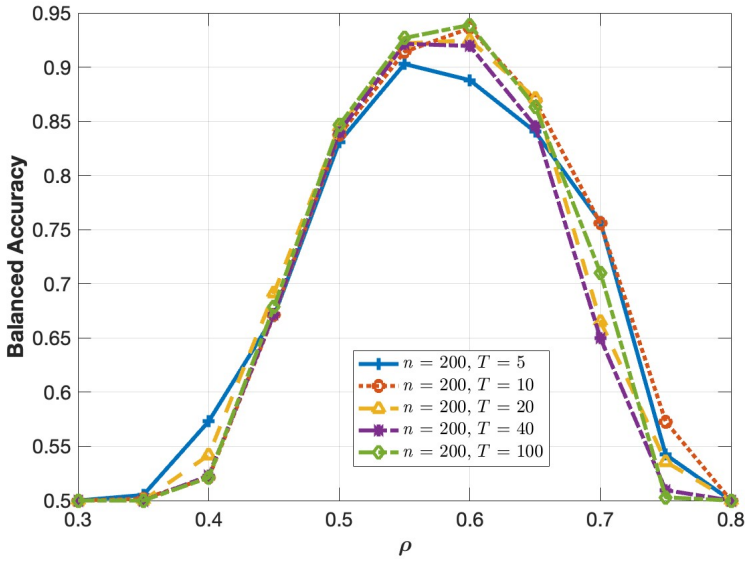
increases with n. For large values of n the impact of ν is not relevant, while for small n values it is preferable to work with higher values of ν, since a higher number of support vectors improves the detection accuracy. However, the value of ν should be limited to limit the computational complexity. Previous considerations can be drawn also from Table 3.2 where BA is reported for different values of n and a selected range of values of ν. Best performance is achieved with a dataset size around $n = 400$ (higher values do not yield relevant benefits).

Table 3.2: OC-SVM: BA vs data-set size n

ν		*0.05*	*0.10*	*0.20*	*0.30*	*0.40*	*0.50*
n	100	0,9046	0,9475	0,9852	0,9880	0,9889	0,9892
	300	0,9411	0,9840	0,9892	0,9902	0,9903	0,9904
	400	0,9918	0,9930	0,9932	0,9932	0,9928	0,9925
	500	0,9928	0,9939	0,9937	0,9936	0,9933	0,9927

iForest requires two parameters, the number of trees T and the threshold $\rho \in [0, 1]$. Fig. 3.4 shows the BA vs ρ for different values

Figure 3.4: iForest. BA vs threshold ρ varying T.



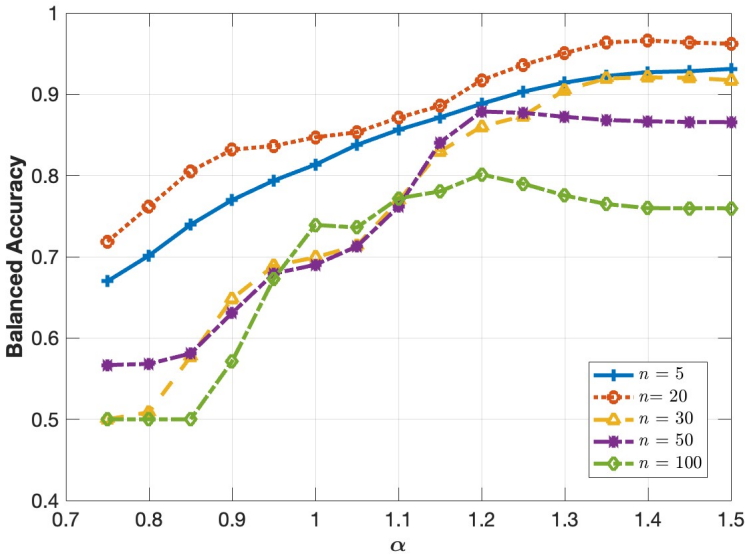
of T and $n = 200$. It can be seen that the optimal threshold is almost the same in all cases and that starting from $T = 10$ increasing the number of trees does not lead to a significant improvement. This is evident also from Table 3.3 where the BA for different dataset sizes is reported varying parameters ρ and T in their optimal ranges. Again, we can see that the number of trees does not significantly affect the performance, while increasing n up to $n = 300$ leads to an improvement, then the performance remains almost constant, thus a further increase would lead only a complexity increase.

Table 3.3: iForest: BA vs data-set size n

ρ		0.55							0.60						
T		5	10	20	40	100	120	140	5	10	20	40	100	120	140
n	50	0.8413	0.8559	0.8610	0.8746	0.8816	0.8881	0.8854	0.8786	0.8863	0.8941	0.8828	0.8924	0.8878	0.8973
	100	0.8384	0.8551	0.8992	0.9069	0.9075	0.9110	0.9095	0.8663	0.8867	0.9025	0.8978	0.9048	0.9097	0.9103
	200	0.90.28	0.9141	0.9219	0.9214	0.9269	0.9289	0.9241	0.8879	0.9359	0.9248	0.9196	0.9388	0.9198	0.9306
	300	0.9021	0.9178	0.9255	0.9277	0.9289	0.9355	0.9304	0.9076	0.9159	0.9217	0.9333	0.9404	0.9414	0.9406
	400	0.8993	0.9101	0.9273	0.9298	0.9344	0.9319	0.9326	0.9199	0.9309	0.9336	0.9290	0.9307	0.9405	0.9321

OC-kmeans algorithm compares clusters' distances using a weight factor α. Fig. 3.5 shows the BA vs α for different values of n. We can

Figure 3.5: k-means. BA vs distance weight α varying n.



see that increasing α there is an advantage up to a certain value, then performance decreases. This worsening is more evident when larger dataset is considered, and, in general, large datasets lead to worse accuracy. This can be seen also from Table 3.4 where BA for different values of n are reported for values of α in its best range.

Table 3.4: OC-kmeans: BA vs dataset size n.

$\boldsymbol{\alpha}$		1.1	1.2	1.3	1.4	1.5
$\boldsymbol{n}$	5	0.8641	0.8799	0.9102	0.9237	0.9272
	20	0.8710	0.9173	0.9506	0.9661	0.9642
	30	0.7687	0.8601	0.9050	0.9207	0.9173
	50	0.7613	0.8786	0.8721	0.8666	0.8655
	100	0.7718	0.8012	0.7753	0.7598	0.7595

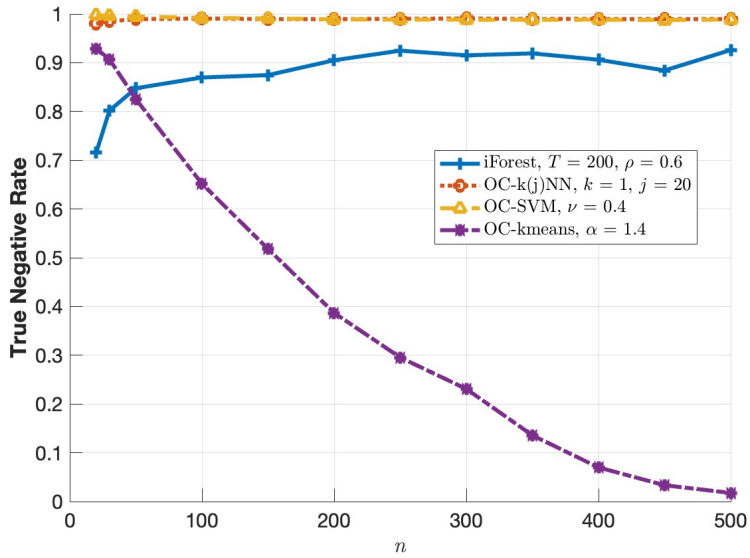
From previous results, we can see that OC-k(j)NN and OC-SVM are those achieving the best BA performance ($\sim$ 99%), while OC-kmeans and iForest reach 96% and 94%, respectively.

It is also interesting to compare the performance of different ML

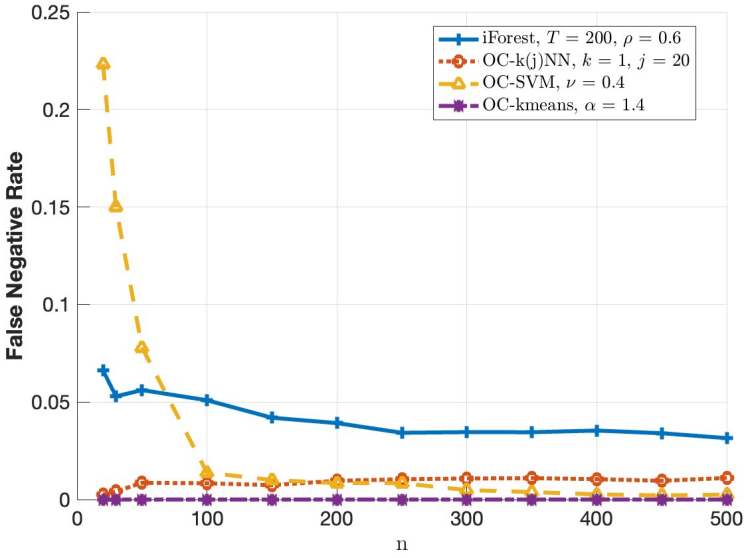
algorithms when their optimal parameters are used. Fig.3.6 shows the *true negative rate* (tnr) and the *false negative rate* (fnr), which are the correct and false detection of Eve, for different algorithms. In terms of fnr, results confirm that OC-SVM and OC-k(j)NN achieve the highest values that are quite independent of the dataset size. For what concerns the fnr instead, OC-k(j)NN achieves values almost constant with n but that does not go to zero, OC-SVM has values of fnr that depend on the dataset size and go to zero around $n = 400$. OC-kmeans presents a good performance in terms of fnr while tnr strongly depends on n and reaches maximum values around 94%. iForest presents the worst performance in both cases, and even increasing n there is a floor to the performance.

3.3 Conclusions

This chapter presented an ML-based PLA framework for identifying IoT nodes belonging to a WSN and detecting potential rogue devices trying to gain unauthorized access. The identification is based on the WF of the received signal, here characterized by various channel attributes. Different ML anomaly detection approaches have been evaluated and compared in terms of different metrics. OC-SVM and OC-k(j)NN resulted to be the algorithms providing the best performance, even if OC-k(j)NN presents a higher fnr. On the other side OC-k(j)NN has the advantage of a reduced time complexity compared to OC-SVM, it does not require training, achieves good performance with a limited dataset $n \in [50 - 100]$, and has linear complexity with n. On the contrary, OC-SVM has a training complexity that exponentially increases with n, and for having low fnr the dataset should be big. The last two algorithms have limited complexity but present a significant performance worsening, especially the iForest.



(a) True Negative Rate.



(b) False Negative Rate.

Figure 3.6: Eve detection rate and Alice misdetection rate vs dataset size.

Chapter 4

PLA approaches comparison

In this chapter we recall, extend, and compare the two classes of solutions described in previous chapters: classification-based and anomaly detection-based. Different ML approaches for both classes of solutions are compared in terms of detection accuracy, complexity, and parameter settings. In particular, we extend previous studies to a multi-device scenario, adding new classification-based algorithms and providing a comprehensive comparison and analysis. The solutions do not require any knowledge of the spoofing node or statistical models that can be difficult to obtain. Multi-device effects are shown, together with those of the training dataset length and the characterizing parameters. Results show that when the number of nodes is high all solutions achieve good detection performance, while the classification-based algorithms do not have good spoofing detection capabilities in small networks.

4.1 System model

We consider here a WSN composed of low-powered nodes unable to support complex security measures. Furthermore, WSN nodes are typically placed in accessible locations, making them vulnerable to physical tampering. In particular, we consider an area, $\mathcal{A}$, where U wireless sensor nodes are deployed. These nodes communicate with a sink node, which is the coordinator in a star-topology network, as shown in Fig. 4.1.

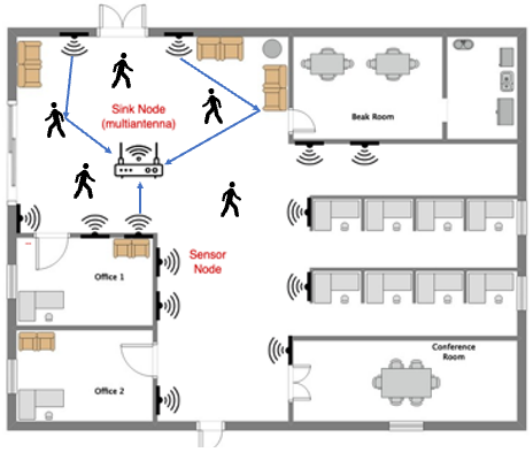


Figure 4.1: System model.

The wireless sensor nodes are presumed to be devices with limited resources in terms of computation, memory, and power. They perform elementary tasks like monitoring physical attributes (e.g. humidity, temperature, vibration, and pressure) and transmitting them to the coordinator using a single antenna transceiver. In contrast, the network coordinator (i.e., the sink node) is a device able to handle more complex functions such as managing network authentication, regulating access for sensor nodes, and undertaking initial data processing from the sensors. Consequently, the sink node is expected to possess greater computational prowess, enhanced energy resources, and expanded memory capacity. Moreover, it is assumed that the transceiver module within the sink node is outfitted with multiple antennas. This configuration allows for spatial information to be accessible at the coordinator's end.

For the channel model, we refer to 2.2.

4.2 Proposed authentication and spoofing detection system

As in previous chapters we present different solutions for providing continuous authentication and spoofing detection in a WSN where the sink

node tries to identify the sensor nodes belonging to its own network, by extracting the physical-layer attributes from each incoming signal (i.e., WF) and applying a ML approach to decide if the node is legitimate or it is not. The WF is here characterized by $M = 4$ different attributes: *(i)* RSS, *(ii)* angle of arrival (AoA), *(iii)* maximum delay, τ_L and *(iv)* the signal energy. Particularly, here we compare the two classes of solutions already presented in previous chapters: *(i) classification*-based and *(ii) anomaly detection*-based. Here, we extend the solutions to a multi-device scenario and we consider additional classification algorithms. Moreover, we try to provide a comprehensive comparison between the two classes and the different ML algorithms. As already stated, the classification-based solutions resort to ML classification algorithms that for definition are multi-class, and hence, can be used in a multi-device scenario where each class corresponds to a node of the network. Indeed, these ML algorithms aim at assigning each element to be tested to one of the known classes based on patterns extracted from data used for training. During the training phase, the classifier is fed with U input datasets with dimension N, that consist of channel attributes and the corresponding classes, (i.e., the IoT node identification code - ID). The classifier analyzes patterns and relationships between channel attributes and the node ID and builds a model that can be used to predict the class of new unseen data. Since assuming knowledge of malicious users' data can be unrealistic, only authorized users' data are used for training and creating classes. Consequently, this kind of algorithm is not able to directly detect a malicious node, that would be in any case classified as belonging to one of the legitimate nodes' classes. An additional step is needed to detect malicious users as detailed later. Conversely, anomaly detection solutions are based on ML algorithms that identify data that do not fit a previously known pattern, so that it is possible to identify spoofing nodes. However, this class of algorithms is designed for a single-class scenario, since differentiates between data of an authorized node and other data. They need to be adapted to the specific scenario where multiple legitimate nodes are involved. For both solutions, to have an exhaustive comparison, we have considered different ML algorithm types: kernel-based, nearest neighbors, clustering, and binary tree.

4.2.1 Classification-based solutions

The first group of considered solutions is based on ML classification algorithms (*classification*-based). These are *native multi-user schemes* able to assign each received data to a predefined class (belonging to an authorized node). However, when data from a malicious node is received, the ML algorithm is not able to identify the anomaly, since the output of the classifier is always one of the authorized classes. The spoofing detection needs a second step: the output of the classification algorithm is crosschecked with the node ID, if the ID corresponds to the class detected by the ML algorithm the node is authorized, otherwise, the node is blocked. More in detail:

1. *INITIALIZATION PHASE*: nodes authenticate using higher layers procedures and a unique ID is assigned to each node.

2. *TRAINING PHASE*: data (i.e., training dataset) from each authenticated node (labeled with its own ID) are collected during this phase and are used to train the ML algorithm.

3. *TEST PHASE*: during normal communications, nodes label the transmitted packets with their ID. The sink node classifies each received packet employing the WF extrapolated by the received signal and compares the classification result with the packet's ID. If the ID class label decided by the ML algorithm and the packet's ID match, the node is authenticated, otherwise the node is blocked and needs performing again a higher layers authentication procedure.

While the identification of authorized nodes benefits from the multi-user nature of the considered ML algorithms, the spoofing detection capability is determined only by the number of sensor nodes in the network. Indeed, considering the worst case in which the attacker knows the IDs used in the network and takes the identity of one of them in its sent packets, the greater the number of nodes the lower the probability that the fingerprint of the malicious node could be classified as the ID exactly corresponding to the ID inserted in the malicious node packet. The probability of a successful malicious attack decreases as U increases. Consequently, this solution performs well in dense networks, such as those expected in future IoT applications characterized by a massive

number of devices. However, in 2.3.2 we proposed a solution also for small-size networks (i.e., with a limited number of nodes). The idea is to deploy U_s dummy nodes, called *sentinel nodes*, alongside network nodes, that cooperate to improve the spoofing detection capabilities of the system. Sentinel nodes are low-complexity nodes that transmit only beacon signals to provide additional fingerprints to the system, but they do not perform any other additional task. Sentinels' fingerprints are added as authorized classes, thus the resulting number of nodes is $U' = U + U_s$ and the spoofing detection capability improves. In what follows we briefly describe the ML classification algorithms that have been considered and compared.

CART - for its description we refer to 2.3.1 The training complexity for the CART algorithm is determined by the sorting operation, which must be repeated up to $NU/2$ times[1] for each attribute, hence it is $O(M(NU)^2 \log_2(NU))$. Test complexity is $O(d)$ where d is the depth of the tree that depends on the number of splits that is selected proportional to the number of nodes as qU.

k-Nearest Neighbor (k-NN) labels the element to be tested with the ID of the class to which most of its k nearest neighbors belong to. It calculates the distance (i.e., here the Euclidean distance is considered) between the element to be tested and all elements of the initial dataset and then takes the k nearest [6, 18]. k-NN algorithm has no training. The test complexity depends on the need to calculate the distances of the element to be tested by the NU elements of the initial datasets using a vector of M attributes and searching the k nearest neighbors, that is $\mathcal{O}(MNU + NUk)$, neglecting the complexity for calculating the distances. Alternatively, the k nearest neighbors can be extracted successively by calculating every time the new distance $\mathcal{O}(MNUk)$.

The first solution requires memory to store the distance for searching the k nearest neighbors but has lower complexity.

Support Vector Machine (SVM) is a supervised algorithm whose aim is finding a hyperplane (a plane in h dimensions) to separate data points into two classes [9]. The training searches for the hyperplane that maximizes the distance between *support vectors* of

[1] With some specific precautions it can be reduced to $O(MNU \log_2 NU)$.

each class, which are the data points with the minimum distance to the hyperplane. Then the hyperplane serves as the decision boundary for classification. Often a linear solution does not exist, hence, elements of the dataset are transformed employing a kernel function into a higher-dimensional space where the data might be linearly separable. Since,SVM is originally a binary classifier, the multi-class problem is broken down into multiple binary classifiers. In particular, in this work, we chose to use the one-to-one model, where a binary classifier for each pair of classes is instanced and trained (i.e., a hyperplane separating every two classes is identified). Being U the number of nodes, $\frac{U(U-1)}{2}$ instances ofSVM are needed. The classification result is given by the majority score of all classifiers. The training complexity ofSVM requires solving a quadratic programming problem, with available solutions having a complexity between $O(N^2)$ and $O(N^3)$ for each binary classifier instance. Once trained, the classification complexity is $O(sMf)$ for each binary classifier instance, where s is the number of support vectors identified by the algorithm, and f is the complexity of the kernel function (in the linear case $f = 1$).

4.2.2 Anomaly detection-based solutions

The second group of solutions considers ML algorithms able to detect anomalous data among received ones. These algorithms usually work in their one-class version, i.e., they are able only to distinguish between data belonging to a single authorized class and the rest. For using these approaches in a multi-device scenario, multiple instances (i.e., one for each authorized node) of the algorithm must be created and each one is trained using only the dataset belonging to a specific node. An instance of the ML algorithm should provide as output a *positive* result if the node is identified as belonging to the authorized node for which the machine has been trained, a *negative* result otherwise. In particular, exploiting the packet's ID, each newly received fingerprint is tested with the algorithm's instance trained with the samples of the node it claims to be. Hence, while U machines must be trained, during the testing phase only one is run.

The anomaly detection algorithms are the same as described in 3.1.1:

- iForest [46].

- OC-$\boldsymbol{k(j)}$NN
- One ClassSVM - OC-SVM
- OC-$k$means

4.2.3 Evaluation metrics

Different metrics are considered to evaluate the authentication/spoofing detection capabilities of different solutions, depending on their specific characteristics. In particular, proposed solutions can perform two tasks, *(i)* identifying nodes of the network and *(ii)* detecting potential spoofing nodes. Hence, possible outputs are:

- *true positive* - the authorized node is correctly identified
- *false positive* - the spoofing node is erroneously identified as an authorized node
- *true negative* - the spoofing node is correctly identified as a malicious node
- *false negative* - the authorized node is erroneously identified as a malicious node

Hence, performance metrics are

- the *probability of correct detection* of an authorized node, P_d^{Auth} (true positive)
- the *probability of missed-detection of a spoofing node*, P_{md}^{Eve} (false positive)
- the *probability of correct detection of Eve*, P_d^{Eve} (true negative)
- the *probability of missed-detection of an authorized node*, P_{md}^{Auth} (false negative). An erroneous classification of the authorized node leads to a *false alarm* since a potential Eve is erroneously detected, and the authorized user is blocked.

For the classification-based solutions, a true positive output is achieved when the classification output matches with the node's ID, and $P_{md}^{Auth} = 1 - P_d^{Auth}$. For what concerns the spoofing detection capability, this

does not depend on the ML classification algorithm but on the number of nodes in the network [52]. Assuming the spoofing node knows the U IDs used in the network, and randomly chooses one of them, the probability of detecting the attack of Eve is $P_d^{Eve} = \frac{U-1}{U}$, hence, the probability of miss detection is $P_{md}^{Eve} = \frac{1}{U}$. In Fig.4.2 is reported the P_{md}^{Eve} in two cases: *1* the spoofing node randomly selects the ID for labeling its packets (*random victim*) and *2* the spoofing node selects the ID of the nearest authorized node *nearest victim*. In the first case, as stated before the P_{md}^{Eve} perfectly matches the curve $1/U$, in the second case the P_{md}^{Eve} is slightly higher, since in this case the selected ID is likely the most similar, hence, it is more probable that the spoofing node is classified in this node class. These results have been presented in [52].

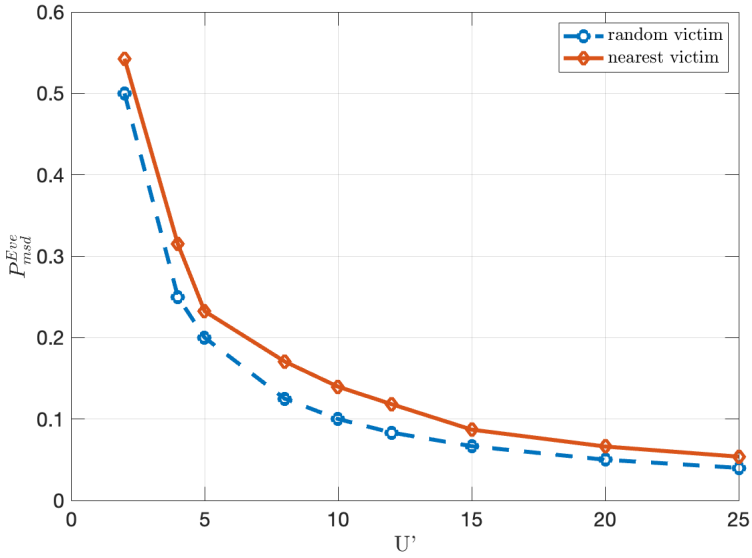


Figure 4.2: P_{md}^{Eve} vs number of nodes U' for the classification-based approaches.

For the anomaly detection-based solutions instead the ML algorithms affect both the probability of correctly identifying a node, P_d^{Auth}, and the probability of correctly identifying Eve, P_d^{Eve}, hence, we define the *Balanced Accuracy (BA)* as the average between the two previous

probabilities

$$BA = \frac{P_d^{Auth} + P_d^{Eve}}{2} \tag{4.1}$$

.

4.2.4 Approaches comparison

In this section the two proposed solutions, *classification*-based and *anomaly detection*-based, as well as different ML algorithms are compared in terms of computational complexity of both training and running phases, and parameters to be optimized. As stated before, in general, classification-based solutions are natively multi-user, except forSVM, so they can distinguish among multiple authorized nodes, but they need a cross-check on the ID to detect spoofing nodes, hence, this capability is strongly related to the number of nodes in the network. In small-network sentinel nodes should be added. The anomaly-based detection solutions are natively able to detect a spoofing node when a single authorized node is present, hence, an ML algorithm instance must be created for each authorized user. Another important element of comparison is how the two solutions react in the presence of changes in the network, in particular when a new node accesses the network. All classification-based solutions require that the ML algorithm goes through training partially or totally, except for the k-NN which has no training.SVM in its One-vs-One implementation requires U new instances, while CART requires training from scratch. As for the anomaly detection-based solutions, only a new instance trained on the new node fingerprint is required.

For what concerns the considered ML algorithms, these present different characteristics, in terms of:

- *parameters* - almost all algorithms require the optimization of some parameters, whose optimal values can be influenced by the training dataset size. Moreover, in general, for the classification-based solution, we can expect that parameters scale with the number of users in the network, while for the anomaly detection-based solutions a trade-off between true negative and false negative occurrences must be found. Indeed, for establishing what is considered an anomaly a decision boundary is needed: a tighter bound usually increases the detection of anomalies (true negative) but it also increases the chance of a legitimate transmission being flagged

Table 4.1: ML algorithms comparison

	Algorithm	Type	Param.	Training Complexity	Test Complexity
Classif.	*CART*	binary trees	# split $q * U$	$\mathcal{O}(M(NU)^2 log_2(NU))$	$\mathcal{O}(d)$
	$k-NN$	NN	k	$\mathcal{O}(1)$	$\mathcal{O}(MNU + NUk)$ or $\mathcal{O}(MNUk)$
	SVM	kernel-based	Kernel type	$\mathcal{O}(\frac{U(U-1)}{2}N^2) - \mathcal{O}(\frac{U(U-1)}{2}N^3)$	$\mathcal{O}(sfM\frac{U(U-1)}{2})$
Anomaly	*iForest*	binary trees	Num.Trees (T), ρ	$\mathcal{O}(UT\psi log\psi)$	$\mathcal{O}(Tlog\psi)$
	OC-$k(j)NN$	NN	k, j	$\mathcal{O}(1)$	$\mathcal{O}(MNkj)$ or $\mathcal{O}(k(MN + Nj))$
	OC-SVM	kernel-based	Kernel type, ν	$\mathcal{O}(UN^2) - \mathcal{O}(UN^3)$	$\mathcal{O}(sfM)$
	$OC-kmeans$	clustering	α	$\mathcal{O}(2UNMI)$	$\mathcal{O}(2NMI)$

as suspicious (false negative). A looser bound will have the opposite effect.

- *training dataset size* - different algorithms require different lengths of the training sequence to achieve good accuracy.
- *computational complexity.*

Table 4.1 details the different characteristics of algorithms.

We underline that assuming that N samples are collected during the training phase for each authorized user, the classification algorithms (with the exception of SVM) require a single machine instance but the training dataset dimension is NU. For theSVM there are $U(U-1)/2$ machines working with a training dataset of dimension $2U$. Differently, for the anomaly detection algorithms, U machines must be trained with a dataset with dimension N, but only one is used for testing.

4.3 Numerical Results

In this section, the detection accuracy of different ML algorithms for both classes of solutions described before is reported. Results are derived using the Matlab software environment through simulations. An area $\mathcal{A} = 20 \times 20\,\mathrm{m}$ representing a large indoor hall with the sink node positioned in the center has been considered. Legitimate and spoofing nodes are randomly placed in the area with a uniform distribution. Numerical results are averaged over several realizations of the same scenario to make the results independent of the specific position of nodes. Each dataset is composed of multiple wireless fingerprints of each node, extracted from channel impulse responses. In particular, the training

dataset is composed of N elements for each authorized node. The radio channel attributes have been stochastically characterized as described in Sec. 2.2 taking into account also their time-variability due to the scatterers' movement. We assume the following channel parameters: scatterers speed falls in the range [0-5] km/h, $\sigma_\tau = 1/\lambda$ with $\lambda = 1.664 \cdot 10^7$ and $\sigma_{AoA} = 1.5849$ [39].

4.3.1 Parameters settings

As stated before some parameters need to be optimized for the ML algorithms. Hence, first of all, the impact and the optimization of parameters and training dataset length have been evaluated.

For the ML classification algorithms, parameters affect only the correct identification of the authorized users (P_d^{Auth}) reported in Table 4.2. The table shows the accuracy of the ML algorithms for different lengths of the training dataset N, as a function of the kernel type for theSVM, the number of neighbors, k, for the k-NN, and the value q that determines the number of splits ($s = q\dot{U}$) for the CART. We can see that using different kernel functions does not significantly change the classification accuracy onSVM, hence, we select the linear kernel since it has the lowest complexity. In k-NN neighbours there is a slight improvement increasing k up to a certain point then benefits tend to reduce. The best choice results to be $k = 5$. Also for the CART algorithm, the optimal choice for the multiplication factor is $q = 4$. For what concerns the input dataset length, N, for all schemes there is an improvement as N increases up to $N \approx 100$, then saturation is reached (i.e., benefits tend to reduce and become even more negligible). This is confirmed even when the number of users changes, as shown in what follows.

Table 4.2: P_d^{Auth} of the classification-based solution as function of ML algorithms parameters, U=20

		ML Classification Algorithms												
		SVM				**k-NN**				**CART**				
		Kernel Type				**#neighbours k**				**split factor q (# splits $q \times U$)**				
		Linear	**Gaussian**	**Quadratic**	**Cubic**	**1**	**2**	**5**	**10**	**1**	**2**	**3**	**4**	**5**
N	20	0.962	0.962	0.959	0.955	0.951	0.951	0.955	0.952	0.911	0.942	0.943	0.943	0.941
	50	0.967	0.966	0.964	0.960	0.958	0.959	0.963	0.962	0.899	0.955	0.956	0.956	0.956
	100	0.971	0.970	0.969	0.967	0.962	0.961	0.966	0.966	0.867	0.959	0.960	0.961	0.961
	200	0.971	0.970	0.970	0.969	0.964	0.963	**0.969**	**0.971**	0.861	0.955	**0.965**	**0.965**	0.964
	500	**0.973**	**0.973**	**0.973**	**0.973**	0.965	0.965	**0.971**	**0.972**	0.851	0.957	**0.966**	**0.970**	**0.970**

For ML anomaly detection algorithms the detection accuracy of both

legitimate and malicious nodes is affected by the algorithms' parameters, hence, the BA is optimized. Table 4.3 shows the BA of the anomaly detection algorithms for different parameters when N varies. For the OC-$k(j)$NN is considered the ratio between the number of neighbors, k/j. The SVM algorithm in its OC version depends on the value of ν that represents an upper bound for the elements of the training dataset outside the hyperplane and the kernel type. The table reports the BA of the OC-SVM when a Gaussian kernel is used. Different kernel types: linear, quadratic, and cubic have been tested but in any case, the BA is significantly lower than for the Gaussian one. In particular, with optimized parameters, the best values achieved are: Linear 0.801, quadratic and cubic 0.499. The iForest algorithm accuracy depends on the threshold ρ and number of trees T while the weight α for OC-$k$means is considered. We can see that for OC-$k(j)$NN it is better to choose a low k/j ratio (1/20) and low values of N. Differently, OC-SVM improves its performance as N increases, while the parameter ν does not significantly affect the performance. Also, iForest requires high values of N and achieves its best performance with a threshold $\rho = 0.6$ and a number of trees $T = 100$. Finally, OC-$k$means best values are achieved with low values of $N \approx 50$ and a weight factor $\alpha = 1.3 - 1.4$. We underline that in the table we have reported only values of parameters in their best ranges due to space limitation. However, simulation have been performed on a wider range of values.

Table 4.3: BA vs dataset size N

		ML Anomaly Detection Algorithms															
		OC-$k(j)$NN k/j				**OC-SVM** Gaussian ν			**iForest** $\rho = 0.55$ T			**iForest** $\rho = 0.6$ T			**OC-$k$means** α		
		1/10	*1/20*	*5/10*	*5/20*	*0.1*	*0.2*	*0.3*	*40*	*100*	*120*	*40*	*100*	*120*	*1.3*	*1.4*	*1.5*
N	50	0.977	**0.990**	0.891	0.961	0.868	0.907	0.940	0.874	0.881	0.888	0.882	0.892	0.887	0.901	**0.918**	**0.913**
	100	0.975	**0.991**	0.878	0.953	0.947	0.985	0.988	0,906	0,907	0,911	0,897	0,904	0,909	0.775	0.760	0.759
	200	0,973	0,989	0,876	0,946	0,984	0,989	0,991	0,921	0,926	0,928	0,919	0,938	0,919	0,618	0,619	0,618
	500	0,971	0,989	0,872	0,939	**0,994**	**0,993**	**0,993**	**0,938**	**0,935**	**0,936**	**0,940**	**0,944**	0,935	0,507	0,507	0,507

4.3.2 Detection Accuracy

In this section, the accuracy of considered solutions is presented. Being a multi-device scenario we want to put in evidence how algorithms behave when the number of nodes varies. In particular, the classification-based solutions are strongly affected by the number of nodes in the area. The detection accuracy of legitimate nodes, P_d^{Auth}, worsens as the number

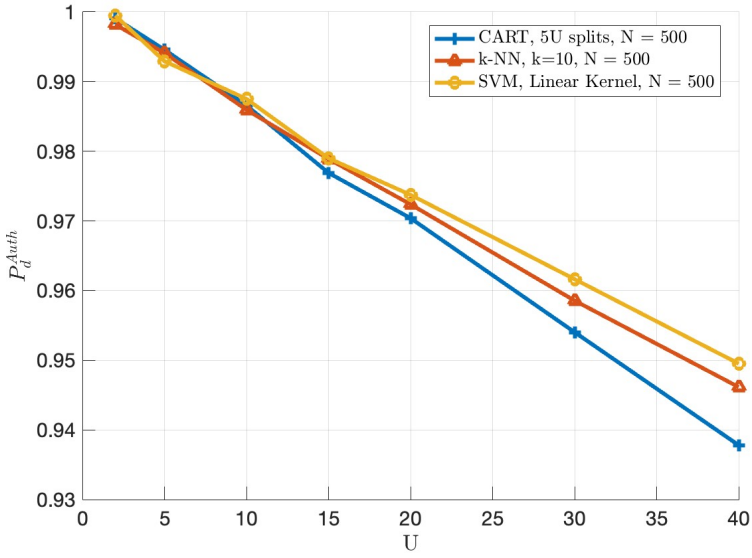


Figure 4.3: P_dAuth vs number of nodes U for the classification-based algorithms.

of nodes in the area increases. This is because the overlapping of the wireless fingerprints of different nodes is more probable if the nodes' density increases, thus, there is a higher probability of misclassification. Fig. 4.3 reports P_d^{Auth} for the classification-based algorithms using the optimal parameters' values seen before.

The figure points out that, for a low number of nodes, all algorithms achieve almost the same performance while, when the number of nodes increases,SVM can achieve better performance, even if all algorithms achieve an accuracy higher than 93% when $U = 40$ (that is 100.000 nodes per km^2). For what concerns the accuracy of the detection of Eve, P_d^{Eve}, it depends only on the number of nodes and not on the selected ML algorithm, as discussed in Sec. 4.2.3 and pictured in Fig. 4.2.

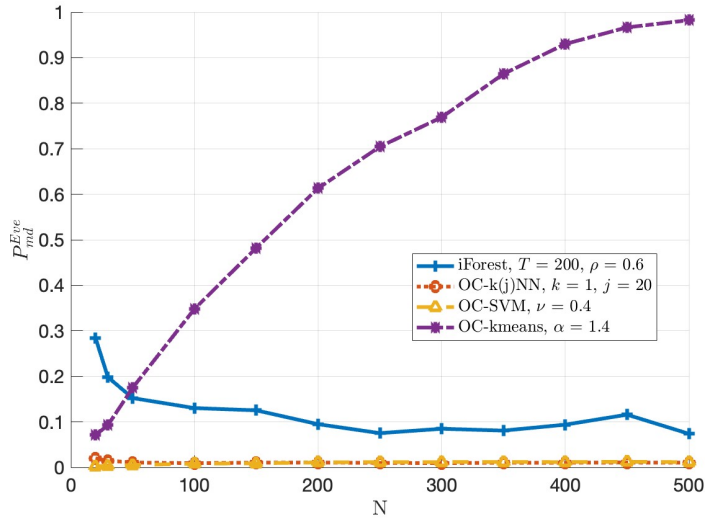
The anomaly-detection algorithms are used in their OC version: the sink node checks the node ID and then uses only the machine instance related to that node. Thus, the wireless fingerprint of the received element is compared only with that of the node with the declared ID. It means that the detection accuracy does not depend on the number

of nodes in the area. Differently, from classification-based approaches, the miss detection of Eve depends on the selected algorithm. Fig. 4.4 shows the miss detection probabilities of spoofing and authorized nodes, P_{md}^{Eve} and P_{md}^{Auth}, respectively. The miss detection of a spoofing node indicates the success of an attack, while the miss detection of an authorized node means an unwanted block of a legitimate node. We can see that OC-SVM and OC-$k(j)$NN get very low values for both probabilities, even if OC-SVM requires a training dataset size higher than $N = 300$. The solution based on binary trees (iForest) is not able to reach very low values in both probabilities, while the OC-kMeans solution is very good in correctly detecting the legitimate node, but the false positive rate increases significantly with N. Finally, for an overall comparison, Fig. 4.5 reports the BA for all considered algorithms (for both classes of solutions). For classification-based solutions, performance is significantly impacted by the number of nodes in the considered area. For a low number of nodes, the spoofing detection accuracy (i.e., P_d^{Eve}) is very poor and this strongly influences the BA even if the authorized users' classification accuracy is high. When the number of nodes increases the BA of all classification-based approaches improves. In particular, when $U = 10$ the performance of iForest is overcome, and for $U = 20$ also the OC-$k$means algorithm performance is reached. However, there is a performance saturation and a slight decrease when U further increases, since the probability of detection of Eve tends to saturate (see Fig. 4.2), while the probability of detection of legitimate nodes decreases (see Fig. 4.3). Anomaly detection solutions based on OC-SVM and OC-$k(j)$NN always achieve better performance around 99% of BA. We can see that, differently from classification-based solutions, the anomaly detection-based ones have constant performance with U.

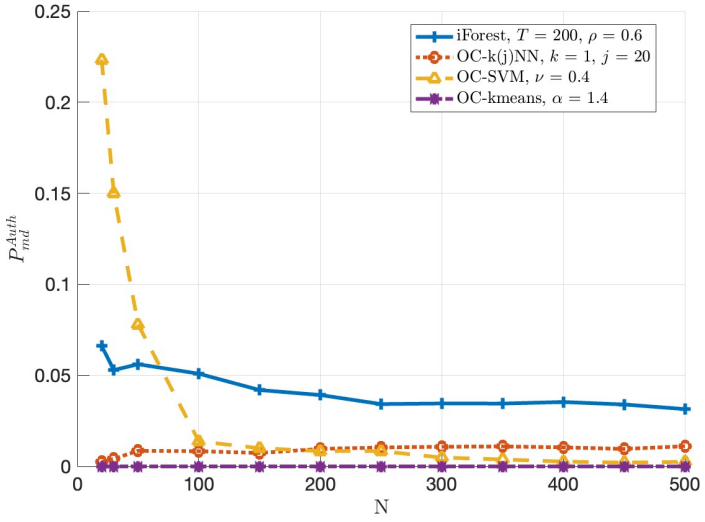
4.4 Conclusions

This chapter presented two frameworks for authentication and spoofing detection, based on wireless fingerprinting. In particular, machine learning classification and anomaly detection algorithms have been analyzed and compared. While classification algorithms are inherently multi-user, they lack in spoofing detection capabilities requiring a cross-check

with the unique node identification code. Differently, anomaly detection algorithms are designed for a three-node scenario and have the inherent capability of detecting spoofing nodes. Results showed that the anomaly detection solutions based on OC-SVM and OC-$k(j)$NN present the best detection accuracy, higher than 99% independently on the number of nodes in the area. Differently, classification-based solutions performance is affected by the number of nodes since on one side increasing the number of nodes improves the spoofing detection capabilities, but on the other side the ability to distinguish among multiple legitimate users decreases. However, classification-based solutions for a number of nodes higher than 20 achieve performance comparable with anomaly-detection solutions based on binary trees and clustering. The binary tree algorithms, in both classes of solutions, provide the worst results. In terms of complexity, even if anomaly detection schemes require the training of one machine for each authorized user, the training dataset size is smaller, and during the test phase, only one machine is run. Consequently, in general, the complexity of the anomaly detection version is lower than the classification version. In particular, OC-SVM presents higher complexity during the training and requires higher values of N if compared with OC-$k(j)$NN that has no training. During the test phase OC-$k(j)$NN has a complexity that is linear with N but requires lower values of N.



(a)



(b)

Figure 4.4: Miss Detection probabilities vs training dataset size N for anomaly-based solutions: (a) False Positive Rate, (b) False Negative Rate

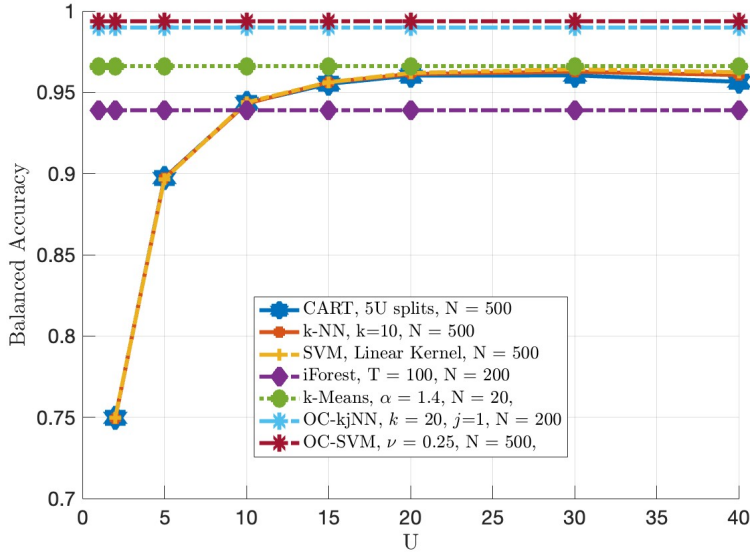


Figure 4.5: BA vs number of nodes U.

Chapter 5

PLA based on ML extended to a mobility scenario

In this chapter we propose a modification of the k-nearest neighbor algorithm to be able to continuously perform physical layer authentication in an environment where the fingerprint changes over time due to the nodes' mobility. This phenomenon is known as concept drift in Machine Learning and requires special attention to be tackled correctly. Also in this case we consider two solutions based on classification and anomaly detection, but we introduce a sliding window to update the dataset and follow the channel variations.

5.1 Introduction

In this chapter the focus is on node mobility and on adapting the ML algorithms to follow the time-variations of the channel that change the wireless fingerprint of legitimate nodes, thus making the correct authentication more challenging. This work delves into the realm of machine learning wireless fingerprinting utilizing the k-nearest neighbor (k-NN) algorithm. The objective is to harness the capabilities of ML to assess the consistency of propagation channel characteristics evolution between previous transmissions by an authorized user and newly received messages. Usually, in the literature, nodes are considered static or the mobility effect is limited to the introduction of the Doppler effect and/or a

variation of the path loss over time. However, these assumptions are not very suitable for actual mobile wireless networks, thus the adaptation of many methods proposed in the literature may be not straightforward. Most of these methods are not capable of following the subsequent channel variations brought on by the changes in the propagation environment due to the changing of the transmitter position. Differently, in this thesis, we have considered an actual mobility scenario. Transmitters move freely while the static receiver stays at the center of the simulated environment. Initially, upper-layer authentication and identity registration are performed and subsequent transmissions are continuously authenticated through ML classification using multifaceted wireless fingerprints. Authentication results are compared to the unique sender IDs attached to received packets.

The key contributions of this work can be summarized as follows:

- Proposing a continuous authentication system for wireless sensor networks with multiple mobile authorized nodes. Unlike prior work on PLA, which often dealt with static nodes or limited mobility, our approach suits dynamic mobile scenarios. Moreover, a multinode environment is rarely considered in the literature.
- Highlighting the challenge of adapting existing PLA methods to mobile scenarios due to difficulties in tracking channel variations caused by transmitter mobility.
- Emulating a wireless sensor network with multiple nodes and one coordinator. Sensor nodes move around while a classification ML algorithm is on, aiming to authenticate the node based on the physical-layer characteristics of its signal at the receiver.
- Authentication accuracy of a sliding window k-Nearest Neighbor (k-NN) algorithm is measured and compared with the performance of the same algorithm in its classical version

5.2 System Model

The Wireless Sensor Network (WSN) is composed of multiple IoT nodes communicating directly with a sink node in charge of gathering elaborat-

ing and possibly forwarding information. The multiple wireless sensor nodes, named $Alice_1$ to $Alice_N$, are transmitting devices tasked with conveying the information gathered by the onboard sensors to the sink node using a radio transceiver. These nodes have to be authenticated by an access point access point (AP), *Bob*. The authentication task is carried out to prevent an illegitimate user, *Eve*, from impersonating any of the legitimate devices, spoofing their identities. Employing the proposed PLA scheme, Bob should be able to identify authorized communications coming from any *Alice* node and to detect those coming from *Eve* and label them as illegitimate. While the access point *Bob* is stationary and equipped with multiple antennas, the devices, *Alices* and *Eve*, can be both static and mobile and have only one transmitting antenna. A variable number of both static and dynamic transmitters has been considered.

5.2.1 Channel Model

The communication channels between Alice and Bob, and Eve and Bob are affected by attenuation due to pathloss, additive white Gaussian noise (AWGN), and time-varying fading. We assume here a multipath fading channel as implemented in the QuaDRiGa Channel Generator. [27,28]. The QuaDRiGa Channel Generator implements the QuaDRiGa channel model as an evolution from the WINNERII channel model. The QuaDRiGa channel model follows a geometry-based stochastic modeling approach allowing the creation of arbitrary double-directional radio channels. The channel parameters, such as delay values and spread, angle spread, and shadow fading are generated stochastically, based on statistical distributions extracted from channel measurements. Different scenarios are modeled by using the same approach, but different parameters for the distributions. The basic features of the model include support of configurable network layouts in the $[0.45 - 100]$GHz frequency range with a supported bandwidth of up to 1GHz, MIMO support, and the capability to simulate the smooth time evolution of large-scale and small-scale channel parameters. The original WINNERII channel model is used to simulate many kinds of wireless communication systems from Local Area Network (LAN) to Metropolitan Area Network (MAN). WINNERII includes different propagation environments for both indoor and outdoor scenarios. It supports both Line of

Sight (LOS) and Non-Line of Sight (NLOS) links and MIMO communication. For a MIMO communication link using T transmitting antennas and R receiving ones the channel matrix H is modeled as:

$$H(t;\tau) = \sum_{m=1}^{TR} H_m(t;\tau)$$

Each H_m is calculated as a TDL with q signal replicas and each replica is expressed as the sum of multiple rays belonging to a cluster.

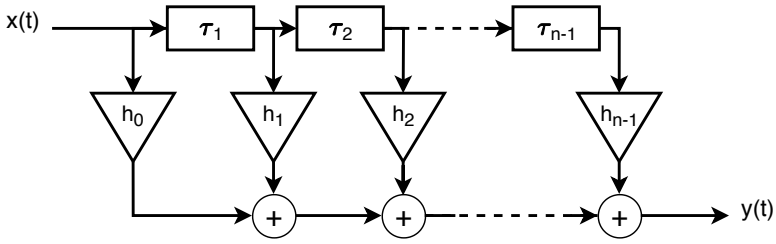


Figure 5.1: TDL model

Because of this, the expression for the i-th ray belonging to the n-th cluster for the link between the s-th transmitting antenna and the u receiving one is

$$H_{u,s,n,i}(t,\tau) = \begin{bmatrix} \boldsymbol{F}_{rx,u,V}(\phi_{n,i}) \\ \boldsymbol{F}_{rx,u,H}(\phi_{n,i}) \end{bmatrix}^T \begin{bmatrix} \alpha_{n,i,VV} & \alpha_{n,i,VH} \\ \alpha_{n,i,HV} & \alpha_{n,i,HH} \end{bmatrix} \begin{bmatrix} \boldsymbol{F}_{tx,s,V}(\varphi_{n,i}) \\ \boldsymbol{F}_{tx,s,H}(\varphi_{n,i}) \end{bmatrix}$$
$$\times \exp(j2\pi\lambda_0^{-1}(\bar{\phi}_{n,i} \cdot \bar{r}_{rx,u})) \exp(j2\pi\lambda_0^{-1}(\bar{\varphi}_{n,i} \cdot \bar{r}_{tx,s})) \exp(j2\pi\upsilon_{n,i}t)\delta(t-\tau_i)$$

where

- $\boldsymbol{F}_{tx,s,V}, \boldsymbol{F}_{tx,s,H}, \boldsymbol{F}_{rx,u,V}, \boldsymbol{F}_{rx,u,H}$ are the antenna radiation patterns for transmitting and receiving antennas and vertical and horizontal polarization,
- $\alpha_{n,i,VV}, \alpha_{n,i,VH}, \alpha_{n,i,HV}, \alpha_{n,i,HH}$ are the polarization gains,
- λ_0 e $\upsilon_{n,i}$ are respectively the carrier wavelength and the Doppler Frequency,

- $\bar{\phi}_{n,i} \cdot \bar{r}_{rx,u}$ e $\bar{\varphi}_{n,i} \cdot \bar{r}_{tx,s}$ are the scalar products between the directions of departure and arrival and the vectors describing the array elements positions.

For what concerns the pathloss, it is modeled with a simple dependency on the distance between transmitter and receiver and the carrier frequency [39]:

$$PL = A \log_{10}(d) + B + C \log_{10} \frac{f_c}{5.0}$$

where $d = 3 < d < 100$ is the distance in [m] between transmitter and receiver, f_c is the system frequency in [GHz] and $A = 18.7$, $B = 46.8$ and $C = 20$.

The proposed ML-based approach also avails itself of channel attributes less commonly used, specifically the delay, which provides information relating to the multipath components, and the Angle of Arrival AoA of the LoS component, which provides additional spatial information. As we operated the QuaDRiGA Channel Generator to simulate the WINNER II [39] channel model we based the implementation of these attributes at the physical layer by relying on the same model.

As such, the delay profile of each channel realization and the AoA are extracted from the simulator and, applying the same solution proposed in 2.2 additional variability is introduced to simulate the effects of noise and and estimation errors. The only difference is that we consider the A1 model from [39] instead of the B3 one, as the previously used B3 scenario is not available in QuaDRiGa, even if an attempt to implement it was made but resulted in failure due to the mathematical implementation behind the simulator.

5.2.2 Mobility & spatial consistency

In the QuaDRiGa Channel simulator, the terminals' mobility is implemented by defining their trajectory over time in the simulation environment. This is done by first defining the trajectory of the node as a union of paths, with each composed of a set of coordinates. Then the overall length of the trajectory is defined and a vector of positions $\bar{p}$ along the track is combined with a vector of time instants $\bar{t}$ resulting in a series of positions over time. Single positions are connected with linear tracks and each track is assumed to be traveled at constant speed

by the terminal to match the $[\bar{p}, \bar{t}]$ pairing. To simulate spatial consistency when a transmitter moves around, the large-scale parameters are initially generated as independent and then correlated by filtering for the entire simulation environment at the beginning of the simulation. Furthermore, in addition to the Doppler shift of the multipath components, present in the WINNERII channel model, the QuaDRiGa channel model extends the mobility support by introducing a concept known as drifting to the small-scale fading model along with a model for the birth and death of scattering clusters. The drifting process involves updating path delays, gains, and angles as the transmitters and receivers move to different locations The birth and death of clusters are implemented by dividing the transmitter trajectory into short overlapping segments. The scattering clusters from the previous segment are smoothly replaced with clusters from the new segment when the transmitter moves from one segment to the next. This process is carried out while keeping the large-scale parameters consistent.

5.3 Proposed System

The overall premises regarding the proposed system are similar to the ones presented in Chapter2 or Chapter 3. In an indoor environment, several nodes transmit towards a central node acting as AP or sink node in LoS conditions. When compared to the previously discussed scenarios the difference is that while some of the nodes are still static, mobile nodes are also present. Specifically, we consider first a single mobile node moving along a designated track at a constant speed as depicted in Fig.5.2, and then we extend the scenario the the case where all transmitting nodes are mobile. The system operates in the same way described in Chapter 2, where after an initial set-up the subsequent messages are authenticated by extracting and classifying their wireless fingerprint. We consider here an algorithm already described in 2.3.1: the *k-Nearest Neighbor (k-NN)* classification algorithm. We choose to consider this algorithm because it does not require training, since it does not have a fixed model but it is evaluated every time it is used. As no real training is actually needed Phase I can be summarized in Fig. 5.3

The main difference in operation is that the mobile nodes' fingerprints are subjected to change over time, due to the node changing its

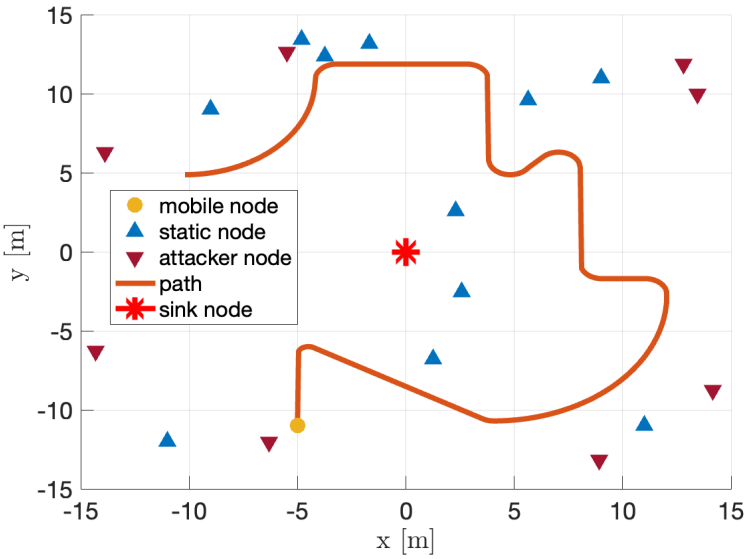


Figure 5.2: Example of a scenario, depicting both legitimate and malicious nodes placement, along with the fixed mobile node trajectory and the network coordinator

location. A general form of this problem is known in the literature as concept drift, and it represents a change in the underlying probability distribution of the phenomena we wish to observe.

5.3.1 Sliding window approach

To allow for adaption to the concept drift caused by the node's mobility the selected ML algorithm was paired with a *Sliding Window* policy. A different approach would be periodically re-training an algorithm. This can introduce excessive overhead and delay. The optimization of such a method would require an evaluation of the performance loss for triggering the re-training. However, this information is not easy to obtain in a real environment. The objective is to maintain a fresh set of samples over which the algorithms are calculated instead of repeating Phase I. The S datasets used for the algorithm implementation are composed of ws samples for each node for a total of Nws samples. Each subset can

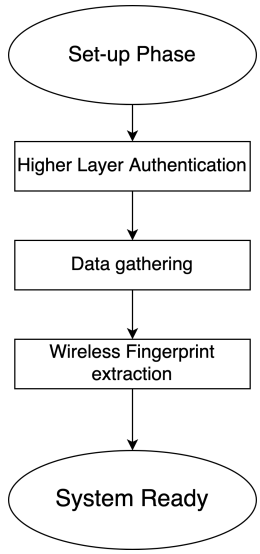


Figure 5.3: Flow diagram summarising Phase I

be described as

$$S^i = \{\bar{s}_1^i, \bar{s}_2^i, \cdots, \bar{s}_{ws}^i\} \tag{5.1}$$

where $\bar{s}_j^i$ is the j^{th} element of the subset of samples S^i belonging to node i. Each element s^i is composed as follows:

$$\bar{s}^i = [a_1, a_2, \cdots, a_m, y_i] \tag{5.2}$$

where a_q is the value of the q attribute and y_i marks $\bar{s}^i$ as belonging to the i class. In the case of a classification algorithm, when the test sample x_t generates a result y_i, matching the sample to the i class, the oldest sample $\bar{s}_{t-ws}^i$ in S^i is discarded and replaced with $\bar{s}_t^i = [\bar{x}_t, y_i]$. In the case of an anomaly detection algorithm, the procedure is the same with the oldest sample being replaced with the new non-anomalous sample. It should be noted that the proposed system features do not feedback on the actual reliability of the outcome of the algorithm, meaning that should the algorithm provide an incorrect solution the data set will absorb the error, affecting all the subsequent decisions until the erroneous sample is discarded and replaced by a newer one.

This procedure is integrated into Phase II as shown in Fig.5.4

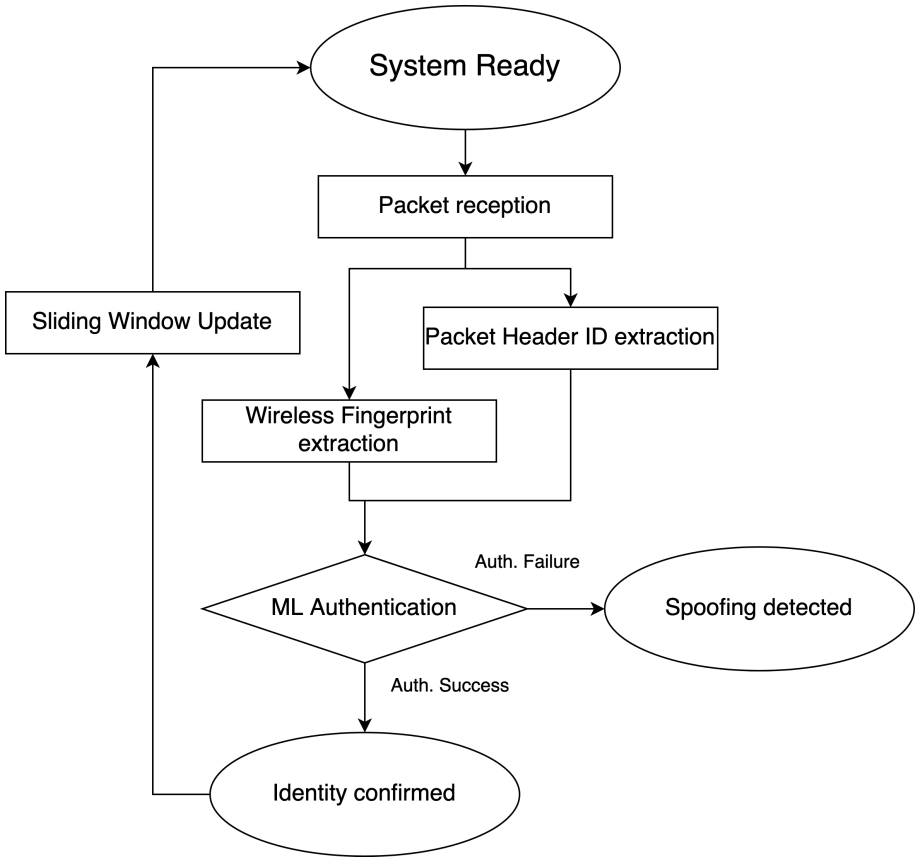


Figure 5.4: Flow diagram summarising Phase II

5.4 Numerical Results

In this section, we numerically assess the proposed approach to adapt to the fingerprint evolution due to nodes' mobility. We evaluate the performance in terms of accuracy, which expresses the system's capability to correctly identify the node by the fingerprints. The core scenario features a variable number of static and dynamic nodes placed in the environment, a 30x30 m indoor open space. The receiver is placed in the middle of the room on the ceiling, 3.5 m from the ground. Although main parameters remain the same we consider different scenarios by

Table 5.1: Parameters for the different scenarios

Scenario	Static nodes	Mobile nodes	Track	Speed	Attackers
A	11	1	fixed, see Fig.5.2	5 km/h	8
B	0	12	linear, random direction, see Fig.5.5	5 km/h	8

varying the number of nodes, their division between static and dynamic, and their movement patterns, as summarized in Tab. 5.1.

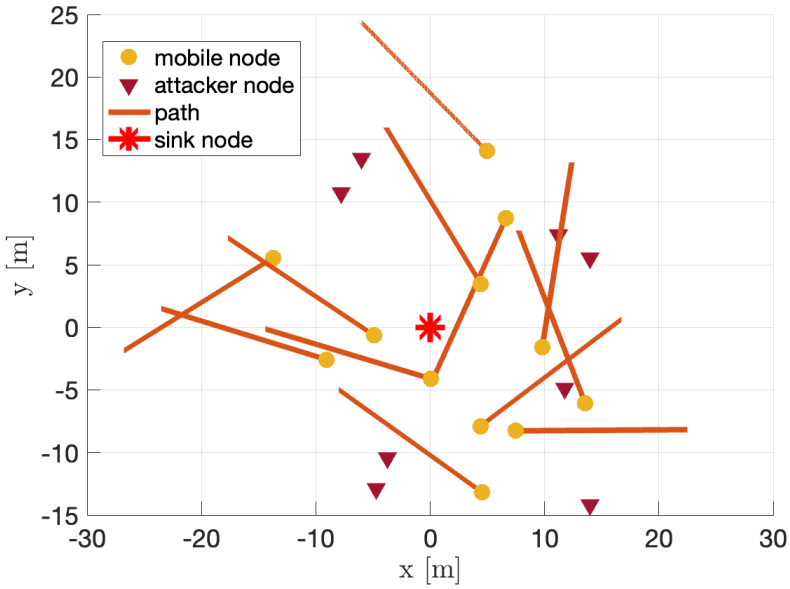


Figure 5.5: Example of scenario B

In scenario A, a single mobile transmitter moves along a predefined trajectory as shown in Fig.5.2. The overall path length for the aforementioned scenarios is 68.23 m. In scenario B, all legitimate transmitters move along a 15 m linear trajectory in a random direction as shown in Fig.5.5.

The node is considered to be transmitting constantly while moving and the channel realizations are generated through sampling with a rate of 10^3 samples each second. The propagation environment's Large Scale Parameters (LSP) are selected from the A1 scenario of the WINNERII channel model. The presence of AWGN is considered at the receiver

with σ^2 selected to guarantee that the node with the worst SNR among all nodes has an SNR value equal to 10dB. All nodes have a single isotropic antenna while the AP has two antennas to perform an AoA estimation. The system operates with a carrier frequency of 5.5 GHz and a 160 MHz bandwidth. We compared the proposed sliding window k-NN algorithm to the original one (i.e., without sliding window).

5.4.1 Scenario A

To begin, we evaluate the classification accuracy of the proposed approach, that is the ability of the system to correctly detect the identity of an authorized node. First of all, we consider the classical k-NN approach using a fixed dataset (i.e., it is not updated). In Fig.5.6a the averaged classification accuracy of the original solution for the static nodes is presented. The average is performed on the whole simulation duration when the values of k and the dataset set size vary. It can be observed that the accuracy is very high, the system has no problem identifying the static nodes in this scenario. In Fig.5.6b the same analysis is presented for the mobile node. A significantly lower accuracy can be observed when compared to the static nodes' performance. This is because when the mobile node moves, the fixed dataset gathered at the start becomes outdated. This is evident in Fig.5.6c that presents the accuracy of the classification over time as the test set is sliced into 200 segments over which the classification accuracy is averaged. The results are presented for different values of k and ws. We can observe that the accuracy is very high at the beginning, then decreases due to the loss of similarity between the received samples and the initial dataset and increases back as the node reaches a region that is closer to the one it was trained on.

Introducing the sliding window solution, we expect a higher ability of the system to follow fingerprint variations. Fig. 5.7a shows the averaged accuracy for all nodes. As we can expect from the proportion between static and mobile nodes, results are very similar to those presented in 5.6a. The proposed system operates with satisfying accuracy in almost all cases considered when classifying static nodes. Moving the analysis to the mobile node, in Fig.5.7b we can see that classification performance varies with the selected parameters. In particular, with a suitable selection of values of k and ws (k=1 and ws greater than 50),

it is possible to achieve almost perfect accuracy. When considering the accuracy over the channel segments as shown in Fig.5.8a, it can be seen that the system is capable of locking on to the fingerprint evolution and show to be capable of continuously authenticating both static and mobile nodes.

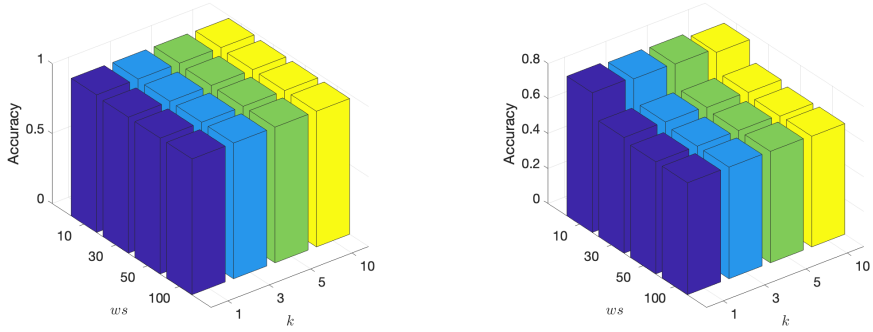
5.4.2 Scenario B

To further demonstrate the effectiveness of the proposed solution we now consider scenario B where, although along less complex tracks, all the nodes are mobile.

In Fig.5.9a the averaged accuracy of the system using a fixed dataset is presented; when compared to Fig.5.9b where the sliding window approach is implemented. A significant difference in performance can be observed. The improvement provided by the proposed solution is even more evident when comparing Fig.5.10a and Fig.5.10b. In the first one, the classification is performed over a fixed dataset, hence, the accuracy over time decreases. Differently, in the second the sliding window solution is adopted, and it shows the capability of correctly following the wireless fingerprint variations even in the presence of more mobile nodes.

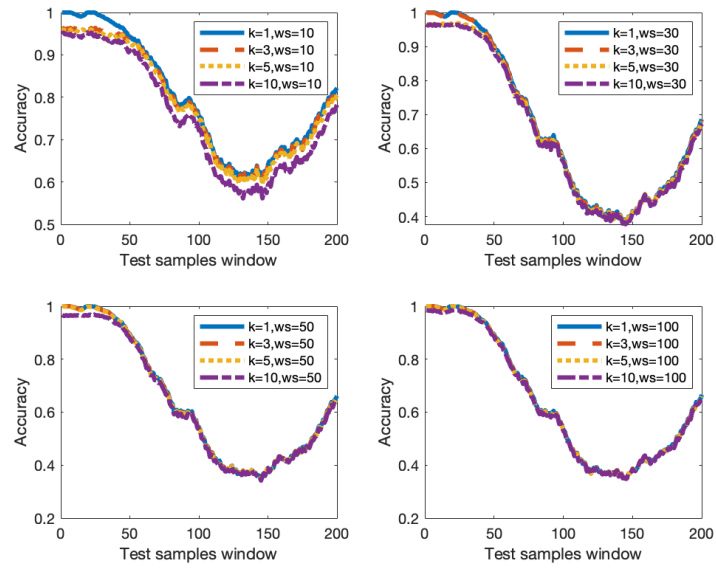
5.5 Conclusion

In this research, the focus has been on addressing the wireless fingerprint evolution issue in PLA due to nodes' mobility. The evolution of the fingerprint in the classification problem represents a case of concept drift traditional algorithms lose accuracy when this phenomenon happens. To counteract the accuracy loss we present a sliding window approach to the k-NN algorithm aimed to update the fingerprint baseline as an alternative to repeating the training. We examined the proposed solution performance in terms of classification accuracy for both the static and mobile nodes and compared it to the original k-NN solution proposed in 2. From the result analysis it emerged that while both solutions perform adequately when classifying static nodes, when the target of the authentication is a mobile node, the proposed solution manages to maintain a high accuracy over time while the original one does not adapt to the concept drift and experiences a significant loss of accuracy.



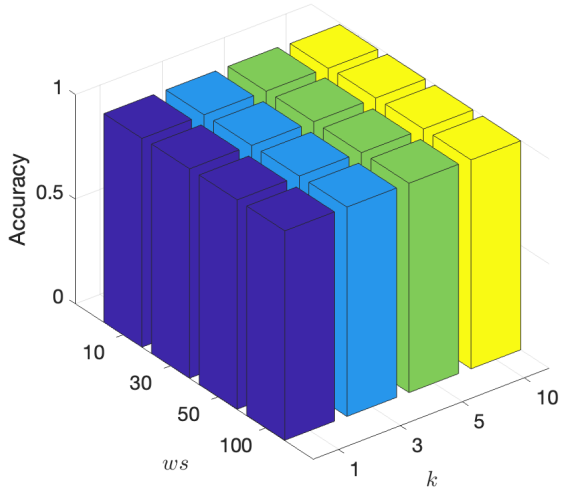
(a)

(b)

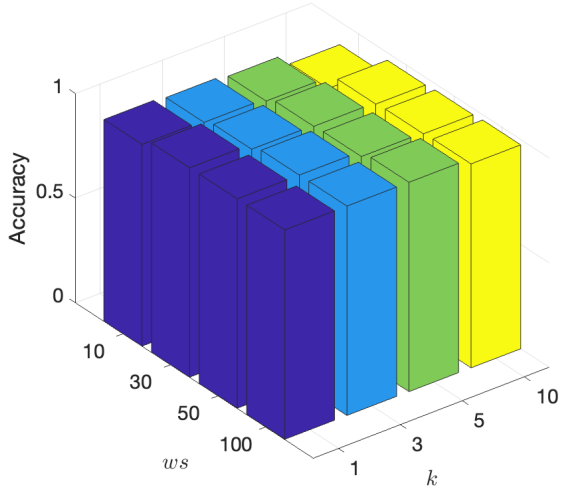


(c)

Figure 5.6: Scenario A - (a): Averaged classification accuracy for all nodes with fixed dataset (b): Averaged classification accuracy for the mobile node with fixed dataset, (c) Classification accuracy for the mobile node over time

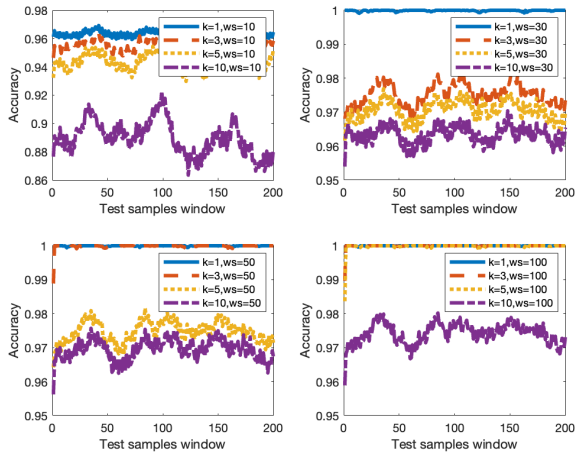


(a)



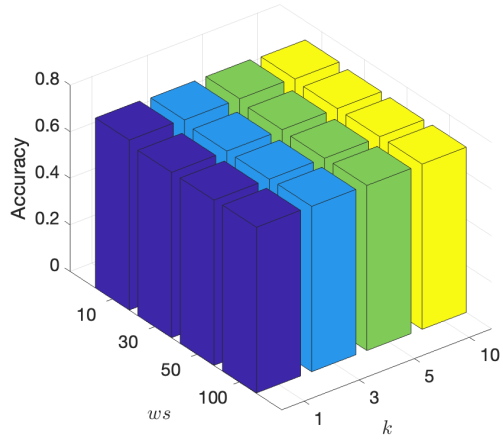
(b)

Figure 5.7: Scenario A - (a): Averaged classification accuracy for the static nodes using the sliding window approach (b): Averaged classification accuracy for the mobile nodes using the sliding window approach

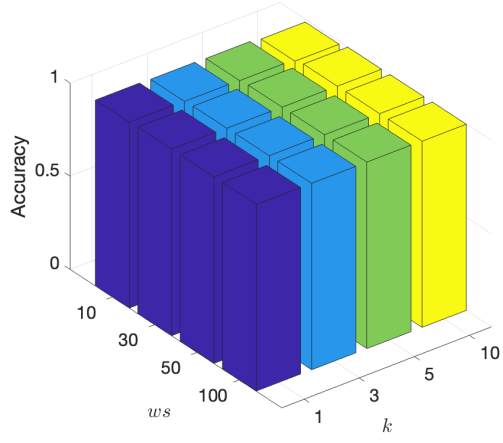


(a)

Figure 5.8: Scenario A - Classification accuracy for the mobile node evaluated over time using the sliding window approach

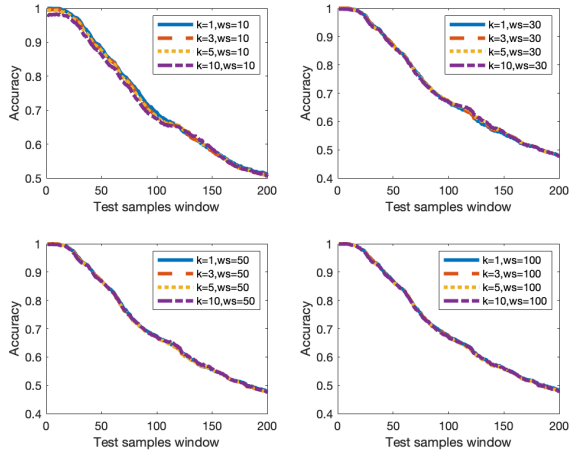


(a)

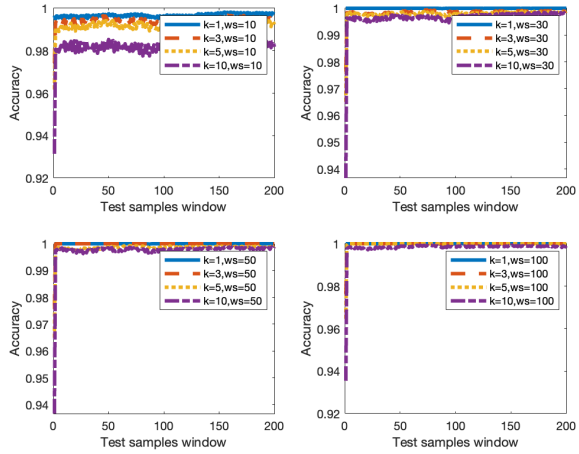


(b)

Figure 5.9: Scenario B - (a): Averaged classification accuracy using the fixed dataset (b): Averaged classification accuracy using the sliding window approach



(a)



(b)

Figure 5.10: Scenario B - (a): Averaged classification accuracy over time using the fixed dataset (b): Averaged classification accuracy over time using the sliding window approach

Part II

Machine Learning for safe and reliable communications

Chapter 6

URLLC

In this chapter we discuss the possibility of applying Machine Learning to channel quality prediction in an Ultra-Reliable Low-Latency Communications (URLLC) context. URLLC is one of the use cases defined by the 5G New Radio standard. Its goal is to provide support for a wide range of services that need very low latency times. The target applications require a quality of service (QoS) that is very different from that provided by older generation mobile networks in terms of both packet error rate, with a probability of 10^{-5} *or less, and a latency, where a less than 1 ms threshold must be met. In the scenario considered the allocation of communication resources in the up-link route between UAV and base-station is examined. To meet the URLLC requirements the Channel Quality Indicator is constantly evaluated. As the information provided by the CQI is an estimate of the current channel condition but communications take place after the CQI's received the information is subjected to aging and might not be descriptive of the actual channel condition when the transmission occurs. The idea examined in the research considers the possibility of training a Long Short-Term Memory (LSTM) network by using multiple sequential instances of the CQI to predict the behavior of the channel during the transmission window from one CQI to the next. The prediction is used to provide extra transmission resources to those*

users whose channel quality is likely to degrade during the next interval allowing them to avoid failing the transmission and achieving greater communication efficiency to meet the URLLC QoS requirements.

6.1 Introduction

Unmanned aerial vehicles UAVs are gaining increasing significance across various domains and, while initially utilized primarily in military applications, they are now capturing widespread attention in a diverse range of fields [1]. These applications include communications, such as relay and mobile base stations (BSs), safety, e.g. fire detection, traffic management, emergency search and rescue, and many more [20, 40]. In recent years, significant research efforts have been devoted to UAV communications, with a primary focus on the data side communications (i.e., payload data) and optimizing UAV trajectories or placements for coverage. The diffusion of UAV applications has presented both opportunities and challenges [96], with a significant one being the effective management of the control traffic, as relatively little attention has been given to issues concerning the control link between UAVs and their ground control station (GCS) such as latency and reliability of the communication occurring over it. This is fundamental as the UAV control link is responsible for supporting safety-critical functions, such as providing real-time information to prevent accidents and collisions. For this reason, the exchange of short control data packets with extremely high reliability and very low latency becomes paramount [71]. To enable the safe, effective, and widespread use of UAVs, the novel challenges presented by control link requirements to support safety-critical functions must be faced. The requirements for the exchange of UAV control information overlap significantly with those of the fifth generation (5G) ultra-reliable low-latency communications (URLLC) service class. URLLC is essential for UAV control applications as it allows for the reliable and timely transmission of data and commands, which is necessary for the proper functioning of the systems [66,71]. URLLC is a term used in the context of 5G and future communication networks and is one of the three main communication services defined by the International Telecommunication Union (ITU) for 5G, alongside enhanced Mobile Broadband (eMBB) and

massive Machine Type Communications (mMTC) URLLC is fundamental for many mission-critical applications, Smart Industry automation, tactile Internet, vehicle communications, e-Health remote surgery, and the aforementioned exchange of UAV control information. In [2] Third Generation Partnership Project (3GPP) has defined the basic URLLC requirements with latency and reliability targets for 1 ms user-plane latency with 10^{-5} reliability in terms of packet loss, although some applications exist with even tighter requirements. Due to the elements characterizing wireless communications like path loss, signal shadowing, and rapid signal fading meeting the URLLC's requirements is an intricate task. As illustrated in [8,42] to meet the stringent requirements for URLLC implementation employing advanced techniques throughout many different parts of the 5G system is paramount. As such, many new technical solutions presented must be adopted, including, but not limited to, new numerologies and Transmission Time Interval (TTI), a different slot/mini-slot structure, link adaptation, the use of Low Density Parity Check (LDPC), and diversity techniques to improve reliability with many of these solutions being native to 5G networks To meet the latency requirements, URLLC services use short block lengths, which limit the coding gain. Since Shannon's capacity bound, which is based on coding performed over an infinite block length, is not applicable in this context, the decode error probability cannot be made arbitrarily small [62], thus affecting reliability. The inverse is also true, as prioritizing reliability demands additional resources, in the form of redundancy for example, which increases latency. Recently, research papers have evaluated URLLC in the context of finite block lengths, in an attempt to optimize block lengths or evaluate error rates, including within the framework of UAV control links [64, 65, 99]. The need to achieve very low target error probability focuses attention on link adaptation, particularly the selection of the ideal modulation and coding scheme modulation and coding scheme (MCS) under the spotlight. Link adaptation operates using the knowledge of the current state of the downlink channel, obtained through the channel quality indicator channel quality indicator (CQI) feedback provided by the user equipment user equipment (UE). The base station base station (BS) uses this information to adjust the transmission rate, therefore an inaccurate or outdated CQI, leading to the selection of a sub-optimal MCS, will im-

pact the performance of URLLCs negatively. CQI aging is, therefore, a critical issue for URLLC, but a trivial solution of increasing the report frequency would both reduce the throughput, due to the increase in signaling, and disjoin URLLC from 5G as a use-case. Existing studies, as highlighted in the review presented in [31], have examined the issue of CQI aging in the context of multiple-antenna and Orthogonal Frequency Division Multiplexing (OFDM) systems, primarily focusing on security and resource management concerns but similar strategies to mitigate the impact of CQI aging on URLLC have not received extensive attention. One proposed approach, discussed in [60], involves providing CQI feedback corresponding to the anticipated worst-case Signal-to-Noise Ratio (SNR) before the next CQI update. This approach entails filtering the channel to estimate the tail of experienced SNR conditions but carries the risk of adopting an overly conservative CQI value, reducing spectral efficiency.

6.1.1 Related Works

The topic of URLLC has gained attention as a use-case included in the 5G standard and to achieve the requirements and improve the overall quality of the communications research has been carried out. In [13] researchers focus on the reliability requirement, specifically for IoT devices, presenting an adaptive K-Repetition (K-Rep) control scheme combined with site diversity reception for uplink Grant Free (GF) URLLC. The use of site diversity reception improves the received signal quality and increases the reliability by using multiple-cell reception while the adaptive K-Rep control scheme adjusts the number of repetitions depending on the UE situation for the K-Rep scheme. Similarly in [70] an approach to select the number of transmission attempts for K-Rep, along with the MCS to provide a high network capacity for uplink periodic and sporadic URLLC traffic is presented. The algorithm employs channel measurements available at the gNB and considers the possible interference between transmissions of different UEs and the features of the gNB receiver such as the possible usage of the SIC mechanism. This paper [21] presents a study on MCS selection and spectrum allocation to support URLLC, exploring the connection between the URLLC requirements, MCS selection, and spectrum allocation to establish bounds for achievable rates. The aim is to exploit said connection to perform

better MCS selection and spectrum allocation to meet the delay and reliability requirements of URLLC. A theoretical analysis model is generated by considering many necessary elements affecting URLLC transmission. Theoretical bounds, such as maximum delay given the allocated bandwidth and minimum required bandwidth given delay and reliability constraint, were obtained. The model was then used to discuss the adaptive MCS selection thresholds and admission region under URLLC constraints. The researchers in [57] have investigated a power and rate adaptation problem for URLLC with Hybrid Automatic Repeat reQuest (HARQ) with statistical CSI. They employed a HARQ transmission scheme combined with Deep Reinforcement Learning (DRL) approach to minimize the long-term average transmit power based on the dynamic queueing system, within the bounds of the URLLC requirements. [35, 100] present a different approach for URLLC, where the link reliability is substituted by service availability, as the focus is shifted towards dependability, to minimize errors burst whose presence might interrupt the communication for a time longer than what the system can stand. This issue is addressed by presenting strategies that are designed to guarantee end-to-end dependable industrial wireless control, including specific scheduling and link adaptation policies. This different paradigm is fit for industrial wireless control systems but its possible extension to all URLLC applications should be investigated with attention. These articles describe a different approach from the one we propose as they operate on a different aspect than ours, as the focus of our research is oriented toward channel prediction to mitigate the discrepancy that might arise from the mismatch between MCS and channel quality. Numerous research papers in the field have tackled channel prediction, with conventional prediction methods relying on knowledge of statistical models, which can be time-consuming, especially in rapidly changing channels. Among these models, Auto-Regressive (AR) models are well-suited since they approximate future channels based on past ones and can directly estimate model coefficients in time-varying channels, even if they are susceptible to noise and interference [29]. As a re-emerging and growing technology solution, the application of Artificial Intelligence (AI) for prediction and forecasts has been widely researched. There are several existing studies on different contexts where the performance of AI-based systems has been

evaluated for predictions, from fault prediction to stock market analysis. Recently, deep learning techniques have gained attention due to their data-driven nature, which does not rely on pre-defined models. Various neural network structures, including Recurrent Neural Network (RNN), have exhibited strong capabilities in time-series prediction. As such, they have found extensive use in channel prediction, as evidenced in studies such as [31] and [29, 38] and a comprehensive review of RNNs for channel prediction can be found in [31]. It's worth noting that these approaches are not explicitly tailored for URLLC and provide estimates of future channel coefficients based on received data, often leveraging multiple features of the data samples and typically designed for systems employing MIMO, OFDM, and Frequency Division Duplexing (FDD) techniques that exploit spatial and/or frequency correlations. In contrast, our approach focuses solely on predicting CQI variations and is based on the use of LSTM. Furthermore, the predictor outlined in [31] and [30] is intended for MIMO channels and operates on data samples to estimate channel coefficients, rather than CQI values. The use of RNN for channel quality prediction and forecast is also considered in [29] where the use of RNN is discussed and compared to other traditional solutions like AR models, showing better performances for the former. In [22] the topic of ML-based channel prediction for URLLC providing insight on the effect of different instances of Geometry-based Stochastic Channel Model (GSCM) on the algorithm performances is explored. The research considers both AR and DL based algorithms, whose prediction is based on a trained neural network. It presents results on how the performance varies significantly depending on the level of abstraction used to represent the channel and on the effects of retraining to keep up with the channel evolution featured in the channel model adopted. This time, the AR model is shown to have better performance due to an easier re-training for channel adaptation. [7] also briefly reviews the application of Machine Learning algorithms for channel prediction. Different approaches for time, frequency, and spatially correlated channels are presented along with a novel approach, similar to triangulation, to extrapolate channel quality information of nodes, for which there is direct information on channel quality. Regarding time-correlated channels, RNN is indicated as better performing thanks to the feedback connections that allow the exploitation of the sequence

properties. [91] explores a variety of models for time series prediction in the context of data traffic volumes. Auto-Regressive Integrated Moving Average (ARIMA), RNN, LSTM, and Hidden Markov Model (HMM) models are compared for the task of predicting changes in traffic volumes starting using the spatial correlation of adjacent BS and channel quality prediction. For the latter, LSTM is shown to perform better than the proposed alternatives. Authors in [37] present a prediction model for the CQI starting from the Received Signal Strength Indicator (RSSI) applied to the vehicular environment of IEEE 802.11p. The prediction is carried out using robust LSTM network to RSSI sequential data and the proposed model is compared with ARIMA, support vector regression, and multi-layer perception models. The proposed model offers improved channel prediction capability than the conventional time-series data prediction models it is compared to. CQI prediction specifically designed for URLLC is explored in [23], where a Convolutional Recurrent Neural Network (CRNN) is employed. The paper describes the implementation of a software-defined radio but does not provide in-depth details regarding the model and framework used, and the analysis is limited to a few specific cases.

6.1.2 Contribution

In this research, the effectiveness of a resource allocation scheme for multi-user UAV networks in the context of URLLC under finite block-length conditions is examined. The proposed scheme relies on a CQI forecast mechanism tailored to support the exchange of UAV control information. For the CQI forecast Recurrent Neural Networks (RNN), in particular, Long Short-Term Memory (LSTM) networks are employed. The outcome of the forecast process is used as a basis for the resource allocation policy to assign the links that are more likely to have a negative mismatch between the selected MCS and the actual channel state when transmitting additional resources to prevent errors. Neural networks are employed for channel forecast using previous CQI information to prevent errors in the URLLC context. This difference in approach is given by the use of a unique set of features based solely on the data of previous CQI values expressed as a time series, as similar studies on channel prediction typically use channel coefficients as inputs and outputs. This approach is well-supported by the 5G infrastructure system,

where UEs provide CQI feedback to the BS and, by implementing the neural network at the BS, it can avail itself of a large pool of computational resources. This research investigates solutions to mitigate the effects of CQI aging based on the combination of a channel quality forecast mechanism using both LSTM classification and regression. The neural network operates using CQI values as inputs and its output, in case of classification, is a simple forecast of whether the future channel condition will support the MCS's choice made on the last received CQI or not. When considering regression the network's output is, instead, the next SNR value. The result of the forecast is not used to select a different MCS, as it would be done in rate adaptation, but to allocate additional resources to prevent packet loss by Maximum Ratio Combining (MRC). An analysis of the scheme's performance is presented for a multi-user scenario to show that the resource allocation scheme based on the channel quality forecast can help reduce the error rate by avoiding a mismatch between the chosen MCS and the actual channel quality. The resource allocation is carried out by considering the output of the LSTM network and assigning the available resources on a side channel to reduce the error rate. When considering the classification, UAVs with a negative outlook and higher classification score are prioritized while, when employing regression, UAVs with the largest gap between current and predicted SNR are given precedence. While initially a scenario where the CQI reports are synchronized was considered and the decision is taken once for all users at the beginning of the slot when the reports are received, the analysis was improved upon by considering unsynchronized CQI reports. The lack of synchronization for the CQI reports raised the issue of information decay, as the LSTM outcome becomes more unreliable the older it gets with a peak just before a new report is received. As the reports are un-synchronized not all the information has the same degree of reliability and to address this issue we also introduce an aging policy to counteract the phenomenon and improve the performance of the system. The aging policy modifies the LSTM outcome obtained when a new report is received by considering as weight the temporal difference between the moment the CQI report is received and the moment the decision is made. The proposed allocation method allows to improve both the latency and rate, by avoiding errors and the resulting retransmission.

Section 6.2 presents the system model, providing insights into CQI modeling, LSTM networks, and the UAV channel model; Section 6.3 introduces the proposed system and Section 6.4 showcases its performance. In Section 6.5 conclusions for this research are drawn.

6.2 System Model

6.2.1 CQI modelling

In a single-cell scenario, the interaction between a GCS and a group of UAVs utilizing URLLC is examined. The UAV transmission rate is dynamically adjusted by employing a closed-loop control system. Through an exchange of known reference signals between the GCS, also serving as the Base Station (BS), and the UAV a Channel Quality Indicator (CQI) is determined. The signal received by the UAV is:

$$y = \alpha x + w \tag{6.1}$$

with x being the known signal, α being the current state of the channel, and w being additive white Gaussian noise (AWGN). The channel between GCS and UAV is considered a flat-fading Ricean-distributed channel, as detailed in the following subsection. Using conventional algorithms, the UAV estimates α and the noise power σ_w^2 from y. Following this operation, the UAV maps the corresponding Signal-to-Noise Ratio SNR, denoted as γ and calculated as $\frac{|\alpha|^2}{\sigma_w^2}$ into the corresponding CQI value. Specifically, the minimum SNR supported by the communication system is indicated by γ_m. For values of SNR smaller than γ_m the transmission is unreliable, and this is represented by a CQI of 0. On the opposite, γ_M represents the maximum SNR for which an MCS is available. For values exceeding γ_M the MCS and transmission rate remain constant, and such values are represented by a CQI of $N-1$. Therefore, the CQI value is an integer in the $[0, N-1]$ range and can be expressed as:

$$q_\gamma = \frac{1}{D} * min\{max\{\gamma - \gamma_m + D_Q, 0\}, \gamma_M - \gamma_m + D_Q\} \tag{6.2}$$

Here, $D_Q = \frac{(\gamma_M - \gamma_m)}{N-2}$ is the quantization step. The CQI is sent back to the BS, and used to select the most suitable MCS. The research operates under the assumption that this communication is error-free so

that the reception of every CQI report is ensured. The quantized SNR value can be recovered using knowledge of CQI and quantization parameters As the quantized value of the SNR is always lesser than the original value a conservative choice of MCS is made. The transmission rate is determined by the selected MCS and should be chosen so that the packet error probability doesn't exceed the target value. Conversely, opting for a more conservative transmission rate may lead to sub-optimal performance in terms of spectral efficiency and may cause network congestion, generating an increase in latency which is the other critical aspect of URLLC.

6.2.2 Long Short-Term Memory Networks

In systems involving re-transmissions, the use of more reliable CQI information allows to reduce resource wastage, and energy consumption and improve latency and data rate by reducing the number of re-transmitted packets. A predictive solution allows to support of URLLC without adopting overly conservative strategies that affect data rates. As the information available to our system is the sequence of past CQI reports the chosen solution must capture dependencies not only in the single values but within the sequence. As common Neural Networks Nearest Neighbors (NN) may struggle with this task, especially when dealing with variable sequence lengths, we opted forRNN who specialize in sequence processing. A fundamental feature of RNNs is that outputs of the network are influenced by previous outputs, in a fashion similar to infinite impulse response filters. However, while basic RNNs are designed for sequence processing, they struggle to capture long-term dependencies as the gradient calculated during training tends to either vanish or explode when propagated over many stages due to feedback action and in the context of channel forecasting, long-term dependencies are crucial. A more advanced evolution in sequence processing architecture is the gated RNN class, which is highly effective. It introduces the concept of "forget" by adding gates that activate as needed by the neural network. Our focus has therefore been on the use of LSTM networks: this kind of network operates by managing the the flow of information using three gates: the forget gate f, the input gate i, and the output gate o. along with two state variables, h and h. Each gate is influenced by the combination of the current input and the previous output. It con-

sists of an affine transformation, a sigmoid activation function, and an element-wise product $\otimes$. Consequently, the output of each gate can be expressed as follows:

$$f_n = \sigma(W_f \cdot [h_{n-1}; x_n] + b_f) \tag{6.3}$$

$$o_n = \sigma(W_o \cdot [h_{n-1}; x_n] + b_o) \tag{6.4}$$

$$i_n = \sigma(W_i \cdot [h_{n-1}; x_n] + b_i) \tag{6.5}$$

Here, the sigmoid function is represented as $\sigma()$ and defined as $\sigma(x) = \frac{1}{1+e^{-x}}$. Furthermore, the intermediate state $\tilde{s_n}$, as well as the new cell states s_n and h_n, are computed as follows:

$$\tilde{s_n} = tanh(W_s \cdot [h_{n-1}; x_n] + b_s) \tag{6.6}$$

$$s_n = f_n \otimes s_{n-1} + i_n \otimes \tilde{s_n} \tag{6.7}$$

$$h_n = o_n \otimes tanh(s_n) \tag{6.8}$$

As described in [60], the new cell state s_n is the result of summing the previous cell state, controlled by the forget gate, with $\tilde{s_n}$. The latter is computed using an affine transformation and a hyperbolic tangent activation function, influenced by the input gate. This cell state is used to calculate the output for the next layer y_n, passing through another hyperbolic tangent activation function and controlled by the output gate. Ultimately, both the output and the updated cell state s are fed back into the system.

6.2.3 UAV Channel Model

Research on air-to-ground channel models for UAV applications has been extensively carried out. While there may be some variations, these models typically share certain features, like LOS propagation, Rician fading, and constraints on the delay caused by signal reflection and scattering [36,53]. The conventional Rician fading model can be represented as:

$$f(|h|) = \frac{2(K+1)|h|}{\rho} \cdot \exp\left(-K - \frac{(K+1)|h|^2}{\rho}\right) \cdot I_0\left(2\sqrt{\frac{K(K+1)}{\rho}|h|}\right) \tag{6.9}$$

In this equation, K represents the Ricean K factor, which signifies the ratio between the LOS component and the NLOS components, while ρ is a scaling factor that depends on the total received power. In this context, the K factor of the Rician fading model holds particular significance, as it quantifies the extent of variations introduced by indistinguishable signal replicas. When the K factor is high, even in the presence of rapidly changing channels, the dynamics remain limited, reducing the impact of delay. However, as documented in previous studies [36], the K factor can span a broad range of values, this can include lower values [93], depending on various factors, including the UAV model, trajectory, altitude, and the frequency band in use. The UAV's speed holds significant importance in this context, as it determines the rate at which the channel undergoes variations and therefore how long before a CQI report becomes outdated. As indicated in [36], Jake's spectrum is a well-established model for the Doppler spectrum of the air-to-ground channel:

$$S(f) = \frac{1}{\pi f \sqrt{1 - \frac{f}{f_d}}} \tag{6.10}$$

where f_d relates to the UAV's speed, as $fd = \frac{v_r}{\lambda}$ with v_r as the UAV's radial speed from the BS and λ as the wavelength.

6.3 Proposed System

In this research, we focus on the application of LSTM networks for channel quality prediction and forecast. The LSTM's output is used as a base of a resource allocation policy for control links in UAV communications. The objective of the resource allocation policy is to reduce the error due to CQI aging by providing additional resources for control communications. The additional resources are represented by Resource Block (RB) on a dedicated side channel and the allocation task is carried out for each slot Let us assume that the network is composed of a BS and several UAVs. The UAVs exchange messages with the BS to evaluate the channel quality and select the MCS. When the selected MCS is no longer valid due to CQI aging, specifically because the channel quality has worsened significantly, all the subsequent communications will be affected by errors. To prevent this from happening we aim to estimate future CQI to prevent the aging from affecting the communi-

cation. Several instances of LSTM networks, trained on CQI sequences of UAV channels with different average SNRs are present at the BS. Each time a new CQI information is available it is fed to the LSTM. We examine solutions based on both classification and regression algorithms. In the classification case, the objective of the LSTM is to provide a forecast $F_c(CQI)$ of what the outlook of the channel quality will be. If the outlook is positive, meaning that the channel quality from the current CQI to the next will not change so much to cause the MCS choice to affect negatively the communication. If the outlook is negative, it means that the channel quality from the current CQI to the next will decrease enough to cause the MCS choice to affect the communication with consequent errors negatively. This can be summarised as the following:

$$F_c(CQI_n) = \begin{cases} +1, & \text{if } CQI_n \geq CQI_{n+1} \\ -1, & \text{if } CQI_n < CQI_{n+1} \end{cases} \tag{6.11}$$

where CQI_n is the n-th CQI received and CQI_{n+1} is the next one, which is yet to arrive. Since our only interest is in the trend of the channel and not in the actual CQI value we treat the forecast as a dichotomic classification problem. In the regression case, we wish to obtain from the LSTM a prediction of the SNR value for the next channel quality indicator CQI_{n+1}. The MCS choice will be done according to both the known current value and the predicted one In both cases, the operation is carried out for every new CQI the BS receives. It should be noted that the system operates on the CQI information: if the channel quality decreases below the CQI change threshold and then increases back again before the next CQI is received, the MCS will still be incorrect for the transmission in the corresponding time slots. Therefore a perfect prediction of the next CQI value or a forecast of the trend cannot completely avoid the mismatch between the MCS and the channel quality. The resource allocation is then carried out, but in a multi-user scenario like the one considered additional actions are required. For the system not to be trivial, we have to assume that the amount of extra resources is limited and, therefore there are not enough resources to assign some to each UAV. Therefore, in the event of multiple forecasts having a negative outlook, or multiple predictions indicating a lower SNR, a criterion for the allocation must be defined. As such, while the allocation policy

in the regression case is to assign the resources available to the users with the worst prediction, we consider a soft approach for classification where instead of the $\{-1; 1\}$ response we use the score of the classification function that is in the $[0; 1]$ range. We then select the highest scores and allocate the side channel resources to the communication with those UAVs. Furthermore, to shield us from channel fluctuations and reduce the error rate, we consider the use of a guard bias when selecting the MCS. The guard bias acts as a safety measure, reducing the CQI and forcing the selection of more conservative MCS so that channel fluctuation in that guard bias range can be withstood without incurring errors. To further contrast the aging effect of the CQI we also introduce an aging process on the LSTM result. The intention is to increase the score as the CQI received becomes older, as it is more likely that the channel will be different from what was reported the more the moment of the report becomes distant in time.

6.4 Numerical Results

In this section, we numerically assess the proposed approach employing LSTM for channel prediction and forecast. We evaluate the performance in terms of both packet loss probability, presenting the mean probability per user achieved by the system as it is the URLLC requirement we aim to meet, and throughput T as the true objective is to maximize it while meeting the aforementioned requirements. The achieved throughput, which is calculated by multiplying the correct decoding probability, calculated with the actual SNR or the equivalent one in case of additional allocation of resources with the transmission rate of the selected MCS is expressed as bit/Hz/#UAV.

The performance of the proposed framework is compared to the following benchmarks:

1. The unaided system operating with an equivalent bandwidth: as overall to allocate the same amount of resources to both systems and given that from Shannon's theorem the channel capacity C is:

$$C = B \log_2(1 + SNR) \tag{6.12}$$

   we note that the allocation cannot be performed smoothly due to resource block size, therefore, we consider the performance of

a system using the same base bandwidth B but an equivalent SNR whose increment makes up for the difference due to the extra resources. This represents the base case we wish to improve upon.

2. A system operating with perfect knowledge of the channel SNR in every transmission slot: this represents an upper bound to the performance, as the allocated resource can be optimally assigned during each transmission slot. While there still may be a rate mismatch it is outside of the scope of the reach of the system due to the limit on the available resources.

3. the system proposed in [] where rate adaptation is carried out based on the information obtained from the LSTM prediction. This method allows us to avoid communication errors by selecting a different MCS to better fit the channel evolution.

The first of the benchmarks represents the base operating conditions with normal while the second one represents an ideal case with perfect channel knowledge. Finally, the last one represents an alternative solution present in the literature. In the following discussion, we consider a 5G system with parameters suitable for the UAV/URLLC context. Specifically, the transmitted packet size is 32 bytes, and the target packet loss is set to 10^{-5}. No considerations are made on retransmissions and a packet is considered lost if it's not delivered if the transmission fails. Additionally, a slot duration of 0.25ms is selected, which is associated with the chosen numerology, reflecting sub-carrier spacing. This sub-carrier spacing is defined as $\delta f = 2^{\mu}\delta f_{LTE}$, with $\delta f_{LTE} = 15$ kHz as the LTE sub-carrier spacing, and we select $\mu = 2$. The sub-carrier spacing is inversely proportional to the OFDM symbol duration, and the slot comprises a fixed number of OFDM symbols (i.e., one slot consists of 7 OFDM symbols). Multiple LSTM instances are trained using a channel sequence of 10^6 samples, which corresponds to around $3.5 * 10^4$ CQIs. In the training sequence, samples are generated from a channel model with fixed parameters, and the network is trained using the corresponding CQI sequence with a batch sequence length of 3. The networks are trained offline for SNR in the $[10, 13]$ range with intervals of 0.5 dB and the appropriate NN can be selected during communication based on extracted channel parameters from data. The

LSTMs share a common structure up to the third layer, after which, the structure is differentiated for the task. The common layers are:

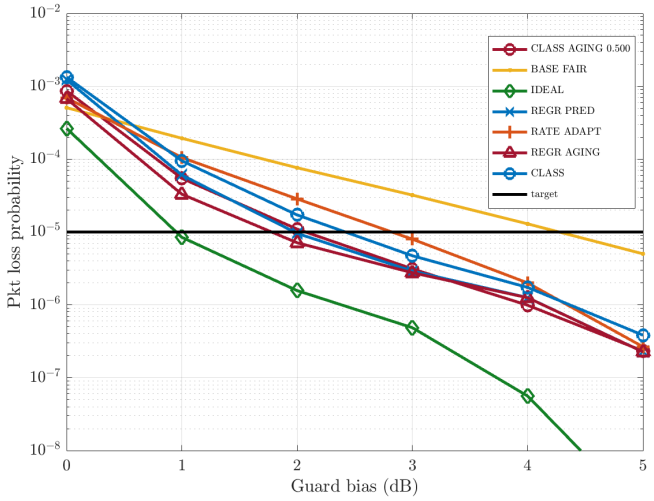
1. an input layer, receiving the current CQI,
2. an LSTM layer with 200 hidden units,
3. a fully connected layer

while the specific layers are:

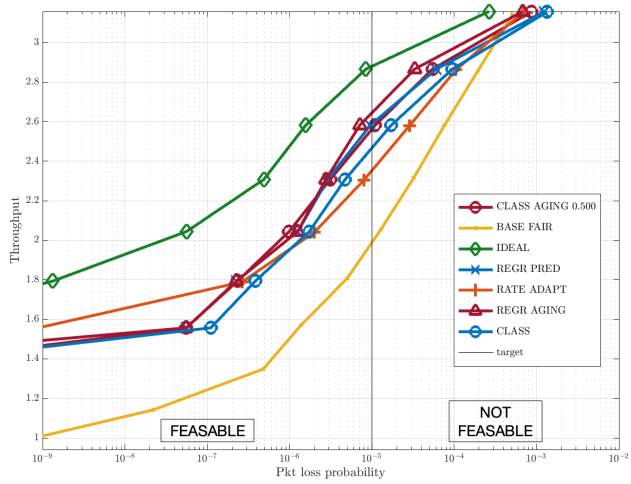
1. a regression layer, for the regression task and
2. a softmax layer, followed by
3. a classification output layer, using cross-entropy as a loss function, for the classification task

In our numerical results, we evaluated through computer simulation a multi-user scenario where U UAVs are present moving at a speed of 70 km/h. The communication occurs on the L-band at a frequency f_0 of 1 GHz with LoS links and a Rician factor of 7dB. The channel is quantized for a SNR in the $[-6, 30]$dB with a 1 dB step range For the classification method, the score aging process is implemented as a multiplicative increment over each slot directly proportional to the CQI age up to a maximum value t. This implies that on the slot the CQI is received the score s is in the $[0, 1]$ range while on the slot immediately before the new CQI arrives it is in the $[0, 1 + t]$ range with it being $s_q = s_0(\frac{q}{T_{CQI}}t)$, with s_0 being the score with no aging, q the considered transmission slot and T_{CQI} the number of slots between each CQI. For the regression method, the aging process on the prediction is introduced as a linear interpolation between the current SNR, extracted from the received CQI, and the predicted SNR value.

Figs. 6.1a6.2a6.3a illustrate the packet loss behavior for a range of values centered around the 10^{-5} URLLC target, represented by the black line, for a system composed by a variable number of UAVs with single side-channel over which one extra resource can be allocated. The proposed methods and the mentioned benchmarks are plotted for variable values of the guard bias, ranging from 0 to 5 dB. It can be observed that, outside of the ideal benchmark cases high guard values are required to meet the target packet loss, and even if their use leads to lower average packet loss it also implies a lower throughput. The proposed system

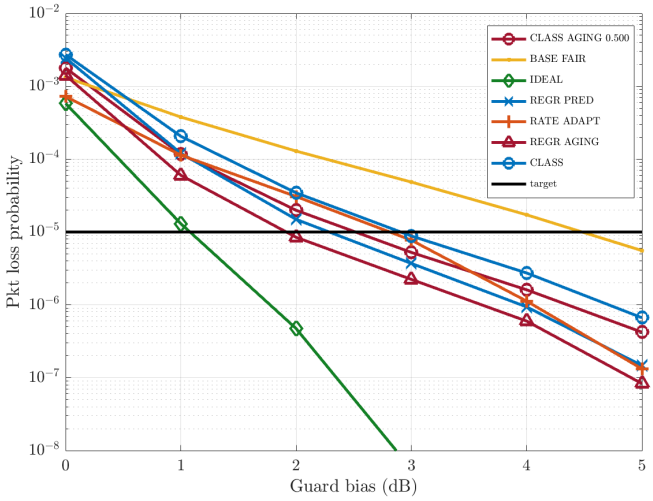


(a)

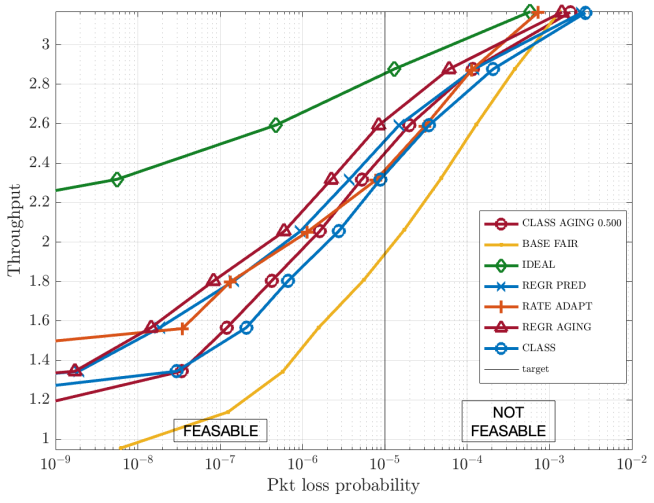


(b)

Figure 6.1: (a): Packet loss probability for different values of the guard bias 3 UAVs respectively and (b): Throughput for the corresponding packet loss probability

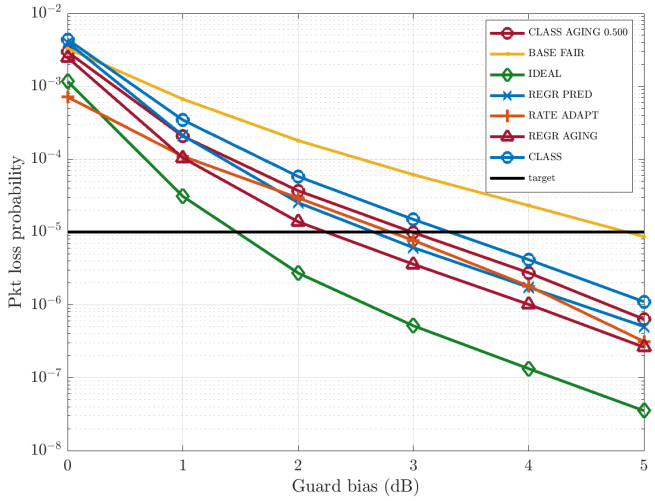


(a)

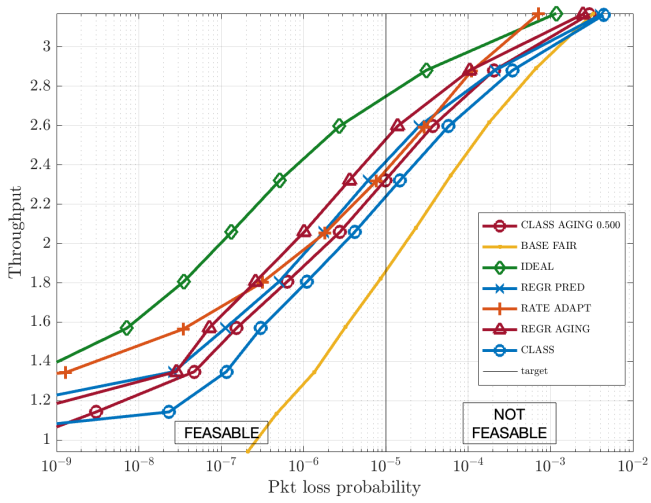


(b)

Figure 6.2: (a): Packet loss probability for different values of the guard bias 6 UAVs respectively and (b): Throughput for the corresponding packet loss probability



(a)



(b)

Figure 6.3: (a): Packet loss probability for different values of the guard bias 12 UAVs respectively and (b): Throughput for the corresponding packet loss probability

without the use of the score aging process offers better performance than the equivalent resources one, allowing to reach the target packet loss with a smaller guard bias. In particular, it can be observed that the regression-based method performs quite well, and its version which includes aging, outperforms in the region of interest all the benchmarks minus the perfect knowledge one. As the number of UAVs increases the adaptive rate solution grows closer, outperforming the classification solution. This is because while the proposed solution is constrained by resource availability, rate adaptation does not require additional resources and can be performed on all channels every time Variations over the aging value t for the classification method provided little to no extra benefit. Figs. 6.1b6.2b6.3b shows how the resource allocation scheme affects the throughput: as a smaller guard bias is required to achieve the required packet loss rate the use of the proposed system, with and without score aging, provides benefits in terms of increased throughput when compared to the base system. The gain for both the packet loss probability and the throughput is lower when the number of UAVs in the system rises. This is because with a limited amount of resources not all the requests for additional resources can be satisfied causing the performance to decrease.

6.5 Conclusion

The stringent requirements for ultra-reliable and low-latency communication in UAV control necessitate an effective link adaptation strategy that ensures the target error rate is achieved while adhering to latency constraints. However, link adaptation faces challenges due to inaccuracies in the MCS selection based on rapidly changing CQI information. This proposed solution, a combination of channel quality forecast and resource allocation strategy aims to address the URLLC error rate requirement, which is fundamental for many applications.

The impact of CQI aging leading to error in communication is addressed by allocating resources from a dedicated side-channel to the links that might require them according to an LSTM classification score evaluating the outlook of the channel from the current CQI to the following one based on the previous CQI values. The aim is to prevent communication errors and loss in both throughput and latency. Along

the proposed solution we implement a score aging policy to further counteract the CQI aging effect by increasing the score as the CQI becomes more unreliable over time.

The proposed solution shows to be able to help support the URLLC requirements. Specifically, both the original scheme and the one including the score aging enhance the system performance in terms of reliability and throughput, with the proposed system including the score aging policy offering further benefits than the one without.

Chapter 7

Conclusion

This chapter briefly summarizes the contribution of the thesis and discusses avenues for future research.

7.1 Summary of contribution

Throughout this Ph.D. the main focus of the research had been the analysis of the application of Machine Learning techniques at the physical level of communication to achieve safety, either in the form of security, in the form of measures for authentication and spoofing detection, or in the form of reliability in communication, when used to improve the quality of communications. For the security aspect, the analysis has evolved to face the different challenges that emerged from the research. Initially, a classification approach was considered to implement a wireless fingerprinting solution in a wireless sensor network that, due to the constraints on the devices' resources could benefit from such a solution. This proposed approach showed promise regarding the capability of recognizing the known transmitters from features extracted from the incoming communication. Different classification algorithms, such as CART, k-nearest Neighbor, and Support Vector Machine were compared with no absolute winner. The focus then shifted over to the system's ability to detect a spoofing attack carried out by a malicious node, aiming to impersonate one of the authenticated nodes of the network. An approach based on the combination of a classification algorithm and a cross-check with the message ID was developed. The performance of this system was linked

to the number of nodes in the network with the detection rate increasing along with the number of nodes. The system's poor performance in a network with few nodes was addressed by introducing sentinel nodes, extra nodes whose only task is to provide additional fingerprints, artificially inflating the number of nodes in the network. However, as the classification has no way to reject a fingerprint as unknown, the focus was moved to the possible application of anomaly detection algorithms that would allow the system to operate in the case of small networks. Several anomaly detection algorithms were considered and in this case, some proved to be more viable. In particular One-Class Support Vector Machine and One-Class k-Nearest Neighbor proved to offer the best results in terms of both detecting the attack and recognizing the legitimate user. When comparing the two proposed approaches, classification, and anomaly detection, it must be noted that while the first lacks a reject option some of the algorithms allow for a multi-user approach, which is its fundamental strength. A comparison between them was proposed with the anomaly detection solution emerging as superior due to the dependence on the number of nodes to achieve good detection performances with the classification-based solution. The mobility issue was then addressed, as all the previous solutions were evaluated on static nodes. As an approach focusing on re-training was not viable, a continuous learning and forgetting solution was proposed in the form of the combination of a k-NN algorithm and a sliding window approach to the dataset. The update of the dataset elements over time proved to be a viable solution to address the wireless fingerprint evolution due to the concept drift introduced by nodes' mobility. As for the reliability aspect, Machine Learning was used to evaluate channel quality and prevent errors during transmission. The case of control communication for UAVs was considered, as their requirements set them in the field of Ultra-Reliable Low-Latency Communications. A resource allocation scheme in a multi-user scenario was proposed to contrast packet loss in case of incorrect selection of the Modulation and Coding scheme. The scheme relies on a Long Short-Term Memory network to perform a channel quality forecast and address the issue of Channel Quality Indicator aging. The scheme was also improved by introducing an aging process on the LSTM forecast to further counteract the CQI aging. By allocating extra resources on a side-channel errors can be avoided and

the tight requirements of URLLC can be met without an excessive loss of performance. The proposed solution proved to be effective in achieving a gain in throughput while still meeting the packet loss probability constraint.

7.2 Directions for future work

The research work conducted so far can be improved upon by future activity in several different ways. As for the security aspect, the classification and anomaly detection solution could be merged by using open-set recognition algorithms that would give the missing reject option to this approach. Alternatively, a reject option could be implemented directly in the classifier as a hypothesis test or similar mathematical expedients. One more point the work could be improved upon could be the analysis of the performance of the proposed solution in a Non-Line-of-Sight scenario. This is particularly relevant as the Line-of-Sight requirement might be excessively constraining for indoor applications or in environments where a change between LoS and NLos is possible, due for example to a moving obstacle. An NLoS analysis might also focus on the selection of new attributes, as the Angle of Arrival which showed to have a strong identifying power, might not be as performing. Finally, a field experiment to validate the results obtained through simulations should be carried out also to verify the presence of unforeseen phenomena and the possible presence of limitations that were not included in the channel model used to implement the wireless fingerprinting solutions

Appendix A

Publications

The research activity has led to several publications in international journals. These are summarized below.[1]

International Journals

1. **Andrea Stomaci**, G.Bartoli, D. Marabissi. "Low-complexity distributed cell-specific bias calculation for load balancing in udns", *EEE Transactions on Vehicular Technology*, vol. 68 in press, 2018. [DOI:10.1109/TVT.2018.2883294]
2. **Andrea Stomaci**, D. Marabissi, L. Mucchi. "IoT Nodes Authentication and ID Spoofing Detection Based on Joint Use of Physical Layer Security and Machine Learning", *Future Internet*, vol. 14 in press, 2022. [DOI:10.3390/fi14020061]

Accepted for Pubblication on International Journal

1. **Andrea Stomaci**, D. Marabissi, L. Mucchi. "Comparison of Machine Learning approaches based on multiple channel attributes for authentication and spoofing detection at the physical layer", *Journal of Communications*

Submitted

1. **Andrea Stomaci**, D. Marabissi, L. Mucchi. *Classification-based and anomaly detection-based machine learning solutions for node identifi-*

[1]The author's bibliometric indices are the following: H-index = 2, total number of citations = 15 (source: Google Scholar on Month 10, 2023).

cation in Internet of Things applications, submitted to "IEEE Open Journal of the Communications Society".

2. **Andrea Stomaci**, D. Marabissi, L. Mucchi, H. Ochiai. *Machine learning based continuous physical layer authentication for wireless networks with mobile nodes*, submitted to "IEEE Access".

Bibliography

[1] "Unmanned aircraft system (uas) service demand 2015 - 2035 : literature review & projections of future usage, technical report, version 1.0 - february 2014," Feb 2014, tech Report. [Online]. Available: https://rosap.ntl.bts.gov/view/dot/12029

[2] 3GPP, "3rd generation partnership project; technical specification group services and system aspects; release 15 description; summary of rel-15 work items (release 15)," 3GPP, Tech. Rep. TR 21.915 V15.0, 2019.

[3] M. Abdrabou and T. A. Gulliver, "Adaptive physical layer authentication using machine learning with antenna diversity," *IEEE Transactions on Communications*, vol. 70, no. 10, pp. 6604–6614, 2022.

[4] ——, "Adaptive physical layer authentication for iot in mimo communication systems using support vector machine," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[5] F. Adamsky, T. Retunskaia, S. Schiffner, C. Köbel, and T. Engel, "Wlan device fingerprinting using channel state information (csi)," in *WiSec '18: Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, June 2018, pp. 277–278.

[6] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[7] Z. Becvar, D. Gesbert, P. Mach, and M. Najla, "Machine learning-based channel quality prediction in 6g mobile networks," *IEEE Communications Magazine*, vol. 61, no. 7, pp. 106–112, 2023.

[8] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proceedings of the IEEE*, vol. 106, no. 10, pp. 1834–1853, 2018.

[9] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, 1992, pp. 144–152.

[10] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[11] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[12] B. Chatterjee, D. Das, S. Maity, and S. Sen, "Rf-puf: Enhancing iot security through authentication of wireless nodes using in-situ machine learning," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 388–398, 2019.

[13] A. Dataesatu, K. Sanada, H. Hatano, K. Mori, and P. Boonsrimuang, "Adaptive k-repetition transmission employing site diversity reception for 5g nr uplink grant-free urllc," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, 2023, pp. 1–5.

[14] V. Erceg, L. Schumacher, P. Kyritsi, A. Molisch, D. S. Baum, A. Y. Gorokhov, C. Oestges, Q. Li, K. Yu, K. N. Tal *et al.*, "Wireless lans indoor mimo wlantgn channel models," 2004.

[15] H. Fang, X. Wang, and L. Hanzo, "Learning-aided physical layer authentication as an intelligent process," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2260–2273, 2019.

[16] H. Fang, X. Wang, and S. Tomasin, "Machine learning for intelligent authentication in 5g and beyond wireless networks," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 55–61, 2019.

[17] H. Fang, A. Qi, and X. Wang, "Fast authentication and progressive authorization in large-scale iot: How to leverage ai for security enhancement," *IEEE Network*, vol. 34, no. 3, pp. 24–29, 2020.

[18] E. Fix and J. L. Hodges, "Discriminatory analysis. nonparametric discrimination: Consistency properties," *International Statistical Review/Revue Internationale de Statistique*, vol. 57, no. 3, pp. 238–247, 1989.

[19] H. Forssell, R. Thobaben, H. Al-Zubaidy, and J. Gross, "Physical layer authentication in mission-critical mtc networks: A security and delay performance analysis," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 4, pp. 795–808, 2019.

[20] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, "Survey on uav cellular communications: Practical aspects, standardization advancements, regulation, and security

challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3417–3442, 2019.

[21] Y. Gao, H. Yang, X. Hong, and L. Chen, "A hybrid scheme of mcs selection and spectrum allocation for urllc traffic under delay and reliability constraints," *Entropy*, vol. 24, no. 5, 2022. [Online]. Available: https://www.mdpi.com/1099-4300/24/5/727

[22] K. Glinskiy, A. Krasilov, E. Khorov, and A. Kureev, "Performance of ml-based channel prediction algorithms for urllc: Channel model matters," in *2023 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. IEEE, 2023, pp. 306–311.

[23] K. Glinskiy, A. Kureev, and E. Khorov, "Sdr-based testbed for real-time cqi prediction for urllc," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021, pp. 1–2.

[24] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, 1995, pp. 278–282 vol.1.

[25] T. M. Hoang, N. M. Nguyen, and T. Q. Duong, "Detection of eavesdropping attack in uav-aided wireless systems: Unsupervised learning with one-class svm and k-means clustering," *IEEE Wireless Communications Letters*, vol. 9, no. 2, pp. 139–142, 2020.

[26] W. Hou, X. Wang, J. Chouinard, and A. Refaey, "Physical layer authentication for mobile systems with time-varying carrier frequency offsets," *IEEE Transactions on Communications*, vol. 62, no. 5, pp. 1658–1667, 2014.

[27] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, "Quadriga: A 3-d multi-cell channel model with time evolution for enabling virtual field trials," pp. 3242–3256, 2014.

[28] S. Jaeckel, L. Raschkowski, K. Börner, L. Thiele, F. Burkhardt, and E. Eberlein, "Quadriga - quasi deterministic radio channel generator, user manual and documentation," Fraunhofer Heinrich Hertz Institute, 2021, tech. Rep. v2.6.1.

[29] W. Jiang and H. D. Schotten, "Neural network-based fading channel prediction: A comprehensive overview," *IEEE Access*, vol. 7, pp. 118 112–118 124, 2019.

[30] ——, "Recurrent neural networks with long short-term memory for fading channel prediction," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.

[31] ——, "Deep learning for fading channel prediction," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 320–332, 2020.

[32] J. Kermoal, L. Schumacher, K. Pedersen, P. Mogensen, and F. Frederiksen, "A stochastic mimo radio channel model with experimental validation," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 6, pp. 1211–1226, 2002.

[33] M. Khalid, R. Zhao, and N. Ahmed, "Physical layer authentication in line-of-sight underwater acoustic sensor networks," in *Global Oceans 2020: Singapore - U.S. Gulf Coast*, 2020, pp. 1–5.

[34] S. S. Khan and A. Ahmad, "Relationship between variants of one-class nearest neighbors and creating their accurate ensembles," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1796–1809, 2018.

[35] S. R. Khosravirad, O. Tirkkonen, U. Parts, L. Zhou, D. Korpi, P. Baracca, and M. A. Uusitalo, "Communications survival strategies for industrial wireless control," *IEEE Network*, vol. 36, no. 2, pp. 66–72, 2022.

[36] A. A. Khuwaja, Y. Chen, N. Zhao, M.-S. Alouini, and P. Dobbins, "A survey of channel modeling for uav communications," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2804–2821, 2018.

[37] J. Kim and D. S. Han, "Deep learning-based channel quality indicators prediction for vehicular communication," *ICT Express*, vol. 9, no. 1, pp. 116–121, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959522000716

[38] A. S. Konstantinov and A. V. Pestryakov, "The new neural network based framework for fading channel prediction for 5g," in *2020 Systems of Signals Generating and Processing in the Field of on Board Communications*, 2020, pp. 1–7.

[39] P. Kyösti, J. Meinilä", L. Hentila, X. Zhao, T. Jämsä, C. Schneider, M. Narandzic, M. Milojević, A. Hong, J. Ylitalo, V.-M. Holappa, M. Alatossava, R. Bultitude, Y. Jong, and T. Rautiainen, "Winner ii channel models," *IST-4-027756 WINNER II D1.1.2 V1.2*, 02 2008.

[40] B. Li, Z. Fei, and Y. Zhang, "Uav communications for 5g and beyond: Recent advances and future trends," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2241–2263, 2018.

[41] X. Li, K. Huang, S. Wang, and X. Xu, "A physical layer authentication mechanism for iot devices," *China Communications*, vol. 19, no. 5, pp. 129–140, 2022.

[42] Z. Li, H. Shariatmadari, B. Singh, and M. Uusitalo, "5g urllc: Design challenges and system concepts," in *International Symposium on Wireless Communication Systems (ISWCS)*, ser. International Symposium on Wireless Communication Systems. IEEE, Oct 2018, international Symposium on Wireless Communication Systems, ISWCS ; Conference date: 28-08-2018 Through 31-08-2018.

[43] R.-F. Liao, H. Wen, S. Chen, F. Xie, F. Pan, J. Tang, and H. Song, "Multiuser physical layer authentication in internet of things with data augmentation," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2077–2088, Mar. 2020. [Online]. Available: https://doi.org/10.1109/jiot.2019.2960099

[44] R. Liao, H. Wen, F. Pan, H. Song, A. Xu, and Y. Jiang, "A novel physical layer authentication method with convolutional neural network," in *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 2019, pp. 231–235.

[45] F. J. Liu, Xianbin Wang, and H. Tang, "Robust physical layer authentication using inherent properties of channel impulse response," in *2011 - MILCOM 2011 Military Communications Conference*, 2011, pp. 538–542.

[46] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *8th IEEE Int. Conf. Data Mining*, 2008, pp. 413–422.

[47] H. Liu, Y. Wang, J. Liu, J. Yang, and Y. Chen, "Practical user authentication leveraging channel state information (csi)," in *Proc. 9th ACM Symp. Information, Computer and Commun. Security*, 2014, pp. 389–400.

[48] J. Liu and X. Wang, "Physical layer authentication enhancement using two-dimensional channel quantization," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 4171–4182, 2016.

[49] Y. Liu, H. Chen, and L. Wang, "Physical layer security for next generation wireless networks: Theories, technologies, and challenges," *IEEE Communications Surveys Tutorials*, vol. 19, no. 1, pp. 347–376, 2017.

[50] Y. Liu, J. Wang, J. Li, H. Song, T. Yang, S. Niu, and Z. Ming, "Zero-bias deep learning for accurate identification of internet-of-things (iot) devices," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2627–2634, 2021.

[51] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. math. statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[52] D. Marabissi, L. Mucchi, and A. Stomaci, "IoT nodes authentication and ID spoofing detection based on joint use of physical layer security and machine learning," *Future Internet*, vol. 14, no. 2, p. 61, Feb. 2022. [Online]. Available: https://doi.org/10.3390/fi14020061

[53] D. W. Matolak and R. Sun, "Unmanned aircraft systems: Air-ground channel characterization for future applications," *IEEE Vehicular Technology Magazine*, vol. 10, no. 2, pp. 79–85, 2015.

[54] L. Mucchi, F. Nizzi, T. Pecorella, R. Fantacci, and F. Esposito, "Benefits of physical layer security to cryptography: Tradeoff and applications," in *2019 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2019, pp. 1–3.

[55] A. Mukherjee, "Physical-layer security in the internet of things: Sensing and communication confidentiality under resource constraints," *Proceedings of the IEEE*, vol. 103, no. 10, pp. 1747–1761, 2015.

[56] A. Ometov, V. Petrov, S. Bezzateev, S. Andreev, Y. Koucheryavy, and M. Gerla, "Challenges of multi-factor authentication for securing advanced iot applications," *IEEE Network*, vol. 33, no. 2, pp. 82–88, 2019.

[57] H. Peng, T. Kallehauge, M. Tao, and P. Popovski, "Power and rate adaptation for urllc with statistical channel knowledge and harq," *IEEE Wireless Communications Letters*, vol. 12, no. 12, pp. 2148–2152, 2023.

[58] L. Peng, A. Hu, J. Zhang, Y. Jiang, J. Yu, and Y. Yan, "Design of a hybrid rf fingerprint extraction and device classification scheme," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 349–360, 2019.

[59] L. Peng, J. Zhang, M. Liu, and A. Hu, "Deep learning based rf fingerprint identification using differential constellation trace figure," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 1091–1095, 2020.

[60] G. Pocovi, A. A. Esswie, and K. I. Pedersen, "Channel quality feedback enhancements for accurate urllc link adaptation in 5g systems," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–6.

[61] A. C. Polak and D. L. Goeckel, "Wireless device identification based on rf oscillator imperfections," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2492–2501, 2015.

[62] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[63] X. Qiu, J. Dai, and M. Hayes, "A learning approach for physical layer authentication using adaptive neural network," *IEEE Access*, vol. 8, pp. 26 139–26 149, 2020.

[64] H. Ren, C. Pan, Y. Deng, M. Elkashlan, and A. Nallanathan, "Joint pilot and payload power allocation for massive-mimo-enabled urllc iiot networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 816–830, 2020.

[65] ——, "Joint power and blocklength optimization for urllc in a factory automation scenario," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 1786–1801, 2020.

[66] H. Ren, C. Pan, K. Wang, Y. Deng, M. Elkashlan, and A. Nallanathan, "Achievable data rate for urllc-enabled uav systems with 3-d channel model," *IEEE Wireless Communications Letters*, vol. 8, no. 6, pp. 1587–1590, 2019.

[67] F. Restuccia, S. D'Oro, and T. Melodia, "Securing the internet of things in the age of machine learning and software-defined networking," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4829–4842, 2018.

[68] B. e. a. Schölkopf, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[69] L. Senigagliesi, M. Baldi, and E. Gambi, "Comparison of statistical and machine learning techniques for physical layer authentication," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1506–1521, 2021.

[70] A. Shashin, A. Belogaev, A. Krasilov, and E. Khorov, "Adaptive parameters selection for uplink grant-free urllc transmission in 5g systems," *Computer Networks*, vol. 222, p. 109527, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128622005618

[71] C. She, C. Liu, T. Q. S. Quek, C. Yang, and Y. Li, "Ultra-reliable and low-latency communications in unmanned aerial vehicle communication systems," *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3768–3781, 2019.

[72] Y.-S. Shiu, S. Y. Chang, H.-C. Wu, S. C.-H. Huang, and H.-H. Chen, "Physical layer security in wireless networks: a tutorial," *IEEE Wireless Communications*, vol. 18, no. 2, pp. 66–74, 2011.

[73] H. Taha and E. Alsusa, "Secret key exchange and authentication via randomized spatial modulation and phase shifting," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2165–2177, 2018.

[74] S. Tomasin, "Analysis of channel-based user authentication by key-less and key-based approaches," *IEEE Transactions on Wireless Communications*, vol. 17, no. 9, pp. 5700–5712, 2018.

[75] W. Trappe, "The challenges facing physical layer security," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 16–20, 2015.

[76] J. K. Tugnait, "Wireless user authentication via comparison of power spectral densities," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 1791–1802, 2013.

[77] N. Wang, T. Jiang, S. Lv, and L. Xiao, "Physical-layer authentication based on extreme learning machine," *IEEE Communications Letters*, vol. 21, no. 7, pp. 1557–1560, 2017.

[78] N. Wang, P. Wang, A. Alipour-Fanid, L. Jiao, and K. Zeng, "Physical-layer security of 5g wireless networks for iot: Challenges and opportunities," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8169–8181, 2019.

[79] Q. Wang, H. Li, D. Zhao, Z. Chen, S. Ye, and J. Cai, "Deep neural networks for csi-based authentication," *IEEE Access*, vol. 7, pp. 123 026–123 034, 2019.

[80] A. Weinand, C. Lipps, M. Karrenbauer, and H. D. Schotten, "Multi-feature physical layer authentication for urllc based on linear supervised learning," in *2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2023, pp. 30–35.

[81] X. Wu, Z. Yang, C. Ling, and X. Xia, "Artificial-noise-aided physical layer phase challenge-response authentication for practical ofdm transmission," *IEEE Transactions on Wireless Communications*, vol. 15, no. 10, pp. 6611–6625, 2016.

[82] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe, "A physical-layer technique to enhance authentication for mobile terminals," in *2008 IEEE International Conference on Communications*. IEEE, 2008. [Online]. Available: https://doi.org/10.1109/icc.2008.294

[83] L. Xiao, L. J. Greenstein, N. B. Mandayam, and W. Trappe, "Using the physical layer for wireless authentication in time-variant channels," *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, pp. 2571–2579, 2008.

[84] ——, "Channel-based spoofing detection in frequency-selective rayleigh channels," *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, pp. 5948–5956, 2009.

[85] L. Xiao, X. Wan, and Z. Han, "Phy-layer authentication with multiple landmarks with reduced overhead," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1676–1687, 2018.

[86] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "Iot security techniques based on machine learning: How do iot devices use ai to enhance security?" *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 41–49, 2018.

[87] L. Xiao, L. J. Greenstein, N. B. Mandayam, and W. Trappe, "Using the physical layer for wireless authentication in time-variant channels," *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, pp. 2571–2579, 2008.

[88] L. Xiao, Y. Li, G. Han, G. Liu, and W. Zhuang, "Phy-layer spoofing detection with reinforcement learning in wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 10 037–10 047, 2016.

[89] N. Xie and C. Chen, "Slope authentication at the physical layer," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 6, pp. 1579–1594, 2018.

[90] N. Xie, Z. Li, and H. Tan, "A survey of physical-layer authentication in wireless communications," *IEEE Communications Surveys Tutorials*, vol. 23, no. 1, pp. 282–310, 2021.

[91] J. Xiong, H. Hu, P. Cheng, C. Yang, Z. Shi, and L. Gui, "Wireless resource scheduling for high mobility scenarios: A combined traffic and channel quality prediction approach," *IEEE Transactions on Broadcasting*, vol. 68, no. 3, pp. 712–722, 2022.

[92] D. Xu, P. Ren, and J. A. Ritcey, "Independence-checking coding for ofdm channel training authentication: Protocol design, security, stability, and tradeoff analysis," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 387–402, 2019.

[93] X. Ye, X. Cai, X. Yin, J. Rodriguez-Pineiro, L. Tian, and J. Dou, "Air-to-ground big-data-assisted channel modeling based on passive sounding in lte networks," in *2017 IEEE Globecom Workshops (GC Wkshps)*, 2017, pp. 1–6.

[94] J. Yoon, Y. Lee, and E. Hwang, "Machine learning-based physical layer authentication using neighborhood component analysis in mimo wireless communications," in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, 2019, pp. 63–65.

[95] P. L. Yu and B. M. Sadler, "Mimo authentication via deliberate fingerprinting at the physical layer," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 606–615, 2011.

[96] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[97] N. Zhang, X. Fang, Y. Wang, S. Wu, H. Wu, D. Kar, and H. Zhang, "Physical-layer authentication for internet of things via wfrft-based gaussian tag embedding," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 9001–9010, 2020.

[98] P. Zhang, Y. Shen, X. Jiang, and B. Wu, "Physical layer authentication jointly utilizing channel and phase noise in mimo systems," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2446–2458, 2020.

[99] X. Zhang, Q. Zhu, and H. V. Poor, "Minimum-energy and error-rate for urllc networks over nakagami-m channels: A finite-blocklength analysis," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[100] L. Zhou, O. Tirkkonen, U. Parts, S. R. Khosravirad, P. Baracca, D. Korpi, and M. Uusitalo, "Dual-mode ultra reliable low latency communications for industrial wireless control," in *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, 2022, pp. 1–7.