University of Genoa

Ph.D. Course in Science and Technology for Electronic and Telecommunication Engineering

DOCTOR OF PHILOSOPHY THESIS

Multimedia Context Awareness for Smart Mobile Environments

Candidate:

Chiara Garibotto

Advisors:

Prof. Fabio Lavagetto

Prof. Igor Bisio

Ph.D. STIET - Cycle XXXI

I am enough of an artist
to draw freely upon my imagination.
Imagination is more important than knowledge.
Knowledge is limited.
Imagination encircles the world.

Albert Einstein

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisors, Prof. Fabio Lavagetto and Prof. Igor Bisio, for their excellent guidance and continuous support.

My deepest thanks also go to Prof. Mario Marchese, whose trust and encouragement inspired me to pursue postgraduate studies in the first place.

I would also like to thank all my friends and colleagues of the DSP and SCNL laboratories, especially Andrea Sciarrone, Fabio Patrone and Giulio Luzzati for their encouragement, good ideas and extensive comments, that helped me to get through the postgraduate study.

I would like to thank my family for their continuous support, trust and love. Their caring help during these years allowed me to accomplish this work.

Finally, special thanks to my fiancé Luca, who joined me in this wonderful journey together. Your love and support have made this all possible.

Chiara

Introduction

Nowadays the development of the Internet of Things (IoT) framework and the resulting huge number of smart connected devices opens the door to exploit the presence of multiple smart nodes to accomplish a variety of tasks. Multimedia context awareness, together with the concept of ambient intelligence, is tightly related to the IoT framework, and it can be applied to a large number of smart scenarios. In this thesis, the aim is to study and analyze the role of context awareness in different applications related to smart mobile environments, such as future smart spaces and connected cities. Indeed, this research work focuses on different aspects of ambient intelligence, such as audio-awareness and wireless-awareness. In particular, this thesis tackles two main research topics: the first one, related to the framework of audio-awareness, concerns a multiple observations approach for smart speaker recognition in mobile environments; the second one, tied to the concept of wireless-awareness, regards Unmanned Aerial Vehicle (UAV) detection based on WiFi statistical fingerprint analysis.

Audio Awareness for Smart Speaker Recognition

One of the topics tackled during this PhD research consists in studying a multiple observation approach to speaker recognition. The main motivation that inspired this research is that nowadays mobile devices are equipped with

advanced technology that allows exploiting such nodes for different types of processing. For example, modern smartphones, embedding high quality microphones to the purpose of speech processing, are offering more and more audio processing features for the emerging context-aware services and applications.

Driven by their potential capabilities and ubiquity, the research community is heading for the design of efficient methods for mobile devices to cooperate together towards a common goal. On the other hand, understanding audio context is a very powerful tool that can be extremely useful in many smart scenarios, especially in the framework of security and user tailored services.

The goal of this thesis is not only to recognize a known speaker among a pre-defined speaker set, but also to be able to understand if a speaker has an unknown identity, providing the functionality to add this new person to the set of known identities. For this reason, this PhD. research tackles digital audio processing aspects and, in particular, its ultimate purpose is to enhance the accuracy in terms of correct recognition of a system able to identify a specific speaker, starting from an unknown audio sample. In order to accomplish this task, we propose to exploit a signal processing algorithm that takes into account multiple observations of the speech signal acquired by different devices. Conference sessions or academic lectures are typical scenarios in which different speakers talk in an environment where many devices are available. For this reason, enhancing the accuracy of a speaker recognition system by exploiting multiple observations collected by different mobile devices may represent an interesting research topic that can be applied to real life scenarios. A key aspect of this research is the definition of different fusion and clustering algorithms, which allow to efficiently exploit the information

acquired by multiple sensors in order to correctly recognize the speaker.

The research approach and methodology follows different important phases. The first step consists in analyzing the performance of a speaker recognition system that performs the classification task with a decision based on a single observation, while considering different conditions of attenuation and noise affecting the receivers. In order to accomplish this task, a proper simulative framework is required. In the first stages of the research, the results were obtained by running simulation tests in the MATLAB computing environment. To this purpose, we developed specific functions able to simulate a real communication and signal processing system. An important phase of the research consisted in the modelling of suitable audio propagation, attenuation, and noise effects, which can deeply influence the performance of the classification system. An important aspect of this phase of the research consisted in developing a smart pre-processing algorithm able to compensate for the detrimental effects of environmental noise on the system performance. In order to tackle this aspect, we propose and test a smart approach to Voice Activity Detection (VAD), which takes into account different parameters that allow to discriminate between voiced and unvoiced audio speech frames. This smart pre-processing algorithm is able to obtain higher classification accuracy with respect to the traditional audio pre-processing approaches, as discussed in [1, 2].

A step forward in this research work consists in introducing a second receiver, in order to verify whether the presence of multiple observations can increase the performance of the system in terms of classification accuracy. The management of multiple observations requires a method to extract data from each node and to merge this information so as to obtain a final result. To this purpose, different approaches can be considered.

This research work focuses on the study and test of different algorithms capable to manage multiple observations of audio signals to the purpose of enhancing the performances of a traditional, single device speaker recognition system. This aspect of the research is especially tackled in [1, 3]. In particular we propose two fusion schemes aimed at jointly exploiting multiple data: the first method is based on the simple average mean of the observations, without taking into account any a-priori knowledge about distance or noise. The second approach consists in performing a weighted average of the contributions of each receiver, depending on the available a-priori information about the position of the nodes and the noise level. An alternative approach regards the development of clustering algorithms that aim at selecting a subset of the existing nodes, whose observations are able to enhance the performance of the system in terms of classification accuracy.

The latest step in this research topic consists in designing an advanced version of the speaker recognition algorithm particularly suited for mobile and vehicular scenarios. In details, the research work presented in [4] focuses on the development of a specific solution, robust to car noise interference, that can be easily employed in the emerging framework of smart in-vehicle applications. The proposed algorithm is tested with real-life car noise, and results show that the speaker recognition accuracy achieves good results even when the environment is corrupted by vehicular noise.

Wireless Awareness for Fingerprint-based Drone Detection

The second aspect of this PhD research focuses on context-awareness in the framework of smart-cities. Among other aspects, important and delicate topics are the emerging safety and security issues related to the great diffusion

of amateur low-cost drones. Indeed, amateur drones are enjoying great popularity in the latest years, and more and more user friendly, easy-to-pilot aerial and terrestrial devices are available off-the-shelf. People can even remotely pilot them using their smartphones. However, the pervasive use of drones also leads to technical and societal concerns and issues that need to be addressed, related to security, privacy, and public safety. In the coming years, a technology able to monitor, identify and keep away drones from sensitive areas will become fundamental. Therefore, in this thesis we propose a novel WiFi statistical fingerprint-based drone detection method, which exploits the inherent characteristics of drone control and First Person View (FPV) transmissions in order to sense the presence of a remotely piloted vehicle in the nearby WiFi coverage area. The proposed technique takes into account the particular features of the WiFi traffic produced by drones and their controllers, and uses machine learning algorithms to detect the presence of such devices in the considered surveillance area. This approach is presented in detail in [5], together with an extensive performance analysis carried out through experimental tests.

The research follows different stages, from the survey of the state of the art to the development of a novel detection algorithm. The proposed technique consists in a *guardian* entity (*e.g.*, a laptop), embedding a network packet sniffer that captures all traffic flows on the WiFi channel. The acquired data are dumped in a *pcap* file, which can be locally processed (*i.e.*, on-board the *guardian*) or sent to a remote station to extract a set of specific features characterizing the traffic. A flow is defined as any wireless stream of network packets, identified by the couple {*Source MAC Address, Destination MAC Address*}. Packets in which source and destination addresses are swapped, are considered as belonging to the same flow. The captured traffic is processed

and partitioned based on a predefined time window, which determines the time interval at which detection is performed. This enables the system to provide real-time drone detection, instead of being forced to wait for the end of the transmission to identify the nature of the considered flows. The partitioned flows are analyzed and specific features are extracted so that each traffic flow is completely defined by the corresponding feature vector, which we call fingerprint. The main features used to identify the drone traffic are mainly related to the duration and behavior of the traffic flows and the distribution of the corresponding packets, which is a consolidated approach for traffic classification in literature [6]. After the feature extraction phase, the actual identification of drone-specific patterns is performed by means of advanced machine learning techniques that allow recognizing different device classes.

The proposed approach has been tested in different experimental scenarios. The devices employed during the tests are: a *Parrot AR.Drone 2.0* UAV, an older *AR.Drone 1.0*, and a *Jumping Race Max* terrestrial drone. All the experimental tests were conducted in an office environment, which includes also other devices sharing the same WiFi channel and producing interfering background traffic. Examples of such ongoing flows are video streaming, audio streaming, video conference calls, web browsing, FTP downloading, and so on.

The proposed detection technique, tested in a variety of real-life scenarios, is able to efficiently detect and identify intruder drones in all the considered experimental set-ups, making it a promising UAV detection approach in the framework of amateur drone surveillance.

The latest phase of this research consists in analyzing the vulnerabilities related to the aforementioned WiFi based technique, by studying the

influence of video streaming packets on the recognition accuracy and precision. In particular, we consider the effects of stealth attacks on the system performance, highlighting the inherent vulnerabilities of the traditional approach, and devising an advanced WiFi-based drone identification method called *Blind Detection*. This novel approach is not influenced by the video streaming component of drone transmissions, and it is able to recognize the presence of UAV independently of the *FPV* functionality. We propose three different variants of the *Blind* technique: *Pure Blind*, *Joint*, and *Separate* methods, depending on the different approach in the training phase of the detection system. The main concepts behind this advanced approach, and a wide discussion on its performance analysis are reported in [7, 8]. The proposed algorithms are tested in different experimental scenarios and the results show that the *Blind Detection* method is able to achieve high accuracy and precision, improving the detection efficiency of the drone surveillance system even in the presence of stealth attacks.

The aim of this PhD research is to exploit the information collected by different devices in order to gain awareness about environmental context and improve the reliability and accuracy of classification systems. Real life scenarios such as smart-spaces and smart-cities are typical environments in which the approaches proposed in this thesis can be successfully exploited. This PhD thesis focuses on the analysis, implementation and performance evaluation of different systems and algorithms, that can be employed in real life scenarios such as smart-spaces and smart-cities, in order to offer context-aware, smart and innovative signal processing solutions.

For what concerns the general structure, this thesis is organized as follows: Chapter 1 introduces the basic concepts on multimedia context awareness and

signal processing in general, highlighting their importance in the framework of ambient intelligence applications. Chapter 2 tackles the aspects of this research related to audio awareness, describing in detail the proposed speaker recognition system, and highlighting the contribution in the different frameworks of smart voice activity detection, multiple observations approaches, and voice-based vehicular applications. Next, Chapter 3 focuses on wireless awareness in the framework of smart cities, providing insight into novel drone detection approaches, and describing the proposed surveillance techniques based on WiFi statistical fingerprint analysis. Finally, in Chapter 4 we draw the final conclusions on this research.

Contents

List of Figures

List of Tables

Chapter 1

Basic Concepts on Multimedia Context Awareness and IoT

The Internet of Things (IoT) can be thought as the main enabling technology of the fourth industrial revolution [11]. There are a number of different and, at the same time, brilliant definitions of this concept. One of the most effective, refers to IoT as *the network of physical objects, or "things", embedded with electronics, software, sensors and network connectivity, which enables these objects to collect and exchange data* [12].

IoT allows objects to be sensed and controlled remotely across the existing network infrastructure, creating opportunity for more direct integration between the physical world and computer-based systems, and resulting in improved efficiency, accuracy and economic benefit. IoT is not the result of a single novel technology, instead several complementary technical developments provide capabilities that, considered together, help to bridge the gap between the virtual and the physical worlds. These capabilities include: communication and cooperation, addressability, identification, sensing, actuation, embedded information processing, localization, user interfaces.

Thanks to these characteristics, the IoT provides the ability of making objects recognizable and smart by enabling context-related decisions. The diffusion of cheap ubiquitous sensors, that can be easily integrated virtually anywhere, has accelerated the growth of Ambient Intelligence. This concept refers to the capacity of a system to sense the environment and to react to certain events or conditions, according to the typical principles and behaviors derived from situation-awareness.

As a result, the IoT has gained a lot of popularity, propelled by the new advancements in mobile information systems. In general, the concept of Ambient Intelligence is strictly related to the definition of the *Data-Information-Knowledge-Wisdom* (DIKW) hierarchy [9], which evolved in many different variants, each characterizing a peculiar aspect of information. This concept is usually represented by a pyramidal structure (see Figure 1.1), illustrating the upward direction followed by the information flow.

The technological evolution and increasing diffusion of tiny smart objects embedded in everyday things, equipped with enough computational capabilities, and easily interconnected with each other and to the Internet, provides the opportunity to design more advanced and innovative cross-domain applications. These services are able to exploit multiple sensors, actuators, and user-generated data, by managing them and interoperating among different contexts. A large number of these applications is based on Context-Awareness, is highly customizable and can be tailored to the user's preferences and needs, relying on the real-time knowledge of the environment.

The concept of *context-awareness* was introduced in 1994 by Schilit and Theimer, who defined context-aware computing as *"the ability of a mobile user's applications to discover and react to changes in the environment they are situated in"* [13]. If we refer to the dictionary, context is generally defined

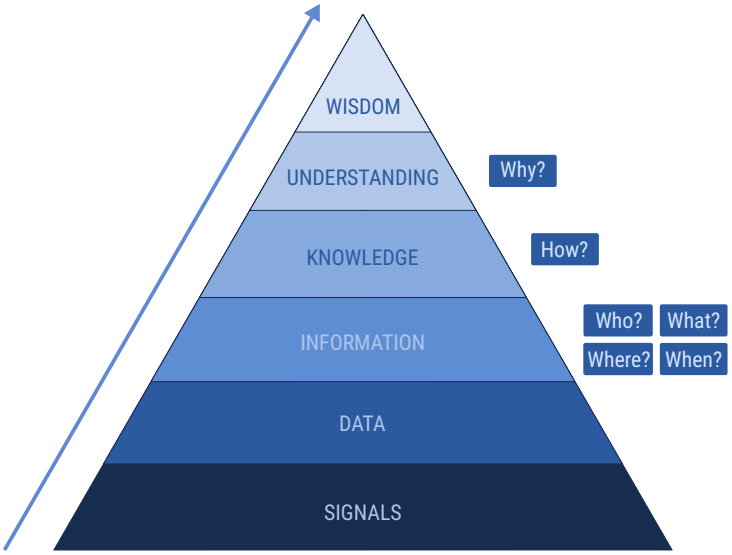


FIGURE 1.1: *The extended DIKW hierarchy [9].*

as the circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood. This means that the context represents the characteristics and the mutual relationships among the different actors of a given situation. This definition has been further modified and refined by researchers to be more suitable for technological applications [14]. As an example, Lieberman et al. proposed another definition where context *"can be considered to be everything that affects the computation except the explicit input and output"* [15]. In contrast with the prior definition of context stated in [13], this description is quite technical and is centred on the application rather than on the user. In this view, context is any information that the system is able to sense beyond the explicit commands and which has an effect on the global state of the system.

Methods to extract context information by employing smart-objects are aimed at providing the answer to the following basic questions about the users' environment: *what*, *who*, *where*, *when*, *why* and *how* [16]. These answers represent a fundamental part of the overall process needed to provide a

complete context-aware service, and they are key steps of the aforementioned DIKW hierarchy. Context refers to the information characterizing the situation of an entity or a group of entities, and it provides knowledge about their current status. The term *context* may assume different meanings based on the scenarios and users involved: activity, geo-spatial information, network state, battery level, energy consumption, environmental parameters, signal-to-noise ratio, and so on. Context awareness also allows for customization and personalized contents to match the preferences of the involved users and entities.

A general scheme for a context-aware system is illustrated in Figure 1.2. Basically, a generic raw signal is acquired into the system through a signal acquisition block, that will be different depending on the nature of the considered signal.

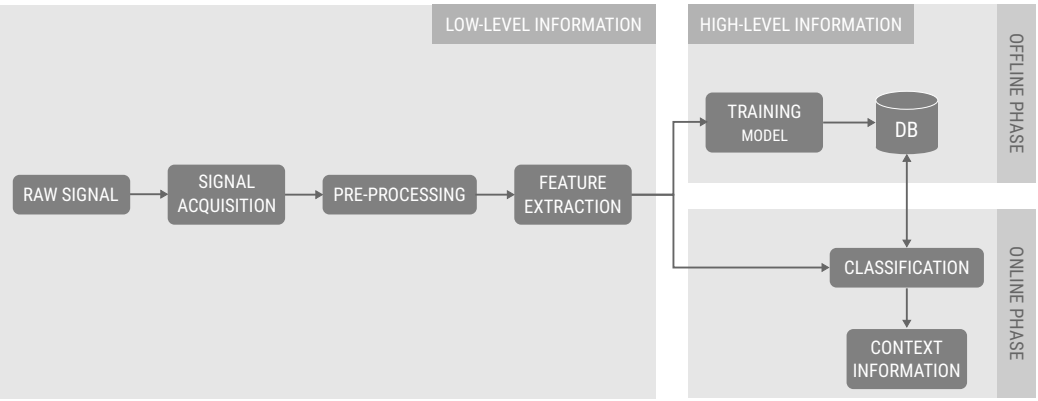


FIGURE 1.2: *General scheme of a context-aware system.*

Next, the acquired signal follows a pre-processing stage, which is meant to prepare the data for the actual processing. This first part of the system refers to the so-called *low-level* information. In order to obtain *high-level* information, after the pre-processing phase specific features are extracted based on the type of signal and the information we need to infer. These

characterizing features are used offline in the training phase to build proper models, which will be stored in a local or remote database, and later used for classification purposes. Finally, during the online phase, the classification system will be used to infer *high-level* context information by comparing the acquired data with the pre-built models. The context information is then used by the context-aware system to discover and smartly react to environmental conditions by adapting its output based on the current scene.

In this thesis we focus on context-aware systems and services, especially designed and implemented as smart products typical of the IoT era. In the following, we will show through practical applications related to the audio and wireless domains, the pervasiveness and ubiquity of smart solutions based on context-awareness, aimed at enhancing ambient intelligence applications driven by the Internet of Things paradigm.

Chapter 2

Audio Awareness for Smart Speaker Recognition

When we consider the Internet of Things, the concept of ambient intelligence can be applied to many real life scenarios. One of the most popular frameworks that is investigated not only by the research community, but also by industrial companies is related to the so called Smart Home/Space environment. In this scenario, multimedia context awareness plays a major role, improving user experience and providing customized applications and user tailored services.

Ambient intelligence related to the Smart Home framework relies on different types of signals and data. Typical examples of signals acquired in this environment are video inputs, audio recordings, temperature measures, location information, and so on. These different raw signals can be processed to extract significant data, which can be employed to estimate context information and infer specific features characterizing the scene.

In this thesis we focus on audio-based signals, which are basic and ubiquitous signals that can be used to obtain important information about context.

In particular, we study and develop a smart context-aware system capable to recognize the identity of an unknown speaker in different environmental conditions. This operation allows gaining wider knowledge about the current scene: information about the speaker, together with a database containing an a-priori known set of identities, can include important information not only about names, but also gender, age, habits, preferences, and other types of characteristics that allow making user-aware decisions and providing customized features.

In the following we will describe in detail the considered speaker recognition system, and we will survey the main related techniques available in the literature. Also, we will present novel approaches to the SR problem, and we will also discuss the main results obtained during our performance tests, carried out in different environmental conditions.

2.1 Speaker Recognition System Architecture

Identifying a person by his or her voice is an important human trait most taken for granted in natural human-to-human interaction/communication. Automatic speaker recognition systems have emerged as important tools to verify identity in many e-commerce applications as well as in general interactions in the framework of business, forensics, and law enforcement.

The problem of Speaker Recognition is to recognize the identity of the speaker who sounds closest to speech analyzed from an audio sample produced by an unknown speaker. It can be tackled in two different scenarios: *closed-set* scenario, when the recognized speaker belongs to a given, a-priori known set, and *open-set* scenario (also called *out-of-set* speaker identification, if the identity of the test subject could also belong to a speaker who is

not part of the predefined known speaker group [17]. We tackle the speaker identification problem in both the aforementioned situations.

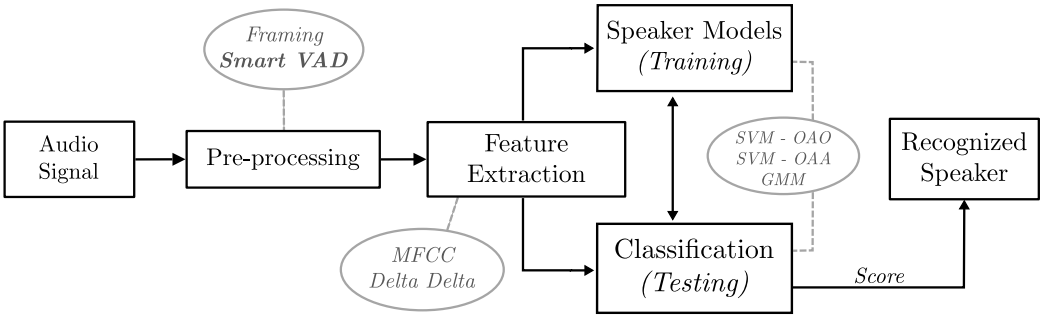


FIGURE 2.1: *The employed speaker recognition system architecture.*

The overview of the employed system architecture is illustrated in Figure 2.1. The proposed system is conceived inside the smart and mobile framework, exploiting ubiquitous connectivity and allowing our system to possibly include a remote server used to store the speaker models or to perform the required computations. Independently of this specific aspect, the general design of the proposed speaker recognition architecture is described in detail in the following.

The acquired audio signal is fed into a pre-processing stage, in which framing and *SmartVAD* are performed. Basically, the raw signal is divided into short segments called frames, during which speech can be considered as stationary. Each frame has a length of $T = 25$ [ms], which is a typical frame duration that allows the speech signal to be considered as stationary. After being divided into frames, the signal undergoes a processing phase, in which frames are analyzed and subsequently kept or discarded based on the proposed smart Voice Activity Detection algorithm. This procedure is described in details in Section 2.1.1.

When the audio signal is conveniently pre-processed, we proceed to the feature extraction phase. At this stage, speech features are computed. In

particular, we employ the first 13 MFCC and the respective 13 DeltaDelta (*i.e.,* the second-order derivative of the MFCC coefficients) for a total of 26 features. For a detailed description and definition of these features we refer the readers to [18] and references therein. After the extraction phase, the aforementioned features will be used to train a supervised classifier, in order to build proper speaker models. Details about the classification process are reported in Section 2.1.4.

2.1.1 Pre-processing: the *SmartVAD*

The first step in order to obtain a good classification system is to pre-process the signal by removing frames that do not contain useful information. When dealing with audio speech signals, this step translates into discarding the audio frames that do not contain speech utterances. This concept is known as Voice Activity Detection (VAD) [19].

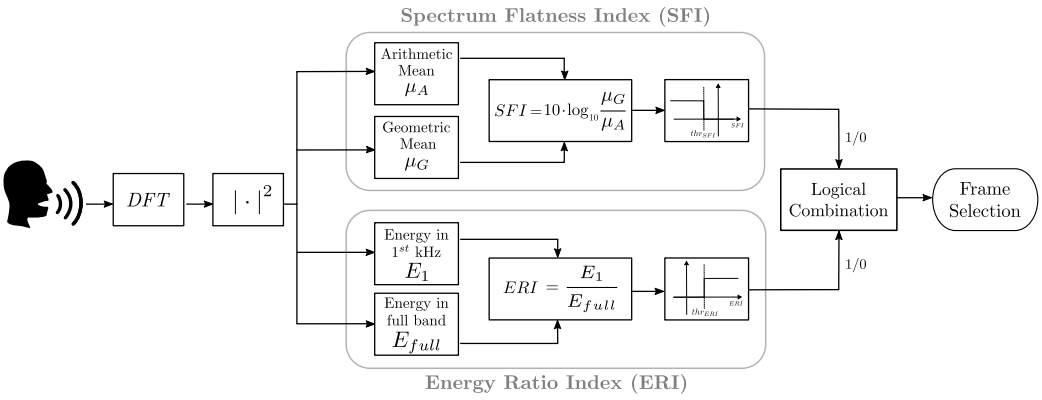


FIGURE 2.2: *The general pre-processing scheme.*

To achieve this goal, many techniques have been proposed in the literature. The simplest consists in applying a Band Pass Filter (BPF) centered over the speech bandwidth (*e.g.,* from about 50 [Hz] to 3500 [Hz]). This action will remove unwanted frequency components that do not fall within

the voice bandwidth, but it does not ensure that the remaining signal components are actually related to speech utterances. In this thesis we use an alternative approach to voice activity detection, which aims at smartly taking into account only actual speech frames. We call this pre-processing method *SmartVAD*, which consists in a short-time spectral analysis pre-processing filtering scheme, reported in Fig. 2.2.

It is based upon two important indicators: *i)* Spectrum Flatness Index (SFI) and *ii)* Energy Ratio Index (ERI) [1]. The rationale behind these two parameters is that a speech frame exhibits a spectrum having most of its energy within the first kilohertz and which should not be flat. Finally, a threshold criterion is applied such that an audio frame is not discarded only if the values of the considered indicators satisfy the threshold condition.

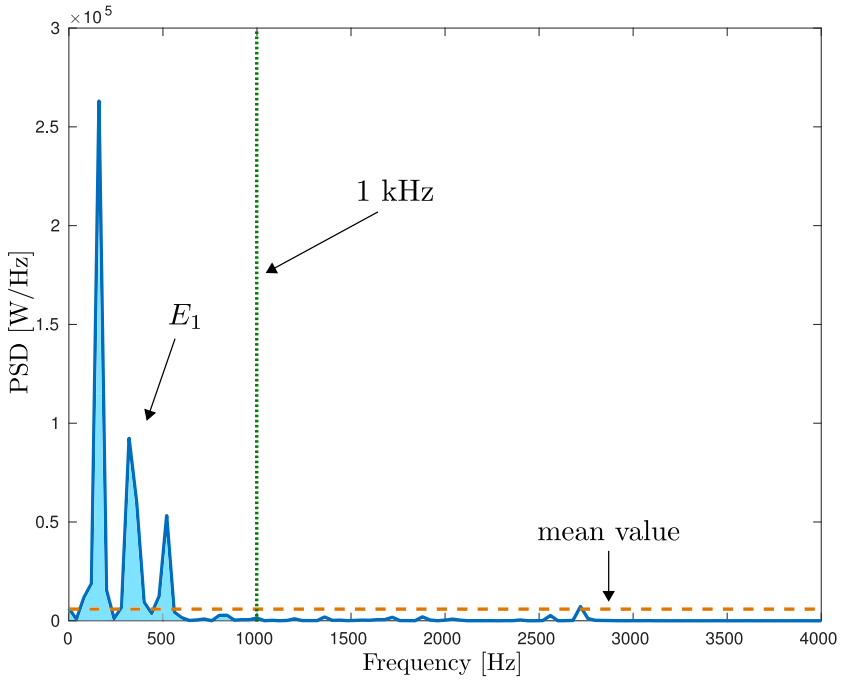


FIGURE 2.3: *PSD of a voiced frame saved by SmartVAD*

Figure 2.3 shows the detail of the Power Spectrum Density (PSD) of an audio frame which is selected by the VAD algorithm as a voiced frame. As it

is possible to notice from the figure, the energy contained in the first kilohertz is a significant percentage of the total energy of the signal. Moreover, the shape of the spectrum is peaky at lower frequencies, in contrast to that of a noise signal, which is flat at all frequencies. These are important indicators of a voiced frame, and they motivate the choice of the proposed VAD indicators, which effectively reflect the nature of the considered signal.

In order to identify the contribution of the two proposed parameters, we conducted a performance analysis comparing the results obtained when the use of the Spectrum Flatness Index and the Energy Ratio Index is combined with different approaches. This study is described in detail in Section 2.1.2. In the following, we will give a detailed explanation of the meaning of the considered VAD parameters.

We denote with $s(n)$, $n \in [1, N]$ the generic numerical audio signal acquired at a sampling frequency f_s and $S(k)$ with $k \in [1, K]$ its Discrete Fourier Transform (DFT).

2.1.1.1 Spectrum Flatness Index (SFI)

SFI is a measure of the noisiness of the signal spectrum and it is a good indicator for voiced/unvoiced/silence frame detection. The Spectrum Flatness Index is determined directly by using the arithmetic and the geometric mean of the spectrum. The arithmetic mean μ_A is computed in the traditional way:

$$\mu_A = \frac{1}{K} \sum_{k=0}^{K} |S(k)|^2. \tag{2.1}$$

In order to be able to obtain a meaningful quantity for the geometric mean also in the case where zero values are present, we compute μ_G as suggested in [20]:

$$\mu_G = w_p \cdot 10^{\frac{1}{K}\sum_{k=0}^{K} log_{10}(|S(k)|^2)}, \tag{2.2}$$

where w_p is a weighting factor that allows to take into account for the computation only the contribution of non-zero values, and it is computed as the number of non-zero values of $S(k)$ over the total number of elements in $S(k)$. These two mean quantities are then combined together to infer a measure of the flatness of the spectrum, as reported in Eq. 2.3.

$$\text{SFI} = 10 \cdot log_{10}\left(\frac{\mu_G}{\mu_A}\right) \tag{2.3}$$

Here the SFI indicator is expressed in $[dB]$ and therefore its values belong to the range $SFI \in (-\infty, 0]$.

2.1.1.2 Energy Ratio Index (ERI)

Audio frames containing speech utterances exhibit the most significant energy components in frequencies belonging to the first kilohertz of the signal spectrum. On the contrary, the majority of unvoiced sounds display spectral concentration energy in a higher frequency range. As a consequence, an effective and simple method to distinguish between speech utterances and background noise consists in taking into account the distribution of energy along the frequency spectrum.

Let $\boldsymbol{\mathcal{F}}$ be the numerical frequency vector, defined as $\boldsymbol{\mathcal{F}} = \left[0, \cdots, f_k, \cdots, \frac{fs}{2}\right]$ with $k \in [1, \frac{K}{2} + 1]$ where the generic element $f_k = \frac{f_s}{K}(k-1)[\text{Hz}]$. The logical block ERI computes the Energy Ratio Index [1] as follows:

$$\begin{cases} ERI = \dfrac{\sum_{k=0}^{K^*} |S(k)|^2}{\sum_{k=0}^{K} |S(k)|^2} \\ K^* = \min\limits_{k} \{f_k\} \text{ s.t. } f_k \geq 1\ kHz. \end{cases} \tag{2.4}$$

From the frames that have not been discarded by the pre-filtering step, the features are extracted as described in [18].

The choice of suitable VAD thresholds is a key aspect that deeply influences the performances of the speaker recognition algorithm. For this reason we performed a preliminary performance analysis, in order to understand the best threshold values to be applied for robust classification. The outcome of this investigation is reported in Section 2.1.3.

2.1.2 VAD Frame Selection Criterion

In order to better understand the influence of the SFI and ERI parameters on frame selection, we ran a performance analysis in which we employ different frame selection criteria, based on the aforementioned indicators. In particular, we compared the speaker classification results obtained when applying the *SmartVAD* with four possible threshold criteria for frame selection. The first two methods take into account only one of the proposed parameters at a time: one method is called *SFI-Only*, and it consists in selecting the frames considering only the threshold related to the SFI parameter; the other approach, on the contrary, compares only the frame values that are related to the ERI parameter (*ERI-Only*), so that only the frames whose ERI parameter value satisfies the threshold condition are saved for classification purposes. The other approaches, instead, are meant to combine together the

contribution of the two indicators. The first joint method is called *AND*, and it selects the audio frame as a voiced signal only if both the VAD parameter values satisfy the respective threshold conditions. The last approach is called *OR* and it is less restrictive: it decides for a voiced frame if at least one of the two parameters is above the threshold.

Classifier	**VAD Approach**				
	SFI-Only	*ERI-Only*	**AND**	*OR*	*NO VAD*
SVM - OAA	73.3	71.8	**74.1**	73,4	62
SVM - OAO	71.3	70.1	**71.9**	71.9	66.3
GMM	61	65	**65.5**	62.9	58.1

TABLE 2.1: *Comparison of the performance obtained with different frame selection criteria.*

A comparison of the average results obtained when employing the different frame selection criteria for speaker classification is reported in Table 2.1. For the sake of brevity, the accuracy values reported in the table represent an average of the results obtained in all simulations, performed by considering all possible combinations of environmental conditions (varying distance, noise, and scenario). Indeed, the global behavior of the system for the various frame selection criteria is confirmed in all simulations, independently of the considered conditions.

As the Table highlights, our study shows that the *AND* frame selection criterion is able to achieve the highest performance with all the considered classifiers. For this reason, we decided to employ this frame selection criterion for the following performance tests. Therefore, in the remainder of this paper the results related to *SmartVAD* performance refer to the values obtained when the *AND* frame selection criterion is applied.

2.1.3 VAD Thresholds Analysis

The choice of suitable thresholds to apply to the SFI and ERI parameters for the evaluation of useful voiced frames is very important for the development of an effective voice activity detection algorithm. For this reason we decided to perform a deep investigation in order to identify the best threshold values to employ for our simulations. The aforementioned tests were run in clean speech conditions, when the speaker lies at a distance of 1 [m] from the receiver.

The performance parameter used for all tests is the classification accuracy α, defined as:

$$\alpha = \frac{N^{corr}}{N^{test}} \times 100, \tag{2.5}$$

where N^{corr} is the number of correctly classified audio files, and N^{test} is the total number of files tested in the considered scenario.

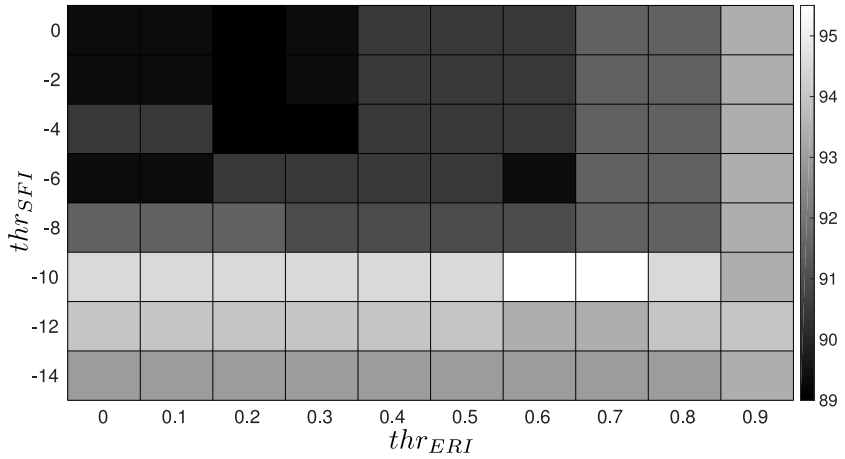


FIGURE 2.4: *Classification accuracy as a function of threshold values averaged for the Closed-Set and Open-Set scenarios.*

A round of classification tests was performed using threshold values varying in the range $thr_{SFI} \in \{-14 \div 0\}$ and $thr_{ERI} \in \{0 \div 0.9\}$. After identifying

the values where accuracy proved higher in *closed-set*, we also investigated the effects of the same range of values in *open-set*, in order to determine a region in which both scenarios can achieve high accuracy. The main results of this joint analysis for the *closed-* and *open-set* scenarios are shown in Figure 2.4. The percentage of accuracy is depicted as a shade of gray, where lighter shades indicate higher performance, as shown in the provided color bar.

The results obtained in this analysis show that we can identify a joint region of threshold values in which the performance of accuracy is remarkably high for both scenarios. We can therefore choose a threshold value both for the parameters SFI and ERI, such that the speaker classification accuracy empirically reaches the highest values. As a consequence, in this paper the following threshold values have been employed: $thr_{SFI} = -10$ and $thr_{ERI} = 0.7$.

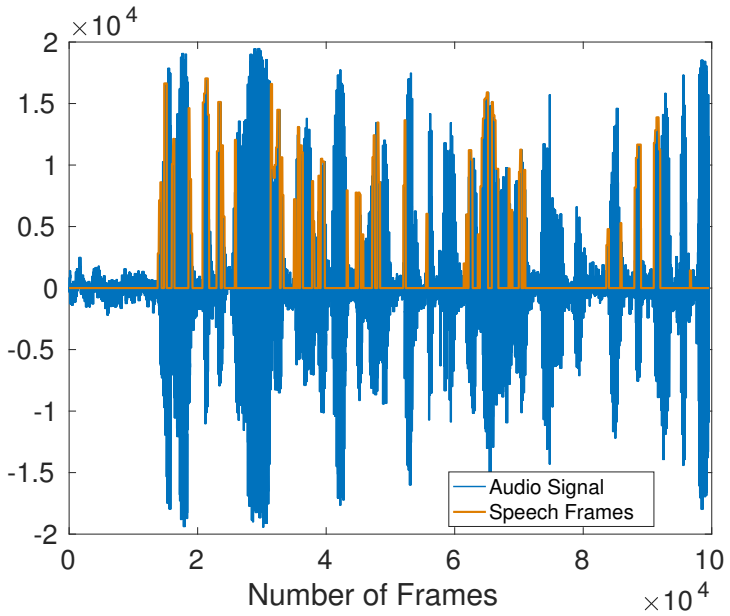


FIGURE 2.5: *Original audio file (blue line) and speech frames selected by the SmartVAD (orange line).*

Fig. 2.5 shows the effects of *SmartVAD* on a sample audio signal. The

dashed red line follows the envelope of the original signal if the *SmartVAD* decides to save the current frame. In all other cases, frames are discarded. It is easy to notice that some frames are discarded by the *SmartVAD* even if there is a significant amount of energy within them.

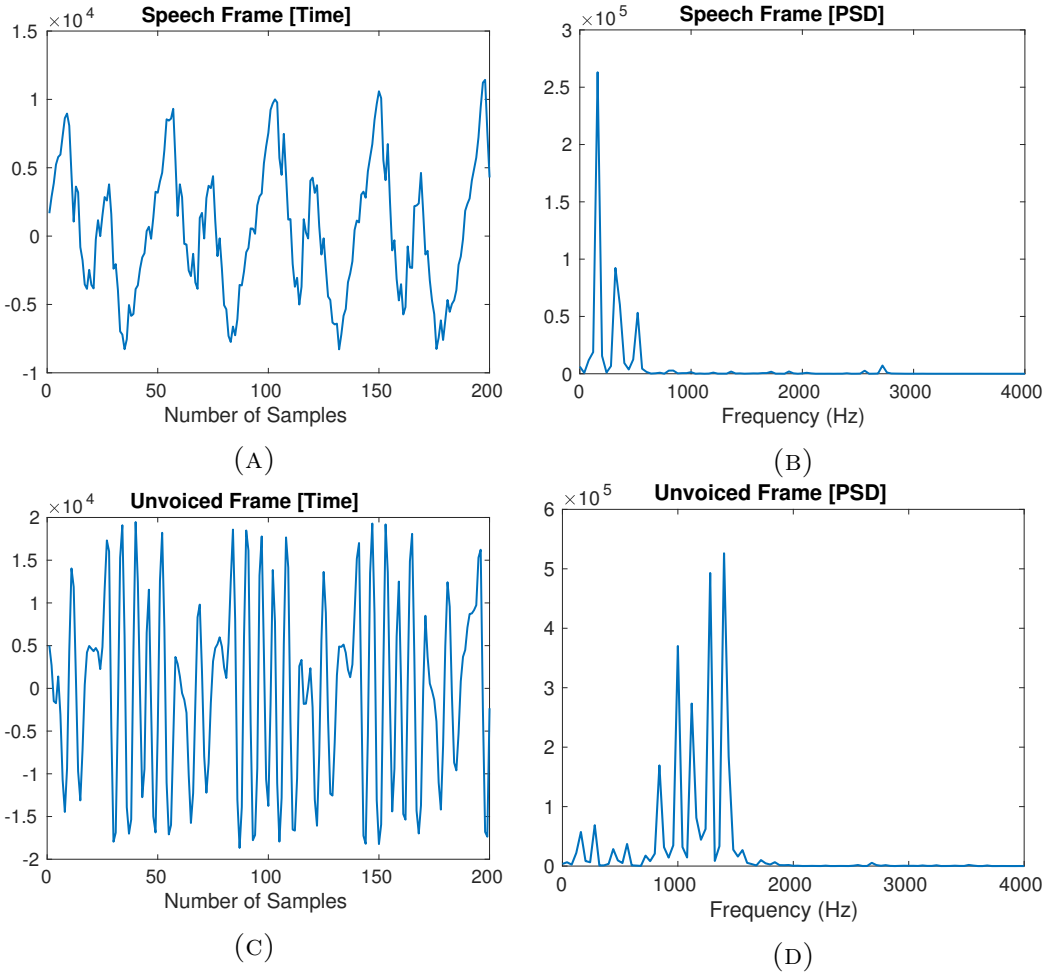


FIGURE 2.6: *Temporal and spectral trends of voiced and unvoiced frames.*

Indeed, a natural speech is composed of both silenced and uttered frames, but unfortunately noise can still affect the signal quality. The aim of *SmartVAD* is to select and save for the final decision only the voiced frames. Figure 2.6 shows the time and frequency domain plot of a voiced frame versus an

unvoiced one. The voiced frame exhibits a regular temporal trend (Fig. 2.6a) and a very clear power spectrum (Fig. 2.6b) in which the three peaks of the main formants are well-separated and evident. In addition, no significant energy is present above the first kilohertz. On the contrary, the unvoiced frame exhibits a more chaotic temporal trend (Fig. 2.6c) and a power spectrum (Fig. 2.6d) showing several peaks at frequencies that are not compatible with the human voice.

2.1.4 Classification: Training and Testing

In this Section, we describe in detail the considered classification approaches used for training the speaker models and for the testing phase. In order to asses the performance of our proposed *SmartVAD* architecture, we compared the effectiveness of the considered pre-processing scheme on top of different classification approaches. Widely used methods for speaker identification in literature are, among others, Support Vector Machines and Gaussian Mixture Models. For this reason we chose these two types of machine learning algorithms to test the performance of our system.

2.1.4.1 Support Vector Machine

SVM is a supervised learning scheme that uses a binary approach to assign samples to a specific class, by dividing the feature space into different regions, one corresponding to each category. The SVM algorithm works in two separate phases: *training* phase and *testing* phase. For the sake of completeness, the details of the employed classification methods are reported in the following with an analogous formalism used in [21, 3] and in the references therein.

Training Phase Being F the total number of frames contained in each recording, we define $\mathbf{X}^f$ the feature vector holding the values of the MFCC and DeltaDelta coefficients for the f-th frame. $\mathbf{Y}$ is the vector containing all the classes (*i.e.,* the names of all speakers). Let $|\mathbf{Y}|$ be the total number of considered speakers and $h \in [1, |\mathbf{Y}|]$ be the classes' index. Y_h is the h-th class (*i.e.,* the single speaker's name). Given the quantity $Y^f \in \mathbf{Y}$ which is the class corresponding to the vector $\mathbf{X}^f$, the association $\left(\mathbf{X}^f, Y^f\right), \forall f \in [1, F]$ is called *observation*.

The main idea of the SVM algorithm is to separate the feature space by means of one single hyperplane: in this paper we employ both the *One-Against-All (OAA)* method that constructs $S_{OAA} = |\mathbf{Y}|$ SVMs, one for each Y_h class of the considered recognition function, and the *One-Against-One (OAO)* approach, which builds $S_{OAO} = \left(\frac{|\mathbf{Y}|}{2}\right) = \frac{|\mathbf{Y}| \cdot (|\mathbf{Y}|-1)}{2}$ models.

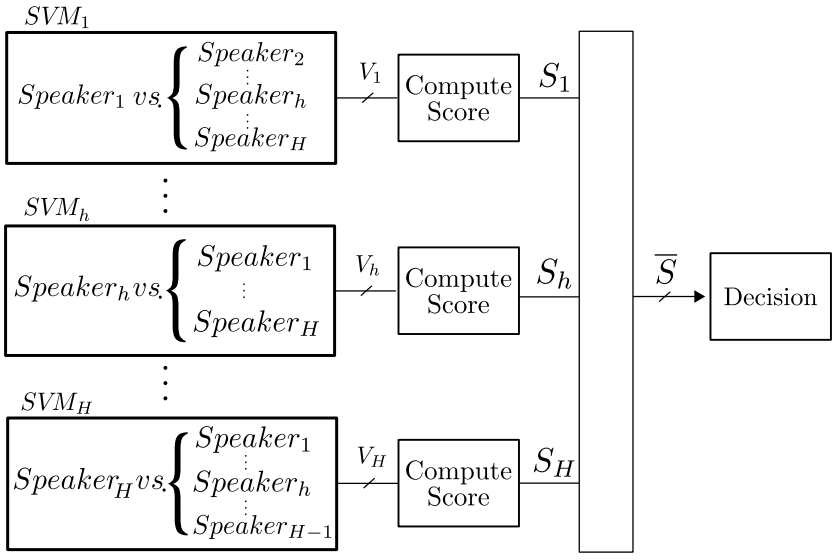


FIGURE 2.7: *Scheme of SVM classification process in One-Against-All approach.*

The inner process with which the model is built is almost the same for both SVM approaches. The main difference between the two types of classification schemes is that, whereas the OAA approach consists in training

one model for each considered known speaker, the OAO approach performs a supervised training of one single model, by assigning a different class to each speaker identity.

In other words, in the OAA approach, the single SVM for the Y_h-th class is trained by employing all the feature vectors of a given training set belonging to the Y_h class with positive labels and all the other feature vectors belonging to the other classes with negative labels [22], performing a sort of binary partitioning (Y_h vs. not Y_h). The OAO approach, instead, consists in training one single SVM for all the considered classes, by assigning different labels to different categories, thus performing a so-called multi-class partitioning. In this way, the total number of labels will be equal to the number of classes we want to identify.

Starting from the audio signals of the training set, the single SVM, built for the class Y_h, can be obtained by computing the aforementioned hyperplane that can be expressed as a function of its orthogonal vector $\mathbf{w}$ that can be obtained from the *Lagrangian Multipliers* computed by solving the problem below. Further details about the analytic description of the SVM have been omitted for the sake of brevity. Eq. 2.6 reports the employed SVM.

$$\underset{Y_h, h\in[1,|\mathbf{Y}|]}{\mathbf{SVM}}^{OAA} : \min_{\boldsymbol{\lambda}} \boldsymbol{\Gamma}(\boldsymbol{\lambda}) = \frac{1}{2}\sum_{f_1=1}^{F}\sum_{f_2=1}^{F}\hat{Y}_{f_1}\hat{Y}_{f_2}\cdot\phi(\boldsymbol{X}_{f_1},\boldsymbol{X}_{f_2})\,\lambda_{f_1}\lambda_{f_2} - \sum_{f_1=1}^{F}\lambda_{f_1}, \sum_{f_1=1}^{F}\lambda_{f_1}\hat{Y}_{f_1} = 0, \quad (2.6)$$
$$0 \leq \lambda_{f_1} \leq C,\ \forall f_1,$$

where $\boldsymbol{\lambda}^Y = \{\lambda_1^Y\ \cdots \lambda_{f_1}^Y\ , \lambda_{f_2}^Y\ \cdots \lambda_F^Y\}$ represents the *Lagrangian Multipliers* vector for $\mathbf{Y}$.

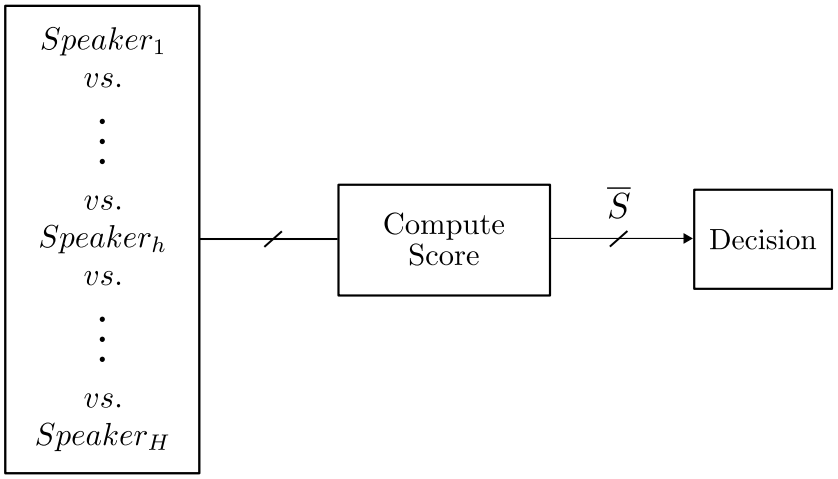


FIGURE 2.8: *Scheme of SVM classification process in One-Against-One approach.*

The SVM performs classification based on a *Kernel function*. In the literature, several kernel functions have been proposed. One of the most common is the *Radial Basis Function* (RBF) or *Gaussian Kernel*, which we employ in this work:

$$\phi\left(\boldsymbol{X}_{f_1}, \boldsymbol{X}_{f_2}\right) = e^{-\gamma\|\boldsymbol{X}_{f_1} - \boldsymbol{X}_{f_2}\|}, \gamma > 0 \tag{2.7}$$

The complexity constant C and the kernel parameter γ have been empirically set, such that $C = 64$ and $\gamma = 0.1$, for both classification schemes.

We denote with F_h and F_k the total number of frames associated to the classes Y_h and Y_k contained within an audio files, respectively. The scalars $\hat{Y}_f \in [-1, 1]$ (see Eq. 2.8) are the numeric binary labels associated to the semantic labels Y_f of the feature vector $\boldsymbol{X}_f$. In practice, we assign a positive label ($\hat{Y}_f = 1$) if the f-th frame belongs to the positive class of the SVM and a negative label ($\hat{Y}_f = -1$) alternatively. In this phase the feature vector is referred to a training set of audio signals acquired and suitably labelled *a priori*.

$$\hat{Y}_f = \begin{cases} 1, & \text{if } Y_f \equiv Y_h \\ -1, & \text{otherwise} \end{cases} \tag{2.8}$$

The SVM used for the research described in this paper has been trained by employing the implementation and guidelines provided in [23].

Testing phase Defining with $|\mathbf{X}^f|$ the number of features for each frame and with F the total number of frames contained within an audio file, the $F \times |\mathbf{X}^f|$ matrix $\boldsymbol{\omega}$, containing all the feature vectors for an audio file is:

$$\boldsymbol{\omega} = \left[\mathbf{X}^1 \cdots \mathbf{X}^f \cdots \mathbf{X}^F\right]^T \tag{2.9}$$

Every time that a speaker must be recognized, the SVM outputs a $F \times 2$ matrix $\boldsymbol{\Omega}^{Y_h}\{\boldsymbol{\omega}\}$, called *Probability Matrix* (PM), one for each trained hyperplane. For what concerns the OAA case, the number of PMs is equal the number of the considered speakers, whereas for the OAO case it corresponds to the number of all possible combinations of all the considered speakers taken in pairs.

Each element $\Omega^{Y_h}_{f,t}$, $f \in [1, F]$, $t \in \{1, 2\}$ of this matrix is the *a-posteriori* probability of the f-th feature vector $\mathbf{X}^f$ belonging to the class identified by the binary label $\hat{Y}_f$, as reported in Eq.(2.10).

$$\boldsymbol{\Omega}^{Y_h}\{\boldsymbol{\omega}\} = \begin{bmatrix} Pr\left(\hat{Y}_1 = 1|\mathbf{X}^1\right) & Pr\left(\hat{Y}_1 = -1|\mathbf{X}^1\right) \\ \vdots & \vdots \\ Pr\left(\hat{Y}_f = 1|\mathbf{X}^f\right) & Pr\left(\hat{Y}_f = -1|\mathbf{X}^f\right) \\ \vdots & \vdots \\ Pr\left(\hat{Y}_F = 1|\mathbf{X}^F\right) & Pr\left(\hat{Y}_F = -1|\mathbf{X}^F\right) \end{bmatrix} \tag{2.10}$$

From all the probability matrices $\mathbf{\Omega}^{Y_h}$, $h \in [1, |\mathbf{Y}|]$, the $F \times |\mathbf{Y}|$ decision matrix $\mathbf{\Psi}\{\boldsymbol{\omega}\}$ is computed, as reported in Eq. 2.11.

$$\mathbf{\Psi}\{\boldsymbol{\omega}\} = \begin{bmatrix} \Psi_{1,1} \cdots \Psi_{1,h} \cdots \Psi_{1,|\mathbf{Y}|} \\ \vdots \\ \Psi_{f,1} \cdots \Psi_{f,h} \cdots \Psi_{f,|\mathbf{Y}|} \\ \vdots \\ \Psi_{F,1} \cdots \Psi_{F,h} \cdots \Psi_{F,|\mathbf{Y}|} \end{bmatrix} \tag{2.11}$$

It is a binary matrix where each element $\Psi_{f,h} = \{-1, +1\}$ is obtained by associating the f-th frame to the binary label $\hat{Y}_f$ which has the highest probability. In formula:

$$\Psi_{f,h} = \arg\max_{\hat{Y}_f} \left\{ Pr\left(\hat{Y}_f = 1 | \mathbf{X}^f\right); Pr\left(\hat{Y}_f = -1 | \mathbf{X}^f\right) \right\} \tag{2.12}$$

From the decision matrix $\mathbf{\Psi}\{\boldsymbol{\omega}\}$, the recognized speaker Y^* is determined. As reported at the beginning of this Section, we evaluate our algorithm both for the *closed-* and the *open-set* scenarios. Both of them require to determine the scoring vector $\overline{\mathbf{S}}$, computed as follows:

$$\overline{\mathbf{S}} = \frac{1}{F} \sum_{f=0}^{F} \Psi_{f,h} \tag{2.13}$$

Each element s_h, $h \in [1, |\mathbf{Y}|]$ represents the scoring of the h-th speaker. It is a measure of the likelihood of the input speech utterance to belong to the h-th speaker. Finally, from the scoring vector $\overline{\mathbf{S}}$ the Maximum Likelihood Index (MLI) h^* is inferred. It is simply the index of the speaker, among the predefined set $\mathbf{Y}$, who has the highest score value (*i.e.,* having the highest *a-posteriori* probability to have produced the input speech signal). In formula:

$$h^* = \underset{h \in [1, |\mathbf{Y}|]}{\arg\max} \left[\overline{\mathbf{S}} \right] \tag{2.14}$$

The recognized speaker is then determined by employing the decision rules reported in Table 2.2, where symbol $Y^{\emptyset}$ indicates that the algorithm has labeled the speech signal as uttered by a speaker that does not belong to the predefined set of speakers (*unknown*).

TABLE 2.2: *Decision rule for the closed- and the open-set scenarios.*

	Decision Rules
Closed-set	$Y^* = Y^{h^*}$
Open-set	$\begin{cases} Y^* = Y^{h^*}, \text{ if } s_{h^*} \geq thr_{U^s} \\ Y^* = Y^{\emptyset}, \text{ otherwise} \end{cases}$

In other words, for what concerns the *open-set* scenario, the maximum score obtained by the speakers belonging to the known speaker set is compared to a predefined threshold, thr_{U^s}. If the highest score is above the considered threshold, the recognized speaker is the one who produced the maximum score, otherwise the classifier chooses for an *unknown* speaker ($Y^{\emptyset}$).

The value of thr_{U^s} strongly depends on the number of speakers belonging to the *a-priori* known speakers group. An empirical analysis, considering up to $|Y| = 6$ speakers, showed that the optimal values for this threshold are:

$$thr_{U^s} = \begin{cases} 0.5, & \text{if } |Y| \leq 2 \\ 0, & \text{otherwise} \end{cases} \tag{2.15}$$

In this paper we set the value of thr_{U^s} to 0, as the number of known speakers used for the training set is 4.

2.1.4.2 Gaussian Mixture Model

The use of GMM to model speaker identity is consolidated in literature as Gaussian components have proven able to represent general, speaker-dependent spectral shapes and have the capability to easily model arbitrary densities. The individual Gaussian components in a speaker-dependent GMM are interpreted to represent acoustic classes which reflect vocal tract configurations that are useful for modeling speaker identity [24].

We therefore propose a GMM-based speaker classification approach which models each speaker identity with a Gaussian mixture. We empirically chose the dimensionality of the GMM, setting $\Gamma = 16$. The approach used to train the Gaussian Mixture Models is the well known Expectation Maximization (EM) algorithm [25]. The details of the proposed method are described in the following.

Training Phase The training phase consists in creating a Gaussian Mixture Model for each speaker belonging to the known speaker set. During this phase, we consider the identity of those speakers we would like to recognize. Each speaker produces a speech utterance used to accurately model the vocal tract conformation through a Gaussian Mixture distribution. The density related to such distribution can be expressed as a weighted sum of Γ component densities, as:

$$p(\overline{x}|\mathbf{\Omega}) = \sum_{i=1}^{\Gamma} p_i g_i(\overline{x}). \tag{2.16}$$

where $\overline{x}$ is a D-dimensional random vector, $g_i(\overline{x},\ i = 1, \ldots, \Gamma$, are the component densities, and $p_i\ i = 1, \ldots, \Gamma$ are the mixture weights. Each component density is a D-variate Gaussian function of the form:

$$g_i(\overline{x}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_i|^{\frac{1}{2}}} exp\left\{ -\frac{1}{2}(\overline{x} - \overline{\mu}_i)' \Sigma_i^{-1} (\overline{x} - \overline{\mu}_i) \right\}. \tag{2.17}$$

where $\overline{\mu}_i$ is the mean vector and Σ_i the covariance matrix. The mixture weights have to satisfy the probability constraint:

$$\sum_{i=1}^{\Gamma} p_i = 1. \tag{2.18}$$

The overall Gaussian mixture density is defined by the mean vectors, the covariance matrices and the mixture weights of all the component densities. In order to identify the distribution, we employ the notation:

$$\mathbf{\Omega} = \{p_i, \overline{\mu}_i, \Sigma_i\}, \ i = 1, \ldots, \Gamma \tag{2.19}$$

For the sake of speaker identification, each model $\mathbf{\Omega}$ corresponds to a Gaussian Mixture and represents the identity of the corresponding speaker belonging to the known speaker set $\mathbf{Y}$. GMMs can be represented using different approaches in terms of covariance matrices. In this work we choose to employ the *full covariance* matrix representation.

The actual training phase consists in creating a Gaussian Mixture model $\mathbf{\Omega}$ for each known speaker, by fitting a GMM distribution over the the set of voice features obtained by processing the audio frames obtained from the training speech samples. In practice, we extract a vector of speech features $\mathbf{X}_f$ (the same coefficients employed in the SVM approach) for each audio frame, and then we build a global matrix ω containing the feature vectors of all the frames belonging to the selected audio file. Later, we use the feature matrix ω to fit a Gaussian distribution $\mathbf{\Omega}$ over the training set data, thus building a model for the considered speaker. This operation is repeated for all the speakers belonging to the known speaker set, leading to a set of $|\mathbf{Y}|$

GMMs.

Testing Phase During the testing phase, speech features are extracted from the test audio files in a similar fashion to the training phase procedure. The obtained features ω are employed in a Maximum Likelihood Posterior Probability test. The objective is to find the speaker model which has the maximum *a-posteriori* probability for a given feature set. The chosen speaker Y^{h*} is then obtained by selecting h^* such that:

$$h^* = \underset{1 \leq h \leq |(Y)|}{\arg\max} \Pr\left(\mathbf{\Omega}_h | \omega\right) = \underset{1 \leq h \leq |(Y)|}{\arg\max} \frac{p\left(\omega | \mathbf{\Omega}_h\right) \Pr\left(\mathbf{\Omega}_h\right)}{p\left(\omega\right)}, \tag{2.20}$$

where the identity is due to Bayes' rule. If we assume that speakers are equally likely, that is $Pr(\mathbf{\Omega}_h) = \frac{1}{|\mathbf{Y}|}$, and noticing that $p(\omega)$ is the same for all speaker models, the final classification rule reduces to:

$$h^* = \underset{1 \leq h \leq |(Y)|}{\arg\max} p\left(\omega | \mathbf{\Omega}_h\right) \tag{2.21}$$

for what concerns the *closed-set* scenario.

Also in this approach, we take into account that the decision can be taken *open-set*, which means that also unknown speakers can be part of the test data. Similarly to the *SVM* approach, the minimum value of the negative log-likelihood of the *a-posteriori* probability is compared to a pre-defined threshold thr_{U^g}, and the outcome of this comparison determines if the recognized speaker belongs to the known speakers group or not. Obviously, the value of this threshold is different from thr_{U^s}, and it was empirically set to -350 for the present work.

2.2 Single Observation Approach to Speaker Recognition

Thanks to the development of the Internet of Things (IoT) framework, the number of connected devices has increased rapidly. The growing interest in speech and speaker recognition is witnessed by the widespread use of IoT applications for security purposes, such as speaker verification and authentication procedures. Moreover, in numerous practical cases a speaker talks in an environment in which many smart devices are present, for example during a seminar presentation, a conference call, or during a lecture in a classroom. For this reason, understanding audio context (e.g., the identity of the current speaker) represents an important tool that can be extremely useful in several realistic IoT scenarios. As the quality of audio signals is deeply influenced by the environmental conditions, an exhaustive study of the performances of the most common speech processing techniques in variable noise conditions and at different source-receiver distances is required.

2.2.1 Related Works

In the literature, many works address the problem of speaker recognition. Among others, valuable ideas are presented in [26], [27] and [28], which employ features such as Mel-Frequency Cepstral Coefficients (MFCC) [21], Perceptual Linear Prediction (PLP) and Linear Predictive Cepstral Coefficients (LPCC) [29, 30], together with some well-known classification tools like Gaussian Mixture Model (GMM) [31, 24], and Support Vector Machine (SVM) [21].

Only some of the literature works, anyway, tackle the problem of speaker identification in noisy environments. Interesting enhancement techniques in

order to take into account the fact that in real scenarios speech spectrum is affected differently in different bandwidths are proposed in [32, 33]. Other approaches might be segmentation and filtering techniques, proposed by [34, 35] to increase robustness in real-world scenarios. The current literature, however, lacks of an exhaustive analysis of the performances of text-independent speaker identification approaches taking into account not only the presence of a noisy channel, but also the joint effects of different types of noise and increasing source-receiver distance on the system accuracy.

2.2.2 Performance Analysis

2.2.2.1 Effects of distance and audio quality on system performance

In this analysis we study the performance of the speaker recognition system when only a single observation is considered. This is a preliminary step to the purpose of analyzing the effects of different environmental conditions on the single observation system. In this way, the obtained results can be used to design proper multiple observation algorithms aimed at improving the efficiency of the speaker recognition system.

Mobile devices are simulated within the MATLAB computing environment on a personal computer running Linux OS. The database used for all tests consists in a set of clean audio signals belonging to different speakers producing utterances in Italian language. For each known speaker 25 audio files have been employed, together with 41 acquired for the *unknown*. Consequently, the overall audio database consists of 141 files divided into 30 female and 111 male speeches.

The position of the audio source signal (*i.e.,* the speaker) is known *a-*

priori. This hypothesis is verified in many practical scenarios (speaker location in a conference, classroom desk, and so on) in which the speaker's position is predefined. The audio signal undergoes an attenuation which depends on the distance d between source and receiver. In order to obtain realistic results, the received signal is also subject to Additive White Gaussian Noise (AWGN). We added the noise to simulate a certain level of Signal-to-Noise Ratio (SNR) at the receiver.

The quantity chosen as performance parameter to compare the classification results of the proposed algorithms is the average classification accuracy α, defined as in 2.1.3.

We analyzed the percentage of the classification accuracy versus a progressively decreasing signal-to-noise ratio SNR $\in \{35, 30, 25, 20, 15, 10, 5, 0\}$ [dB], at different distances $d \in \{1, 2, 3, 4, 5\}$ [m] from the source.

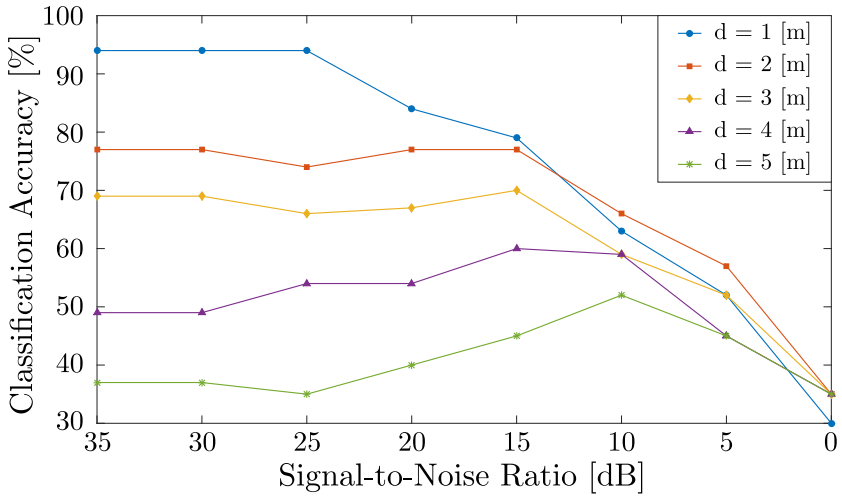


FIGURE 2.9: *Classification accuracy against snr at different source-receiver distances in closed-set.*

Figure 2.9 and 2.10 show the results related to the *closed-set* and *open-set* scenarios, respectively. The performance in terms of accuracy clearly decreases as the distance between source and destination increases, as it

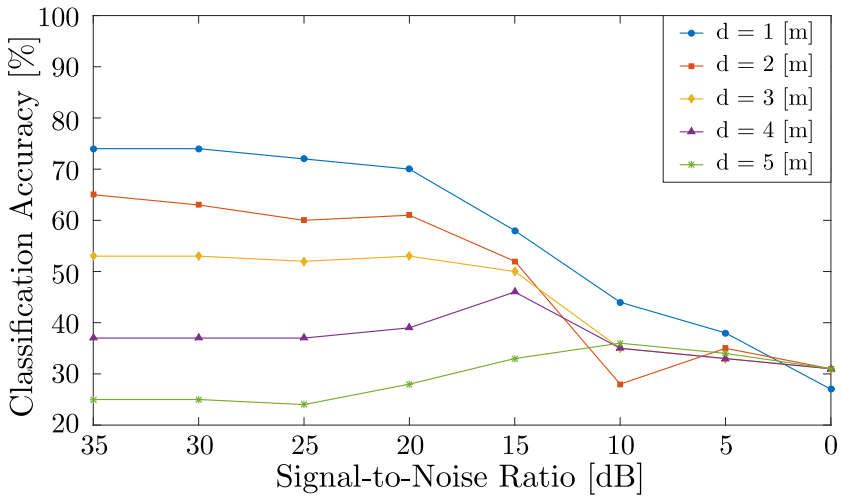


FIGURE 2.10: *Classification accuracy against snr at different source-receiver distances in open-set.*

would seem reasonable. Furthermore, the percentage of correct classifications strongly decreases as the SNR decreases: when the signal-to-noise ratio drops under 15 [dB] the system performance tends to converge independently of the considered distance, and it progressively becomes unsatisfactory (with an average accuracy dropping under 50%). In these conditions the system outcome is unpredictable as the recognition is excessively corrupted by the noise.

2.2.2.2 Effects of noise and classifier on system performance

In this analysis we want to study the influence of different types of noise on speaker recognition accuracy. Also, we compare the performance of the system when using different machine learning approaches to classify the speech.

The simulative setup is the same employed in the previous analysis, but the database used for our tests is different. The considered dataset consists in a set of clean audio signals produced by different speakers (both male and female) in Italian language. For the *closed-Set* scenario, we considered a total

of 64 audio files, 16 for each of 4 speakers (half male and half female) belonging to the known speaker set. For what concerns the *open-Set* scenario, we added to the previously mentioned files, 16 audio files belonging to unknown speakers with the same gender partitioning of the training set.

During the tests we took into account 5 different source-receiver distances in the range $d \in \{1 \div 5\}[m]$, spaced by $1[m]$. The audio signal undergoes an attenuation which depends on the distance d from the source. In order to consider a realistic scenario, we analyzed the performances of the system both in clean speech conditions and when the signal is corrupted by environmental noise of different types. In particular, we tested the proposed algorithm in presence of Additive White Gaussian noise (AWG), Flicker noise ($PINK$) and Speech-Shaped noise (SS), which are typical environmental noise types that can corrupt a speech signal [32, 36]. The noise source is supposed at the distance of $1[m]$ from the receiver and exhibiting a root mean square value of 0.03. The frame selection criterion employed in this analysis is the AND approach described in Section 2.1.2. The threshold values for both the parameters we employed are equal to $thr_{SFI} = -10$ and $thr_{ERI} = 0.7$, chosen as a result of the empirical analysis reported in Section 2.1.3. Also in this case, we use as performance parameter the classification accuracy, defined as above (see Eq. 2.1.3).

The main objective of the test consists in the comparison of different machine learning algorithms (SVM with One-Against-All approach, SVM with One-Against-One approach, and GMM) for the purpose of speaker recognition in different noisy conditions, when the speaker lies at increasing distance from the receiver. Figure 2.11 shows the performance obtained by employing the SVM classifier in One-Against-All approach. In all figures, the bars showing darker color faces refer to the tests performed in *closed-set* scenario,

while the stacked bars with a lighter shade are related to the corresponding *open-set* case. As one can note, if the signal is affected by white Gaus-

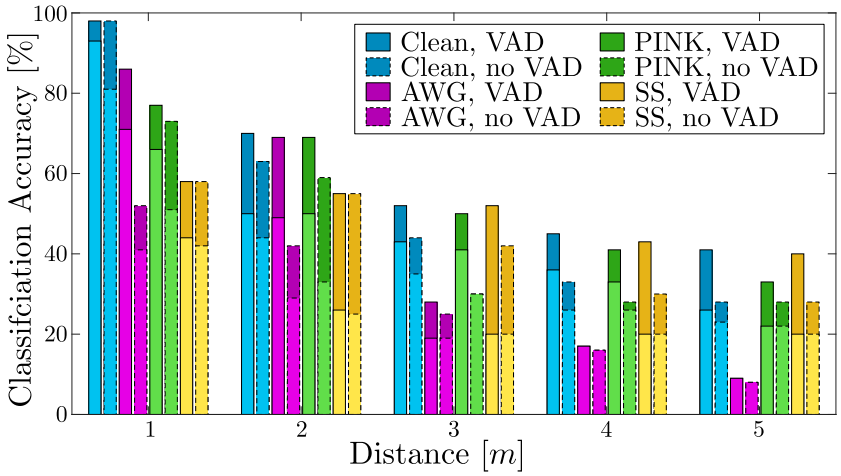


FIGURE 2.11: *SVM-OAA classifier performance with and without VAD in different noisy conditions.*

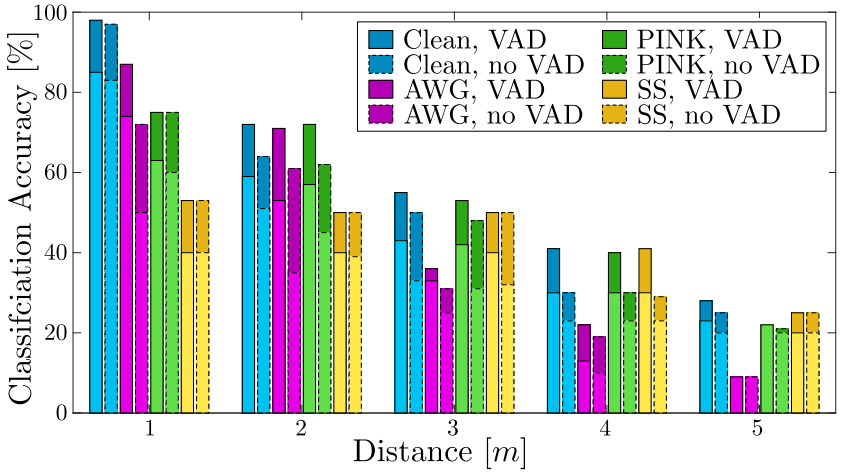


FIGURE 2.12: *SVM-OAO classifier performance with and without VAD in different noisy conditions.*

sian noise the performance of the system drops under 40% as the distance increases over 2 meters. Moreover, the results show that applying our proposed *SmartVAD* approach is able to reduce the corrupting effects of noise on the voice signal. As the lighter bars highlight, in the *open-set* case the

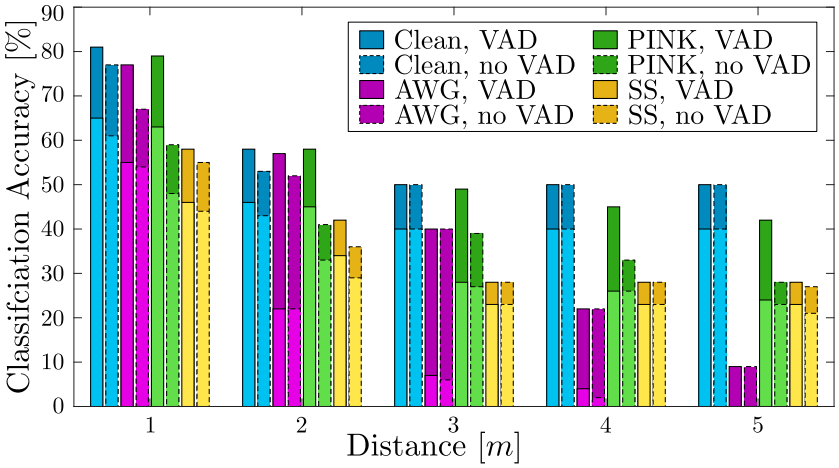


FIGURE 2.13: *GMM classifier performance with and without VAD in different noisy conditions.*

speech-shaped noise (similar to an interfering voice) has deeper effects on the global classification performance, since the scenario also includes speech utterances that do not belong to known speaker voices. However, the pre-processing phase is able to enhance the accuracy of the system in most of the cases. Another round of tests was run when employing the SVM classifier configured with a *OAO* approach (Figure 2.12). This bar plot confirms the behavior exhibited by the corresponding results of the SVM algorithm in *OAA* configuration, where *AWG* noise detrimental effects are stronger as the distance increases.

Finally, system performance was tested when employing a Gaussian Mixture Model classifier (Figure 2.13). In this case, the GMM classifier performs slightly worse that the SVM approach when the speaker lies near the receiver. Also in this case the use of the *SmartVAD* proves fundamental for boosting the system accuracy, especially when the signal is corrupted by flicker noise. It is worth noticing that in the *open-set* case the white Gaussian noise strongly corrupts the recognition performance, the more as the distance increases. However our *SmartVAD* scheme is able to enhance the

accuracy level of the system when the speech signal is affected by pink noise and lies within 3 meters from the receiver.

Considering the performances of the different classifiers, the results show that both SVM approaches perform well when the speech source is near the receiver, whereas the GMM classifier works better than SVM at a farther distance. Indeed, clean speech recognition accuracy is around 50% even at a distance of 5 meters. We can see that the most performing classifier in noisy conditions is the Support Vector Machine in OAA approach, followed by the SVM in OAO configuration. The latter, however achieves lower values of accuracy with respect to the former, when the distance between speaker and receiver exceeds 4 meters. The GMM method can obtain comparable results when source and receiver are at 1 meter of distance, but the performance degrades rapidly as the voice signal experiences attenuation.

2.3 Multiple Observations Approach to Speaker Recognition

When multiple observations of an audio signal are available, the need of efficiently managing the information gathered by different devices arises. Thanks to the latest innovations, different Mobile Devices (MDs) can even communicate without a network infrastructure, by using the Device-to-Device (D2D) communication paradigm, exploiting technologies such as WiFi Direct or Bluetooth. The multiple observations approach consists in fusing acquisitions coming from different but homogeneous devices, that experience different distances and SNRs with respect to the speech source. The general scheme of the employed approach is sketched in Fig. 2.14. Its key idea is to improve the performance of a system with respect to the case in which only

a single sensor is employed.

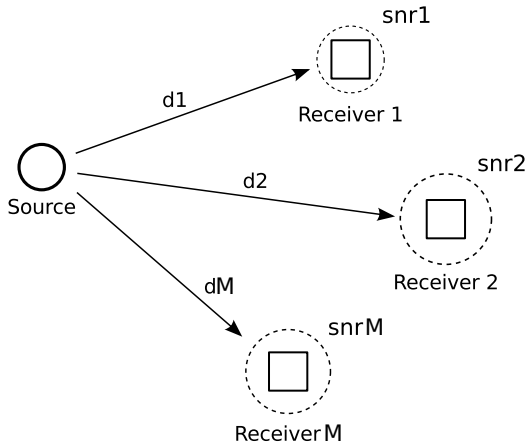


FIGURE 2.14: *The multiple observations approach.*

In the literature, the majority of the works that rely on multiple observations to enhance the performance of the systems employ a multi-dimensional array that implements audio beam-forming. For example, [37] exploits multiple observations from two dimensional audio sources for speaker localization applications. [38] considers the multiple observation approach for extracting the desired speech signals uttered by moving speakers in a reverberant environment. Some works also present speech applications in which separate audio sources are considered. [39] tackles the problem of multiple audio sources data fusion for the purpose of speaker diarization, by exploiting the time-delay-of-arrival of the features coming from different sources. These delays are input as an extra feature vector and are modeled by an HMM classifier. Finally, [40] enhances the robustness of a speaker recognition system by using *i-vectors*, a compact representation of a speakers utterance, from multiple speech sources. Differently from our work, [40] employs speech segments with an average duration of 2.5 minutes and audio data taken from professional speech databases and, consequently, does not considered audio data acquired with *off-the-shelf* devices. The aim of this thesis is to investi-

gate the feasibility of enhancing the accuracy (*i.e.*, the percentage of correct recognition) of SR algorithm by taking into account multiple observations of speech signal coming from several receivers (*e.g.*, smartphones or other smart-objects). In order to achieve this task, we propose different fusion and clustering algorithms to the purpose of jointly exploiting data coming form different devices in both the *open-* and *closed-set* scenarios.

2.3.1 Multiple Observations Algorithms

Multiple receivers can be involved in the recognition process by exploiting different schemes in order to improve the speaker classification accuracy through a collaborative approach. When we consider audio signals in a real-life environment, the concept of *context* translates into information about distance and noise. In this thesis, we assume that each receiver knows the Signal-to-Noise Ratio *a-priori*. It is object of ongoing research to obtain such SNR directly, by estimating the power of both signal and noise from the frame selected or discarded by the VAD, respectively. The approaches proposed in this thesis consist in employing information at different levels:

- *score-level*
- *feature-level*

We develop different methods to manage multiple observations, that aim at obtaining an enhanced speaker recognition algorithm by jointly exploiting data coming from different receivers.

2.3.1.1 Score-Level Fusion

Average (*AVG*) The first method performs a simple arithmetic mean of the scores obtained at the different receivers. Each device performs the

classification independently of the other MDs and it produces an output vector $\mathbf{S}$, whose elements refer to the scores of the speakers belonging to the *a-priori* speaker set. The final score vector $\mathbf{S}^*$, obtained by applying the fusion criterion, is the following:

$$\mathbf{S}^* = \frac{1}{M} \sum_{i=1}^{M} \mathbf{S}_i \tag{2.22}$$

where M is the total number of receivers and $\mathbf{S}_i$ is the score vector produced by the i-th receiver.

Weighted Average (*S2D*) The second approach computes a weighted average of the contributions of the various receivers, by taking into account the attenuation and noise conditions related to the specific MD. As confirmed by the single observation analysis reported in Section 2.2, the classification accuracy decreases as the source-destination distance increases and as the signal-to-noise ratio decreases. We define a context parameter p_i that is proportional to the SNR and to the distance such that:

$$p_i = \frac{(SNR_i)^{\xi}}{d_i} \tag{2.23}$$

where SNR_i is the signal-to-noise ratio affecting the i-th receiver and d_i is the distance between the source of the audio signal and the i-th MD. The quantity ξ is a power factor that can enhance the contribution of the noise with respect to the attenuation. The weighting factor w_i is obtained by normalizing the parameters computed for the different nodes:

$$w_i = \frac{p_i}{\sum_{i=1}^{M} p_i} \tag{2.24}$$

The overall output scoring vector $\mathbf{S}^*$ resulting from this approach is therefore:

$$\mathbf{S}^* = \sum_{i=1}^{M} w_i \cdot \mathbf{S}_i \tag{2.25}$$

which is the weighted sum of the contributions obtained by each receiver.

2.3.1.2 Score-Level Clustering

As the MDs' conditions become more challenging, the contribution of the observations will be more corrupted by noise, so affecting the overall classification performance. To efficiently tackle this issue, we also propose two clustering methods that aim at exploiting only a subset of the available receivers. Considering the parameters defined earlier, we select the MDs that will contribute in recognizing the speaker based on pre-defined thresholds.

Parameter-based Clustering (PCL) The first clustering algorithm computes the parameter p_i for each available MD as reported in Eq. (2.23) and then compares this value to a threshold ϵ_p. If it is lower than the threshold, the i-th node is turned off and its contribution is not considered for computing the final scoring vector $\mathbf{S}^*$. In formula:

$$w_i = \begin{cases} \frac{p_i}{\sum_{i=1}^{M} p_i}, & \text{if } p_i \geq \epsilon_p \\ 0, & \text{otherwise} \end{cases} \tag{2.26}$$

Weight-based Clustering (WCL) The last clustering algorithm proposes an alternative selection criterion, which consists in using the weighting factor w_i as clustering parameter: if w_i is lower than a pre-defined threshold

ϵ_w, the MD is excluded from the computation of the final score.

$$w_i = \begin{cases} \frac{p_i}{\sum_{i=1}^{M} p_i}, & \text{if } w_i \geq \epsilon_w \\ 0, & \text{otherwise} \end{cases} \tag{2.27}$$

In both the clustering methods if all the receivers are in conditions such that their parameters have values below the thresholds, these methods produce no output. The rationale is not to take any decision if the observations is excessively corrupted by noise.

2.3.1.3 Feature-Level Concatenation (*FCAT*)

This approach consists in considering the contributions of the various MDs at the feature level. In particular, the features are extracted from the frames that our VAD algorithm considered as valid speech frames at each receiving node, thus generating a single feature matrix $\boldsymbol{\omega_i}$ for each available MD. All the feature matrices are then concatenated as described in Eq. 2.28.

$$\boldsymbol{\omega}_{\text{cat}} = [\boldsymbol{\omega}_1 \cdots \boldsymbol{\omega}_i \cdots \boldsymbol{\omega}_M]^T \tag{2.28}$$

The final decision is then taken in a centralized way, by considering for the testing phase the global feature vector $\boldsymbol{\omega}_{\text{cat}}$. The trained SVMs will therefore output a global score $\boldsymbol{S}$, which will determine the prediction of the speaker identity.

2.3.2 Performance Analysis

The experiments proposed in this thesis focus on the application of the proposed methods at two different levels: *i)* score fusion and clustering and *ii)*

feature concatenation. For each of them, we consider both the cases in which an unknown speaker is considered or not.

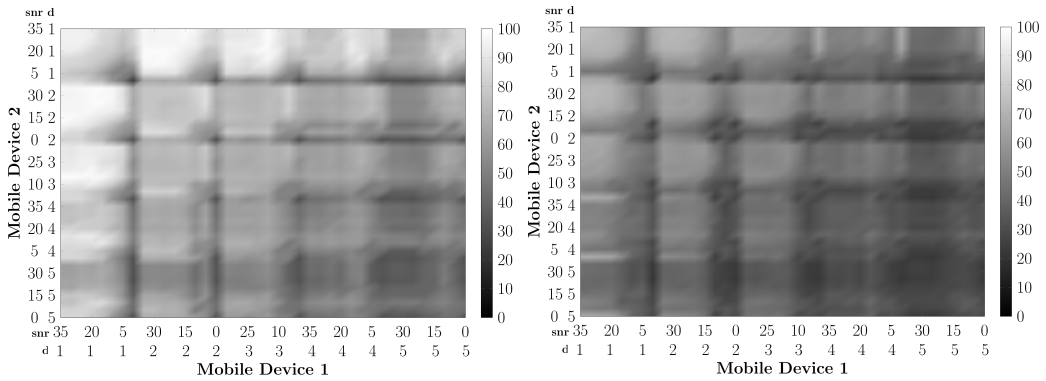
Mobile devices are simulated within the MATLAB computing environment on a personal computer running Linux OS. The database employed for all tests consists in a set of clean audio signals belonging to different speakers producing utterances in Italian language. We considered 25 audio files for each known speaker, and 41 files for the *unknown*. Consequently, the overall audio database consists of 141 files divided into 30 female and 111 male speeches. For the sake of replicability, the whole database of speech signals is made available at [41].

The position of the audio source signal (*i.e.,* the speaker) is known *a-priori*. The audio signal is received by different devices and it undergoes an attenuation which depends on the distance d_i between source and the *i-th* MD. In order to obtain realistic results, the received signal is also subject to Additive White Gaussian Noise (AWG). We have added noise to simulate a certain level of Signal-to-Noise Ratio (SNR), so that the i-th MD exhibits a specific SNR_i. Also in this case, we use as performance parameter the above-defined average accuracy α.

The performance tests show the classification accuracy when multiple observations are considered. We ran several tests in order to understand the behaviour of the proposed approaches in an exhaustive range of conditions. All simulations are performed in a MATLAB computing environment with the same characteristics as described above. In all performance tests we consider a number up to 5 receivers, lying at different distances from the audio source $d \in \{1 \div 5\}$ [m] and experiencing a signal-to-noise ratio $SNR \in \{35 \div 0\}$ [dB]. The analysis is performed by simulating all possible combinations of the conditions of each receiver. Since the proposed algorithms exhibit very

similar behaviours for all the tested scenarios, for the sake of brevity we only report the results related to the case when 2 receivers are involved.

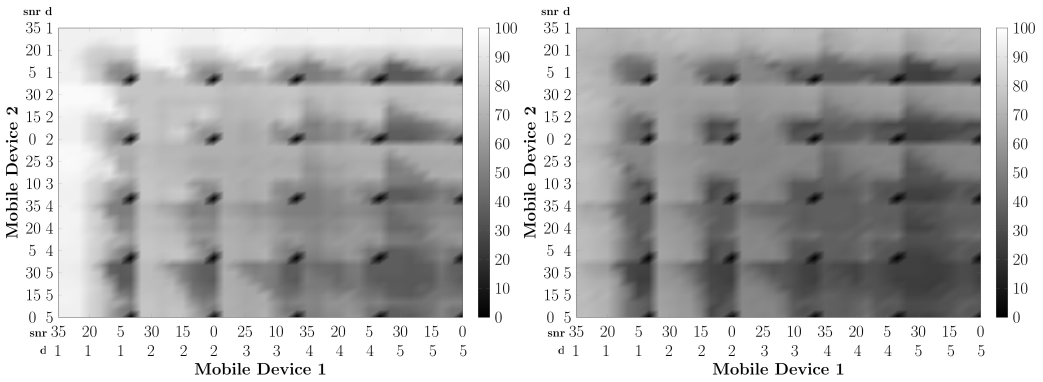
A set of audio signals produced by speakers belonging to the speaker set is received at each MD independently, and the respective score vectors $\mathbf{S}_i$ are produced. The final score is then obtained by applying one of the proposed fusion or clustering algorithms. Consequently, the classification accuracy for that specific method is computed. The results are shown as a set of shaded matrices representing the accuracy obtained by using the different fusion and clustering algorithms in a multiple receiver scenario when 2 MDs are involved. To better understand the obtained performance, we present the results when the conditions of the first and second node vary in the same range we considered for the single observation case. The pictures shows the classification accuracy as a shade of gray: the lighter the shade, the higher the percentage of correct recognition, as stated in the provided color bar. For the sake of clarity, the accuracy matrices are presented with a full gray scale $\{0 - 100\}$ in all cases.



(A) (B)

FIGURE 2.15: *Accuracy matrices obtained with AVG in closed-set (2.15a) and open-set (2.15b).*

All the proposed algorithms are first tested in a *closed-set* scenario, which



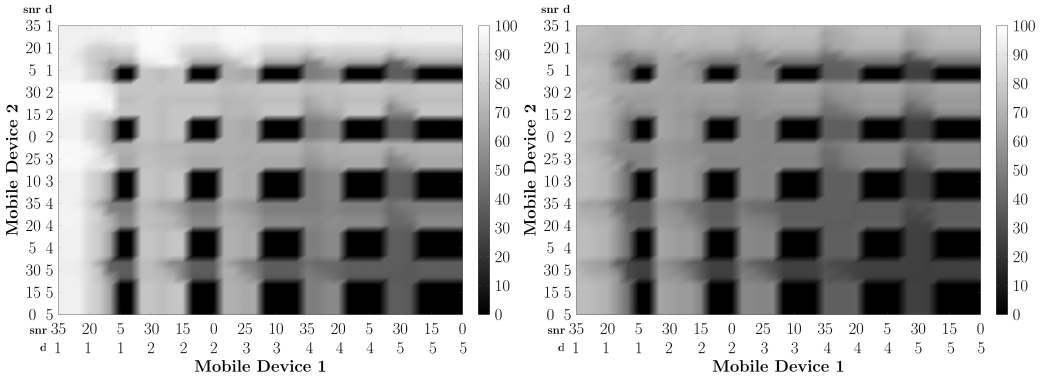
(A) (B)

FIGURE 2.16: *Accuracy matrices obtained with S2D in closed-set (2.16a) and open-set (2.16b).*

means that all the test files belong to speakers from the *a-priori* known set. Figure 2.15a shows the percentage of classification accuracy when the *AVG* score fusion method is applied. As mentioned in Section 2.3.1, this method consists in considering the contribution of each receiver by computing the average mean of the scores produced by each receiver (Eq. 2.22). This approach works fine when both receivers exhibit good conditions in terms of signal-to-noise ratio and attenuation, but it produces low accuracy values when the conditions become more challenging.

To tackle this effect, we apply the *S2D* fusion method. This approach takes into account *a-priori* knowledge about noise and distance in order to appropriately weight the contributions of each receiver. The scheme consists in using the parameter defined in Eq. 2.23, with $\xi = 2$. The results obtained using this algorithm are presented in Figure 2.16a. The image shows that this method is able to improve the classification accuracy in most cases, and the performance enhancement is particularly evident when the conditions of the receivers are more challenging. It is also important to note that, due to the weighting procedure, the contribution of the receivers whose signal-to-noise

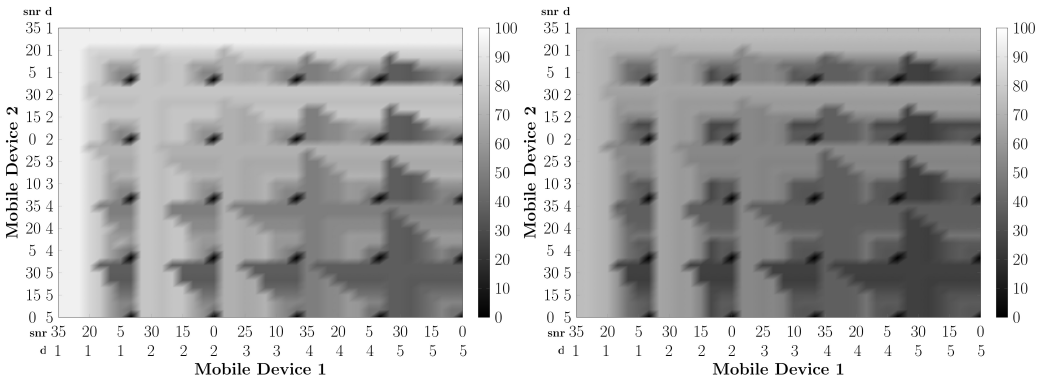
ratio lies at 0 [dB] is cut off by the algorithm. This effect is represented in the picture as a black spot, which does not indeed correspond to 0% accuracy, but it must be intended as a lack of output from the system. If the available receivers are strongly affected by noise and the signal experiences deep attenuation, the classification accuracy decreases critically. For this reason we propose, as an alternative to the weighting solution, a cluster selection of the nodes based on their context conditions. The general idea behind all clustering methods consists in considering only the data collected by to those nodes whose clustering parameter lies above a pre-defined threshold (*i.e.*, $p_i \geq \epsilon_p$ or $w_i \geq \epsilon_w$, depending on the considered method).



(A) (B)

FIGURE 2.17: *Accuracy matrices obtained with PCL, $\epsilon_p = 100$, in closed-set (2.17a) and open-set (2.17b) scenario.*

The first clustering scheme employs the selection criterion defined in Eq. 2.26, with $\xi = 2$. This method is the cluster-based version of the *S2D* fusion approach and it is called *PCL*. The main difference between this approach and the weighted average is that, when the parameter value of a node lies under threshold, its contribution is ignored. Figure 2.17a shows the classification accuracy obtained by applying this criterion, with an *a-priori* clustering threshold ϵ_p empirically set to 100. The black areas displayed in the figure



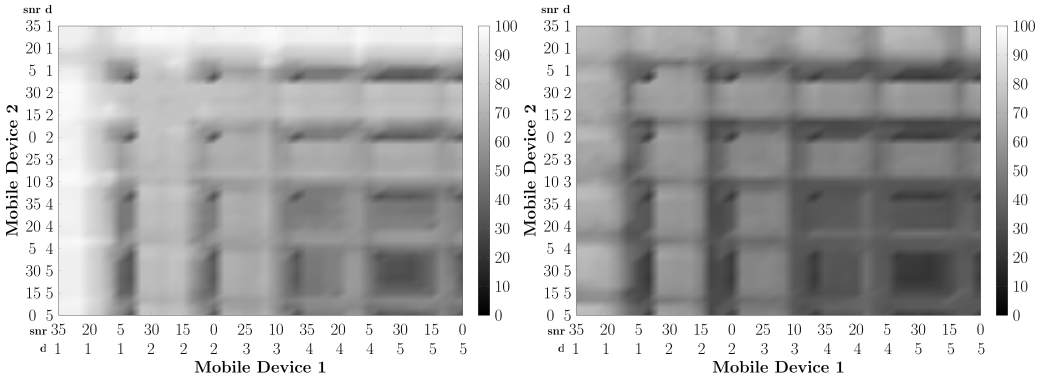
(A) (B)

FIGURE 2.18: *Accuracy matrices obtained with WCL, $\epsilon_w = 0.5$, in closed-set (2.18a) and open-set (2.18b) scenario.*

highlight the fact that the clustering technique introduces cases in which all the nodes lie under threshold and consequently the system produces no output. The rationale is that we choose not to produce any output, rather than making a decision based on excessively corrupted information.

The second proposed clustering method uses the weighting factor w_i as clustering parameter and for this reason it is called *WCL*. This approach is defined in Eq. 2.27 and is computed by setting $\xi = 2$ in Eq. 2.23. The clustering threshold ϵ_w employed for excluding the nodes in worse conditions is empirically set to 0.5. The accuracy obtained by applying this method is shown in Figure 2.18a. This clustering algorithm exhibits a behaviour which is quite similar to that of the *S2D* technique in terms of response to the nodes' conditions. The main difference between these two schemes, however, is that the clustering approach allows achieving higher performance with respect to the corresponding fusion method, as shown by the lighter shades of gray reported in the picture.

The last tested method is related to the features concatenation technique and it is therefore called *FCAT*. This approach consists in sharing the ob-



(A) (B)

FIGURE 2.19: *Accuracy matrices obtained with FCAT, in closed-set (2.19a) and open-set (2.19b).*

Accuracy	*AVG*	*S2D*	*PCL*	*WCL*	*FCAT*
above	19.1%	22.2%	9%	0.3%	29.5%
between	79.6%	73.7%	71.4%	98.5%	67.9%
below	1.3%	2.8%	0%	0%	2.6%
no output	0%	1.3%	19.6%	1.3%	0%

TABLE 2.3: *Comparison between multiple and single observation values in closed-set.*

servations of the different receivers at feature rather than at score level, as described in Eq. 2.28. This means that the classification is performed only once, with a centralized approach, putting together the information collected by the various receiving nodes. The accuracy results related to this technique are shown in Figure 2.19a. This method exhibits a global behaviour similar to that of the *AVG* fusion algorithm, but it provides in most cases better accuracy results, as confirmed by the lighter shades of the picture. The results of this round of tests confirm that the multiple observation approach can indeed enhance the performance with respect to the single observation method. In many cases the classification accuracy is improved if we consider observations coming from multiple devices undergoing different signal-to-noise ratios and

Device	*snr [dB]*	*dist [m]*	Accuracy (α)					
			single obs.	*AVG*	*S2D*	*PCL*	*WCL*	*FCAT*
MD_1	35	1	94%	96%	98%	98%	94%	98%
MD_2	25	2	74%					

TABLE 2.4: *Classification accuracy (α) of the proposed techniques in closed-set.*

source-receiver distances. This trend is evident both in the *closed-* and in the *open-set* scenarios, even if the former offers more significant improvements. These interesting results are highlighted in Table 2.3 for what concerns the *closed-set* approach, while Table 2.4 shows the exact values of the accuracy for a representative case of study.

We tested the proposed algorithms also in the *open-set* scenario in order to make a comparison of the performance results obtained when the two different settings are considered. The classification accuracy values show that the system exhibits the same behaviour in both scenarios, but the global accuracy is higher is the *close-set* case. This is an expected result, as the performance of the *close-set* system proved higher also in the preliminary single observation analysis. This is explained by the fact that, in the *open-set* case, the number of speakers is increased with respect to the *closed-set* scenario and so is the complexity of the SVM, thus leading to lower performance. In order to make an intuitive comparison, the figures of all the tests related to the *open-set* scenario have been presented next to the respective *close-set* case: Figure 2.15b, 2.16b, 2.17b, 2.18b and 2.19b show the accuracy matrices related to applying the *AVG*, *S2D*, *PCL*, *WCL* and *FCAT* algorithms to the *open-set* case, respectively. Even in this scenario, the classification accuracy results show that in various cases the performance of the system can be improved by considering the contribution of multiple receivers in heterogeneous conditions, as reported in Table 2.5. As above,

Accuracy	*AVG*	*S2D*	*PCL*	*WCL*	*FCAT*
above	4.2%	13.7%	3.8%	0%	11.4%
between	86.2%	80.4%	75.6%	98.7%	85.3%
below	9.6%	4.6%	0.9%	0%	3.3%
no output	0%	1.3%	19.6%	1.3%	0%

TABLE 2.5: *Comparison between multiple and single observation values in open-set.*

Device	*snr [dB]*	*dist [m]*	Accuracy (α)					
			single obs.	*AVG*	*S2D*	*PCL*	*WCL*	*FCAT*
MD_1	15	1	58%	65%	66%	66%	63%	61%
MD_2	30	2	63%					

TABLE 2.6: *Classification accuracy (α) of the proposed techniques in a open-set scenario.*

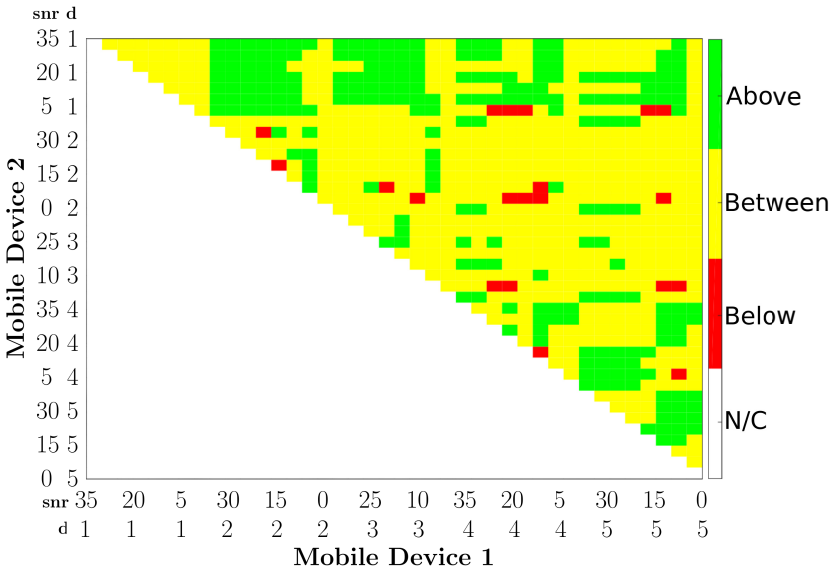


FIGURE 2.20: *Distribution of classification accuracy for FCAT in closed-set.*

exact values of the accuracy for a representative *out-of-set* case of study are reported in Table 2.6. In Figure 2.20 it is possible to visualize the distribution of the classification accuracy values related to the results shown in

Figure 2.19b, obtained by applying the *FCAT* multi-observation method in *closed-set* scenario. The corresponding matrix is symmetric, therefore values lying under the main diagonal (labelled as *N/C* values in the figure) have been omitted for the sake of readability. Analyzing this distribution, we can argue that the multiple observation approach proves advantageous in most of the cases where one of the devices lies in favourable conditions ($d \leq 1$ [m]). Unfortunately, when both MDs are farther from each other, the multiple-device approach does not introduce any performance gain. Finally, when both receiving nodes experience challenging conditions ($d \geq 4$ [m]) the contributions of the different receivers merge in a constructive way, thus leading to improved accuracy in speaker classification.

2.4 Speaker Recognition for Smart In-Vehicle Applications

In the latest days, the number of connected vehicles has increased rapidly. The Internet of Things (*IoT*) framework and the diffusion of related enabling technologies, such as Device-to-Device (*D2D*) and Vehicle-to-Vehicle (*V2V*) communications, cloud and fog computing and big data analysis have strongly improved the feasibility of connecting and communicating through a large number of mobile nodes, often in non-ideal environmental conditions [42, 43]. In the IoT vision, even the conventional web services are enriched with physical world services by penetrating smart things into the existing web, which leads to the concept of the so called Web of Things (*WoT*) [44, 45]. The emerging D2D framework offers a combination of ad-hoc and centralized communication mechanisms, thus bringing the gap between two previously disjoint domains [46, 47]. Also, the great improvements to cellular

infrastructure, with the design of LTE-Advanced and the emerging 5G networks, are a stepping stone to new opportunities and new challenges in terms of ubiquitous connectivity in the framework of smart vehicles and smart mobility [48, 49].

In this framework, great effort is put in the development of the future smart and connected cars, devised to embed the latest technologies in order to provide driving functionality, user comfort and customized on-board entertainments. One of the fundamental features of these technologies is to offer ubiquitous connectivity, in order to be able to communicate among different devices (*i.e.* on-board computer, smartphones, smartwatches, navigators, environmental sensors), but also towards the Internet. Indeed, many services and applications often rely on cloud services, which offer the possibility to exploit remote capabilities, both in terms of computation and storage.

Another important aspect to take into account is that the pervasiveness of technology and connectivity in vehicular environments can be also very sensitive in terms of security issues. This is especially true if we consider the safety and privacy of people on-board. To overcome these issues, researchers proposed different approaches and techniques, trying to limit the vulnerabilities of smart mobile systems [50, 51].

Multimedia hands-free applications and wearable technologies are a key feature for this type of applications [52, 53], and they have experienced a large commercial growth lately, drove by the fact that the concept of smart-vehicles and smart-spaces in general is becoming more and more popular every day.

The importance of audio speech processing in this framework is demonstrated by its employment in a wide variety of commercial applications. This growing interest in speech and speaker recognition is witnessed by the

widespread use of security services, such as speaker verification and authentication procedures. Other common applications of speech processing techniques lie in the range of accessibility solutions: the most remarkable examples of this kind are the speech-to-text and text-to-speech functionalities, especially useful in the framework of in-vehicle hands-free applications. Next-generation human-vehicle interfaces will likely incorporate biometric person recognition, using speech, video, images, and analog driver behavior signals to provide more efficient and safer vehicle operation, as well as pervasive and secure in-vehicle communication [54]. Practical cases could be for example a driver giving commands to the car on-board computer, passengers using customized on-board entertainment during a trip, or even driver monitoring and profiling applications (see Figure 2.21). Understanding audio context, such as the number of speakers in a conversation, the identity of the current speaker, the mood of the users involved, represents an important tool that can be extremely useful in several realistic scenarios.

Yet, technical and deployment issues limit the performances of these systems in real-world settings under adverse conditions. Furthermore, as the quality of audio signals is deeply influenced by the environmental conditions, an exhaustive study of the performances of the most common speech processing techniques in variable noise conditions and at different source-receiver distances is required. The in-vehicle environment is indeed highly subject to noise sources, both related to engine or traffic sounds, and to people chatting inside the car.

Many literature works address the issue of speech processing in challenging environmental conditions, proposing noise robust audio processing techniques, able to provide good performances even if noise is corrupting the audio signal. To the author's knowledge however, none of them addresses

FIGURE 2.21: *Examples of context-aware applications in a smart vehicle environment.*

the problem of distant, noisy speaker recognition in an exhaustive way.

The main idea is to think of our work inside this framework, having in mind the future possible integration of the proposed system into available on-board technologies. This integration offers the possibility to take advantage of the ubiquitous connectivity provided by the framework of connected cars, which would allow our proposed system to comprise for example a remote server used to keep the speaker models or to perform the required computations. To the purpose of enhancing the Speaker Recognition (SR) performances in challenging environmental conditions, in this thesis we propose a noise and distance robust speaker identification algorithm, which embeds a smart pre-processing method employing Voice Activity Detection (*VAD*), suitable to be employed in the framework of smart and connected vehicles. We also report a comprehensive performance analysis of the most

common speech audio processing approaches in different scenarios. In particular, we analyze the speaker recognition accuracy as the source-receiver distance varies and also when additional noise of different types is introduced in the environment. An exhaustive comparison of the performances of the traditional and the proposed speaker recognition algorithms is then presented.

2.4.1 Related Works

In literature, many works are aimed at recognizing the identity of people on-board vehicles, but their main focus is on the driver. These solutions usually exploit information about the driving style and behavior, in order to understand the identity of the current driver. Some interesting works proposing this type of solution are [55, 56]. The basic idea is to analyze the steering, accelerating, breaking, and even distance-keeping signals in order to infer who is driving the vehicle. Among others, some authors propose to identify the driver by analyzing the characteristics of his/her sitting posture [57]. However, these solutions are not feasible for our purposes, since they have two important drawbacks: they require a significant amount of time to collect driving data from the car, and they focus only on the identity of the driver, not considering other passengers on-board. From the industrial point of view, more and more car manufacturer are producing new vehicles providing smart functionality. There are two main issues addressed by smart and connected car solutions: the first is user-centric and related to safety and driver monitoring, while the second is typical of companies managing large vehicle fleets. When it comes to safety, the available solutions aim at understanding the state of the driver, by monitoring parameters such as driving style, gaze, eye blink patterns, and lane keeping [58, 59]. Solutions

related to fleet management usually leverage smart tags and readers, which allow recognizing the driver with the same principle of an employee card or badge [60, 61, 62]. More complex vehicle authentication systems rely on biometric sensors, implementing solutions such as fingerprint recognition [63].

Signal processing solutions for in-vehicle scenarios are a widely analyzed topic in literature, since the importance of this type of systems is growing in the latest period. For example, an overview of multimodal in-vehicle person recognition technologies is reported in [54]. Also, [64] proposes a feature extraction approach based on Minimum Variance Distortionless Response (MVDR) to perform reliable speech recognition for in-vehicle applications. Another interesting solution is described in [65], where authors present a GMM/SVM-supervector system for age and gender recognition, in order to provide flexible in-car dialog taking into account the specific needs and preferences of users.

The type of environment we consider in this thesis is very peculiar, since it is strongly affected by noise (think, for example, of a conversation in a moving car or train). For this reason it is important to assess the performance of speaker recognition systems in noisy environments. The effects of Additive White Gaussian noise on the short term cepstrum are highlighted in [66]. Other works propose interesting enhancement techniques [32, 33] in order to take into account the fact that in real scenarios speech spectrum is affected differently in different bandwidths. Other approaches might be segmentation and filtering techniques, proposed by [34, 35] to increase robustness in real-world scenarios. Some works suggest to employ the so called *multi-style* or *multi-conditional* training approach, which consists in using voice samples produced with different styles or corrupted by different types of noise during the training phase, in order to improve the classification performance

when noisy speech utterances are considered for testing [67]. Most of these techniques, however, require some prior knowledge about the characteristics of the noise corrupting the audio samples. Also, the concept of far-field speaker recognition is not new to the literature: interesting investigations are reported in [68], which proposes reverberation compensation and feature warping as solutions to mitigate the effects of mismatched training-testing conditions.

2.4.2 Performance Analysis

The general architecture of the in-vehicle speaker recognition system is conceived as described in Chapter 2.1. For this specific application, the SR functionality is implemented in an smartphone application running on Android mobile platforms and devised to be employed for in-vehicle applications, both using smartphone platforms and on-board technology.

In order to simulate in-vehicle environmental conditions, the performance evaluation presented in this thesis is carried out within the MATLAB computing environment on a personal computer running Linux OS. We considered different types of environmental noise influencing the system, which are typical of an in-vehicle scenario [32, 36, 69]. In particular, we tested the proposed algorithm in presence of Additive White Gaussian noise, which is comparable to the background sound of a motor engine, Flicker noise, typical of electronic devices and Speech-Shaped noise, that can be caused by the chatting of other passengers on-board. The audio database used for all tests consists in a set of audio signals produced by different speakers (both male and female) in Italian language. For the *closed-set* scenario, we considered a total of 64 audio files, 16 for each of 4 speakers (half male and half female) belonging to the *a-priori* known speaker set. For what concerns the *open-set*

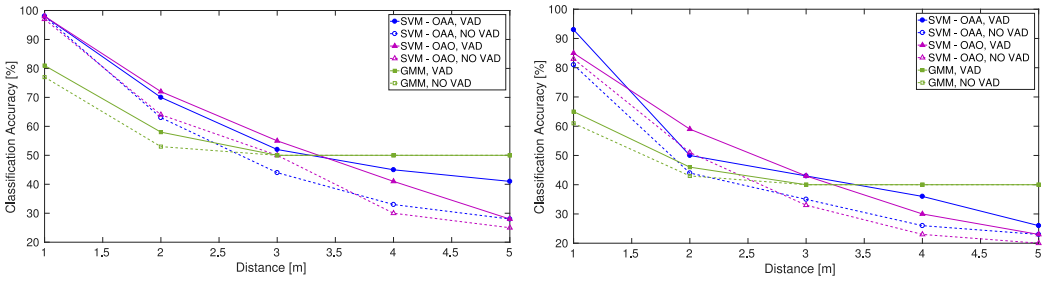
scenario, we added to the previously mentioned files, 16 audio files belonging to unknown speakers (*e.g.*, an unusual passenger of the vehicle), with the same gender partitioning of the training set. For the sake of replicability, the audio speech database is made available at [70]. Also, the system is conceived such that the audio signal experiences an attenuation which depends on the distance d from the source.

In the simulations we considered the speaker (source) placed at an increasing distance d from the microphone (receiver). In particular, we ran simulations with distance $d = \{1, 2, 3, 4, 5\}[m]$. This distance is compatible with the size of many different vehicles, from small mini-cars to larger vans and motorhomes. The set-up is conceived in order to simulate a scenario with a number of people inside a vehicle, such as a car, train, bus, or plane, and lying at different distances from the SR system. We started by performing all possible tests in clean speech conditions. Then, in order to consider a realistic scenario, we also analyzed the performances of the system when the signal is corrupted by the aforementioned in-vehicle noise types. Noise sources are supposed at the distance of $1[m]$ from the receiver and exhibiting a root mean square value of 0.03.

We chose as performance parameter the classification accuracy α, defined as in Eq. 2.1.3.

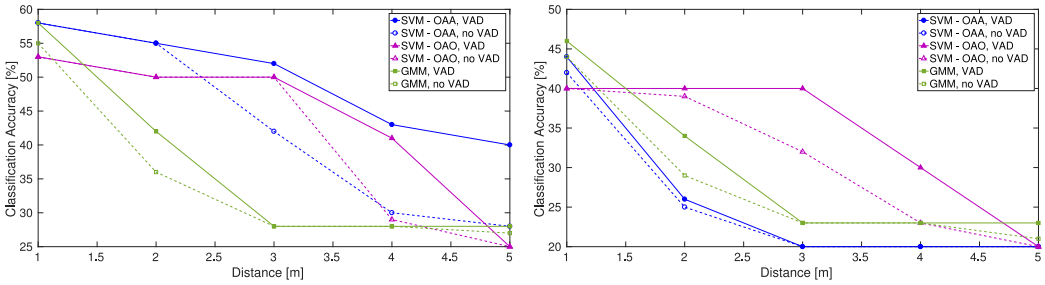
2.4.2.1 Classifiers Comparison

The first round of simulations focuses on the comparison of different machine learning algorithms for the purpose of speaker recognition in clean speech conditions. We built our proposed pre-processing scheme on top of 3 different classification approaches: SVM with One-Against-All approach, SVM with One-Against-One approach, and GMM. Figure 2.22a shows the performance



(A) (B)

FIGURE 2.22: *Classifier performance with and without SmartVAD in clean speech conditions for closed-set (2.22a) and open-set (2.22b).*



(A) (B)

FIGURE 2.23: *Classifier performance with and without SmartVAD in speech-shaped noise for closed-set (2.23a) and open-set (2.23b).*

obtained by employing the aforementioned classifiers for *closed-set* speaker identification when the speaker lies at different distances from the audio receiver. The results show that both SVM approaches perform well when the speech source is near the receiver, whereas the GMM classifier works better than SVM at a farther distance (over 3.5 meters). In all considered cases, the proposed smart voice activity detector is able to improve the speaker recognition accuracy.

The same consideration is valid for the *open-set* scenario, where the global behavior of the algorithms is similar to the *closed-set* one, but the accuracy levels prove slightly lower. This effect, reported in Figure 2.22b is explained by the fact that the identification task is harder, since unknown speakers are

added to the classification tests.

An analysis of the performances of the different classifiers in noisy conditions was also conducted. The behavior of the different approaches is similar when considering different types of noise. For the sake of brevity we only report the results related to the speech-shaped noise, as it is the most common type of noise that can corrupt a speaker recognition system. The obtained accuracy values are depicted in Figure 2.23.

We can see that the most performing classifier in noisy conditions is the Support Vector Machine in One-Against-All approach, followed by the SVM in *OAO* configuration. The latter, however achieves lower values of accuracy with respect to the former, when the distance between speaker and receiver exceeds 4 meters. The GMM method can obtain comparable results when source and receiver are at 1 meter of distance, but the performance degrades rapidly as the voice signal experiences attenuation.

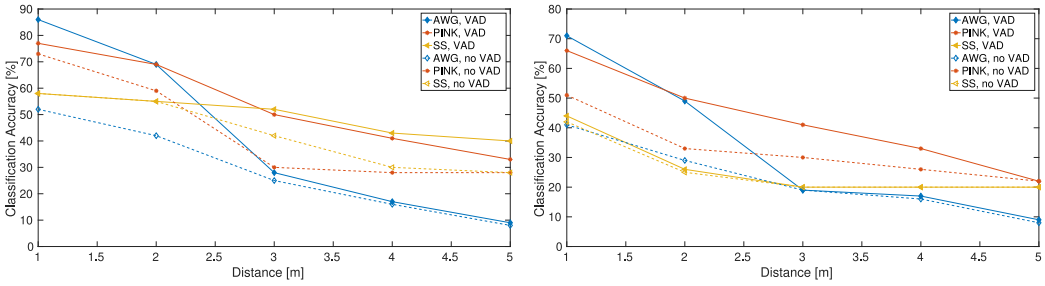
2.4.2.2 Audio Processing Time Analysis

In the following we present a performance analysis on the processing time required by our system to recognize the current speaker. The proposed solution is designed to be employed on-board vehicles, in order to provide customized services and personalized applications, such as on board entertainment and comforts. Thus, even if in-vehicle applications often require real-time performance, since our system is not intended for driving or safety purposes, there are no strict requirements on the processing time. Nonetheless, applications should provide a smooth and transparent human-computer interaction, so as to offer high quality user experience. For this reason, we show that our proposed solution is able to obtain good results in terms of the processing time required for speaker recognition. Table 2.7 shows the mean and standard

deviation of the execution time required by the system to perform speaker recognition. We computed the required execution time by considering the difference of two *timestamps*, the first obtained at the beginning of the audio processing phase, and the second at the end, when the system outputs the recognized speaker.

Statistics	**Classifier**			
[ms]	*SVM-OAA*	*SVM-OAO*	*GMM*	*Average*
Mean	728	800	805	778
Standard Dev.	128.5	148.3	116.6	131

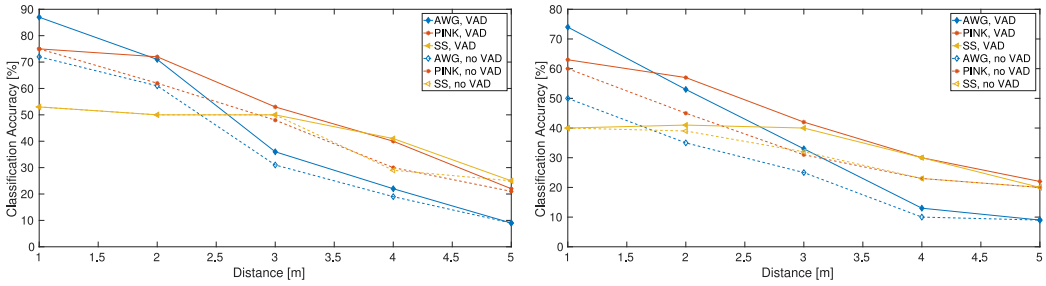
TABLE 2.7: *Average processing time required by the proposed speaker recognition system.*



(A) (B)

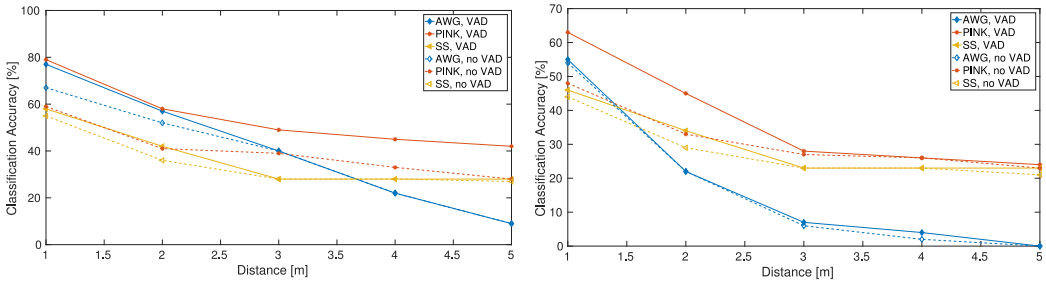
FIGURE 2.24: *SVM-OAA performance with and without VAD in different noisy conditions for Closed-Set (2.24a) and Open-Set (2.24b).*

The average processing time employed for audio processing and decision making is 728 $[ms]$ when considering the SVM-OAA classifier, which appears the fastest method. The slower approach is the GMM one, with an average execution time of 805 $[ms]$. This amount of time is considered suitable for target applications such as customized entertainment and personalized on-board services.



(A) (B)

FIGURE 2.25: *SVM-OAO speaker recognition performance with and without VAD in different noisy conditions for the Closed-Set (2.25a) and Open-Set (2.25b) scenarios.*



(A) (B)

FIGURE 2.26: *GMM speaker recognition performances with and without VAD in different noisy conditions for the Closed-Set (2.26a) and Open-Set (2.26b) scenarios.*

2.4.2.3 In-Vehicle Noise Effects on Accuracy

In this round of simulations, we try to understand how our proposed voice activity detection algorithm is able to mitigate the effects of different types of noise on the speech signal in a simulated in-vehicle environment. In order to consider a realistic scenario, we evaluate the performance of our system in the presence of environmental noise of different types. In particular, so as to reproduce car-like conditions, we tested the proposed algorithm in presence of Additive White Gaussian (*AWG*) noise, which reproduces the typical sound of a motor engine, Flicker noise (*PINK*), usually produced by electronic de-

vices and Speech-Shaped noise (*SS*), which introduces an interference caused by the chatting of other passengers on-board. All tests also consider the behavior of the algorithm as the speaker lies at increasing distance from the audio receiver.

Figure 2.24a shows the effects of the three types of noise on a speaker recognition test with *SVM-OAA* classifier in *closed-set* scenario. As one can note, the performance of the system when the signal is affected by white Gaussian noise becomes worse as the distance increases over 2 meters. Moreover, the results prove that the use of our proposed *SmartVAD* approach is able to effectively reduce the corrupting effects of noise on the voice signal.

Figure 2.24b shows the behaviour of the speaker identification algorithm in *open-set* scenario. It is evident that in this case the speech-shaped noise (similar to an interfering voice) has deeper effects on the global classification performance as the scenario includes also speech utterances that do not belong to known speaker voices. However, the pre-processing phase is able to enhance the accuracy of the system in most of the cases.

Another round of simulations was run when employing the SVM classifier configured with a *OAO* approach. In Figure 2.25 the results concerning the Closed-Set and Open-Set scenario are shown. This plots confirm the behaviour exhibited by the corresponding curves of the SVM algorithm in *OAA* configuration, where *AWG* noise effects are stronger when the distance increases.

Finally, system performance was tested when employing a Gaussian Mixture Model classifier. In Figure 2.26a, it is possible to analyze the results obtained in *closed-set* scenario. In this case, the GMM classifier performs slightly worse that the SVM approach when the speaker lies near the receiver. Also in this case the use of the *SmartVAD* proves fundamental to

boost the system accuracy, especially when the signal is corrupted by flicker noise.

The classification results obtained in the *open-set* scenario are reported in Figure 2.26b. It is evident that in this case the white Gaussian noise strongly corrupts the recognition performance, the more as the distance increases. However our *SmartVAD* scheme is able to enhance the accuracy level of the system when the speech signal is affected by pink noise and lies within 3 meters from the receiver.

Our detailed analysis proves that, almost in all cases, the proposed preprocessing VAD algorithm is able to enhance the speaker recognition performance of the system. The main contribution of the *SmartVAD*, as described in details in Section 2.1.1, consists in identifying the audio frames that contain a voiced signal, with respect to those which are silent or noisy. The audio frames which are identified as not suitable for the speaker identification task are then excluded from the classification process. An analysis of the percentage of discarded frames is reported in Figure 2.27. As one can see, the highest percentage of corrupted frames corresponds to additive white Gaussian noise. Also flicker noise contributes to a strong frame corruption, slightly increasing with distance.

2.4.2.4 Real-life Evaluation

In order to analyze the behavior of our proposed system in a realistic scenario, we tested our algorithm not only with artificial additive noise, but also in the presence of noise generated by real-life vehicles. In particular, we ran performance evaluation tests using as corrupting noise real-life audio recordings of different types of engine [71, 72]. In order to run our tests, we extracted four different audio files of the same length, and we tested the

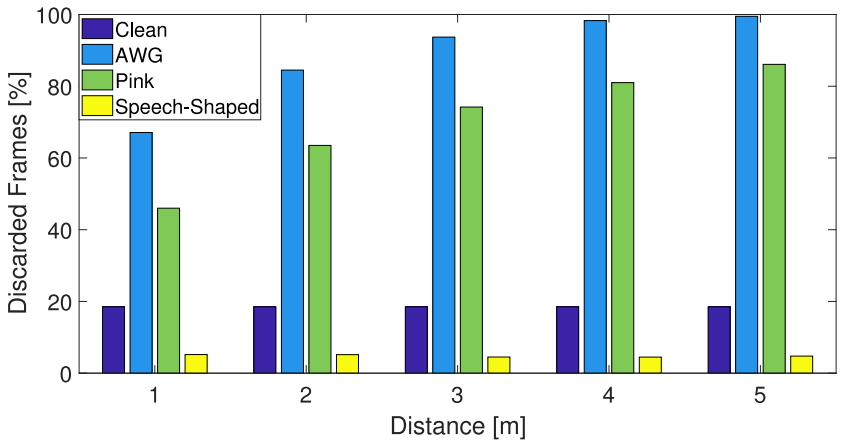


FIGURE 2.27: *Percentage of audio frames discarded by SmartVAD in different noisy conditions.*

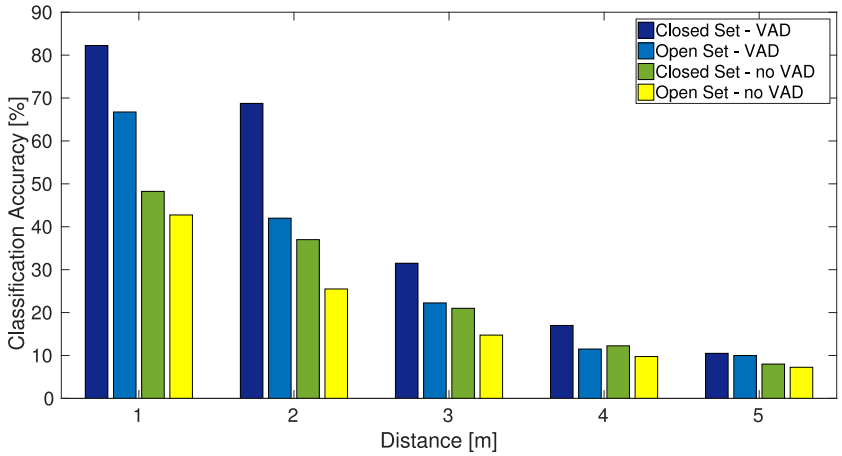


FIGURE 2.28: *Average speaker recognition accuracy in real-life evaluation testbed.*

effects of these types of noise on the dataset, both in the *closed-set* and *open-set* scenarios. Moreover, we compared the results obtained when applying our proposed *SmartVAD* algorithm with respect to those obtained with no signal pre-processing.

In Figure 2.28 we report the accuracy in the considered scenarios as an average of the values obtained with the different tests. The results show the recognition performance as the distance of the speaker from the receiving microphone increases from 1 to 5 meters. The performance analysis shows

that the behavior of the system is quite similar to that obtained when the audio files are corrupted with additive white Gaussian noise. As expected, we can notice a detriment of the performance as the distance increases. It is important to notice that the results related to the *closed-set* scenario exhibit higher accuracy with respect to the *open-set* case, thus confirming the trend of the previous simulation tests. In detail, the speaker recognition accuracy can reach 83% if we apply *SmartVAD* pre-processing in a *closed-set* framework. Moreover, the reported evaluation confirms that, even in the case of real-life noise, the use the proposed *SmartVAD* approach is effective at improving the percentage of correct recognition.

2.5 Conclusions on Audio Awareness

In this Chapter we analyzed the concept of audio awareness in the Internet of Things framework. In particular, we tackled the problem of speaker recognition in challenging environmental conditions and with different approaches. In Section 2.1 we proposed a robust speaker identification algorithm, which embeds a smart pre-processing method for Voice Activity Detection and we empirically investigated the optimal configuration and frame selection criterion in order to obtain a solid and effective voiced/unvoiced/silence audio speech detection. In Section 2.2 we tackled the problem of recognizing the identity of a speaker both in *closed-* and *open-set*, when the audio acquisition is performed at different distances and when different types of environmental noise are considered. We also presented a detailed analysis of the performances of common speaker identification systems, compared to those obtained with our novel proposed *SmartVAD* speaker recognition approach in various challenging environmental conditions. The results prove that our

solution is able to improve the accuracy of a traditional speaker recognition system in terms of correct classification, even if we consider distant speech signal acquisition and different types of noisy environments. In Section 2.3 a multiple-observation approach for audio speaker recognition on mobile devices has been proposed. We presented different fusion and clustering algorithms in order to exploit signals coming from multiple receivers to the purpose of enhancing the speaker classification performance with respect to a single device case. Numerical results comparing the accuracy obtained by applying the different methods confirms that in a significant percentage of cases the multiple-receiver approach can achieve higher classification accuracy with respect to those obtained when a single device is employed. Section 2.4 presents a detailed analysis of the performances of our proposed speaker identification system, especially designed for the typical challenging conditions that can be experienced inside an in-vehicle environment. The results prove that our solution is able to boost the accuracy of a traditional speaker recognition system in terms of correct classification, even if we consider a typical mobile vehicular framework, characterized by distant speech signal acquisition and different types of noise sources.

Chapter 3

Wireless Awareness for Fingerprint-based Drone Detection

One of the main aspects concerning the Internet of Things is related to the future Smart City. In the latest years, many researchers, technology companies and car manufacturers have been working together to the purpose of developing ad-hoc devices and technology specifically designed to offer smart services in the framework of the future connected city.

In this scenario, smart vehicles are one of the keywords that drive the research community, especially if we consider the frameworks of remote piloting or autonomous driving. For what concerns the former, nowadays an increasing number of smart devices that can be piloted remotely, the so called *drones*, can be easily found off the shelf. The fashion of wireless drones is spreading, and consequently the number of Unmanned Aerial Vehicles (UAV) is increasing every day. Unfortunately, the misuse of these devices can represent a major threat for the society, and therefore we need proper methods

to monitor and contain this potentially dangerous emerging technology.

Multimedia context awareness is a key concept that can be used to tackle this delicate issue, finding intelligent ways to manage and regulate the presence of drones in our cities.

Generally, remotely piloted vehicles rely on the device-controller communication for movements and actions. In particular, UAVs use WiFi or Radio Frequency communications to send and receive control, navigation, and video data. The large majority of commercial drones use the WiFi 802.11 protocol to exchange data with their controller, leveraging the fact the most users already own a smartphone that can be used to easily pilot their brand new drone through the provided FPV functionality.

For this reason, in this thesis we decided to explore this particular framework, and we studied and developed a smart system able to detect the presence of an unauthorized drone by exploiting statistical fingerprint analysis on WiFi traffic flows. Our proposed technique allows identifying the presence of drones in the nearby area, and it proved to be robust to possible stealth attacks performed by malicious users.

In the following, we will first introduce the problem of drone surveillance and survey the main existing techniques and the related literature. Next, we will describe in detail our novel proposed drone detection method and we will discuss the results obtained in experimental validation tests carried out in different real-life scenarios.

3.1 Introduction to Drone Surveillance

Enabled by the recent technological advances, and the great availability of affordable devices on the market, UAVs have received growing interest world-

wide thanks to their numerous applications in many different fields, such as logistics, surveillance, photography, agriculture, communications, public services, and so on [73]. In addition, commercial drones can usually be piloted through the WiFi connection of personal smartphones and tablets, thus increasing their user-friendliness and easiness of use. At the same time, the emerging Internet of Things framework and the related enabling technologies, such as the $5G$ standard, Device-to-Device networks, cloud and fog computing, have strongly increased the pervasiveness of connectivity and communications [74, 75]. Moreover, the ubiquitous use of smart devices and objects, together with the latest trends in deploying drone swarms, triggers the need for more efficient and smarter network solutions, able to cope with dynamic and evolving topologies [76, 77]. As a consequence, the need for surveillance methods able to detect the presence of unauthorized drones naturally arises, driving the research community towards the design of efficient monitoring systems able to integrate machine learning methods and data mining techniques in this evolving environment.

Drones will likely play a major role in the future smart cities, for example by serving as mobile access points for broadband wireless access and Location Based Social Networks (LBSN), delivering goods and merchandise, and maintaining surveillance and security [78, 79, 10]. Unfortunately, this future picture poses security, privacy, and public safety concerns that need to be discussed. For this reason, designing a technology able to monitor, identify and keep away drones from restricted areas becomes a major target for the research community.

The main idea behind drone detection consists in having a device specifically set up to patrol a sensitive area. This device is usually called *guardian*, and its core consists in the drone surveillance technology, which is used to

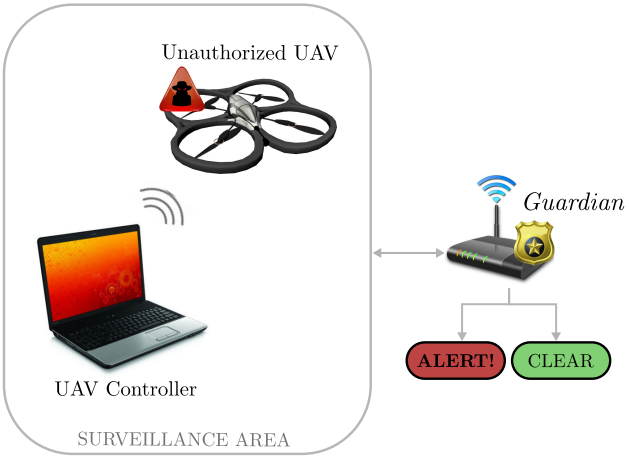


FIGURE 3.1: *General scheme of a drone surveillance system.*

detect the presence of possible unauthorized devices in the nearby area. This concept is illustrated in Figure 3.1. The surveillance system can be implemented on different hardware devices, strongly depending on the considered application and deployment scenario. For example, drone sensing systems could be implemented on different types of devices and located as depicted in Figure 3.2:

- **Fixed ground system**, continuously monitoring the nearby area (*e.g.*, a security camera).
- **Fixed aerial system**, always scanning the surrounding flying zone (*e.g.*, a radar or sonar system)
- **Monitoring drone**, actively moving throughout the sensitive area in search of possible threats (*e.g.*, a security patrol).

Solutions suitable to be employed in such scenarios must comply to some basic characteristics:

- **lightness**: to be easily embedded on small devices, such as a small UAVs.

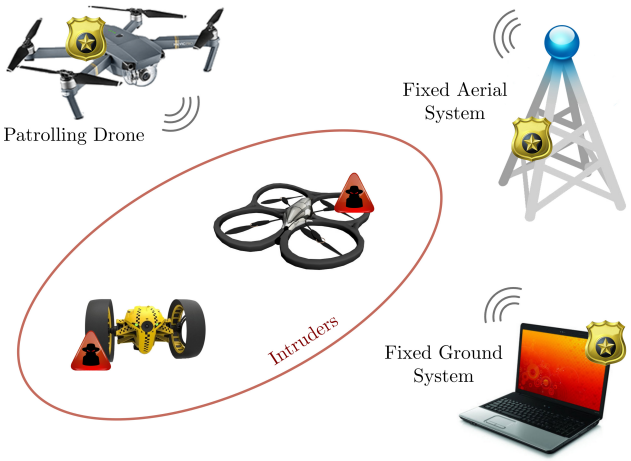


FIGURE 3.2: *Alternative practical implementations of a guardian entity.*

- **resiliency**: to be robust to background interference.
- **energy efficiency**: to provide long battery autonomy.

A brief survey of the main solutions proposed in the literature to identify nearby drones is provided in Section 3.2.

Moreover, a large part of low-cost miniature drones is usually controlled via D2D communication to directly route data traffic between close objects providing energy efficiency, high throughput, low delay, as well as spectrum efficiency [42]. A list of commercially available drones relying on WiFi communication is provided in Table 3.1.

Most of these drones include also the so called *First Person View (FPV)* mode, which allows the owner to pilot the UAV remotely by using the video stream of a real-time camera that provides the first-person perspective of the drone itself. This type of streaming, together with the control signals, has particular characteristics that allow UAV detection through traffic analysis.

Drone Model	Brand	Frequency [GHz]	Weight [g]	Price [$]
AR.Drone	Parrot	2.4	420	320
Bebop 2	Parrot	2.4	500	580
Disco FPV	Parrot	2.4	750	1390
Jumping Race	Parrot	2.4, 5	205	170
Mambo	Parrot	2.4	100	130
Swing	Parrot	2.4	295	150
Typhoon H	Yuneec	5	1950	1280
H501A X4 Air Pro	Hubsan	2.4	1300	230
H507A X4 Star Pro	Hubsan	2.4	162	99
X8SW	Syma	2.4	1500	155
X8SC	Syma	2.4	1500	150
X5UW	Syma	2.4	127	75
U818A Discovery	UdiRc	2.4	132	100
U29W	UdiRc	2.4	100	60
U845 Voyager	UdiRc	2.4	98.5	70
U36W	UdiRc	2.4	23	25
U28W	UdiRc	2.4	53	75
U31W	UdiRc	2.4	85	100

TABLE 3.1: *A list of off-the-shelf amateur WiFi drones.*

3.2 Related Works on Drone Detection

The reliable detection of drones is a challenging task due to the great diversity in the device characteristics and also to presence of many interfering objects in the environment. The literature offers a large number of different methods used to perform UAV detection. In the following, a general taxonomy of the main existing drone detection techniques is presented.

First of all, anti-drone systems can be divided into four main categories [80], depending on their final purpose, as depicted in Figure 3.3:

- **Warning Methods**, where the *guardian* device is employed to detect unauthorized drones and perform early warning on their presence [5]
- **Spoofing Methods**, where ad-hoc technologies are used in order to

send fake signals to hijack drone threats (*e.g.*, GPS spoofing [81]).

- **Jamming Methods**, where the control and navigation system is disturbed by a strong interference (*e.g.*, electromagnetic pulse) [10]
- **Neutralizing Methods**, where drones can be directly destroyed or captured, mainly in emergency situations or in highly sensitive areas.

Moreover, the latest proposed solutions are pushing towards the deployment of UAV detection systems on-board drones themselves, thus introducing the distinction between the concept of *Monitoring Drone* (MDr) and *Intruder Drone* (IDr) [82]. Unfortunately, some of the available techniques are not suitable to be implemented on-board UAVs, due to logistics constraints and/or possible interference.

Since the first step towards the neutralization of a threat is a robust and reliable detection, in this thesis we will focus on the *Warning Methods*.

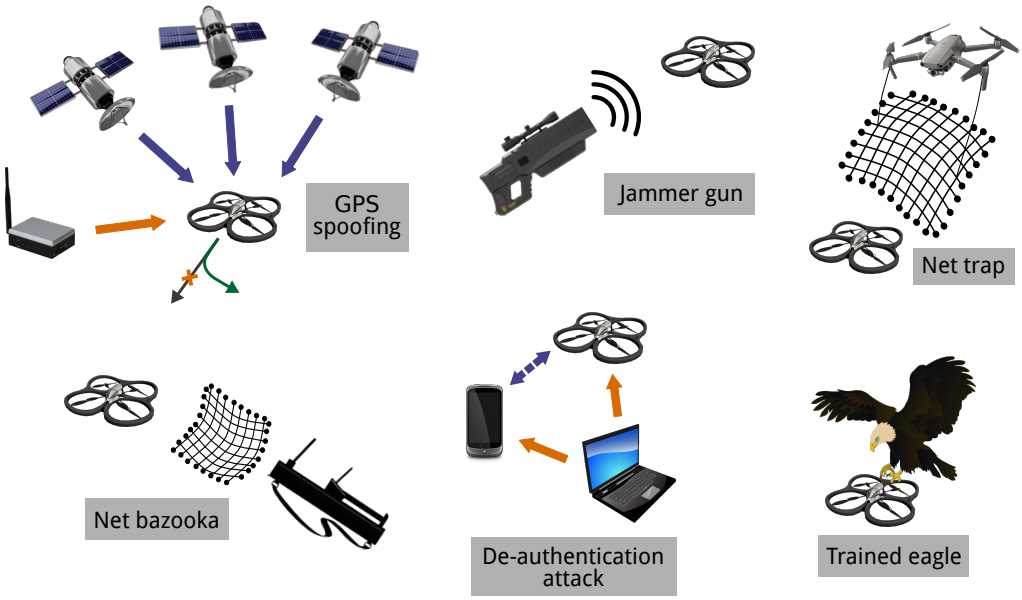


FIGURE 3.3: *Taxonomy of existing anti-drone surveillance systems [10].*

In the following, a brief survey of existing surveillance systems especially devised for the early detection of drones is presented.

3.2.1 Acoustic Detection

These approaches consist in using microphones in order to capture ambient sounds. Most microphones have a working range of about $25-30\,ft$. The rationale is that UAVs usually produce a typical hissing sound around $40\,kHz$, due to their brushless direct current motors. In [83] the authors propose an audio classification system for drone detection based on Hidden Markov Models (HMM) in order to perform phoneme analysis and identify drones by their emitted audio sounds. In [84], authors propose an acoustic sensor composed of a 120-element microphone array. An acoustic imaging algorithm determines in real-time the sound power level coming from all directions, using the phase of the sound signals. A tracking algorithm uses beamforming to follow and select the sound sources, that can be used to identify sound signatures in order to detect IDrs. Another technique is proposed in [85], where a different architecture consisting of a small tetrahedral microphone array is used for the same purposes. Unfortunately, the acoustic approach works well in quiet environments, but it usually fails in crowded urban areas or noisy conditions. Another drawback of sound-based detection is that audio sensors are not suitable to be mounted on-board a MDr, due to the interference produced by the monitoring UAV itself.

3.2.2 Video Detection

This type of detection employs camera sensors, both visual and thermal, in order to identify moving objects in the scene. Commercial cameras can reach an operative range of about $350\,ft$, which leads to a quite reasonable surveil-

lance area. This technique exploits features such as color, contour lines, shapes and edges to characterize a typical UAV object with respect to other items [86]. Also, motion detection algorithms can be used to analyze features across consecutive frames [87]. This can help in distinguishing objects with similar shapes, such as UAVs and birds, by their typical movements (*e.g.*, mechanical movements of drones *vs.* natural movements of birds). Cameras are also very sensitive to lighting conditions, and they can identify objects only when the target is in line of sight.

3.2.3 Thermal Detection

Thermal drone detection is based on the principle that drones equipped with motor engines employ methods such as turbo-gases as propulsion system. This type of engine produces hot gases from the exhaust, which results in heating up the drone [88]. An example of this kind of drones is represented by fixed-wing UAVs, which can be detected by thermal sensors due to their high temperature during the flight. Unfortunately, the large majority of commercial drones consists in electric quad-copters, which do not radiate enough heat to be detected by this method. The implementation costs of this technique are high compared to the identification rate, therefore this method is usually employed as an add-on to other UAV detection systems. An important drawback of this technique is that it cannot be mounted on-board a MDr, as the temperature of the drone itself would interfere with the measurements of the thermal sensors.

3.2.4 RADAR Detection

This method exploits the electromagnetic principle of backscattering for drone detection. The traditional RADAR approach relies on the fact that

Detection	Target Principle	On-board	Pros & Cons
Video	Object and motion detection	✓	Fails with other flying objects Range $< 350\,ft$ Needs line of sight
Audio	Hissing frequency $40\,kHz$	✗	Fails in noisy urban areas n/a on-board drones Range $25 - 30\,ft$
RADAR	Backscattering Doppler/micro-Doppler	✓	Fails with small quad-copters
Temperature	Engine gases	✗	Fails if drones have electric motors High cost and low reliability Range $< 350\,ft$
RF	Control/video over RF	✓	Difficult when altitude is high Depends on power/sensitivity Range $< 1400\,ft$
WiFi	Control/video over WiFi	✓	Available for WiFi drones only WiFi working range

TABLE 3.2: *Summary of existing drone detection methods.*

aircraft usually expose a large RADAR cross section (RCS), but most of commercial drones are electric quad-copters, with a rather small RCS [89]. A major drawback of this approach depends on the building materials, since plastic, having dielectric properties close to air, results in little reflection back to the transmitter. For this reason, the latest literature works employ modified types of RADAR sensors, which exploit the energy backscattered from propeller and rotors. On this topic, multi-static radars have been recently proposed to analyze the micro-Doppler signature of micro-drones, and to classify UAVs with different payload size [90]. Moreover, in [91] authors employ frequency-modulated continuous-wave (FMCW) radar to extract information about the drone's size, type, range, and radial velocity. Another interesting approach, presented in [92], uses multiple-input multiple-output (MIMO) radar to generate 256 virtual elements to detect and track drones. The system is used to decide whether the drone exists within a certain range or angular cell. The RADAR based technique can be implemented in a rather

portable format, therefore it can be deployed also on-board a MDr[93].

3.2.5 RF Detection

The Radio Frequency (RF) based detection methods exploit the fact that UAVs communicate with the ground controller using radio frequency signals. Drone communication protocols commonly employ the same frequency bands also used for WiFi transmissions, in particular in the 2.4 and $5\,GHz$ range. Moreover, drones equipped with cameras usually transmit a video stream to their control unit on the same wireless channel. In [94], morphological filtering in frequency domain is used to devise an algorithm for detecting UMTS, LTE, and drone communication signals in adverse environments. RF detection has a very long effective range, covering over $1400\,ft$. A minor issue of this technique is that the target detection rate strongly depends on the transmitter power and receiver sensitivity. Lately, new approaches have been proposed, which rely on Software Defined Radio (SDR). An interesting solution is presented in [95], where authors propose a distributed system to identify the appearance and the approximate position of unwelcome drones by integrating SDR transceivers and wireless acoustic sensors. A general overview on passive drone detection is provided in [96], where authors also develop a SDR-based portable universal software radio peripheral prototype for the detection of drones in different scenarios.

3.2.6 WiFi Detection

Most commercial drones are designed to be piloted through WiFi signals, to allow amateurs to use their own smartphone to control the drone. Also, these types of UAVs usually provide a First Person View video functionality, to stream the feed of their embedded camera directly to the display of the

smartphone. In the literature few contributions investigate on the idea of exploiting the WiFi signal to detect the presence of unauthorized drones. The main idea is to capture packet flows belonging to drone control and video transmissions, by using a wireless packet sniffer on the WiFi channel. This approach is sometimes embedded on special drones themselves to locate other types of nearby devices [97]. However, these techniques are usually based on *a-priori* knowledge of the remotely piloted vehicle, such as information about the vendor *Organizationally Unique Identifier (OUI)* used to identify the sender/receiver of specific packets [98]. In this framework, we provide an innovative approach based on the analysis of the statistical fingerprint of WiFi traffic in order to identify the presence of a drone. Differently from previous works, this method does not require a-priori knowledge of UAVs to successfully detect a drone. Starting from this solution, we discuss the vulnerabilities of the WiFi-based approach, and we propose and advanced method aimed at overcoming possible issues arising during surveillance. The motivations and the details of the detection method proposed in this thesis are described in the following.

3.3 WiFi Statistical Fingerprint Based Drone Detection System

The WiFi fingerprint based drone detection method aims at exploiting specific characteristics related to drone control, navigation and video packets. The final goal is to detect an unauthorized UAV flying in a given surveillance area.

In detail, WiFi traffic flows are captured by a network packet sniffer (*e.g.*, *Wireshark* [99]), which is installed on board a *guardian* device. All captured

packets are dumped and stored in a *pcap* file. The acquired data is then processed locally on-board the *guardian* entity or sent to a remote station for cloud processing in case of lack of resources.

The processing stage works as follows: we define a time window τ_w, such that the captured traffic is partitioned in small portions of τ_w seconds. This operation determines the time interval at which classification is performed, thus enabling real-time drone detection. Each portion of traffic is analyzed in order to extract specific features that allow to completely define the corresponding flow. The feature vector used for flow identification is hereby called *fingerprint*. The features used to characterize the data are mainly related to the duration and behavior of the traffic flows and to the distribution of the corresponding packets, which is a consolidated approach for traffic classification in literature [6].

In the following, we will describe the analytic quantities related to the considered flows, in order to better explain the choice of the features used to identify the traffic classes.

Let us define a generic flow f as any wireless stream of network packets, identified by the source and destination Medium Access Control (MAC) addresses, $\mathcal{M}_s$ and $\mathcal{M}_d$. We apply a mapping operator Ψ in order to associate MAC addresses to specific flows, such that:

$$\Psi(\mathcal{M}_s, \mathcal{M}_d) = k, \quad k \in [1, ..., K] \tag{3.1}$$

where K is the total number of flows captured in the considered time window. Therefore, the generic flow can be described as the quantity:

$$f(\mathcal{M}_s, \mathcal{M}_d) = f^k = \{p_1, \ldots, p_i, \ldots, p_{N^k}\}, \tag{3.2}$$

where p_i is the i-th packet belonging to the considered flow, and the quantities $\mathcal{M}_s$ and $\mathcal{M}_d$ refer to the source and destination MAC address, respectively.

The total number of packets belonging to the k-th flow is therefore given by the quantity N^k. Packets in which source and destination addresses are swapped, are considered as belonging to the same flow, thus defining the symmetry property of the mapping operator Ψ:

$$\Psi(\mathcal{M}_s, \mathcal{M}_d) = \Psi(\mathcal{M}_d, \mathcal{M}_s) \tag{3.3}$$

A packet p_i is further defined as a sequence of L^i bytes, such that:

$$p_i^k = [b_1, \ldots, b_j, \ldots, b_{L^i}], \tag{3.4}$$

where b_j is the j-th byte of the considered packet.

Consequently, the average packet length $\bar{L}^k$ for the k-th flow is computed as:

$$\bar{L}^k = \frac{1}{N^k} \sum_{i=1}^{N^k} L^i \tag{3.5}$$

The root mean square value Q of the packet length for the k-th flow is given by:

$$Q^k = \sqrt{\frac{1}{N^k} \sum_{i=1}^{N^k} |L^i|^2} \tag{3.6}$$

The *duration* D^k of the k-th flow is the time taken by the destination to receive the entire stream of N^k packets belonging to such flow. In particular, let us define t_i^k the generic time instant in which the i-th packet of the k-th

flow arrived at the guardian. The total duration of the flow is computed as:

$$D^k = \left(t^k_{N^k} - t^k_1\right), \tag{3.7}$$

where $t^k_{N^k}$ and t^k_1 are the arrival time of the last and first packets of the flow, respectively. Another important feature is related to packets inter-arrival time. We can compute the average inter-arrival time $\Delta\bar{t}^k$ of packets belonging to the k-th flow as:

$$\Delta\bar{t}^k = \frac{1}{N^k - 1} \sum_{i=1}^{N^k-1} \left(t^k_{i+1} - t^k_i\right) \tag{3.8}$$

The root mean square value of the packets inter-arrival time is therefore given by:

$$\Delta Q^k = \sqrt{\frac{1}{N^k - 1} \sum_{i=1}^{N^k-1} |\Delta t_i|^2}, \tag{3.9}$$

where $\Delta t_i = (t^k_{i+1} - t^k_i)$ is the time difference between the arrival time of two generic consecutive packets pertaining to the flow f^k.

In order to compute the number of embedded packets, we need to denote as $\mathcal{M}^i_{tx}$ and $\mathcal{M}^i_{rx}$ the MAC address of the transmitter and receiver devices of the i-th packet, respectively. We define embedded packets those packets whose source and destination addresses coincide with the transmitter and receiver addresses. Mathematically:

$$\mathcal{E}^k = \{p^k_i \in f^k : \ \mathcal{M}^i_{tx} \equiv \mathcal{M}_s \wedge \mathcal{M}^i_{rx} \equiv \mathcal{M}_d\} \tag{3.10}$$

The number of embedded packets belonging to the k-th flow is therefore the cardinality of the above defined set, $|\mathcal{E}^k|$.

Let us define ϕ^i_h the mathematical quantity associated to the wireless DS

status flags of the i-th packet, with $h \in [1, 4]$. This is a binary quantity that can assume only the values 0 or 1. The number of packets z_h^k in each traffic flow that exhibit a specific DS status can be therefore computed as:

$$z_h^k = \sum_{i=1}^{N_k} \phi_h^i \,,\; h \in [1, 4] \tag{3.11}$$

Consequently, the last feature used to characterize the k-th traffic flow can be defined as the set Z^k containing the amount of packets exhibiting the possible DS status flags:

$$Z^k = \{z_h^k :\; h \in [1, 4]\} \tag{3.12}$$

The fingerprint describing the generic flow is finally defined as the feature vector:

$$\mathcal{F}(k) = \{N^k, D^k, \bar{L}^k, Q^k, \Delta\bar{t}^k, \Delta Q^k, |\mathcal{E}^k|, Z^k\} \tag{3.13}$$

After the feature extraction phase, the actual identification of drone-specific patterns is performed by means of consolidated machine learning techniques that allow recognizing different classes.

In Section 3.4, we will provide a detailed analysis of the performance of the proposed WiFi-based detection system, and in Section 3.5, we will discuss the inherent vulnerabilities of this approach, introducing an advanced method especially devised to overcome the possible issues related to a traditional WiFi-based detection system.

3.4 Traditional Approach to WiFi-based Drone Detection

In this Section, we present a drone detection method based on several features extracted from WiFi network traffic flows related to drone control and video transmissions. The final goal is to detect the presence of an unauthorized UAV in a given surveillance area. The system is conceived as an implementation of the design presented in Section 3.3. A sketch of the employed architecture is illustrated in Figure 3.4.

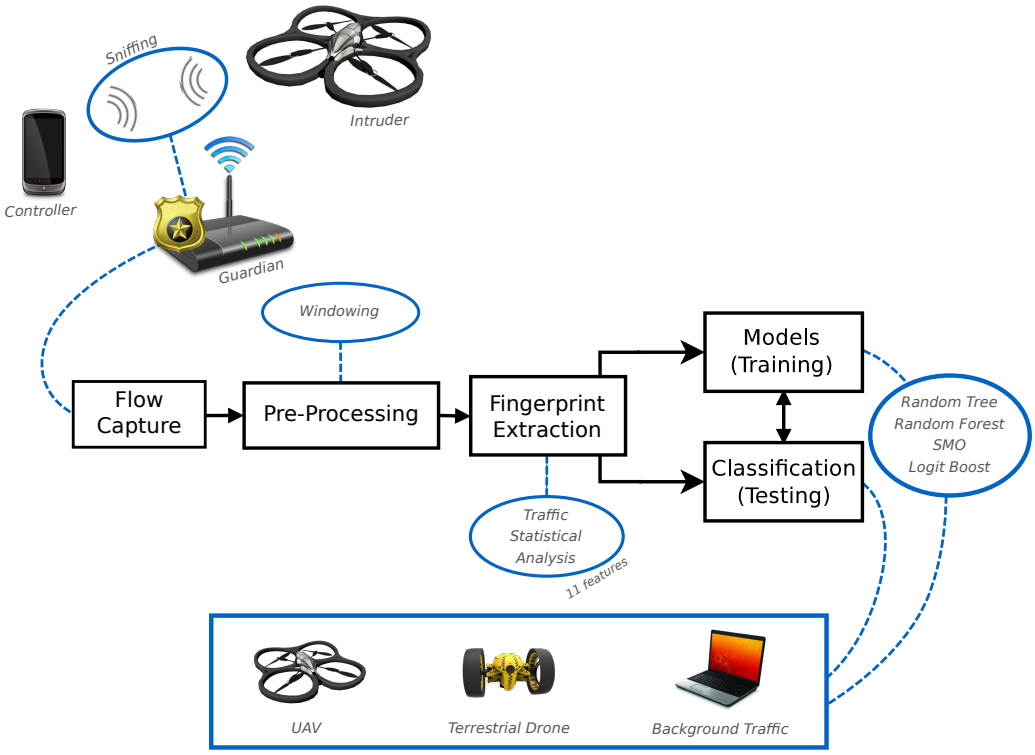


FIGURE 3.4: *Overview of the traditional WiFi fingerprint-based drone detection scheme.*

In the following, we will describe the results obtained when evaluating our proposed method in experimental tests. In all the considered scenarios, the

surveillance entity (the *guardian*) we employed is a laptop running Mac OS, with the Wireshark sniffer in monitor mode installed on-board. The laptop includes our proposed WiFi-based drone detection algorithm, which uses the aforementioned statistical analysis on wireless traffic to identify the presence of unauthorized drones in the surrounding area.

The devices employed during our tests are: a *Parrot AR.Drone 2.0* UAV, an older model of flying *Parrot AR.Drone*, and a *Parrot Jumping Race Max* terrestrial drone. The UAVs are two typical amateur *FPV* electric quadcopter weighting about $400\,g$, equipped with a high definition $720p$ camera and a flying range of $50\,m$, which can be controlled over WiFi at $2.4\,GHz$ by a remote *Android* or *iOS* device. The second type of device is a two-wheel terrestrial drone, also capable of jumping, which can be piloted via WiFi at both 2.4 and $5\,GHz$ by a remote mobile device. This drone weights about $200\,g$ and is equipped with a VGA camera for *FPV* drive control. The devices used to remotely pilot the drones are two *Samsung Galaxy Tab 2* tablets for the UAVs and a *Xiaomi Redmi 4 Pro* smartphone for what concerns the terrestrial drone control.

All the experimental tests were conducted in an office environment, which includes also other devices sharing the same WiFi channel. Examples of such ongoing traffic flows are video streaming, audio streaming, video conference calls, web browsing, FTP downloading and so on.

For what concerns the recognition phase, we employed the well-known software tool *Weka* [100], which provides an implementation of many efficient machine learning classifiers. In this evaluation, we consider different types of classifier to establish the best performance. In particular, the employed classifiers are:

- **Random Tree:** a decision tree classifier that considers randomly cho-

sen attributes at each node. It performs no pruning.

- **Random Forest:** a classifier that consists of a forest of Random Trees.
- **SMO:** a Sequential Minimal Optimization algorithm for training a Support Vector Machine with polynomial kernel.
- **Logit Boost:** a machine learning approach that performs Additive Logistic Regression.

The model related to each classifier has been constructed by using a large set of flow data obtained by applying a $\tau_w = 5\,s$ partitioning *window* to a long WiFi recording. The capture was related to a scenario where one UAV, one terrestrial drone and a number of background TCP/IP traffic sources were present. It is worth noting that these sources generated both audio/ video streams and variable length data exchanges. Each model was trained to recognize three different device classes: *UAV*, *Terrestrial Drone (TD)*, and *Background Traffic (BG)*. The obtained models were used to classify the traffic flows, related to the scenarios described below.

Sc-1: Single Flying Drone The scenario depicted in Figure 3.5a includes a single flying *AR.Drone 2.0* controlled by an Android tablet. The traffic produced by this UAV shares the same channel used by flows related to *YouTube* and live web radio streaming, together with a background WLAN traffic coming from working devices in the nearby area.

Sc-2: Flying and Terrestrial Drones This experimental setup, shown in Figure 3.5b is quite similar to the previous one. However, in this scenario a terrestrial *Jumping Race Max* controlled by an Android smartphone is also moving in the area. As above, the testing environment is disturbed by

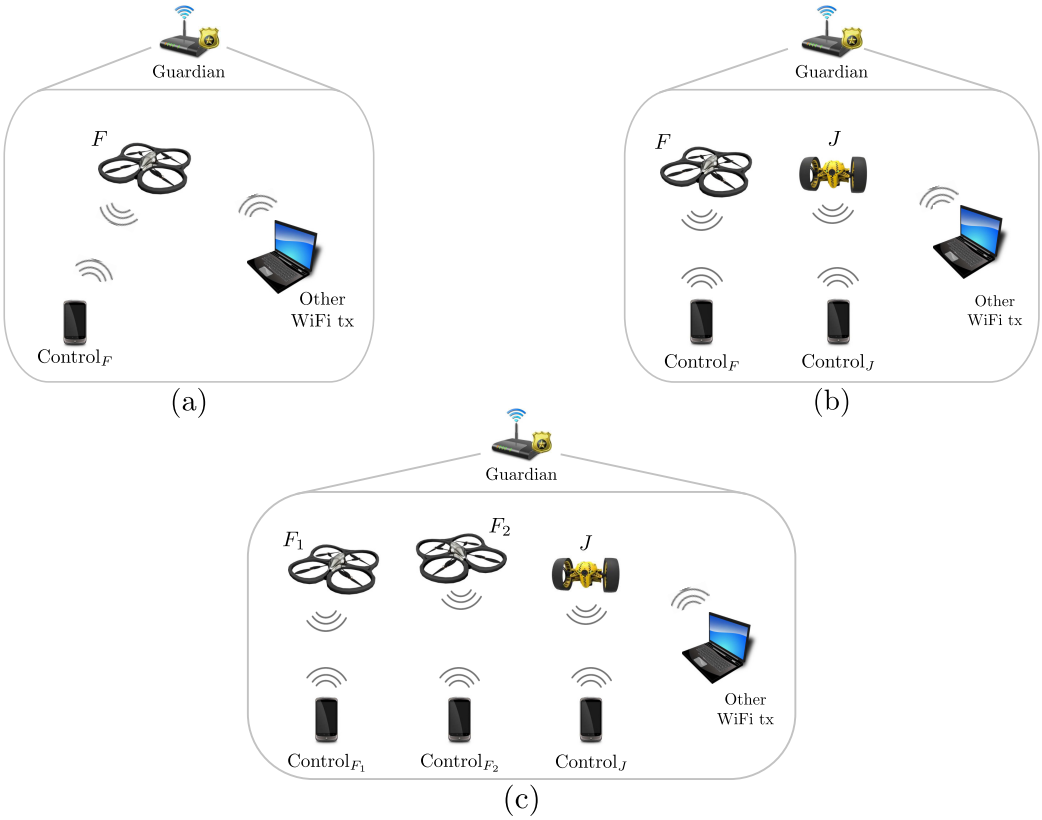


FIGURE 3.5: *Scheme of the Single Flying Drone (a), Flying and Terrestrial Drones (b), and Multiple Flying and Terrestrial Drones (c) scenarios.*

different types of transmissions and multimedia streaming traffic produced by other devices working nearby.

Sc-3: Multiple Flying and Terrestrial Drones This scenario consists of two flying *AR.Drones*, plus a terrestrial *Jumping Race Max* violating the same no-access area. Background interfering flows coming from other devices are also considered in this experimental test. The corresponding scheme is illustrated in Figure 3.5c.

In all the considered scenarios the UAV, terrestrial drone, and WLAN communications took place on the same WiFi channel at $2.4 GHz$ band. However, the detection algorithm could be easily modified so that the *guardian*

can cyclically scan on the different available WiFi channels. The percentage of drone flows is about 1% of the total number of flows. Table 3.3 reports the results achieved by employing different machine learning approaches. As performance parameters, we consider the following quntities:

- True Positive Rate relative to the *UAV* class, defined as the number of correctly classified *UAV* flows (*UAV-TPR*);
- Precision on the *UAV* class, defined as the ratio between true positives and the sum of true and false positive *UAV* instances (*UAV-PPV*);
- Percentage of UAV instances misclassified as terrestrial drones (*UAV2TD*);
- Percentage of UAV traffic misclassified as normal background flows (*UAV2BG*).

Classifier	Scenario	UAV-TPR	UAV-PPV	UAV2TD	UAV2BG
Random Tree	*Training*	95.5	100	4.5	0
	Sc-1	89.9	100	10.1	0
	Sc-2	93.3	100	4.4	2.2
	Sc-3	87	100	11.3	1.4
Random Forest	*Training*	95.5	100	4.5	0
	Sc-1	87.4	100	12.4	0
	Sc-2	93.3	100	4.4	2.2
	Sc-3	84.5	100	14.1	1.4
SMO	*Training*	95.5	100	0.0	4.5
	Sc-1	63	69.4	33.6	3.4
	Sc-2	91.1	100	2.2	6.7
	Sc-3	60	100	59.2	5.6
Logit Boost	*Training*	95.5	100	4.5	0
	Sc-1	82.4	74.8	17.6	0
	Sc-2	93.3	100	4.4	2.2
	Sc-3	83	100	15.5	1.4

TABLE 3.3: *Comparison of performance results (%) related to UAV detection using different classifiers.*

Almost all classification approaches allow achieving comparable results for what concerns the precision level of the flying drone class; only the *SMO* classifier shows slightly lower performance. Indeed, the *UAV* class precision is always above 70% for all the considered scenarios. It should be noted that, in case of error, all the considered algorithms tend to misclassify UAVs as terrestrial drones (*UAV2TD*). In this sense, the class relative to terrestrial drones may be considered as a warning of a possible presence of undesired UAVs. Rarely the classifiers confuse the traffic produced by a UAV as background traffic generated by other authorized devices transmitting in the considered WLAN. The only exception is given by the *SMO* classifier, which is more prone to misclassifying drones as background traffic (*UAV2BG*) than the other machine learning approaches. Our experiments highlight that the highest recognition performance can be obtained when employing decision tree learning methods. We also analyzed the case where the acquired WiFi capture is partitioned with a *window* of 10 s. The obtained results are quite similar to the ones related to the case shown in Table 3.3; however, the true positive rate is slightly lower and this is due to the smaller number of flows available for describing the UAV traffic, thus reducing the accuracy of the trained model. Another related aspect to consider is that classifiers are significantly affected by the length of the WiFi capture used as training set: a long WiFi capture strongly tends to stabilize the behavior of all the considered classifiers.

In the next Section we will discuss the vulnerabilities of this traditional approach, and we will propose an advanced method devised to be robust to possible malicious attacks.

3.5 Advanced Techniques for Drone Surveillance: *Blind Detection*

In this Section, we discuss the vulnerabilities of the classic WiFi detection technique with respect to malicious attacks consisting in hacking the drone firmware configuration. In particular, we investigate the impact of video streaming flows on correct drone identification. To this aim, we run experimental tests consisting of simulated malicious attacks. We call these occurrences *stealth* attacks, since they are performed by using a drone with disabled video stream, in order to deceive the surveillance system. To tackle this problem, we devise proper strategies to make the WiFi-based detection method more efficient and robust to possible threats. In particular, we propose an advanced WiFi-based approach, which we call *Blind Detection*. This technique accounts for the particular characteristics of the WiFi control traffic produced by drones and their remote controllers, minimizing the influence of video packets in the recognition process. In order to classify flows, we apply data mining algorithms so as to recognize different classes of traffic and detect the presence of unauthorized devices. Finally, we report a performance analysis of the proposed techniques, carried out by performing an extensive experimental campaign.

3.5.1 Vulnerabilities to Stealth Attacks

The traditional WiFi statistical fingerprint based UAV detection technique described in detail in Section 3.4 is devised for the detection of amateur drones that can be piloted via WiFi. Users commonly employ their own smartphones or tablets as controllers for such UAVs, thus relying on the transmission of the FPV video streaming for piloting. This is an important

aspect that can have a deep impact on the detection of this type of drones. Indeed, a major concern that can arise is related to the fact that the video streaming data might be a predominant component of the drone wireless traffic flow. As a consequence, the recognition task might be strongly influenced by the statistical characteristics of video streaming packets. An additional concern is related to the security issues of the *Parrot AR.Drone*, which is reported to be especially subject to major vulnerabilities. Indeed, authors in [101] report that the operating system of the drone can be hacked without particular effort, since no security protection is employed. Given these considerations, the video streaming functionality of a drone could be shut down on purpose, so providing to an attacker the chance to bypass WiFi-based detection systems.

For this reason, we propose a thorough study to assess the actual contribution of the video streaming component on the correct drone recognition, by testing the traditional detection approach with hacked *Parrot AR.Drones*. Then, we propose an improved version of the considered algorithm, which makes the drone detection system more robust and reliable in case of *stealth* attacks.

3.5.2 Our approach: Blind Detection

As discussed above, a possible attack that could be implemented in order to deceive a drone surveillance system consists in disabling the FPV video streaming functionality. For the sake of clarity, we first define three possible configurations for a drone device, which produce the related testing scenarios:

- *Classic configuration*: the UAV runs the original firmware, and its FPV video streaming is enabled.

TABLE 3.4: *Description of the proposed detection techniques.*

Algorithm	**Approach**	**# Models**	**Drone Video Streaming**
Traditional	Classic	1	enabled
Pure Blind	Blind	1	disabled
Joint	Blind	1	enabled + disabled
Separate	Blind	2	enabled, disabled

- *Stealth configuration*: the drone runs a custom firmware, and its FPV video streaming is disabled.
- *Mixed configuration*: the firmware of the UAV has been hacked, and the video streaming can be turned on or off at will using the controller.

We refer to the configuration of a drone with disabled video streaming as *Stealth* mode. This particular setting has an effect on the statistical distribution of WiFi traffic flows pertaining to drone transmissions, which can cause the surveillance system to wrongly classify part of the captured WiFi packets. For this reason, we propose an improved version of the drone detection algorithm, that tries to address the aforementioned problem, taking into account the presence of hacked UAVs. In detail, the proposed technique is based on a specific approach for the training phase of the detection system, involving a large database of WiFi flows belonging to drones in *Stealth* configuration, together with other traditional wireless background flows. Thanks to this phase, the surveillance system is able to learn the statistical distribution of packets belonging to UAV control and navigation traffic, minimizing the influence of FPV video streaming packets on classification, and increasing the recognition performance with respect to the traditional method. We define this advanced detection approach *Blind Detection*, since it does not need video information to correctly detect the presence of drones. Starting

from this concept, we propose three different variants of *Blind* UAV detection (Table 3.4):

- *Pure Blind Detection (PD)*: this approach uses for system training only flows belonging to video-disabled drones. We employ our custom piloting software to shut down the FPV video functionality of the drone, and we capture WiFi traffic produced when the UAV is controlled by line-of-sight.

- *Joint Detection (JD)*: in this case, both UAV video-enabled flows and video-disabled data are used as dataset in the training phase. Again, we employ the customized piloting software to arbitrarily turn on or off the video functionality.

- *Separate Detection (SD)*: as in the previous case, both video-enabled and video-disabled drone transmissions are considered. The main difference is that two different classification models are built, one trained using only video-enabled data flows, and the other employing only video-disabled flows. Finally, a drone is detected if at least one of the two surveillance models identifies a flow classified as UAV.

In Section 3.5.3, we describe in detail the implementation of the aforementioned approaches, and we report a performance analysis in order to assess the effectiveness of the proposed solutions.

3.5.3 Experimental Tests

The general scheme of the proposed drone detection system is reported in Figure 3.6. In all the considered scenarios, the surveillance entity (*i.e.*, the *guardian*) we employed is a laptop running Linux OS, with the WiFi interface configured in monitor and promiscuous mode. The laptop runs the

WiFi-based drone detection algorithm on-board, and uses the *Wireshark* [99] packet sniffer and statistical analysis on wireless traffic to identify the presence of unauthorized drones in the surrounding area.

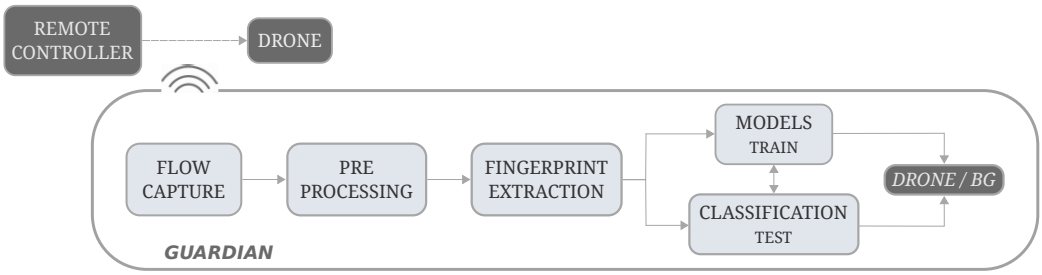


FIGURE 3.6: *Overview of the advanced WiFi fingerprint-based drone detection scheme.*

The UAV employed during our tests is a *Parrot AR.Drone 2.0*, a commercial electric quad-copter, equipped with a high definition $720p$ camera for *FPV* video streaming and a flying range of $50\,m$. The drone can be controlled over WiFi at $2.4\,GHz$ by using a remote device such as a smartphone, tablet, or PC. In order to pilot the drone, we used another laptop computer running Linux OS, with our custom version of piloting software installed on-board. All experimental tests were carried out in a working environment, which includes also other office devices sharing the same WiFi channel. Different types of background traffic is considered, such as video streaming, audio streaming, video conference calls, web browsing, LAN printers, and FTP downloading. For what concerns the detection phase, we employed the well-known data mining tool *Weka* [100], which provides an implementation of many consolidated machine learning classifiers. In this evaluation, we compared the performances of different classifiers, whose discussion is reported in Section 3.5.5. The experimental tests were performed by employing a custom web-based *AR.Drone* controller developed starting from the open-source code of the *ardrone-webflight* software [102]. This custom firmware allowed

controlling the UAV through the browser of a laptop computer running Linux OS. In this customized version of the software, we included the possibility to arbitrarily turn on or off the FPV video functionality with a simple command. This setup allows defining three possible scenarios related to the drone configuration: the *Classic* scenario, where the UAVs involved are in *Classic* configuration, the *Stealth* scenario, where the drone is in *Stealth* configuration, and the *Mixed* scenario, in which the drone is alternatively turned into *Classic* or *Stealth* configuration randomly.

We carried out different experiments, in order to test various conditions in terms of background traffic flows traveling on the WiFi channel that might interfere in the recognition process. The drone detection system is a binary classifier that allows distinguishing two possible classes of traffic flow:

- *Drone*: positive, traffic belonging to drone transmissions.
- *BG*: negative, background traffic.

We consider as performance parameters the following consolidated statistical quantities:

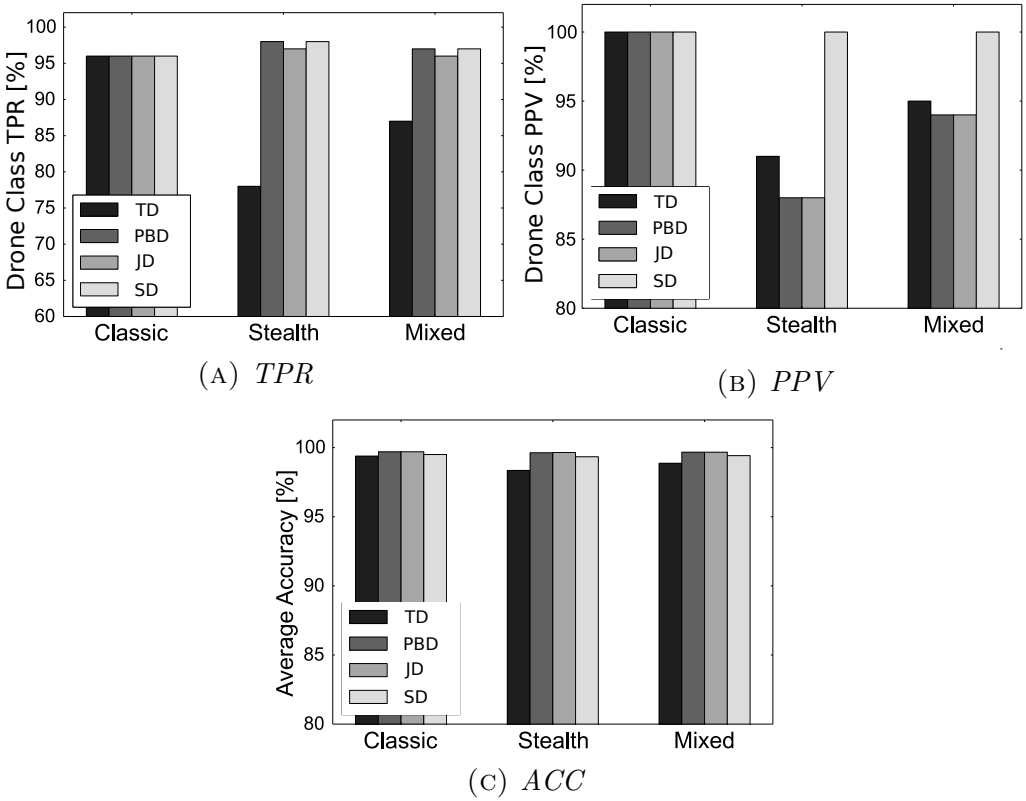
- True Positive Rate (TPR): it is the number of *Drone* instances classified as *Drone*.
- Positive Predicted Value (PPV): also called Precision, it is the number of true positive instances over the total number of instances classified as positive.
- Accuracy (ACC): it is the number of correctly classified instances over the total number of tested instances.

We ran all experimental tests evaluating the aforementioned performance

metrics, but for the sake of brevity in the following we report only the most relevant results as average.

3.5.4 Simulating a stealth attack

The first experimental step consists in assessing the vulnerabilities of the traditional approach in the case of a *stealth* attack. To this aim, we performed a practical experimental test using a *Parrot AR.Drone 2.0* by intentionally turning off the FPV video streaming functionality. In this way, we simulated a restricted area violation performed by a malicious user with a UAV configured in *Stealth* mode. The interdicted area was patrolled by a *guardian* implementing the different WiFi drone detection techniques.



(A) *TPR* (B) *PPV*

(C) *ACC*

FIGURE 3.7: *Comparison of performance for Traditional and Blind techniques in different scenarios.*

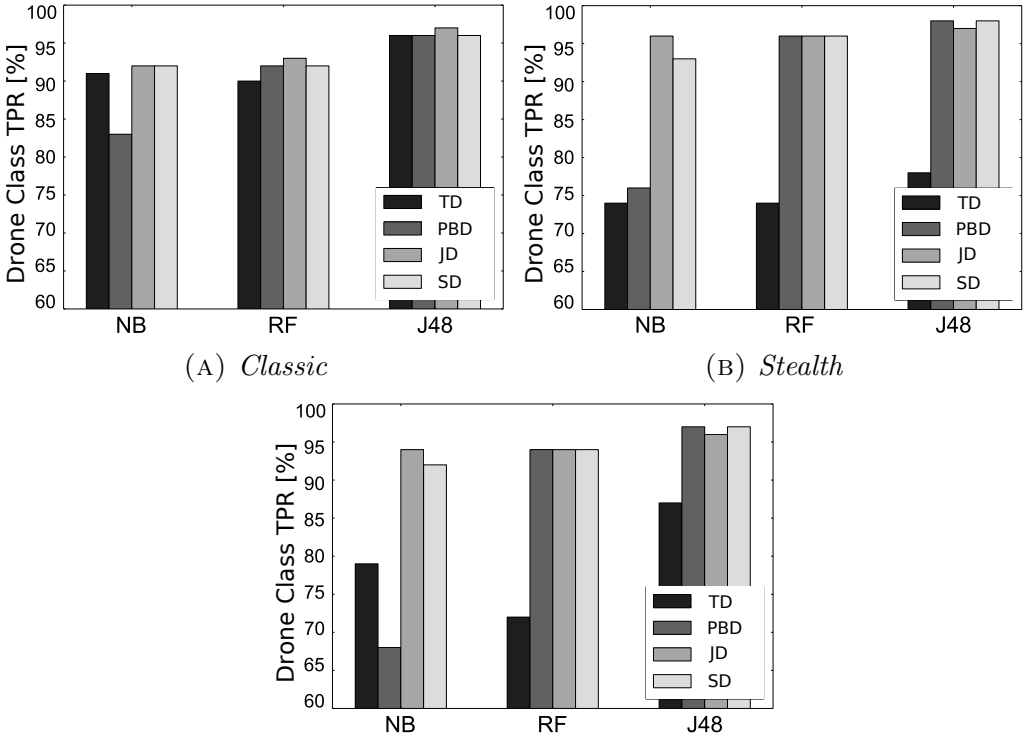
In the following we report the average values of the results obtained in the different test runs, with varying background conditions.

The bar plots highlight that, considering a *Stealth* scenario, when we employ the *Traditional* drone detection method (*TD*) the performance of the system is highly reduced with respect to the *Classic* one (see Figure 3.7a). Indeed, the surveillance system is prone to failing in the identification of a significant number of drone traffic flows, which increases the risk of missing the detection of unauthorized devices. Numerically, the Drone *TPR* achieved by the *Traditional* method reaches 96% in a *Classic* video-enabled scenario, but it experiences a drop to 78% in the case of *Stealth* UAV configuration. This has also effect on the drone class precision: the system scores 100% (*i.e.*, all *Drone* instances correctly classified) in the *Classic* scenario for all the detection approaches, but it decreases by about 10% in the *Stealth* case. Only the *Separate* approach is able to maintain a *PPV* of 100% in all scenarios (Figure 3.7b). At the same time, all the considered algorithms allow achieving the same performance in terms of average classification accuracy. Finally, the performed experiments show that the *Traditional* detection method is not able to efficiently recognize the presence of a flying drone in the area if a *Stealth* attack is performed. On the contrary, the proposed approaches can always achieve a high rate of success, thus assuring efficient surveillance against unauthorized drones.

3.5.5 Performance comparison with different classifiers

A comparison of the performance of the considered drone detection algorithms obtained with different machine learning classifiers is reported in Figure 3.8. The bar plots report the Drone True Positive Rate in the *Classic* (3.8a), *Stealth* (3.8b), and *Mixed* (3.8c) scenarios, respectively. We compared

the performance of the drone detection system when employing Naive Bayes (NB), Random Forest (RF) and J48 classifiers.



(A) *Classic* (B) *Stealth*

(C) *Mixed*

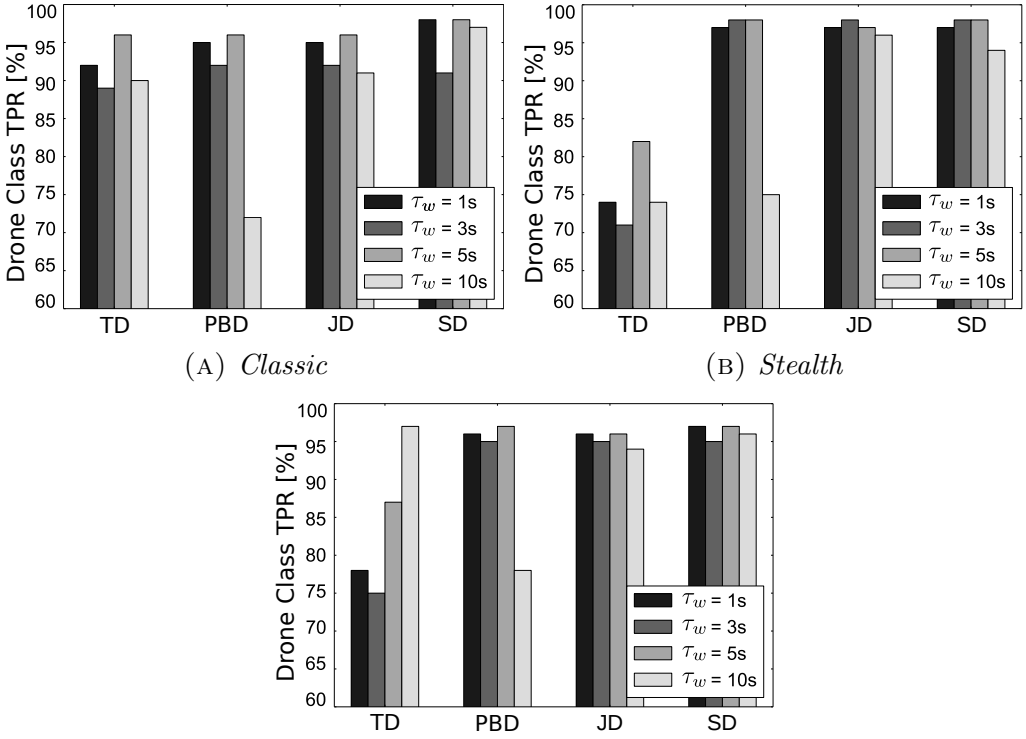
FIGURE 3.8: *Comparison of TPR obtained using different classifiers in the considered scenarios.*

The results show that the NB classifier is not able to provide satisfactory results in terms of correct recognition, even if the use of the proposed JD and SD algorithms allows improving the performance of this classifier almost in every scenario. The results highlight that the RF classifier has some weaknesses for what concerns the *Traditional* method, mostly in the *Stealth* and *Mixed* scenarios (see Figure 3.8b and 3.8c). Finally, one can notice that the proposed *Blind* detection techniques are able to provide very good and promising results independently of the classifier in all the considered scenarios. Indeed, the *Joint* and *Separate* techniques can always achieve

TPR values higher than 92%.

3.5.6 Effects of time window size

Another important parameter to take into account is the time window τ_w used to perform real time detection. The choice of a sub-optimal duration might deeply influence the efficiency of the algorithms. Therefore we decided to run a set of practical tests to identify the best time window to use for detection.



(A) *Classic* (B) *Stealth*

(C) *Mixed*

FIGURE 3.9: *Comparison of TPR obtained using different detection time windows τ_w in the considered scenarios.*

Figure 3.9 shows a comparison of the Drone *TPR* obtained when running the algorithms with a time window of $\tau_w = \{1, 3, 5, 10\}[s]$. It is evident that a $10s$ time window has a great impact on the performance of the system, prob-

ably waiting too much time before classification. However, the *Joint* and *Separate* methods allow achieving good results, since they can compensate the weaknesses concerning the sole *Traditional* or *Pure Blind* approaches. Also, when $\tau_w < 5s$ the *Traditional* detection technique cannot provide satisfactory results when considering the *Stealth* and *Mixed* scenarios. On the other hand, the *Blind* approaches are able to achieve a high true positive rate and good recognition performance also in this case.

3.5.7 Effects of attribute selection

An interesting aspect to consider is the choice of the features used to compose the fingerprint. In all the previous test, we used as fingerprint the whole vector of features described in Eq. 3.13 of Section 3.3. In order to understand the contribution of the different features to the characterization of the traffic flow, we ran an attribute selection algorithm based on the Information Gain (IG) evaluator on the dataset. Consequently, we obtained a ranking of the considered features, and we decided to keep only the most informative 8 attributes out of 11. The outcome of this analysis allowed us to obtain a new feature vector $\mathcal{F}'(k)$, defined as:

$$\mathcal{F}'(k) = \{N^k, D^k, \bar{L}^k, Q^k, \Delta\bar{t}^k, \Delta Q^k, |\mathcal{E}^k|, Z'^k\}\,, \tag{3.14}$$

where Z'^k is a modified version of the set Z^k defined in Eq. 3.12 of Section 3.3, which is reduced to:

$$Z'^k = \{z_h^k :\ h = 3\} \tag{3.15}$$

We ran the experimental tests using as fingerprint the reduced feature vector $\mathcal{F}'(k)$, and we considered the same scenarios used for the complete

Feature Set	Precision [%]				Scenario
	TD	Pure Blind	Joint	Separate	
Complete	100	100	100	100	Classic
Reduced	100	94	91	100	
Complete	91	88	88	100	Stealth
Reduced	91	71	69	100	
Complete	95	94	94	100	Mixed
Reduced	95	82	80	100	

TABLE 3.5: *Comparison of PPV using the complete and the reduced feature set in the different scenarios.*

fingerprint tests. The analysis shows that the overall performance in terms of drone *TPR* and accuracy obtained after applying the attributed selection is almost the same achieved with the original set of features. At the same time, some differences can be noticed in the values related to the precision metric. In Table 3.5 a comparison of the PPV obtained using the fingerprint $\mathcal{F}^k$ (*Complete* feature set) with respect to $\mathcal{F}'^k$ (*Reduced* feature set) in the different scenarios is reported. It is easy to notice that there is a slight drop in the precision achieved by the *Pure Blind* and *Joint* methods with respect to the original setup. However the *Separate* technique is able to achieve 100% precision in all considered cases, also when the reduced fingerprint is employed.

3.6 Conclusions on Wireless Awareness

In this Chapter we studied important aspects related to wireless awareness in the framework of the emerging Internet of Things. In particular, we tackled the problem of detecting the presence of unauthorized drones in restricted

areas by exploiting statistical information about the WiFi traffic. First, in Section 3.2 we presented the main approaches available in the literature considering the problem of drone surveillance, by discussing the main advantages and drawbacks deriving from their employment.

In Section 3.3, we proposed a WiFi statistical fingerprint technique for UAV identification, which analyses the traffic flows available on the WiFi channel in order to recognize possible nearby threats. The usefulness of this method is motivated by the increasing amount of commercial amateur drones using WiFi as control and FPV video streaming protocol. We have tested the performance of our method in different scenarios, also in the presence of interfering wireless traffic due to nearby devices. The results presented in Section 3.4 show that our approach is able to efficiently detect the presence of unauthorized drones in all the considered conditions with an average precision greater than 96%.

Moreover, in Section 3.5 we discussed the main vulnerabilities that characterize the considered system, and evaluated the impact of video streaming flows on drone detection by performing experimental tests. To tackle this problem, we proposed an advanced WiFi-based approach called *Blind Detection*, making the surveillance system more robust and effective also in the case of possible stealth attacks. Finally, we tested the performance of our proposed approach with different drone configurations in typical attack scenarios. The results show that the proposed technique is able to improve the detection efficiency of the surveillance system with respect to the traditional approach, and suggest that our solution could be a promising technique in the framework of drone surveillance.

Chapter 4

Final Conclusions

In this thesis we studied and analyzed the role of multimedia context awareness in the Internet of Things framework, by applying signal processing and ambient intelligence techniques in order to devise novel and interesting solutions suitable for smart scenarios. The main aspects tackled by this research are related to audio and wireless awareness. In particular, this thesis focused on two main research topics: smart algorithms for speaker recognition in challenging mobile environments, and unauthorized drone detection based on WiFi statistical fingerprint analysis.

For what concerns audio awareness, in Chapter 2 we tackled the problem of speaker recognition in different environmental conditions and with different approaches. We proposed a speaker identification technique, embedding a smart pre-processing algorithm for Voice Activity Detection, and we investigated the most appropriate parameter configuration in order to obtain an efficient and robust VAD operation. We also proposed a solution to the problem of recognizing the identity of a speaker both in *closed-* and *open-set*, when the audio acquisition is performed at different distances and when different types of challenging environmental conditions are considered.

As further contribution, a multiple-observation approach for audio speaker recognition on mobile devices has been proposed. We presented different fusion and clustering algorithms able to exploit signals coming from multiple devices, so as to enhance the performance of a single receiver speaker recognition system. Finally, we presented a speaker identification system especially designed for the typical challenging conditions that can be experienced inside an in-vehicle environment. In all the considered cases, the results prove that our solution is able to boost the accuracy of a traditional speaker recognition system in terms of correct classification, even if we consider distant speech and challenging environmental conditions.

Concerning the study on wireless awareness, in Chapter 3 we tackled the problem of detecting the presence of unauthorized drones in restricted areas by exploiting statistical information about the WiFi traffic. First, we presented the main approaches available in the literature facing the problem of drone surveillance, and we discussed the main advantages and drawbacks deriving from using the different techniques. Also, we proposed a WiFi statistical fingerprint approach for UAV identification, which analyses the traffic flows available on the WiFi channel in order to recognize possible nearby threats. Moreover, we discussed the main vulnerabilities that characterize the considered system, and evaluated the impact of video streaming flows on drone detection by performing experimental tests. To tackle this problem, we proposed an advanced WiFi-based approach called *Blind Detection*, making the surveillance system more robust and effective also in the case of possible *stealth* attacks. Considering the different proposed techniques, we tested the performance of our methods in specific scenarios, also in the presence of background interfering traffic produced by nearby wireless devices. The results show that our approach is able to efficiently detect the presence

of unauthorized drones in all the considered conditions. Also, the advanced *Blind Detection* technique was tested with different drone configurations in typical attack scenarios, and results showed that the proposed technique is able to improve the detection efficiency of the surveillance system with respect to the traditional approach. These analysis suggests that our solution could be a promising technique in the framework of drone surveillance.

In conclusion, the analysis carried out in this thesis tackled important and hot topics in the framework of Internet of Things applications. Our study suggests that signal processing techniques, together with innovative multimedia context-aware solutions, can play a leading role in the framework of the future Smart Spaces and Smart City scenarios.

Bibliography

[1] I. Bisio, F. Lavagetto, C. Garibotto, and A. Sciarrone, "Speaker Recognition Exploiting D2D Communications Paradigm: Performance Evaluation of Multiple Observations Approaches," *Mobile Networks and Applications*, vol. 22, no. 6, pp. 1045–1057, Dec 2017. [Online]. Available: https://doi.org/10.1007/s11036-017-0876-z

[2] I. Bisio, C. Garibotto, A. Sciarrone, and F. Lavagetto, "Performance analysis of smart audio pre-processing for noise-robust text-independent speaker recognition," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Nov 2017, pp. 848–852.

[3] I. Bisio, C. Garibotto, F. Lavagetto, and A. Sciarrone, "Enhancing Speaker Recognition with Multiple Observations over Mobile Networks," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[4] I. Bisio, C. Garibotto, A. Grattarola, F. Lavagetto, and A. Sciarrone, "Smart and Robust Speaker Recognition for Context-Aware In-Vehicle Applications," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2018.

[5] I. Bisio, C. Garibotto, F. Lavagetto, A. Sciarrone, and S. Zappatore, "Unauthorized Amateur UAV Detection Based on WiFi Statistical Fingerprint Analysis," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 106–111, APRIL 2018.

[6] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys Tutorials*, vol. 10, no. 4, pp. 56–76, Fourth 2008.

[7] I. Bisio, C. Garibotto, F. Lavagetto, A. Sciarrone, and S. Zappatore, "Improving WiFi Statistical Fingerprint based Detection Techiques Against UAV Stealth Attacks," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, to appear.

[8] ——, "Blind Detection: Advanced Techniques for WiFi-based Drone Surveillance," *IEEE Transactions on Vehicular Technology*, 2018, under review.

[9] R. L. Ackoff, "From data to wisdom," *Journal of applied systems analysis*, vol. 16, no. 1, pp. 3–9, 1989.

[10] I. Guvenc, F. Koohifar, S. Singh, M. L. Sichitiu, and D. Matolak, "Detection, tracking, and interdiction for amateur drones," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 75–81, APRIL 2018.

[11] I. Bisio, C. Garibotto, A. Grattarola, F. Lavagetto, and A. Sciarrone, "Exploiting Context-Aware Capabilities over the Internet of Things for Industry 4.0 Applications," *IEEE Network*, vol. 32, no. 3, pp. 101–107, May 2018.

[12] C. Dobre and F. Xhafa, *Pervasive Computing: Next Generation Platforms for Intelligent Data Collection*. Morgan Kaufmann, 2016.

[13] B. N. Schilit and M. M. Theimer, "Disseminating active map information to mobile hosts," *IEEE Network*, vol. 8, no. 5, pp. 22–32, Sept 1994.

[14] J. J. Bisgaard, M. Heise, and C. Steffensen, "How is context and context-awareness defined and applied? a survey of context-awareness," *Aalborg university*, 2004.

[15] H. Lieberman and T. Selker, "Out of context: Computer systems that adapt to, and learn from, context," *IBM systems journal*, vol. 39, no. 3.4, pp. 617–632, 2000.

[16] D. R. Morse, S. Armstrong, and A. K. Dey, "The what, who, where, when, why and how of context-awareness," in *CHI '00 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '00. ACM, 2000, pp. 371–371. [Online]. Available: http://doi.acm.org/10.1145/633292.633518

[17] J. H. L. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74–99, Nov 2015.

[18] I. Bisio, F. Lavagetto, M. Marchese, A. Sciarrone, C. Frá, and M. Valla, "Spectra: A speech processing platform as smartphone application," in *Communications (ICC), 2015 IEEE International Conference on*, June 2015, pp. 7030–7035.

[19] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE signal processing letters*, vol. 6, no. 1, pp. 1–3, 1999.

[20] E. A. E. Habib, "Geometric mean for negative and zero values," *International Journal of Research & Reviews in Applied Sciences*, vol. 11, no. 3, Jun 2012.

[21] I. Bisio, A. Delfino, F. Lavagetto, M. Marchese, and A. Sciarrone, "Gender-driven emotion recognition through speech signals for ambient intelligence applications," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 2, pp. 244–257, Dec 2013.

[22] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, Mar 2002.

[23] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[24] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 1, pp. 72–83, 1995.

[25] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, Nov 1996.

[26] A. Barghi and H. Bayani, "Design and impelmentation of a speaker verification system using i-vector and support vector machines," in *Robotics and Mechatronics (ICRoM), 2014 Second RSI/ISM International Conference on*. IEEE, 2014, pp. 434–439.

[27] M. M. Homayounpour and I. Rezaian, "Robust speaker verification based on multi stage vector quantization of mfcc parameters on narrow

bandwidth channels," in *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on*, vol. 1. IEEE, 2008, pp. 336–340.

[28] Y. Liu, T. Fu, Y. Fan, Y. Qian, and K. Yu, "Speaker verification with deep features," in *Neural Networks (IJCNN), 2014 International Joint Conference on*. IEEE, 2014, pp. 747–753.

[29] H. Hermansky, "Perceptual linear predictive (plp) analysis of speech," *the Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[30] H. Li, B. Ma, K.-A. Lee, H. Sun, D. Zhu, K. C. Sim, C. You, R. Tong, I. Kärkkäinen, C.-L. Huang *et al.*, "The i4u system in nist 2008 speaker recognition evaluation," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 4201–4204.

[31] H. C. Bao and Z. C. Juan, "The research of speaker recognition based on gmm and svm," in *System Science and Engineering (ICSSE), 2012 International Conference on*. IEEE, 2012, pp. 373–375.

[32] S. Kamath and P. Loizou, "A multi-band spectral subtraction method for enhancing speech corrupted by colored noise," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, May 2002, pp. IV–4164–IV–4164.

[33] X. Menéndez-Pidal, R. Chen, D. Wu, and M. Tanaka, "Compensation of channel and noise distortions combining normalization and speech enhancement techniques," *Speech Communication*, vol. 34, no. 1–2,

pp. 115 – 126, 2001, noise Robust {ASR}. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639300000492

[34] Z. Q. Wang and D. Wang, "A joint training framework for robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 796–806, April 2016.

[35] K. S. Rao, A. K. Vuppala, S. Chakrabarti, and L. Dutta, "Robust speaker recognition on mobile devices," in *2010 International Conference on Signal Processing and Communications (SPCOM)*, July 2010, pp. 1–5.

[36] M. Wu, D. Wang, and G. J. Brown, "A multipitch tracking algorithm for noisy speech," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 3, pp. 229–241, 2003.

[37] A. Tripathy, L. Kumar, and R. M. Hegde, "Robust two dimensional source localization using the music-group delay spectrum," in *International Conference on Signal Processing and Communications (SPCOM)*, 2012, pp. 1–5.

[38] S. M. Golan, S. Gannot, and I. Cohen, "Subspace tracking of multiple sources and its application to speakers extraction," in *International Conference on Acoustics, Speech and Signal Processing*, March 2010, pp. 201–204.

[39] A. Stolcke, G. Friedland, and D. Imseng, "Leveraging speaker diarization for meeting recognition from distant microphones," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 2010, pp. 4390–4393.

[40] M. McLaren and D. van Leeuwen, "Source-normalized lda for robust speaker recognition using i-vectors from multiple speech sources," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 755–766, 2012.

[41] "DSP speech audio samples," http://www.dsp.diten.unige.it/owncloud/public.php?service=files&t=5830ebe809f4d0260396fcdcbf7b2516.

[42] J. Liu, N. Kato, H. Ujikawa, and K. Suzuki, "Device-to-device communication for mobile multimedia in emerging 5g networks," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 12, no. 5s, 2016. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84994495822&doi=10.1145%2f2983641&partnerID=40&md5=d07a32d63938b2cd6a4ef976ff11a712

[43] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645 – 1660, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X13000241

[44] D. Zeng, S. Guo, and Z. Cheng, "The web of things: A survey," *Journal of Communications*, vol. 6, no. 6, pp. 424–438, 2011, cited By 114. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-80052060563&doi=10.4304%2fjcm.6.6.424-438&partnerID=40&md5=0b14f05c62b528865a8c4b53441af337

[45] S. Duquennoy, G. Grimaud, and J. J. Vandewalle, "The web of things: Interconnecting devices with high usability and performance," in *2009 International Conference on Embedded Software and Systems*, May 2009, pp. 323–330.

[46] J. Liu, N. Kato, J. Ma, and N. Kadowaki, "Device-to-device communication in lte-advanced networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 1923–1940, Fourthquarter 2015.

[47] J. Liu, Y. Kawamoto, H. Nishiyama, N. Kato, and N. Kadowaki, "Device-to-device communications achieve efficient load balancing in lte-advanced networks," *IEEE Wireless Communications*, vol. 21, no. 2, pp. 57–65, April 2014.

[48] J. Liu, H. Nishiyama, N. Kato, and J. Guo, "On the outage probability of device-to-device-communication-enabled multichannel cellular networks: An rss-threshold-based perspective," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 1, pp. 163–175, Jan 2016.

[49] J. Dai, J. Liu, Y. Shi, S. Zhang, and J. Ma, "Analytical modeling of resource allocation in d2d overlaying multihop multichannel uplink cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 6633–6644, Aug 2017.

[50] J. Liu, S. Zhang, W. Sun, and Y. Shi, "In-vehicle network attacks and countermeasures: Challenges and future directions," *IEEE Network*, vol. 31, no. 5, pp. 50–58, 2017.

[51] J. Liu and W. Sun, "Smart attacks against intelligent wearables in people-centric internet of things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 44–49, December 2016.

[52] S. Nordholm, I. Claesson, and B. Bengtsson, "Adaptive array noise suppression of handsfree speaker input in cars," *IEEE Transactions on Vehicular Technology*, vol. 42, no. 4, pp. 514–518, Nov 1993.

[53] W. Sun, J. Liu, and H. Zhang, "When smart wearables meet intelligent vehicles: Challenges and future directions," *IEEE Wireless Communications*, vol. 24, no. 3, pp. 58–65, 2017.

[54] E. Erzin, Y. Yemez, A. M. Tekalp, A. Ercil, H. Erdogan, and H. Abut, "Multimodal person recognition for human-vehicle interaction," *IEEE MultiMedia*, vol. 13, no. 2, pp. 18–31, 2006.

[55] C. Miyajima, Y. Nishiwaki, K. Ozawa, T. Wakita, K. Itou, K. Takeda, and F. Itakura, "Driver modeling based on driving behavior and its evaluation in driver identification," *Proceedings of the IEEE*, vol. 95, no. 2, pp. 427–437, 2007.

[56] C. Miyajima, Y. Nishiwaki, K. Ozawa, T. Wakita, K. Itou, and K. Takeda, "Cepstral analysis of driving behavioral signals for driver identification," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 5. IEEE, 2006, pp. V–V.

[57] A. Riener and A. Ferscha, "Supporting implicit human-to-vehicle interaction: Driver identification from sitting postures," in *The First Annual International Symposium on Vehicular Computing Systems (ISVCS 2008)*, 2008, p. 10.

[58] H.-B. Kang, "Various approaches for driver and driving behavior monitoring: A review," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, June 2013.

[59] Q. Wang, J. Yang, M. Ren, and Y. Zheng, "Driver fatigue detection: A survey," in *2006 6th World Congress on Intelligent Control and Automation*, vol. 2, 2006, pp. 8587–8591.

[60] "Frotcom - automatic driver identification system," http://www.frotcom.com/features/automatic-driver-identification-system?set_lang=en, accessed on: 13th December, 2017.

[61] "Synx and driver id - identify individual drivers and monitor their driving style," https://www.transpoco.com/driver-identification, accessed on: 13th December, 2017.

[62] "Nedap identification systems - driver identification," http://www.nedapidentification.com/driver-identification.html, accessed on: 13th December, 2017.

[63] "Bayometric - automatic driver identification ssystem," https://www.bayometric.com/biometric-driver-identification-system/, accessed on: 13th December, 2017.

[64] U. H. Yapanel and J. H. L. Hansen, "A new perspective on feature extraction for robust in-vehicle speech recognition," in *ISCA Proc.: Eurospeech, 2003*, 2003, pp. 1281–1284.

[65] M. Feld, F. Burkhardt, and C. A. Müller, "Automatic speaker age and gender recognition in the car for tailoring dialog and mobile services," in *Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010*, 01 2010, pp. 2834–2837.

[66] J. P. Openshaw and J. S. Masan, "On the limitations of cepstral features in noise," in *Acoustics, Speech, and Signal Processing, 1994.*

ICASSP-94., 1994 IEEE International Conference on, vol. ii, Apr 1994, pp. II/49–II/52 vol.2.

[67] J. Ming, T. J. Hazen, J. R. Glass, and D. A. Reynolds, "Robust speaker recognition in noisy conditions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1711–1723, July 2007.

[68] Q. Jin, T. Schultz, and A. Waibel, "Far-field speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2023–2032, Sept 2007.

[69] J. Wang, S. Furui, and B. Juang, *Real World Speech Processing*. Springer US, 2013. [Online]. Available: https://books.google.it/books?id=KPrTBwAAQBAJ

[70] "DSP audio speech database," http://www.dsp.diten.unige.it/images/download/speech_db_jstsp.zip.

[71] "White noise - car sounds - driving," https://www.youtube.com/watch?v=tTWfGST8rW4.

[72] "Car driving sound effect interior," https://www.youtube.com/watch?v=30EXeN9wHxg.

[73] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, Dec 2016.

[74] J. Liu, H. Nishiyama, N. Kato, and J. Guo, "On the Outage Probability of Device-to-Device-Communication-Enabled Multichannel Cellular Networks: An RSS-Threshold-Based Perspective," *IEEE Journal*

on Selected Areas in Communications, vol. 34, no. 1, pp. 163–175, Jan 2016.

[75] M. N. Tehrani, M. Uysal, and H. Yanikomeroglu, "Device-to-device communication in 5g cellular networks: challenges, solutions, and future directions," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 86–92, May 2014.

[76] D. Takaishi, Y. Kawamoto, H. Nishiyama, N. Kato, F. Ono, and R. Miura, "Virtual Cell Based Resource Allocation for Efficient Frequency Utilization in Unmanned Aircraft Systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3495–3504, April 2018.

[77] F. Tang, Z. M. Fadlullah, N. Kato, F. Ono, and R. Miura, "AC-POCA: Anticoordination Game Based Partially Overlapping Channels Assignment in Combined UAV and D2D-Based Networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1672–1683, Feb 2018.

[78] X. Li, D. Guo, H. Yin, and G. Wei, "Drone-assisted public safety wireless broadband network," in *2015 IEEE Wireless Communications and Networking Conference Workshops*, March 2015, pp. 323–328.

[79] F. Tang, Z. M. Fadlullah, B. Mao, N. Kato, F. Ono, and R. Miura, "On A Novel Adaptive UAV-Mounted Cloudlet-Aided Recommendation System for LBSNs," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2018.

[80] G. Ding, Q. Wu, L. Zhang, Y. Lin, T. A. Tsiftsis, and Y. D. Yao, "An amateur drone surveillance system based on the cognitive internet of

things," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 29–35, Jan 2018.

[81] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via gps spoofing," *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014.

[82] Z. Kaleem and M. H. Rehmani, "Amateur drone monitoring: State-of-the-art architectures, key enabling technologies, and future research directions," *IEEE Wireless Communications*, vol. 25, no. 2, pp. 150–159, April 2018.

[83] M. Nijim and N. Mantrawadi, "Drone classification and identification system by phenome analysis using data mining techniques," in *2016 IEEE Symposium on Technologies for Homeland Security (HST)*, May 2016, pp. 1–5.

[84] J. Busset, F. Perrodin, P. Wellig, B. Ott, K. Heutschi, T. Rühl, and T. Nussbaumer, "Detection and tracking of drones using advanced acoustic cameras," in *Unmanned/Unattended Sensors and Sensor Networks XI; and Advanced Free-Space Optical Communication Techniques and Applications*, vol. 9647. International Society for Optics and Photonics, 2015, p. 96470F.

[85] M. Benyamin and G. H. Goldman, "Acoustic detection and tracking of a class i uas with a small tetrahedral microphone array," ARMY RESEARCH LAB ADELPHI MD, Tech. Rep., 2014.

[86] Z. Zhang, Y. Cao, M. Ding, L. Zhuang, and W. Yao, "An intruder detection algorithm for vision based sense and avoid system," in *2016*

International Conference on Unmanned Aircraft Systems, June 2016, pp. 550–556.

[87] S. R. Ganti and Y. Kim, "Implementation of detection and tracking mechanism for small uas," in *2016 International Conference on Unmanned Aircraft Systems*, June 2016, pp. 1254–1260.

[88] R. Stolkin, D. Rees, M. Talha, and I. Florescu, "Bayesian fusion of thermal and visible spectra camera data for mean shift tracking with rapid background adaptation," in *2012 IEEE Sensors*, Oct 2012, pp. 1–4.

[89] M. Ritchie, F. Fioranelli, H. Griffiths, and B. Torvik, "Micro-drone rcs analysis," in *2015 IEEE Radar Conference*, Oct 2015, pp. 452–456.

[90] F. Fioranelli, M. Ritchie, H. Griffiths, and H. Borrion, "Classification of loaded/unloaded micro-drones using multistatic radar," *Electronics Letters*, vol. 51, pp. 1813–1815(2), October 2015.

[91] J. Drozdowicz, M. Wielgo, P. Samczynski, K. Kulpa, J. Krzonkalla, M. Mordzonek, M. Bryl, and Z. Jakielaszek, "35 GHz FMCW drone detection system," in *2016 17th International Radar Symposium (IRS)*, May 2016, pp. 1–4.

[92] J. Klare, O. Biallawons, and D. Cerutti-Maori, "Detection of UAVs using the MIMO radar MIRA-CLE Ka," in *Proceedings of EUSAR 2016: 11th European Conference on Synthetic Aperture Radar*, June 2016, pp. 1–4.

[93] A. Moses, M. J. Rutherford, and K. P. Valavanis, "Radar-based detection and identification for miniature air vehicles," in *2011 IEEE*

International Conference on Control Applications (CCA), Sept 2011, pp. 933–940.

[94] M. Witschi, J. Schild, B. Nyffenegger, C. Stoller, M. Berger, R. Vetter, G. Stirnimann, P. Schwab, and F. Dellsperger, "Detection of modern communication signals using frequency domain morphological filtering," in *2016 24th European Signal Processing Conference (EUSIPCO)*, Aug 2016, pp. 1413–1417.

[95] X. Yue, Y. Liu, J. Wang, H. Song, and H. Cao, "Software defined radio and wireless acoustic networking for amateur drone surveillance," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 90–97, APRIL 2018.

[96] H. Fu, S. Abeywickrama, L. Zhang, and C. Yuen, "Low-complexity portable passive drone surveillance via SDR-based signal processing," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 112–118, APRIL 2018.

[97] Z. Liu, Z. Li, B. Liu, X. Fu, I. Raptis, and K. Ren, "Rise of mini-drones: Applications and issues," in *Proceedings of the 2015 Workshop on Privacy-Aware Mobile Computing*. New York, NY, USA: ACM, 2015, pp. 7–12.

[98] S. Kamkar, "Skyjack: autonomous drone hacking w/ Raspberry Pi, aircrack & Javascript," http://www.samy.pl/skyjack/, Accessed on: 18/04/2017.

[99] W. Fundation, "Wireshark: Go deep," https://www.wireshark.org/, Accessed on: 10/07/2018.

[100] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD*

Explor. Newsl., vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278

[101] J.-S. Pleban, R. Band, and R. Creutzburg, "Hacking and securing the AR. Drone 2.0 quadcopter: investigations for improving the security of a toy," in *Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications 2014*, vol. 9030, 2014.

[102] L. Eschenauer, "AR.Drone WebFlight - Flying robots in your browser," https://github.com/eschnou/ardrone-webflight, Accessed on: 23/07/2018.