UAV DETECTION AND LOCALIZATION SYSTEM USING AN INTERCONNECTED ARRAY OF ACOUSTIC SENSORS AND MACHINE LEARNING ALGORITHMS

by

Facundo Ramiro Esquivel Fagiani

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science

Department of Computer and Information Technology

West Lafayette, Indiana

May 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Eric T. Matson, Chair

Department of Computer and Information Technology

Prof. Anthony H. Smith

Department of Computer and Information Technology

Dr. John C. Gallagher

Department of Electrical Engineering and Computer Science, University of Cincinnati

Approved by:

Dr. John Springer

To my mother,

everything I am, I owe it to her.

ACKNOWLEDGMENTS

I wish to acknowledge, with great gratitude, every individual who helped me on the path of developing this thesis. First, to my advisor Prof. Eric T. Matson, for providing the project idea, the resources to develop it, and his constant support at each step of the process. To the other members of my committee, Prof. John C. Gallagher and Prof. Anthony H. Smith for their help on solving issues that appeared along the way. To Prof. Michael J. Dyrenfurth, for teaching me how to properly carry out a research and how to begin with this project. To Dana Utebaieva for introducing me to the topic of sound processing and providing me the tools to start exploring this research area. Finally, to Yaqin Wang, for assisting me on the experiments, even on the coldest days at McAllister park, without her help I would not have completed this project on time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AC-WGAN	Auxiliary Classifier Wasserstein Generative Adversarial Network
AS	Acoustic Sensor
CAGR	Compound Annual Growth Rate
CNN	Convolutional Neural Network
CRNN	Convolutional Recurrent Neural Networks
CUAV	Counter Unmanned Aerial Vehicle
DOA	Direction of Arrival
KNN	K-Nearest Neighbor
LAN	Local Area Network
LOS	Line-of-Sight
LRCS	Laser Radar Cross-Section
LSTM	Long Short-Term Memory
MFCC	Mel-Frequency Cepstral Coefficients
NLOS	Non-Line-of-Sight
OC-SVM	One Class Support Vector Machines
PIL	Plot Image Learning
RCS	Radar Cross Section
RF	Radio Frequency
RMSE	Root Mean Squared Error
RQ	Research Question
SNR	Signal-to-Noise Ratio
STFT	Short-Time Fourier Transform
SVDD	Support Vector Data Description
SVM	Support Vector Machines
TDOA	Time Delay of Arrival
UAV	Unmanned Aerial Vehicle
USRP	Universal Software Radio Peripheral
YOLO	You Only Look Once

ABSTRACT

The Unmanned Aerial Vehicles (UAV) technology has evolved exponentially in recent years. Smaller and less expensive devices allow a world of new applications in different areas, but as this progress can be beneficial, the use of UAVs with malicious intentions also poses a threat. UAVs can carry weapons or explosives and access restricted zones passing undetected, representing a real threat for civilians and institutions. Acoustic detection in combination with machine learning models emerges as a viable solution since, despite its limitations related with environmental noise, it has provided promising results on classifying UAV sounds, it is adaptable to multiple environments, and especially, it can be a cost-effective solution, something much needed in the counter UAV market with high projections for the coming years. The problem addressed by this project is the need for a real-world adaptable solution which can show that an array of acoustic sensors can be implemented for the detection and localization of UAVs with minimal cost and competitive performance.

In this research, a low-cost acoustic detection system that can detect, in real time, about the presence and direction of arrival of a UAV approaching a target was engineered and validated. The model developed includes an array of acoustic sensors remotely connected to a central server, which uses the sound signals to estimate the direction of arrival of the UAV. This model works with a single microphone per node which calculates the position based on the acoustic intensity change produced by the UAV, reducing the implementation costs and being able to work asynchronously. The development of the project included collecting data from UAVs flying both indoors and outdoors, and a performance analysis under realistic conditions.

The results demonstrated that the solution provides real time UAV detection and localization information to protect a target from an attacking UAV, and that it can be applied in real world scenarios.

CHAPTER 1: PURPOSE & PROBLEM

1.1 Introduction

It is common knowledge that Unmanned Aerial Vehicles (UAVs) represent a threat in battlefields nowadays, but they can also put institutions' and civilian's safety at risk [1]. This problem has aroused the attention of researchers who have dedicated multiple studies to the quest of detecting potentially malicious drones.

Visual, acoustic, radar, and radio-frequency solutions have been proposed, some of them with promising results, but with limitations [2]. Sound recognition solutions stand out between the other mentioned approaches because it is a potentially cost-effective approach [3], which can be implemented with limited computational resources [4], executed in real time [5] and provide accurate results [4]–[7].

Detecting an attacking UAV is the first step in the mitigation scenario. To implement counter measures, the position of the attacking drone relative to the protected target is key. For that reason, this work focuses on the design and implementation of a system that can identify the direction of arrival of a UAV relative to a protected target. A low-cost implementation which works in real time is the goal of this project.

1.2 The Problem

Unmanned Aerial Vehicles (UAVs) have certainly become a trending topic in the latest years [8]. Their growth in popularity can be attributed to the multiple potential applications of this technology. From commercial uses to homeland security, the range of possibilities is wide, but as it can be useful, it also represents a potential threat. As the technology evolves, drones are becoming cheaper and smaller each year, and they can carry larger payloads. This poses a risk for civilians and institutions, as it becomes easier for UAVs to invade restricted air space passing undetected [9], [10], or to carry potentially harmful payloads, as weapons or explosives [11], [12]. This is one of the main motivations behind the fast and accurate detection of these threats becoming the center of several studies.

Multiple works have addressed promising results on the detection of UAVs [2]. Either using visual, acoustic, radar, or radio-frequency technologies, each research area has its merits, but they all have their limitations too. Image and lidar recognition devices have problems when the visibility is reduced (e.g., by fog, light, crowds, etc.), sound recognition devices when situated on noisy environments, radars when the object has small radar cross sections (RCS), and radio-frequency devices when trying to identify autonomous flying drones that do not emit identifiable frequencies [2].

Despite their limitations related to the presence of noise, sound recognition solutions are a cost-effective approach [3], the sensors can be located far away from the target, and by pre-processing the training data and using machine learning or deep learning algorithms, authors have achieved good results in the differentiation between UAV signals and other sounds [13], [14]. But these good results are not exempt from questioning. The lack of public available datasets, the diverse and unclear experimental conditions, and the scarce amount of civilian studies which actually use microphone arrays for detection and localization [2], make it difficult to execute a proper comparative analysis, and even more challenging to replicate the results for real world applications.

Even if a system can detect the presence of a malicious UAV, in order to implement counter measures needed to protect a given target, first the UAV must be located. The problem addressed is the need for real world adaptable solutions which demonstrate that an array of acoustic sensors can be used to detect and estimate the direction of arrival of potentially harmful drones under realistic environmental conditions, with minimal cost and competitive performance.

1.3 Significance

The significant threat UAVs represent has been demonstrated in multiple occasions by incidents of alarming risk, as the case of a domestic UAV landing on the United States White House [9], or the attacks to German chancellor Angela Merkel [15] and Venezuelan president Nicolás Maduro [12]. Situations like these have led to an exponential increment on “anti-drone” technology investment, which is projected to reach a market size of $2.315 billon USD by the year 2025 [16], thus it is key to reduce the costs of these technologies without sacrificing effectiveness.

Sound recognition solutions are expected to be cost-effective [3], and they can potentially provide effective results [13], [14], but for a solution like this to be released and marketed as a usable product, first a replicable proof of concept implementation must be delivered, hence the significance of implementing and testing a model under real world conditions. Based on this, the key indicators of the significance of this study are its relative implementation cost and performance benchmarks (response time, accuracy, false alarm rate, classification error, precision, F1-score, etc.) compared with other existing solutions, plus the feasibility of packaging the solution as a replicable and marketable product, to profit from the potential business opportunity.

1.4 The Purpose

The purpose is to develop and validate a low-cost acoustic detection system to alert in real time about the position and direction of arrival of a potentially harmful UAV, relative to a target. The project includes the development of an interconnected array of acoustic detectors which use machine learning classification to recognize the presence of a potentially harmful UAV and estimate its direction of arrival.

The significance of this proposal is given by the need to demonstrate the feasibility of a real-world UAV acoustic location implementation. As an emerging area of research, most of the evidence of success on acoustic detection is experimental, moreover, most related works are more focused on the detection and classification of drones rather than in their localization [2], and those works who focus on locating UAVs tend to increase the costs by using several microphones [6], [17], or are tied to a specific environment [4], [5]. Implementing a low-cost real-world model which demonstrates the effectiveness of acoustic detection using machine learning is relevant because it is the kickoff for the mass replication, marketing, and usage of UAV acoustic detection systems. This need is clear when analyzing the rising amount of investment on anti-drone solutions. In 2018 the anti-drone market size value was USD 576.7 million [18]. Reports indicate that the market size will continue to grow at a Compound Annual Growth Rate (CAGR) of between 24.04% [16] and 29.9% [19], it means it is expected to reach a market size of around USD 2.3 billion by the year 2025 [16], or even more [19]. Acoustic detection being a low-cost solution [3] can help reduce the costs significantly, generating a profitable business opportunity.

1.5 Research Questions and Deliverables

This project is based on the idea that acoustic detection is an effective solution for UAV threat localization, moreover, that a low-cost implementation is feasible by using an array of interconnected acoustic sensors running machine learning classification algorithms. Based on this, the following research questions (RQ) arise:

- *RQ-1:* How accurate, precise, and cost-effective is the proposed model for locating potentially harmful UAVs in real time?

- *RQ-2:* What error level can be achieved on the identification of position and direction of arrival of a UAV using an array of acoustic sensors?
- *RQ-3:* What is the response time that can be achieved on UAV detection using an array of acoustic sensors running machine learning algorithms?
- *RQ-4:* What is the minimum cost an acoustic detection and location system can achieve while keeping an acceptable performance?

Based on the research questions, the deliverables of this project are:

- A working acoustic sensing device which records signals and sends them to a central server to be processed.
- The software installed in a central server to process sound signals, calculate the results related to UAV threat and visualize them in real time.
- The network protocols to communicate components.
- A performance and cost analysis of the proposed model using different configurations.

1.6 Assumptions

- There will be just one UAV flying at a time.
- The UAV does not implement any detection prevention measures.
- The target to be protected will not be moving.

1.7 Delimitations

This project addresses the presence of a potential threat (UAV) and a possible range for its direction of arrival relative to a target. The following items are off the limits of what is going to be delivered and will not be considered:

- Proposing counter measures to stop the attacking UAV.
- Identifying all types of drones.
- Effectiveness under different environmental conditions.
- Physical phenomena such as the Doppler effect.

1.8 Limitations

- Results may vary if a drone emits a frequency that is too different from the samples used for training and testing, although literature says they should not [6].
- The solution may not work if the environment noise overwhelms the drone sound.
- The weather or other environmental noise can have an impact on the effectiveness of the solution.
- The model will be implemented using low-cost computational devices, this can have an impact on the computational time, thus in the response time.
- The connectivity range between nodes can be limited by the networking hardware used.
- The range of detection can be limited by the types of microphones used.
- The accuracy of the results can be limited by the recording quality of the microphones used.

1.9 Definitions

- Unmanned Aerial Vehicle, UAV or Drone: “An aircraft that is operated from a distance, without a person being present on it” [20].
- Machine Learning: “Machine Learning is the science (and art) of programming computers so they can learn from data” [21, p. 10]
- Acoustic sensor: An electronic device that can record sound signals (Operational Definition).
- Node: An element within the network model which includes an acoustic sensor and the means to process the sound signal and communicate the results (Operational Definition).
- Target: An element in the model that is the aim of an attacking UAV (Operational Definition).
- Real-time: A period of time that is considered enough to take an immediate response action (Operational Definition).
- Payload: “Goods that a vehicle is carrying or can carry” [22].
- UAV Detection System: A system that aims to identify if a UAV is present within a given range (Operational Definition).

- Acoustic/Sound Detection System: A UAV Detection System that uses sound signals as the main input (Operational Definition).
- Detection Prevention Measure: Any measure implemented with the goal of having a drone passed undetected (Operational Definition).

1.10 Summary

The problem as identified at this moment is that UAVs represent a threat for civilians' and institutions' safety, for that reason a detection and location, cost-effective system that works in real time is needed. The significance of this problem was demonstrated by several example situations where UAVs jeopardized the safety of civilians and institutions, and the assumption that under current market trends, the presence of UAVs will increase over the coming years.

Acoustic detection models using machine learning classification algorithms are deemed as a possible low-cost solution to detect the presence of an attacking UAV and to identify its possible position and direction of arrival, relative to the target that needs to be protected. A cost-effective real time implementation under this approach is the purpose of the current project, the importance of this is given by the market trends on anti-drone technology which mark an exponential increase for the coming years.

To address the mentioned purpose, a proof-of-concept implementation, which meets the mentioned requirements was developed and tested. This project answers questions related with the performance and feasibility of the approach and delivers working interconnected acoustic sensing devices and the associated software to alert in real time about a potential threat.

In the following chapters, a literature review about the concepts under which this project is based will be presented to construct the reliability of the study, and the methodology to achieve the proposed goals will be explained, including the details about the testing and evaluation process that validates the proposed model.

CHAPTER 2: REVIEW OF THE LITERATURE

2.1 Introduction

The research problem, as perceived at this stage, is that UAVs represent a potential safety problem for civilians and institutions as they can access restricted zones and carry potentially harmful payloads. A real time model for detecting these flying objects is key, and acoustic detection models using machine learning techniques emerge as viable solutions. Moreover, UAV detection systems are generally deployed to protect a target, so locating the attacking UAV is essential to implement counter measures. A proof-of-concept low-cost implementation for UAV detection and location using an array of acoustic sensors, which implement machine learning algorithms to classify them, is needed in order to demonstrate that this technology can meet the requirements and confirm its viability, in that way, this approach could be widely implemented in real world scenarios.

Concepts relevant to this study are:

- Unmanned Aerial Vehicle, UAV, or Drone: An autonomous flying object. It can represent a threat to civilians and institutions safety.
- Acoustic sensors: Microphone devices that can capture sound signals. A set of these elements form an array of acoustic sensors.
- Machine Learning: algorithms that generate a classification or prediction model based on training data. Neural Networks or Deep Learning algorithms are encompassed in this term.
- Sound Classification: Method that uses a classification algorithm to classify sound signals.

2.2 Search methodology

To start with the literature search, IEEE Xplore and Scopus were identified as the most promising library databases. IEEE Xplore provides "full text access to the world's highest quality technical literature in electrical engineering, computer science, and electronics" [23]. Scopus is the "largest abstract and citation database of peer-reviewed research literature" [23]. Both include articles from conferences, journals, magazines, and standards.

Considering the problem statement, a graphic sketch that illustrates the relationships between important concepts is presented in Figure 1. These concepts are then grouped and organized on a Venn Diagram on Figure 2. Using this as a basis, an initial search strategy was proposed. It includes five search terms with Boolean logic, which can be visualized on Figure 2 as S#1 to S#5. The period was restricted to 2017 – 2020 to find just the most recent works on the area. On IEEE Xplore database, the search (Table 1) resulted to be effective, finding a total of 55 articles deemed useful, from which 48 were found using the search term A1 (corresponding to S#1 in Figure 2). Search term A4 had to be restricted to only journals and magazines because it returned too many results to be analyzed. The same applies to search term A5, but the word "threat" was removed instead. The search in Scopus database (Table 2) provided similar results, with a total of 39 articles deemed useful, from which 37 correspond to search term B1. Searches B4 and B5 could not be analyzed because they provided too many results.

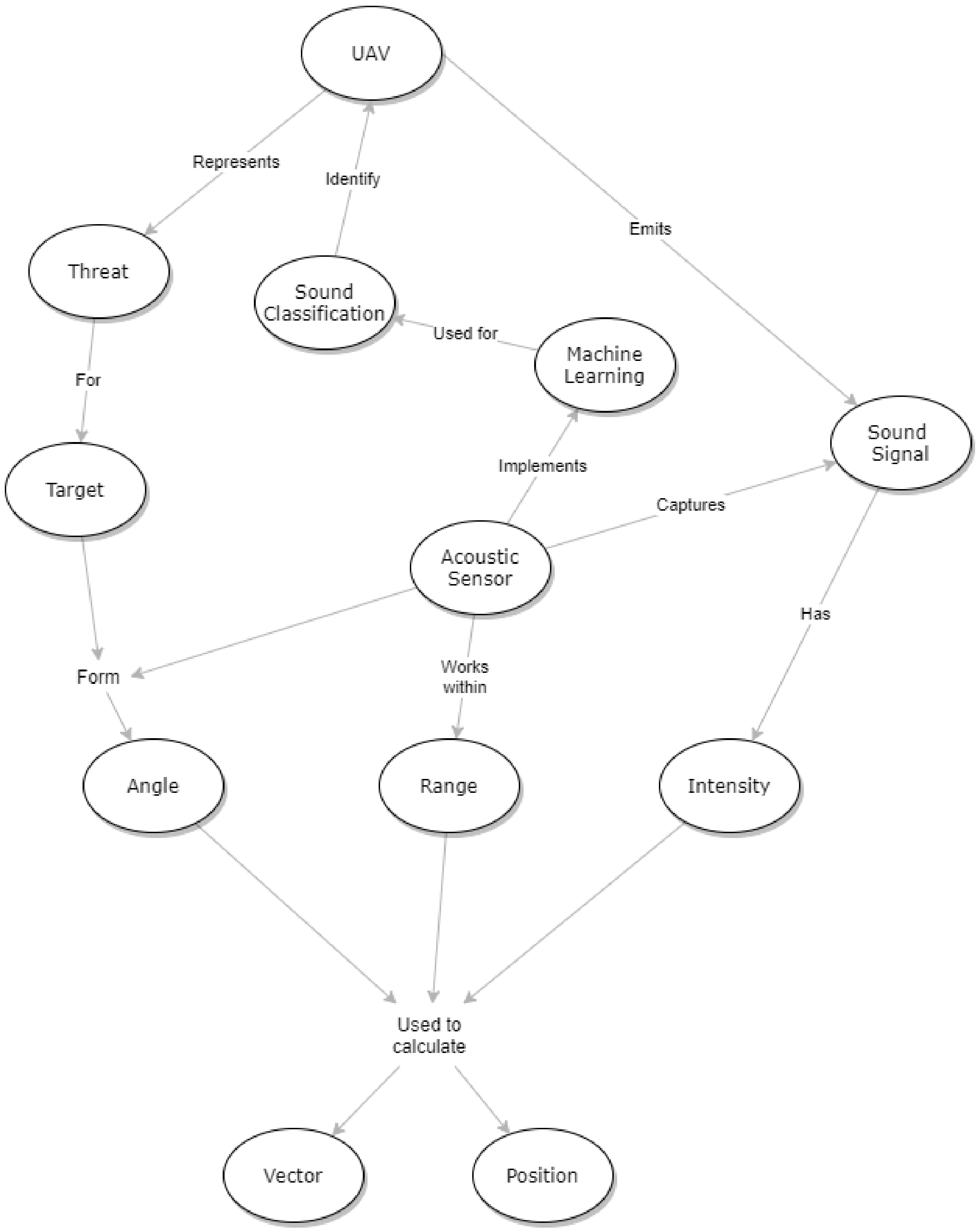

Figure 1. Concept Map that illustrates the relationships among key concepts

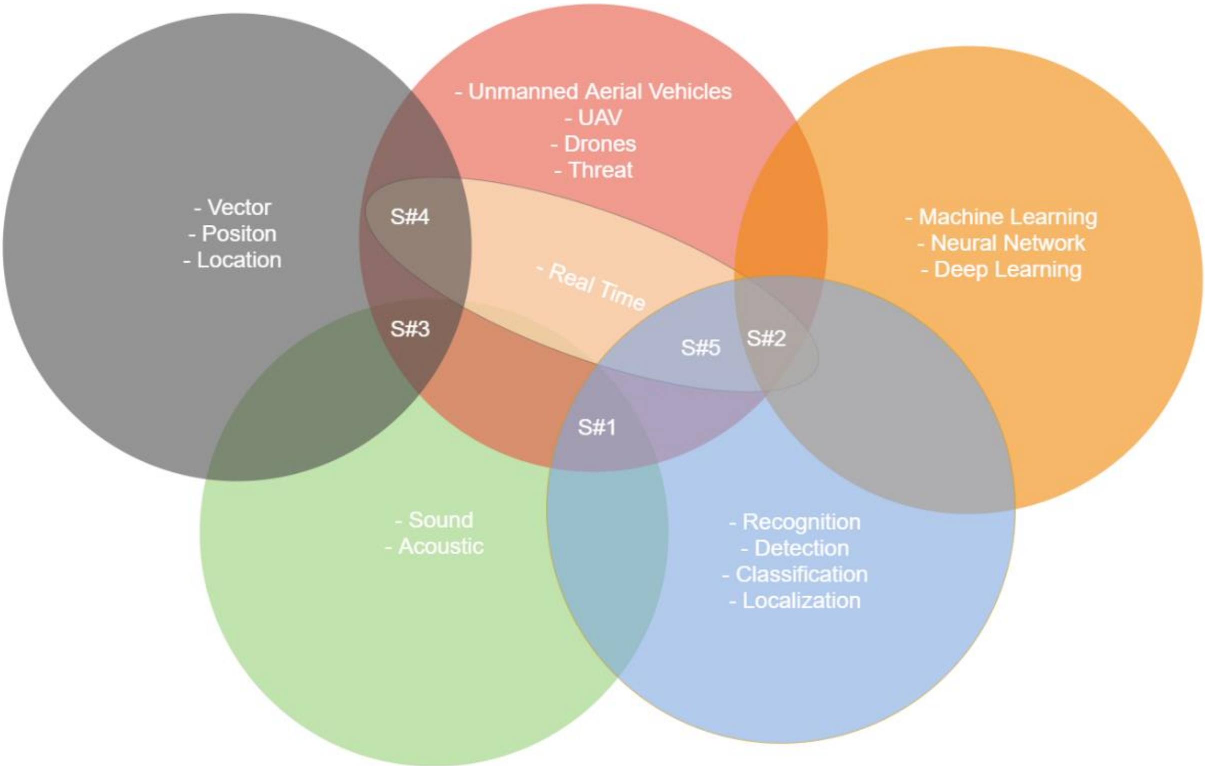

Figure 2. Venn Diagram that illustrates the relationships among key concepts

Table 1. Database A (IEEE Xplore) results – October 27, 2020

S#	Search Term(s)	Period	Filter	Hits	Repeated	Useful
A1	(Sound OR Acoustic) AND (Recognition OR Detection OR Classification OR Localization) AND ("Unmanned Aerial Vehicles" OR UAV OR Drones OR Threat)	2017 - 2020	-	158	0	48
A2	("Unmanned Aerial Vehicles" OR UAV OR Drones OR Threat) AND ("Machine Learning") AND (Recognition OR Detection OR Classification OR Localization) AND "Real Time"	2017 - 2020	-	155	3	3
A3	("Unmanned Aerial Vehicles" OR UAV OR Drones OR Threat) AND (Sound OR Acoustic) AND (Vector OR Position OR Location)	2017 - 2020	-	83	54	2
A4	("Unmanned Aerial Vehicles" OR UAV OR Drones OR Threat) AND (Vector OR Position OR Location) AND "Real Time"	2017 - 2020	Journals & Magazines	133	8	0
A5	("Unmanned Aerial Vehicles" OR UAV OR Drones ~~OR Threat~~) AND (Recognition OR Detection OR Classification OR Localization) AND "Real Time"	2017 - 2021	-	126	43	2

Table 2. Database B (Scopus) results – October 28, 2020

S#	Search Term(s)	Period	Filter	Hits	Repeated	Useful
B1	(Sound OR Acoustic) AND (Recognition OR Detection OR Classification OR Localization) AND ("Unmanned Aerial Vehicles" OR UAV OR Drones OR Threat)	2017 - 2020	-	279	61	37
B2	("Unmanned Aerial Vehicles" OR UAV OR Drones OR Threat) AND ("Machine Learning") AND (Recognition OR Detection OR Classification OR Localization) AND "Real Time"	2017 - 2020	-	251	57	0
B3	("Unmanned Aerial Vehicles" OR UAV OR Drones OR Threat) AND (Sound OR Acoustic) AND (Vector OR Position OR Location)	2017 - 2020	-	286	105	2
B4	("Unmanned Aerial Vehicles" OR UAV OR Drones OR Threat) AND (Vector OR Position OR Location) AND "Real Time"	2017 - 2020	-	1257	-	-
B5	("Unmanned Aerial Vehicles" OR UAV OR Drones ~~OR Threat~~) AND (Recognition OR Detection OR Classification OR Localization) AND "Real Time"	2017 - 2021	-	1425	-	-

Based on the experience from previously mentioned searches, search term S#1 outperforms all the other terms. Most of the deemed useful articles found using other search terms are already included by S#1, as the number of repeated articles show. To continue the search, only S#1 search term was used to search on seven more databases (Tables 3-9). From those, Web of Science provided the most significant amount of new useful articles, including

17 patents that will result useful to analyze. A search for theses and dissertations was also carried out on "ABI Inform Collection" (Table 9) and "Purdue University Graduate School" database (Table 8), providing 3 useful theses.

Table 3. Database C (Web of Science) results – October 29, 2020

S#	Search Term(s)	Period	Filter	Hits	Repeated	Useful
C1	TS=(Sound OR Acoustic) AND TS=(Recognition OR Detection OR Classification OR Localization) AND TS=("Unmanned Aerial Vehicles" OR UAV OR Drones)	2017 - 2020	-	309	90	24

Table 4. Database D (Engineering Village) results – October 29, 2020

S#	Search Term(s)	Period	Filter	Hits	Repeated	Useful
D1	(Sound OR Acoustic) AND (Recognition OR Detection OR Classification OR Localization) AND ("Unmanned Aerial Vehicles" OR UAV OR Drones)	2017 - 2020	NOT IEEE	248	149	5

Table 5. Database E (ProQuest) results – October 29, 2020

S#	Search Term(s)	Period	Filter	Hits	Repeated	Useful
E1	(Sound OR Acoustic) AND (Recognition OR Detection OR Classification OR Localization) AND ("Unmanned Aerial Vehicles" OR UAV OR Drones)	2017 - 2020	-Scholarly Journals - UAV (Subject)	276	19	4

Table 6. Database F (Knovel) results – October 29, 2020

S#	Search Term(s)	Period	Filter	Hits	Repeated	Useful
F1	(Sound OR Acoustic) AND (Recognition OR Detection OR Classification OR Localization) AND ("Unmanned Aerial Vehicles" OR UAV OR Drones)	-	-	30	0	0

Table 7. Database G (Techstreet Enterprise) results – October 29, 2020

S#	Search Term(s)	Period	Filter	Hits	Repeated	Useful
G1	(Sound OR Acoustic) AND (Recognition OR Detection OR Classification OR Localization) AND ("Unmanned Aerial Vehicles" OR UAV OR Drones)	-	-	5	0	0

Table 8. Database H (Purdue University Graduate School) results – November 09, 2020

S#	Search Term(s)	Period	Filter	Hits	Repeated	Useful
G1	(Sound OR Acoustic) AND (Recognition OR Detection OR Classification OR Localization) AND ("Unmanned Aerial Vehicles" OR UAV OR Drones)	-	- Dissertations & Theses - Categories: Computer Engineering, Applied Computer Science, Computer System Security, AI and Image Processing, Autonomous Vehicles	275	0	3

Table 9. Database J (ABI Inform Collection) results – November 09, 2020

S#	Search Term(s)	Period	Filter	Hits	Repeated	Useful
G1	(Sound OR Acoustic) AND (Recognition OR Detection OR Classification OR Localization) AND ("Unmanned Aerial Vehicles" OR UAV OR Drones)	2010-2020	Dissertations & Theses	64	0	0

The final count of articles deemed useful to analyze is 130. 63 articles are conference papers, 47 are journal or magazine articles, 3 are theses, and 17 are patents. The articles were then classified based on the main topic they can provide information about during the literature review (Table 10). Although many could belong to more than one category, they were put only on their most prominent category. It is worth mention that methods other than acoustic detection were deemed useful to gain context, but they were not the main focus of this search, for that reason there are many less articles found on those topics than in acoustic detection.

Table 10. Article classification

Topic	Count	Detail
Acoustic Detection	40	Articles focused on detection of UAVs using sound signals
Drone Classification	7	Articles focused on classifying UAV sound
Drone Localization	12	Articles that use acoustic detection to locate UAVs
Multi Method Detection	7	Articles that use more than one method to detect UAVs
Radar Detection	11	Articles focused on detection of UAVs using radars
Radio-Frequency Detection	8	Articles focused on detection of UAVs based on radiofrequency
Review	11	Articles which review existing literature and methods
Sound Analysis/Processing	7	Articles focused on the analysis or processing of sound signals
Use Acoustic Sensor Array	16	Articles that use acoustic detection and mention the implementation of an array of acoustic sensors.
Visual Detection	11	Articles focused on detection of UAVs using image detection or laser beams.

2.3 Literature summary pertaining to the problem

Unmanned aerial vehicles have exponentially gained popularity over the recent years. With an annual growth of 66.8%, the global shipments of this technology are expected to reach 2.4 million units by 2023 [24].

The reduction on UAV costs has made them available for more people in more application areas. This democratization of the access to UAVs brings benefits to society when used responsibly, but as with any technology, it can also represent a threat if used with malicious intentions. On civil scenarios, UAVs have been used by burglars to make reconnaissance and target homes [25], to smuggle drugs into prisons [26], and more. Battlefields are probably the terrain where malicious drones are more widely used. Terrorists can use them to carry weapons or explosives, representing a serious threat to infantry [11], [27]. But that is not the only threat to homeland security, drone attacks to important government heads [12], [15] and UAV access to restricted zones [9], [10] have also raised the alarms and made evident the need to detect and locate potentially harmful UAVs.

2.3.1 UAV Detection

UAV detection has become the focus of many studies which have approached the problem in different ways. Detecting the presence of a UAV can be a challenging task due to the small size and small speed and altitude at which some drones can fly [6]. This section focuses on the different approaches to UAV detection, each one with their benefits and limitations.

2.3.1.1 Radar Detection

Radar devices “radiate electromagnetic energy and detect the echo returned from reflecting objects (target)” [28, p. 1.1]. Based on the echo returned, radars collect information about the position and nature of the target. The capacity of these devices to detect an object is highly conditioned by the target’s Radar Cross-Section (RCS), an attribute of objects that describes the intensity of the echo they return when exposed to an electromagnetic wave, and which depends on the physical attributes of the object, such as composition, size, shape, radiation, and polarization, among others [28, p. 11.2-11.18]

The detection of drones and especially micro-drones represents a challenge for radars since they can have a small RCS, and they can fly at low altitudes [29]. Despite this limitation, multiple articles have approached the problem of detecting and classifying UAVs by using radar recognition and have provided good results [2], even claiming that this approach has proven to be viable [30].

In [31] authors addressed the issue that it is not viable to implement a continuous transmission radar system since it would mean a high operational cost, and it raises concerns to human safety due to possible excess of radiation, so they proposed a passive radar alternative and tested it with favorable results.

In [32] the study takes the approach of a binary classification between drones and birds. Since these animals share similar RCS and motion patterns with UAVs, they tend to confuse radars. With a simple KNN approach, they obtained accuracy results close to 100% for close range tests (0.3-0.4 km).

A more detailed approach is taken in [33] where authors used radars to detect if a drone was carrying a payload. They classified in a close range of 60m if drones were carrying payloads of 0g, 200g and 500g, and got accuracy results above 90% using a Naive Bayes algorithm.

Even though UAV radar detection studies have provided promising results, they still show important limitations. A lot of the work on the topic which presumes positive results, can be considered as just experimental, and in some cases the experimental conditions seem limited, i.e., the experiments were performed at low altitudes or with limited ranges [2]. Summarizing, it is not possible to state that this technology has overcome its limitations related with UAVs' low altitude flight, slow speed, small size, and small RCS, although it may do it in the future.

2.3.1.2 Radio-Frequency Detection

UAVs are usually remotely commanded using a Radiofrequency (RF) signal, so by capturing those command signals, or any RF signals emitted/received by the drone, it is possible to detect and track UAVs, this is the basic concept behind UAV RF detection [2].

In [34] a hash fingerprint and a distance-based support vector data description (SVDD) algorithm are used on the detection of UAVs. Using this method and a small number of elements in the training set, authors were able to recognize UAV signals in the 2.4GHz frequency band, obtaining good results on an indoor environment, although the system performance deteriorates as the noise increases.

In [35] a non-line-of-sight (NLOS) RF solution is proposed with low-cost hardware. Authors state that the NLOS condition generates an amplifying effect on the RF signatures produced by UAV's movement, so they captured the features from NLOS RF signals using a deep learning model (i.e. an LSTM network), and then they implemented a binary classification using Support Vector Machines (SVM). They were able to detect UAVs with an accuracy rate of 96% on NLOS scenarios, and 98.4% on line-of-sight (LOS) scenarios.

In [36] a model using Auxiliary Classifier Wasserstein Generative Adversarial Networks (AC-WGANs) is proposed. They use a Universal Software Radio Peripheral (USRP) oscilloscope and an antenna to collect UAVs' signals and environmental RF noise, which are used to train their AC-WGANs model. They achieved a classification rate around 95% on indoor environment test. Even augmenting the noise ratio the results were still promising, but in the outdoor environment experiment the model did not perform as good, and quickly deteriorates with the distance.

It can be concluded that RF detection techniques have shown promising results, but they still show limitations. The results deteriorate quickly with the presence of environment noise and when increasing the distance. Another important drawback is that despite most drones are remotely commanded, there exist UAVs that fly autonomously by presetting the flight path or using preprogrammed GPS, limiting RF detection systems possibilities since no RF signals are exchanged [2].

2.3.1.3 Visual Detection

Visual detection is the use of image or video data and computer vision techniques to detect UAVs [2]. Some of the advantages of using visual detection methods are its medium detection range, good localization perspectives and the easy interpretation of data by humans, unlike other methods which require an expert eye to interpret the data. In addition, visual data provides more information about the object, like the model, dimensions and if it is carrying payload [37].

Some of the challenges of visual detection methods are that it is difficult to detect drones at high speeds in real time, and that UAVs' shapes can be confused with other flying objects, like planes or birds. For that reason in [37] the problem is divided in two sections, first they focus on the detection of moving objects, then they try to classify the object between drone, bird or background. For the detection of moving objects, they used a method called "two-points background subtraction algorithm", in which the pixels that change their value from one frame to the next one are analyzed. For the classification part, they used a Convolutional Neural Network (CNN) algorithm, achieving an F1-Score of 0.742 overall. The main limitation found on this implementation was that moving backgrounds heavily affect their performance.

In [38] an implementation consisting of two cameras is considered. First, a static wide-angle camera is used to make a primary flying object detection and tracking in a long range of up to ~1km, then the objects that are deemed suspicious due to visual and motion signatures are further analyzed with a narrow-angle RGB camera. Both cameras are installed in a rotating turret, and both detection processes are done together overlaying the frames coming from both cameras and implementing a "You Only Look Once (YOLO)" deep learning algorithm. They achieved to reduce false alarms almost to 0, although this method fails to detect some positive cases, with a 0.91 true positive rate, which is less than other methods mentioned in the paper.

In [39] the focus is put on the distinction between UAVs and aircrafts for the implementation of a UAV detection system in airports. The basis of this design is that UAVs have different motion patterns, so curvature and turn based features are extracted to train a binary classification algorithm. Even using a simple K-Nearest Neighbor (KNN) algorithm, this work achieved an accuracy of around 90%. Of course, this implementation is quite limited since it only differentiates the flight patterns of UAVs and aircrafts.

Regarding the drawbacks of image detection, it can be mentioned that the accuracy of these methods is heavily correlated with image quality, which means that an image detection implementation would require high quality cameras and more computational time, meaning an increment on costs [1]. Another drawback is that image detection performs poorly when the visibility is low due to time and weather [2], [37]. Thermal cameras are an alternative to bypass this limitation, however they represent an increase in costs and they still have some problems on humid environments [37].

On the task of tracking UAVs, optical sensors can spot and trace drones, but having an accurate estimation of the actual spatial velocity and position in real time is a complex task [6].

2.3.1.4 Lidar Detection

Another line-of-sight approach for UAV detection is the use of lidars. Although it bears some similarities with visual detection, lidar implementations provide some advantages. Since it is not affected by a moving or noisy background, it still works on dark environments, and the position of an object is known as soon as it is detected [40].

Talking about the drawbacks, lidars are still limited by the line of sight, meaning that fog, rain, or other environmental obstructions may add noise. Lidars are also quite expensive devices, so a cost-effective implementation is hard to implement [41]. A particular difficulty on UAV detection is that UAVs have a small laser radar cross-section (LRCS), representing a real challenge for this method [40].

In [40] and [41] authors implement tracking systems using lidars to detect the position of UAVs, but both achieved mixed results, and how to distinguish UAVs from other flying objects is not so clear. There exist also a patent which uses a lidar to track UAVs' position [42], but again, the detection part of the process lacks of clarity. It can be concluded that this technology is just in its initial steps, better results may be found in future works.

2.3.1.5 Acoustic Detection

Acoustic detection is a prominent area within UAV detection. The method is based on the idea that UAVs emit different noises (propeller blades, engine, wind, etc.), among which propeller blades is the one that stands out the most and can be detected [43].

Figure 3. [6] shows the sound signal emitted by a UAV in the frequency and time spectrum. Fig. 3 (b) and Fig. 3 (d) show in different colors the different spectrum amplitudes of the signal. Matching them with the harmonics shown in Fig. 3 (a) and Fig. 3 (b), it is visible that what marks the difference between UAVs' sound signal and noise is the presence of harmonics, they are the main features to be detected by an acoustic detection technique.

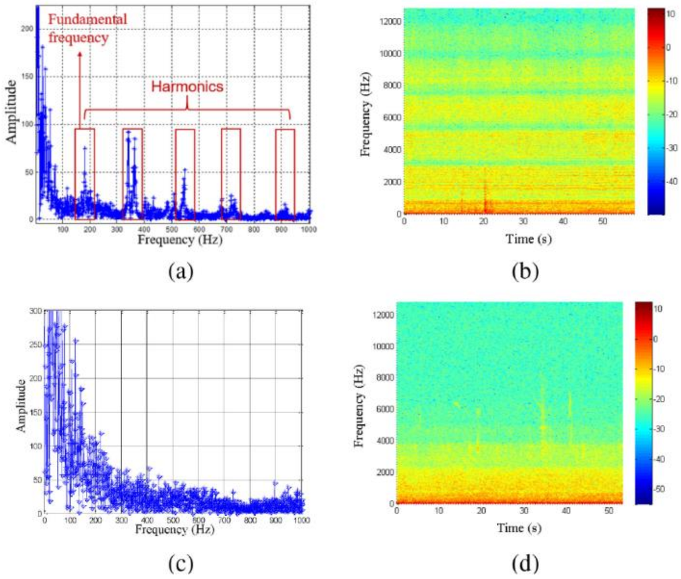

Figure 3. "Time-Frequency analysis of the drone's signals and background noise"[6, p. 2733]

There are several reasons for using this approach. Acoustic sensors can be placed at any distance from the target so UAVs can be detected at wider ranges, and acoustic sensors can also detect a threat at any angle [4]. Price is also a significant factor since an array of acoustic sensors could be a low-cost solution, although it depends on the quality of the microphones used [44].

Multiple authors have provided promising results when it comes to UAV Detection using acoustic sensors [4]–[7], [45], but acoustic detection comes with some shortcomings as well. The detection rate can be affected by several factors, including "the sensitivities of microphones, surrounding noise, the distances between the drone and the arrays" [6, p. 2736]. The model proposed in [14], for example, failed when the environmental noise (originated by planes in this case) dominated the UAV sound.

The success in this detection method has even led to the publication of a few patents. Some examples include a method for distinguishing drone sound from other sounds by producing two sets of feature vectors [46], and a method that uses both sound and shape information to identify low-altitude UAVs [47].

2.3.1.6 Payload Detection

UAVs are widely used with recreational purposes, which means that not every time a UAV is detected it poses a threat, especially on civil scenarios. A prominent research area encompassed within UAV detection is the classification of drones carrying payload, because a payload could potentially be weapons or explosives. Acoustic detection appears as a possible solution for this problem since adding payload to a drone increments its mass, altering its acoustic signature [7].

The goal in [7] was to classify drones between “loaded”, “unloaded” and “noise”, using CNN algorithms. Authors managed to achieve 99.5% of accuracy on their tests, although it is mentioned that the response time may be large, making it non-viable for real time implementations.

In [48] the approach was to separate the problem into two binary classifications. One classifier was trained to detect if a sound came from a loaded Phantom 2 drone, and the other if the sound was an unloaded Phantom 2 drone. Both classifiers were Convolutional Neural Networks (CNN), and the final prediction is taken using a voting system between the results provided by the two classifiers. Using this method, a composite accuracy of 99.92% was achieved. One thing to mention is that, as proposed in this paper, the model must be trained again for each new UAV model that it wants to support.

In conclusion, although there is still work to be done on the generalizability and real time processing of acoustic methods for payload detection, it is a promising research area.

2.4 Literature summary pertaining to the purpose & its significance

In the current study, the focus is put on identifying in real time the position of a UAV by using acoustic sensors, specifically on the angle and direction of arrival of the drone relative to a target.

Implementing a solution that uses acoustic sensors could bring significant benefits. As previously stated, acoustic detection is a potentially low-cost approach to UAV detection if microphones are properly chosen. An inexpensive application using acoustic sensors was demonstrated in [3]. Another benefit of acoustic detection is the reduction on computational resources it requires [4], making it not only cheaper to implement, but also reducing the response time of classification algorithms, making it more suitable for real time requirements.

Having a low-cost implementation for UAV detection and localization is key under the current market context, where the investment on anti-drone technologies is expected to take

$2.315 billion USD of the market size by the year 2025 [16], with some predictions going even further, saying it will have a Compound Annual Growth Rate (CAGR) of 29.9%, reaching $4.5 billion USD by the year 2026 [19]. Having the anti-drone product with the most competitive costs and best results will be the challenge of many companies in the near future.

2.4.1 UAV Localization

Detecting the presence of UAVs is a challenging task, but the speed and direction of arrival of a drone are even more complex to calculate [5]. Despite that, some works have made progress on the localization and tracking of UAVs in real time.

One approach to UAV localization is to turn the problem into a binary classification, and find the position based on the presence or absence of a UAV in a zone. Authors in [4] managed to trace the trajectory of the UAV by plotting the presence of drone sounds over time. In [5] authors used a single node, consisting of two acoustic sensors with 10m of separation, to estimate the direction of arrival (DOA) of a UAV. The implementation consisted of separating a field into different sections and using CNN and CRNN algorithms to predict if a drone is present in each section. They achieved an accuracy of 97.6% with an inference time of 0.429 seconds, demonstrating that a real time implementation is possible. A disadvantage of their approach is that the arrangement of acoustic sensors is tied to that specific configuration for any future implementations.

Another approach to find the UAV direction of arrival is the use of beamforming. Beamforming is a signal processing technique which uses propagating wave fields to estimate the direction of arrival of a radio or acoustic signal, by filtering the signals with overlapping frequency content that come from different locations [49]. A method like this is used in [17] where they used the delay between the channels on the recordings to find the angle of arrival of the sound. They then needed to confirm the nature of the object, so they focalized the recording on that direction and applied a binary classification to identify if the object was a UAV or not. Authors mention that low elevation angle and multi-source are problems that need to be solved in the future. Beamforming technique has even been used in a patent of 2018 to determine the location of a drone by its sound [50].

A method similar to beamforming was used in [6]. This method uses a Time Delay of Arrival (TDOA) algorithm to avoid the problem of multipath effect. Authors mention that in a TDOA analysis, multiple peaks can appear, leading to wrong localization results, so they implemented a Bayesian framework, a method that iteratively predicts the state of a parameter based on the current status of the system and historical estimated states. They mention they could achieve an estimation error below 5 meters on 90% of the times.

2.5 Literature summary pertaining to the methodology

2.5.1 Feature Extraction

Sound signals need to be transformed from the frequency domain, as captured by an acoustic sensor, to digital information that can be interpreted by a learning algorithm. This means that the signal must be preprocessed, and features need to be extracted from it before being used in a training or prediction process.

The first step is determining the time length that each sample will have. Short time samples are more sensitive to noise [7], while long time samples take more time to process and can reduce the response time to a potential threat. In [5] it is mentioned that length an accuracy are not linearly related, but a strong correlation exists between them. Another sampling technique to gather more information and increase precision is to overlap the sequences, as in [17] where they propose a 50% of overlapping.

When dealing with sound frequencies, it is found that high frequencies tend to have smaller magnitudes than low frequencies, for that reason a pre-emphasis filter is usually applied to have a relatively constant frequency response among different frequency bands [7]. Formula 1 is a commonly applied formula for pre-emphasis filter [7] where α is the filter coefficient, x is the input signal and y is the filtered signal. Both x and y are in the time domain.

$$y(t) = x(t) - \alpha x(t-1)$$

Formula 1. Pre-emphasis filter

As previously mentioned, what makes the UAV sound distinguishable are its harmonics. Three typical methods to extract features from the harmonics are the Short-Time Fourier Transform (STFT), Filter Banks, and Mel-Frequency Cepstral Coefficients (MFCC).

STFT is a Fourier related transform used to obtain local properties of a frequency (f), in particular, “to obtain some ‘local frequency spectrum’, f is restricted to an interval and the Fourier transform of this restriction is taken” [51, p. 37], meaning that a signal is divided in short segments to compute the Fourier transform of each of those short segments. This feature extraction technique was used in many successful implementations [4], [6].

Filter banks are “an array of band-pass filters that separates the input signal into multiple components, each one carrying a single frequency sub-band of the original signal” [52, p. 2]. This feature extraction technique is inspired in the way the human auditory system processes audio signals [53], and it is an intermediate step on the calculation of MFCCs.

MFCCs are short-term spectral based features which can represent sounds’ amplitude in a compact form, and for that reason it is a popular method for speech recognition [54]. Using MFCC can reduce the size of the data, making computation faster. In [7] it is mentioned that it can help reduce the size of the data to 1/90.

Although MFCC is more widely used for audio classification tasks, authors in [4] found that STFT is better than MFCC for UAV distinction, because “both wind and UAV have stronger amplitude on lower frequency bands” and “MFCC contains more dense information of sounds as it represents sounds with several coefficients, while STFT is relatively an intermediate feature” [4, p. 497].

A normalization function can be applied at different points during signal processing. In [7] the normalization function is applied on the input data to “find an optimization point quickly for the gradient descent method” and to “perform adequate learning instantaneously by eliminating the small learning rate set disadvantage” [7, p. 863]. In [4] it is mentioned that the signals should be normalized, but a scaling normalization, which is “a technique to divide the signals by the maximum value of the total audio file” [4, p. 498], is not possible because in real-time environments the maximum value changes as new signals are captured.

2.5.2 Machine Learning algorithms

On the detection and classification of UAVs using sound recognition, different machine learning algorithms have been used, some of them providing good results, as will be discussed below.

Plot image machine learning algorithms (PIL) were trained with pre-recorded UAV sounds obtaining an accuracy of 83% on binary classification in [1], although authors mention that PIL algorithms require large data sets to get accurate results.

K-Nearest Neighbors (KNN) is one of the simplest machine learning algorithms, for that reason it is also so popular. It was implemented in [1] with pre-recorded UAV sounds, obtaining 61% of accuracy on a binary classification, a poor performance if compared with other algorithms in this review. Authors in [1] mention that although fast and simple, KNN is “not capable of building the hierarchies of internal representations likely necessary to support proper classification of similar, yet distinct, target” [1, p. 4].

Support Vector Machines (SVM) are used in [4] with data collected in person using an array of acoustic sensors. For a binary classification (drone vs noise), an F-1 Score of between 0.779 to 0.787 was obtained. In [17] an SVM and a semi-supervised One Class SVM (OC-SVM) are used, on a binary classification as well. They achieved an accuracy of 99.5% and 95.6% respectively, meaning that OC-SVM is not an improvement on traditional SVM.

Convolutional Neural Networks (CNN) is arguably the most popular machine learning algorithm nowadays. It is used for binary classification in [5] and it resulted in an accuracy between 92.88% and 98.23%. A variant of CNNs are Convolutional Recurrent Neural Networks (CRNN), and they were implemented in [5] as well, obtaining even better results (between 95.43% and 97.6%)

More complex implementations like ensembles of different machine learning algorithms or neural networks with several layers may provide better results, but they also imply a significant increase on computational costs. For example, a ResNet-50 Convolutional Neural Network (CNN) was implemented for binary classification in [5] providing more accurate results (98.47%) than a simpler CNN implementation (97.6%), but it took around 16 times longer to predict the results.

Another approach to improve detection performance is detection fusion, meaning the use of several microphones to individually detect the presence of UAVs, and then fusing all individual results in one consensual prediction. An approach like this was used in [6] where they used 8 microphones, each one running an SVM algorithm to detect the presence of a UAV, then the predictions were prioritized and fused using a weigh vector, obtaining an almost perfect detection rate, with a false alarm rate of 6.44% in the worst case scenario. The disadvantage of this method is that having 8 detectors at the same time represents 8 times more hardware needed, increasing the costs.

It is worth mentioning that although some results are really promising, the characteristics of the training data (features, number of samples, parameters, amount of noise, etc.) are not always deeply described, so implementing the same methods may not provide the same results.

2.5.3 System Design

For the experiment setup, the acoustic sensors can be arranged in multiple ways. Angle, range, alignment, number of nodes, number of microphones per node, and height of the acoustic sensors are some of the variables to take into consideration, and these configurations can have an impact on the results.

In [4], six acoustic sensors were configured surrounding the target, the angle and range between acoustic sensors and the target were changed in each of the four experiments as shown in Figure 4.

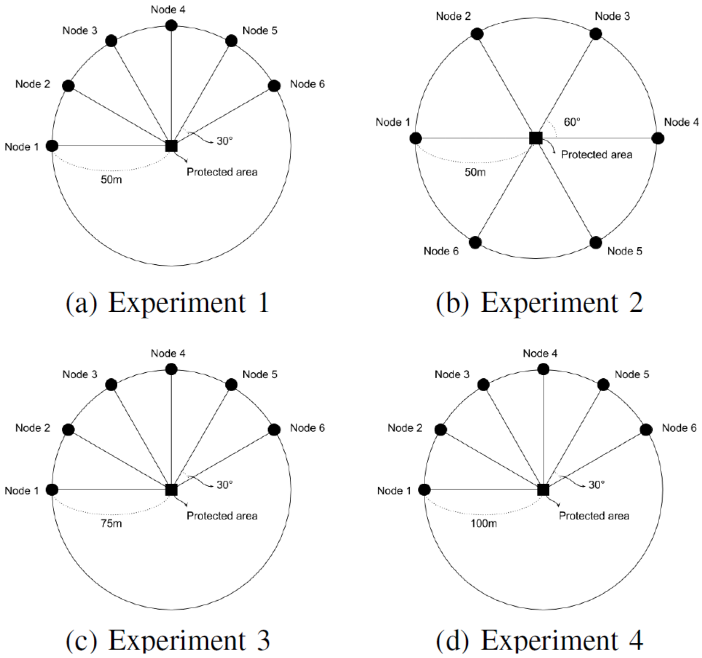

Figure 4. “Experimental Configurations” [4, p. 495]

Each node can have a different setup as well. In [5] for example, each node consists of two acoustic sensors with 10 meters of separation between each other, each of them with stereotype input, but which can record as a single channel. Having multiple microphones per node is necessary for using methods like beamforming or TDOA since the angle of arrival is calculated based on the difference between the signal recorded by each sensor [6], [17]. But having more sensors also means a higher cost and is an additional computational challenge to process the signals together.

The connection between elements is another aspect to consider. The fastest way to connect devices seems to be through an optic fiber as in [6], but having the devices connected through a wire reduces the flexibility of the design. A more flexible approach is networking the devices using a wireless communication as described in [4].

2.5.4 Evaluation

During evaluation, there are some factors that can affect the outcome of the model implemented and must be considered. Microphone sensitivity, microphone quality, surrounding noise, and distance between nodes or to the attacking UAV are some examples.

The quality of a recording, is positively correlated with the accuracy of a sound classification model, so results can change based on the quality of the sample used for training and testing [4].

Noise is probably the main difficulty for acoustic detection method as previously stated, so measuring the signal-to-noise ratio (SNR) could be helpful to understand the impact it is having on the evaluation. In [6] the SNR is measured by collecting surrounding noise for a long period of time in a specific surveillance region, then the collected noise was divided in several segments and “the average spectrum of the stationary noise” [6, p. 2736] was obtained. It is worth mentioning that collecting noise only from one place could be helpful to provide better results on that specific setup, but it would reduce the adaptability of the model to other scenarios.

Different UAVs have been used to test the results in the literature review. DJI Phantom 1 [1], DJI Phantom 2 [1], [5], [7], DJI Phantom 3 [6], [55], or Parrot AR Drone 2.0 [4] are some examples of drones used, but since the acoustic signal frequency emitted by most amateur drones are close to 200 Hz, using different types of drones should not affect the validity of the experimental results too much [6].

Finally, regarding statistic measures to evaluate system performance, the most frequently used are accuracy [5], [6], [45], F1-Score [4], false alarm rate [6], [34], inference time [5], confusion matrices [17], [36] and classification error [17], among others.

2.6 Summary

In the current review of the literature, some examples of the misuse of UAV technology were presented. They emphasize the need for a real time implementation that detects and identifies the direction of arrival of an attacking UAV.

Some approaches to UAV detection were presented as well, including radar detection, radio-frequency detection, visual detection, lidar detection and acoustic detection, and the virtues and limitations of each method were explained. The possibility and importance of identifying if a drone carries payload was introduced as well.

As this work focuses on acoustic detection and finding the direction and angle of arrival of an attacking UAV relative to a target, the significance of using acoustic detection was explained, and some works which attempt to locate UAVs were presented. Among the possible techniques to be used, studies using beamforming, TDOA and binary classification were explored.

Finally, different methodologies to process acoustic signals, extract features, implement classification algorithms, deploy the system, and evaluate the results, were mentioned. In the following section, the decision over which methodologies are used in the current project is explained.

CHAPTER 3: METHODOLOGY

3.1 Introduction

The recent advances on UAV technology have led to the democratization and exponential market growth of these devices [24]. Widely used for recreational purposes, UAVs allow several possible applications, but as they can be used for humans' benefit, they also pose a threat, as now it is possible to load them with explosives or weapons [11], [12]. Acoustic sensors combined with machine learning classification algorithms emerge as a possibility for the quick detection of UAVs [13], [56], but detecting the presence of an attacking UAV is just the first step. For an "anti-drone" system to be implemented, there is also the need to locate the UAV threat. The problem addressed by this study is the need for an effective low-cost UAV detection and localization system which works in real time, with replicable results, to demonstrate that an array of acoustic sensors is a viable solution.

The specific purpose of this project is to engineer and validate a model that combines an interconnected array of acoustic sensors with machine learning algorithms to alert in real time about the presence, position, and direction of arrival of a potentially harmful UAV in relationship with a target that needs to be protected. The significance of this project is given by the impact a cost-effective and performant solution for UAV threats would have in an "anti-drone" market which is expected to have a huge expansion in the coming years [18], [19].

This chapter explains the methodology used during the proposed developmental research and provides an in-depth description of the solution's design.

3.2 Research Approach and Scope

The type of research conducted in this project is a developmental research. The product developed is a system which can detect the presence of an approaching UAV, calculate its direction of arrival relative to a target (specifically a range for the position of the UAV when it passes through a microphone array barrier), and alert in real time about this information. The scope of this project included the development and configuration of:

- A set of electronic devices with acoustic sensors and networking capabilities to sense the sound produced by a UAV and communicate to a central server.

- The software, installed in a central server, necessary to classify between background noise and UAV sound, to calculate the tentative direction of arrival of the UAV, and to visualize these data.
- The network protocols to communicate between the components in the system.

The experiments include testing four different types of UAVs both indoors and outdoors. This is expected to be representative of the entire population of amateur drones because the type of UAV should not be a high impact factor to the results of this research due that most amateur drones emit frequencies close to 200Hz [8].

3.3 System Design

The main criteria for designing the system were to reduce the costs (price of all the components in the system) and response time (time that passes between when the UAV enters the restricted zone and when the system displays the alert) as much as possible, without sacrificing too much accuracy. These criteria are considered because the alert must be produced in real time to provide useful information to protect a target, and because the model needs to be cheap to be widely used in the market.

The developed system design consists of three main parts: an array of acoustic sensors, a central server, and the network connection between them. The design of this system including the mentioned elements is shown in Figure 5.

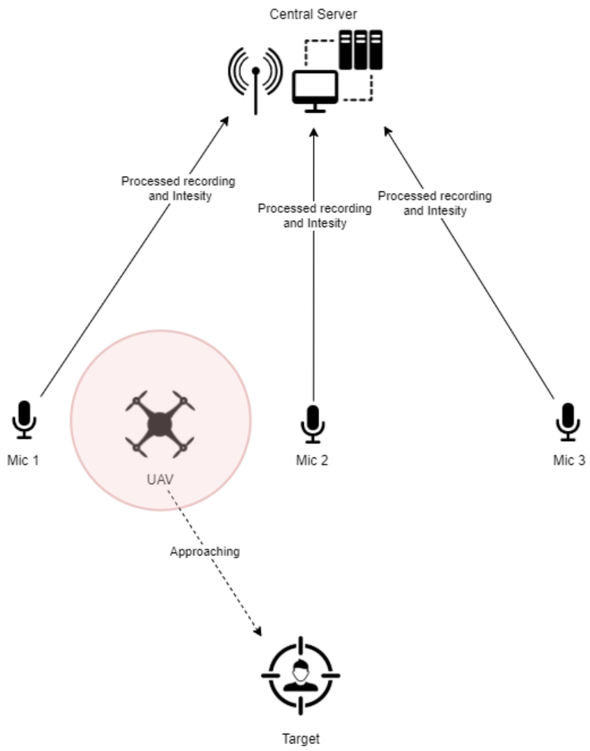

Figure 5. System Design.

The acoustic sensors are designed to be positioned as a defensive barrier between the target and external UAV threats in a way that, when a drone passes between the acoustic sensors, it is recognized as a threat and its position relative to the two closest microphones is predicted based on the difference in sound intensity it produces. The details about this implementation will be explained in the following sections.

3.3.1 Acoustic Sensors

Each node is composed of a single-board computer with integrated networking capability to connect to a standard Wi-Fi network, and a single generic microphone connected to it. This project was tested using a Raspberry Pi [57] 3 Model B V1.2, but any single-board computer with Wi-Fi connectivity and which runs Python 3.7 or superior should be able to run the solution with similar results. The system is designed to work with any generic microphone as well.

As mentioned, the project runs on Python 3.7. When started, it records a sound sample of a fixed duration (which can be configured), then that sample is processed to get either an STFT, Filter Banks or MFCC transformation result. This transformed recording and its metadata are sent to the central server in real time to execute the computations needed. This process is executed in an infinite loop until it is stopped.

Regarding the metadata included in the mentioned transmission, it includes sound intensity, range of detection and geographic position of the acoustic sensor. Range of detection and geographic position are passed by parameters while setting up the node, but the sound intensity of the recording is calculated as the Root Mean Square (RMS) of the signal. For the purpose of this project, intensity is defined as how big the amplitude of the sound signal is, that is the reason for choosing RMS, it is a simple and fast-to-calculate representation of the mean amplitude of the sound wave.

It is worth mentioning that the recorded sample can be stored in the local memory of the single-board computer to be used for future analysis or for training classification algorithms, which is the process explained in section 3.3.2.1.

3.3.1.1 Sound Processing

Each recording captured by the acoustic sensor is stored in an array in memory which contains the sound signal. The sample rate is set at the default value of 44100 Hz, and all recorded sounds are signals with two channels (stereo). As each file has a different length, the sound signals are separated into chunks of a fixed length. Short time samples are more sensitive to noise [7] and long time samples take more time to process, reducing the possibilities for a real time response, so a balance had to be found running different tests. A pre-emphasis filter is applied to have a relatively constant frequency response among different frequency bands [7] using Formula 1 (see section 2.5.1). The frequencies in a signal may vary over time, so to get a more representative depiction of the signal when applying Fourier transform later, the signal is separated in short time frames of 25 ms with a 10 ms stride (15 ms overlap). To reduce spectral leakage, a hamming window [58] is applied over each of the mentioned frames. The next step is to transform the signal into the frequency spectrum, to do that, a Short-Time Fourier-Transform (STFT) [59] is calculated for each short-frame. Finally Filter Banks and Mel-frequency Cepstral Coefficients (MFCCs) are computed. The features are sent in an array to be processed by the central server, and they are the input for machine learning classifiers as well (see section 3.3.2.1). It is considered that Short Time Fourier Transform (STFT) features are better than Mel Frequency Cepstral Coefficients (MFCC) [4], either way, the three feature extraction methods (STFT, Filter Banks and MFCC) are evaluated during experimentation (see chapter 4). A flow chart of this sound transformation process is shown in Figure 6.

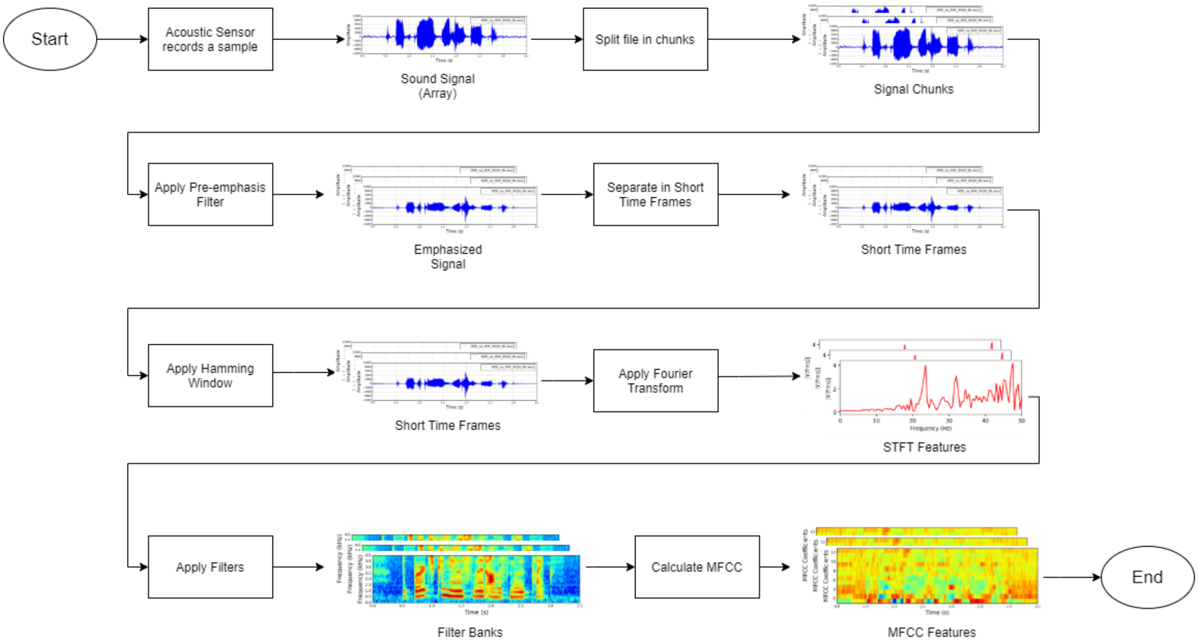

Figure 6: Sound Transformation process

The general structure for this transformation from signal to STFT, Filter Banks or MFCC, the values for the parameters used and the captions of the signal at its different stages included in Figure 6 were extracted from [60] and tested with success in previous unpublished works, for that reason they were chosen for this project.

This sound processing could have been done on the server side, an option that makes sense if expecting the processing to be as fast as possible, but experiments done in the lab have shown that this process is not very computationally expensive. Even cheap nodes as the mentioned Raspberry Pi 3 (which is not the latest model) can handle the computational requirements of this process in real time. This approach, which can be related with the Edge Computing [61] approach, has the main advantage on this project of reducing considerably the network traffic, at the same time that it relieves the central server from all the computational responsibility, allowing it to scale considerably more, that is to say, it is expected to work properly even with a high number of nodes.

3.3.2 Central Server

The central server oversees processing the sound signals sent by the acoustic sensors in real time, predicting the position of the UAV threat based on the parameters received, and displaying and logging the alerts. The central server receives sound features from each signal sample previously processed by the acoustic sensor nodes, and implements machine learning algorithms to identify the presence of a UAV, similar to what was shown in previous studies [1], [4], [17]. It also integrates all the metadata sent by the acoustic nodes (intensity of the signal, node position and detection range), in that way, it can estimate a range for the UAV position.

3.3.2.1 Classifier Training

The module that trains classification algorithms is run separately from the normal execution flow of the system. It takes sound sample files in ".wav" format as an input, it applies the sound processing described on section 3.3.1.1 which generates features in either STFT, Filter Banks or MFCC form, and it uses them to feed different classification algorithms. An 80% of the samples are used for training and the remaining 20% for testing, and they are split using a Stratified Shuffle Split [62] with no re-shuffling (n_splits = 1).

The classification module is prepared to run most of the machine learning algorithms available on the scikit-learn library [63] by just passing the name of the algorithm as a parameter. The options implemented are: (1) k-Nearest Neighbor, (2) Linear Models Classification, (3) Linear Models Multiclass Classification, (4) Decision Trees, (5) Random Forests, (6) Gradient Boosted Regression Trees, (7) Kernelized Support Vector Machines and (8) Neural Networks (Multi-layer Perceptron), (9) Stochastic Gradient Descent (SGD), (10) Gaussian Process Classification (GPC) and (11) Gaussian Naive Bayes (GNB). Even though the system is prepared to work with any of the mentioned classification algorithms, the analysis of each one of them escapes the scope of this project, so only GNB, SVM and Neural Networks are analyzed. The reason for using SVM and Neural Networks (MLP) is because of their simplicity and promising results in previous studies [2], [5], [7], [56], while the reason to use GNB is because it is fast on training and prediction, and has shown good results in previous unpublished studies.

The classification executed is a binary classification, between "uav" or "noise", but the system can train models with multiple classes by configuring a CSV file that contains the list of WAV files and their corresponding label.

Once the execution has finished, the system prints a performance report of the resulting model over the test data, and stores such model for its future use in a pickle file [64].

3.3.2.2 UAV Detection

When the system starts, it loads a pre-existing machine learning model stored in a pickle file. Each set of sound features received from the acoustic sensors are processed by this model which labels the sample either as a “uav” or a “noise”.

The general test scenario is that the sample received has the exact length allowed by the model, so there would be one prediction for each sample, but there is a second test scenario considered which is that the sample is a fixed number of times longer than what a model allows. This scenario is handled with a voting method in which the predicted value needs to represent a certain percentage of the total predictions to be labeled as positive. For example, if a sample of 2.5 seconds is provided to a model trained to handle samples of 0.5 seconds, there will be 5 predictions for that single sample, if 3 of them are labeled as “UAV” and 2 of them are labeled as “noise”, with a criterion of “more than 50%” of the total samples, the result will be that the 2.5 seconds sample is labeled as “UAV” since it represents a 60% of the total samples. This feature is especially useful to avoid false positives as will be shown on Chapter 4.

Finally, all the meta data received from the acoustic sensor is stored on an in-memory dictionary where the status of each one is maintained and updated each time a new prediction is generated. If the prediction is a “uav”, an alert flag is set for that acoustic sensor.

3.3.2.3 Position Prediction

Once all acoustic sensors have been updated, and if any of them is flagged with an alert, then it is time to predict the position of potential threats.

Some methods, like beamforming [49], use the time delay of arrival (TDOA) to calculate the position of a UAV, and they need either a very synchronized clock in each node, or at least 2 acoustic sensors per node (or even more if they want to provide better accuracy), not to mention the complex calculations to estimate the position. In this case the approach is simpler, it is designed to have only one microphone per node and to work asynchronously. The solution predicts a range for the UAV position, which is calculated using the intensity of the signal, and the position and range of the acoustic sensor. When a UAV approaches a covered zone and two acoustic sensors detect it, there is only a limited area where both sensors have coverage, so the UAV should be either located between them or approaching on that direction. Using the intensity of the signal it is possible to reduce the range even more. The advantage of using this approach is a reduction on computational cost and time. Since calculating the TDOA is not necessary, the computation is simpler, and by using a single microphone, the hardware requirements are lower, hence less expensive. An extra advantage is that it can work asynchronously as previously mentioned, simplifying any implementation and maintenance.

Talking about the details of how the mentioned approach was implemented for this project, the core element to consider is the “intensity” of the signal received by each microphone compared with the others. The term “intensity” for this project refers to the amplitude of the signal and must not be confused with “Sound Intensity” which refers to the rate of energy that flows across a unit area. This intensity is calculated as the RMS of the signal (as mentioned in section 3.3.1), but this intensity is not considered in absolute terms, it is relative to the environmental noise already existing. Whenever a UAV approaches a node, it produces a change in the amplitude of the signal recorded by the microphone, so what is relevant in this case is the magnitude of that change, not the absolute value of the signal amplitude. To calculate this, the new intensity received is compared with the median value of the last 50 intensities of signals considered as “noise”, in this way, the change in intensity produced by the UAV is obtained, which for the purpose of this project will be called “intensity change”. Figure 7 graphically explains this calculation.

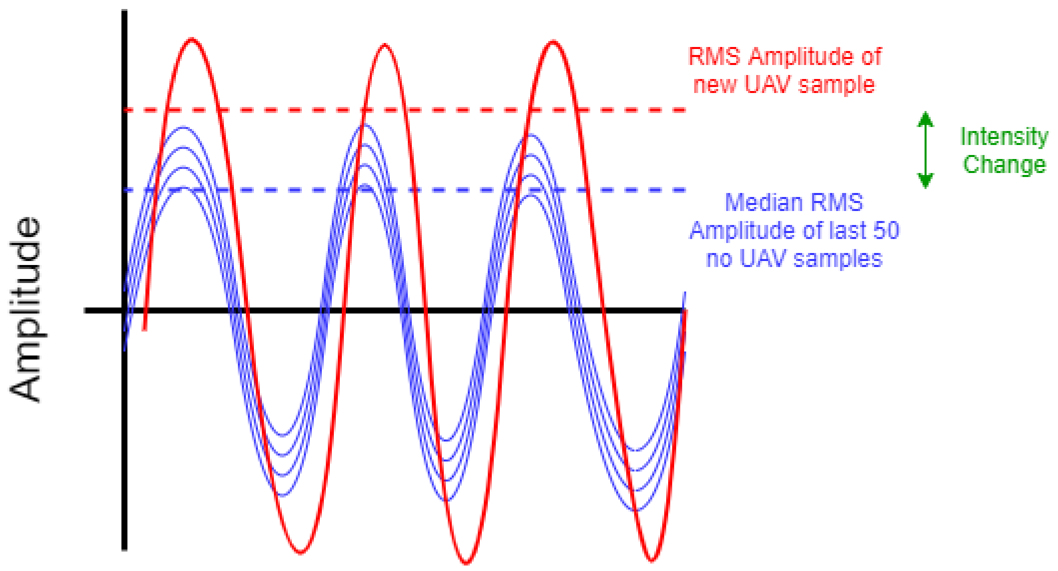

Figure 7: Intensity Change Example

The proposed solution only considers a threat relative to two acoustic sensors, that means that if three acoustic sensors detect a threat, only the two with the higher intensity change will be considered. This situation should not happen normally since the acoustic sensors are designed to be positioned within a distance equivalent to the maximum range they can cover, so the coverage of three acoustic sensors should not overlap normally.

Having two acoustic sensors “A” and “B”, the predicted position is calculated using a proportion between the intensity change in “B” and the whole intensity change A + B, so the calculation for that proportion is:

$$prop_{A,B} = \frac{intensityChangeB}{intensityChangeA + intensityChangeB}$$

Formula 2. Intensity change proportion.

The predicted latitude and longitude are calculated based on the mentioned proportion:

$$predictedLat_{A,B} = lat_A + (lat_B - lat_A) * prop_{A,B}$$

Formula 3. Predicted latitude for two acoustic sensors.

$$predictedLon_{A,B} = lon_A + (lon_B - lon_A) * prop_{A,B}$$

Formula 4. Predicted longitude for two acoustic sensors.

Finally, the predicted latitude and longitude are logged and displayed for the user. The UAV is expected to approach in a direction contained between the predicted latitude and longitude ± an error range which is based on the range of the acoustic sensor. The value for this range of error will be discussed in chapter 4.

3.3.2.4 Visualization

As previously mentioned, the predictions obtained are logged for deeper analysis, but visual information is provided to the user as well in the form of a webpage which displays in real time what is the expected direction of arrival of the UAV. Figures 6 and 7 show how the system displays the presence or absence of UAV threats.

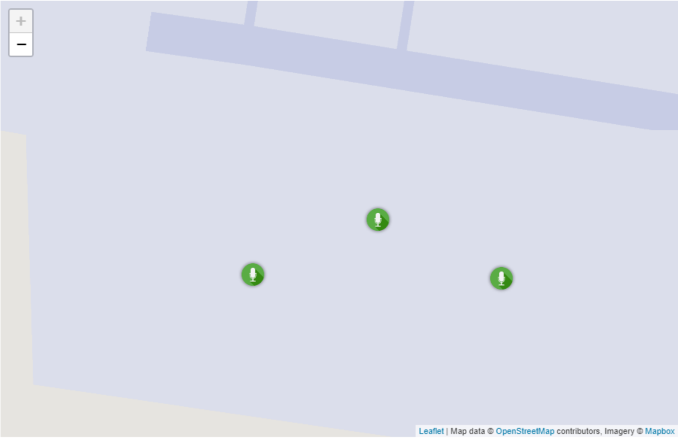

Figure 8: System Screenshot with no UAV detected

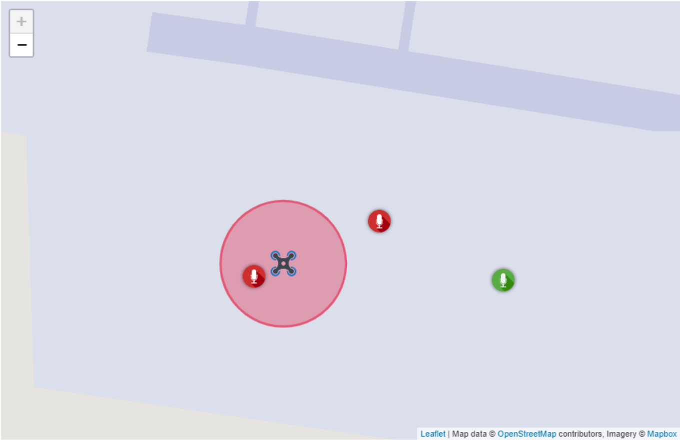

Figure 9: System Screenshot a UAV detected

The web page was developed in plain JavaScript. It displays markers generated using Leaflet [65] over a Mapbox map [66]. The markers are updated in real time by listening to events in a pipeline implemented with the event streaming platform Apache Kafka [67], which is fed by the system implemented in Python. The relevance of using Apache Kafka is that it provides a high throughput “with latencies as low as 2ms” [67], meaning that the result can be shown in real time and that the visualization delay is almost imperceptible.

3.3.3 Network Configuration

Regarding the connection between acoustic sensors and the central server, a previous work [68] has used a local area network (LAN), implementing an access point with Wi-Fi connection and devices using a single-band on 2.4 GHz with the protocol IEEE 802.11 b/g/n.

The same approach has been taken for this project, connecting the nodes to the same Wi-Fi network, and connecting between each other via HTTP requests. The reason for using this approach is that it provided good results in the mentioned previous work, and that the other option would be a wired connection using optic fiber like in [6], but this approach is harder to setup and reduces the possibility to adapt the system to different environments.

Regarding implementation specifications, the position of the access point is irrelevant as long as every node has good quality connection to it, both the personal computer used as a central server and the Raspberry Pi nodes have built-in networking capabilities, and the HTTP configuration is setup for the project using Flask library in Python [69].

3.4 Development

As previously mentioned, there are three basic components that have been developed: the acoustic sensing devices, the software for the central server, and the network protocols to connect them.

Although no formal development methodology (i.e. Extreme Programming, Scrum, Lean, etc.) was used, some tools and concepts from these were. The development process consisted of a flexible iterative prototyping approach with incremental development, in which each element is tested and validated in the lab, and changes are made based on the feedback provided by the research project committee. An approach resembling a Kanban board [70] for tracking pending tasks in the project was used as well.

3.5 Data Collection

The data collection process carried out can be divided in three phases: lab data (phase 1), training data (phase 2), and performance data (phase 3).

The first phase was executed with the purpose of generating a bank of acoustic signals to train a testing version of the machine learning algorithm and help the development of a prototype. The samples consist of sound recordings of two small UAVs flying indoors.

The second phase was executed for training the working version of the system with real world data, so the samples belong to two commercial models of UAVs being flown outdoors with natural noises and voices on the background. These UAVs were flown both unloaded and carrying a payload, to replicate real world scenarios.

The third phase is for evaluating the performance of the final model, so the conditions are similar to phase 2, but this time the system was fully functional and different performance indicators were registered for deeper analysis.

The details about the data collection process and its corresponding analysis are described in chapter 4.

3.6 Evaluation

The data collected during phases 1 and 2 were used for training and fine tuning the machine learning model, and for designing and validating the position estimation algorithm. Accuracy, precision, recall and F1-Score are the metrics used to evaluate the prototype at this stage.

The final evaluation is based on the data collected during phase 3. The metrics used to evaluate the effectiveness of the proposed solution are:

- *False Positives and True Positives:* the proportion of false positives and true positives was calculated considering if the UAV was flying when a prediction was generated.
- *Mean Response Time:* the mean time between when a UAV crossed a certain position and when the prediction was logged by the system.
- *Root Mean Squared Error:* the root mean of the squared errors (or the mean of the absolute values for the errors) between the position predicted by the system and the actual position of the UAV at that given time.
- *Cost summary:* an analysis of the costs of the system components against the protection it can potentially provide.

Referring to research questions (section 1.5), each of the mentioned metrics help answer them in the following way:

- *RQ-1:* Accuracy, precision, cost summary.
- *RQ-2:* Root mean squared error.
- *RQ-3:* Mean response time.
- *RQ-4:* Accuracy, precision, false positives rate, true positives rate, mean response time and root mean squared error associated with the cost summary.

3.7 Reliability and Validity

The main instrument for reliability of the measures is a “Test-Retest Method” [71, p. 224]. The final evaluation test (data collection phase 3) was executed repeatedly and under different conditions, in that way, a correlation between experimental results can be calculated.

About validity, content validity is constructed by using the same measuring and statistical methods that are the state of the art and are repeatedly used throughout the literature, while criterion validity is ensured by statistical analysis, in this case, it is expected that results on scenarios where the UAV is absent should be significantly different from scenarios where the UAV is present.

3.8 Summary

This chapter defined a scope for the project, which is the development and configuration of three main components: a set of acoustic devices, a central server that computes and displays the results, and the network to connect these components. The design for each of these components, and the development methodology for the whole project was explained as well.

The data collection process was divided in three phases, the first and second ones producing data to develop and internally validate the necessary software, and the third one to evaluate if the project meets its goals. The criterion for the success of the project was settled to be the minimization of costs and response time while keeping an acceptable performance, and the specific metrics to evaluate this performance were settled to be accuracy, precision, false alarm rate, true positive rate, mean response time and mean squared error.

CHAPTER 4: EXPERIMENTS AND DATA ANALYSIS

4.1 Introduction

In chapter 3 the resulting design of the solution was explained, but to reach that point, several steps were taken. In the current chapter, the process to arrive to that final design is explained in chronological order, including the failures and successes that forged the way there.

The goal of the project is to provide good UAV detection and localization results, in real time, and with the cheapest possible components, so all the efforts and decisions taken point towards that goal.

The experiments that shaped the project include flying UAVs both indoors at a lab, and outdoors at a park under realistic conditions. Both tests served to find a strong machine learning model, while keeping the implementation simple. Outdoor tests served to test the performance of the solution as well. All the results for these experiments and the deductions taken at each step are explained in the following chapter.

4.2 Equipment

For the current project, each node consists of a single board computer and a microphone connected to it. The single Board computers are Raspberry Pi 3 Model B V1.2 (Figure 10), with a market price of less than 45 USD [72], and the microphones are USB “Zaffiro” (Figure 11) with 2,2 KΩ of impedance, -58 dB ± 3 dB of sensitivity and 30Hz to 16000Hz of response frequency, with a market price of less than 30 USD [73]. The Raspberry Pi were powered using a generic USB Power Bank and the USB ports of two laptops, but any clean power source (a power source that does not introduce noise) is a viable option. The total cost for each node is as low at 75 USD for any person, but for a company the price can significantly be reduced. It is worth mentioning that these are not minimal requirements, less expensive equipment could also provide a solution with the same effectiveness level.

Figure 10: Raspberry Pi 3 Model B V1.2

Figure 11: Microphone Zaffiro

Regarding the UAVs to be flown, for indoor tests this project used a small sized UAV Syma X20P (Figure 12) and a medium sized Syma X5UW (Figure 13), while for outdoor tests the models used were DJI Phantom 4 (Figure 14) and EVO 2 Pro (Figure 15).

Figure 12: Syma X20P

Figure 13: Syma X5UW

Figure 14: DJI Phantom 4

Figure 15: EVO 2 Pro

About the computer that worked as a central server, it was a Laptop Dell Inspiron 15 3000 Series with a processor Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz, 8GB of DDR3 RAM memory and a graphic card NVIDIA GeForce 820M. For training the Machine Learning algorithms, this same computer was used on initial stages, but then a more powerful computer was used to save time. The other computer used was a Dell Alienware M17 R3 with a processor Intel Core i7-10750H (6-Core, 12MB Cache, up to 5.0GHz w/ Turbo Boost 2.0), 16GB of DDR4 RAM Memory, and a graphic card NVIDIA(R) GeForce RTX(TM)2070 8GB GDDR6.

4.3 Phase 1: Lab Data

4.3.1 Initial approaches.

To better understand and justify the model proposed on Chapter 3, first there is need to explain the initial ideas and approaches taken, including those discarded.

Based on the literature reviewed on Chapter 2, it is possible to assume as a fact the feasibility of identifying UAVs with the help of machine learning algorithms and sound recognition techniques, so the main challenge for this project was to find a localization model that complements the mentioned method.

The first approach to locate a UAV was to use the Time Delay of Arrival (TDOA) which was successfully used in previous studies [6]. The general idea to make it perform fast enough to provide real time results with limited equipment was to apply some sort of data reduction on the signal the microphone records. With this idea in mind, a simple code for finding the lag between two signals by using cross correlation was implemented. The microphones were put at around 5 meters apart from each other and a sound sample of 3 seconds was recorded. The timestamp at which the microphones started recording the sample was used to fix the time difference between them, since they were not fully synchronized. The delay between microphones calculated using cross correlation and fixing with the timestamp approach was of 0.118492 seconds, which at a speed of 343.21 m/s (the speed of light) gives a separation of 40.66m, which is clearly not accurate. The conclusion for this experiment is that, to have an accurate estimation, either both clocks should be perfectly synchronized, or both microphones should be connected to the same computer sharing the same clock, which is not viable in an asynchronous and single-microphone-per-node implementation like the one desired. The main problem with synchronism is that it is hard to maintain and scale if several nodes are desired. Other factors that seem to have an impact on the accuracy of this approach are the quality of the signal, the sample rate, the distance, and the noise. This does not mean that TDOA is not a valid approach, in fact under the right conditions it could be the approach which provides the best precision on acoustic localization, the problem is that due to the mentioned constraints it is not the right fit for the current project.

Another approach that was shortly analyzed was to predict the location based on which node detects the UAV first, but since the UAVs start their recording at different times, this is not an accurate estimator, the problem is asynchronism again.

In this way, after analyzing the problem thoroughly, the idea of a prediction based on the sound level or intensity of the sound appeared. The basis of this idea is that when a microphone records and detects a UAV, this UAV produces a change in the intensity of the signal perceived by the microphone. If the UAV is closer to the microphone, then the sound is louder, and the change perceived by the microphone is bigger. Given two microphones with similar characteristics, the one that perceives a bigger change in intensity should have the UAV closer to it, so by a simple proportion analysis, it should be possible to estimate a range for the position of the UAV relative to the mentioned microphones. This approach probably has a bigger error range than other approaches like TDOA, but it is also easier to implement asynchronously and more robust, so it was considered that this is the approach that fits the project better.

4.3.2 Indoor UAV Tests

After defining the localization method, indoor tests started with two models of UAVs (Syma X20P and Syma X5UW), with the goal of defining the detection algorithm.

First, samples of background noise and a Syma X20P flying indoors were collected. The samples were collected using the system as designed for final implementation, which means a Raspberry Pi with a microphone connected to it. The samples consisted of a 10 second recording in ".wav" format. Each sample was manually checked to confirm that none of them were incorrectly labeled and to remove sounds like the UAV landing or crashing. A total of 364 background noise and 92 Syma X20P 10-second samples were recorded.

Some electromagnetic noise was perceived in the recordings, the strength of this noise was different in each microphone but constant on the same microphone for all recordings, so it can be attributed to the microphone and not the system. The samples were kept with this noise since both UAV and background samples had them.

One of the criteria to follow throughout this project is time response reduction, so the first algorithm used was GNB, since it is the fastest one from the three to be analyzed (GNB, SVM and Neural Networks), and the feature type used was MFCC since it is the one that provides the most data compression. For this case and all future cases, 80% of the data is used for training and 20% for testing with a stratified shuffle split. These tests resulted in 3 classifiers trained, with samples of 0.5, 1 and 1.5 seconds (10 seconds samples are split in smaller short time samples). Table 11 shows the results for this configuration. With an accuracy of up to 98.72% on the test set, the results are promising for an initial test. It is possible to observe that the accuracy increases as the sample size increases, which is expected. The short time samples do not show bad results either.

Table 11: Results for GNB with MFCC (Syma X20P1 vs Background noise)

Sample Size	**Accuracy Train**	**Accuracy Test**	**Noise**			**UAV**		
			Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
0.5 sec	0.9667	0.9501	0.97	0.96	0.97	0.86	0.89	0.88
1 sec	0.9704	0.9572	0.98	0.97	0.97	0.89	0.90	0.89
1.5 sec	0.9977	0.9872	0.99	0.99	0.99	0.97	0.96	0.97

At this point it was noticed that even microphones of the same model show different mean amplitudes on average, causing the predictions to be closer to one microphone even when the UAV is not. This could be due to the electromagnetic noise previously mentioned or because the microphones have different gain. To solve the problem, the RMS amplitude of the last 50 recordings is stored for each microphone, then every new RMS amplitude labeled as "UAV" is compared with the median of these last 50 noises for each specific microphone, in that way, even if the microphone has different internal noise or different gain, the results reflect the change in the intensity more evenly. Median was chosen over average because it is resistant to outliers, such as sudden environment noises like cars, screams, etc. It was observed through the UI of the system that this change had a big positive impact on UAV localization.

To expand the previous model, 73 samples of a medium sized UAV (Syma X5UW) were collected using the same criteria and methodology used previously. The new 10-second samples were added to the existing ones, and a new model was trained. Table 12 shows the results for this model. Accuracy was reduced when the new UAV sound was added, but results are still good with up to 91.81% of accuracy on the test set, meaning that the approach works for more than one type of UAV. One thing to observe is that the difference in accuracy between models of different sample size was reduced.

Table 12: Results for GNB with MFCC (Syma X20P1 and Syma X5UW1 vs Background noise)

Sample Size	Accuracy Train	Accuracy Test	Noise			UAV		
			Precision	Recall	F1-Score	Precision	Recall	F1-Score
0.5 sec	0.9109	0.8993	0.91	0.95	0.93	0.88	0.78	0.83
1 sec	0.9177	0.8932	0.90	0.95	0.92	0.87	0.77	0.82
1.5 sec	0.9508	0.9181	0.92	0.96	0.94	0.91	0.82	0.86

Continuing with the construction of the model, it was noticed that some voices were considered positive by the model, so 77 voice sound samples were added to the collection. Adding these voice samples reduced the accuracy considerably as can be seen in Table 13.

As the results were not good enough, it was time to explore other options for the model design. Filter Banks and STFT models were trained at this point. Initial results in Tables 14 and 15 show that accuracy improves using Filter Banks rather than MFCC or STFT, with an accuracy of up to 94.55% on the test set.

These test results also inverted the relationship between accuracy and sample size, probably because short samples have less change over time and provide more amount of data (each 10-second sample generates 20 0.5-second samples but only 6 1.5-second samples). Using short time samples could provide more accuracy, but it also generates a lot more “false positives” as it was observed through the system’s UI.

Table 13: Results for GNB with MFCC (Syma X20P1 and Syma X5UW1 vs Background noise and voices)

Sample Size	Accuracy Train	Accuracy Test	Noise			UAV		
			Precision	Recall	F1-Score	Precision	Recall	F1-Score
0.5 sec	0.9277	0.9096	0.95	0.93	0.94	0.82	0.86	0.84
1 sec	0.9028	0.8712	0.94	0.88	0.91	0.72	0.86	0.78
1.5 sec	0.8810	0.8132	0.89	0.84	0.87	0.64	0.73	0.68

Table 14: Results for GNB with Filter Banks (Syma X20P1 and Syma X5UW1 vs Background noise and voices)

Sample Size	Accuracy Train	Accuracy Test	Noise			UAV		
			Precision	Recall	F1-Score	Precision	Recall	F1-Score
0.5 sec	0.9458	0.9455	0.99	0.94	0.96	0.85	0.97	0.91
1 sec	0.9490	0.9480	0.99	0.93	0.96	0.85	0.98	0.91
1.5 sec	0.9525	0.9437	0.98	0.94	0.96	0.86	0.95	0.90

Table 15: Results for GNB with STFT (Syma X20P1 and Syma X5UW1 vs Background noise and voices)

Sample Size	Accuracy Train	Accuracy Test	Noise			UAV		
			Precision	Recall	F1-Score	Precision	Recall	F1-Score
0.5 sec	0.8660	0.8527	0.84	0.99	0.91	0.95	0.48	0.64
1 sec	0.8705	0.8705	0.85	0.99	0.92	0.96	0.55	0.70
1.5 sec	0.8824	0.8791	0.87	0.98	0.92	0.94	0.60	0.73

As the performance of the model decreased, more data was collected to analyze if by doing that it was possible to improve the model performance in a significant way, especially to try to reduce the number of false positives perceived through the UI. 160 samples of the Syma X20P1 and 156 samples of the Syma X5UW were added to the ones already in the collection for a total of 481 UAV samples and 441 noises (background and voices).

With these new data, the Filter Banks models (Table 16) did not show much improvement, although they were still the most accurate ones. MFCC models (Table 17), on the other hand, showed a big progress, especially on 1.5 second samples. STFT models (Table 18) surprisingly showed a reduction on their accuracy, although their F1-Score for UAV was better, meaning that the accuracy reduction could be due to it adapting better to UAV recognition. Despite the accuracy results improved in general, the false positives problem was not solved, still too many false positives could be observed through the UI.

Table 16: Results for GNB with Filter Banks (Syma X20P1 and Syma X5UW1 vs Background noise and voices – More Data)

Sample Size	Accuracy Train	Accuracy Test	Noise			UAV		
			Precision	Recall	F1-Score	Precision	Recall	F1-Score
0.5 sec	0.9554	0.9572	0.97	0.94	0.95	0.94	0.98	0.96
1 sec	0.9532	0.9555	0.97	0.94	0.95	0.94	0.97	0.96
1.5 sec	0.9516	0.9431	0.94	0.95	0.94	0.95	0.94	0.95

Table 17: Results for GNB with MFCC (Syma X20P1 and Syma X5UW1 vs Background noise and voices – More Data)

Sample Size	Accuracy Train	Accuracy Test	Noise			UAV		
			Precision	Recall	F1-Score	Precision	Recall	F1-Score
0.5s	0.8898	0.8774	0.86	0.89	0.87	0.89	0.87	0.88
1s	0.9162	0.8921	0.87	0.91	0.89	0.91	0.88	0.89
1.5s	0.9451	0.9115	0.90	0.92	0.91	0.93	0.90	0.91

Table 18: Results for GNB with STFT (Syma X20P1 and Syma X5UW1 vs Background noise and voices – More Data)

Sample Size	Accuracy Train	Accuracy Test	Noise			UAV		
			Precision	Recall	F1-Score	Precision	Recall	F1-Score
0.5s	0.7638	0.7611	0.67	0.99	0.80	0.99	0.55	0.71
1s	0.7739	0.7711	0.68	0.99	0.81	0.99	0.57	0.72
1.5s	0.7855	0.7777	0.69	0.98	0.81	0.98	0.59	0.73

Since results with GNB are not good enough, especially considering the number of false positives it provides, it was time to try with more complex Machine Learning algorithms.

SVM models were trained with the existing collection of samples, and it resulted in better accuracy results and a dramatical reduction of the false positives perceived through the UI. Using SVM, MFCC results (Table 19) improved from a maximum of 91.15% of accuracy in the test set to 95.81%, Filter Banks (Table 20) from 95.72% to 98.00%, and STFT (Table 21) surprisingly reduced its accuracy from 77.77% to 73.04%, still being the worst feature type.

SVM with Filter Banks seemed like the best approach at this point, its only setback was that it takes considerably more time to train and predict results. It takes 0.17 seconds in average to generate a prediction, which is 8.5 times more than GNB (0.02 seconds), although it is still a good enough time anyways. Talking about training and prediction time, Filter Banks take more time than MFCC as well, which is expected since MFCC reduces data dimensionality. That same reason may be why Filter Banks work better than MFCC. Since MFCC compresses the information, some useful information for recognizing UAVs can get lost.

It is worth mentioning that unlike GNB, SVM was not used with the default configuration provided by scikit-learn library, the parameters used were C=1.3, kernel='rbf' and gamma='scale'. These parameters were used with success in previous unpublished works, so they were used again in this project.

Table 19: Results for SVM with MFCC (Syma X20P1 and Syma X5UW1 vs Background noise and voices – More Data)

Sample Size	**Accuracy Train**	**Accuracy Test**	**Noise**			**UAV**		
			Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
0.1s	0.9831	0.9366	0.96	0.90	0.93	0.92	0.97	0.94
0.2s	0.9951	0.9537	0.98	0.92	0.95	0.93	0.98	0.96
0.4s	0.9985	0.9581	0.98	0.93	0.96	0.94	0.98	0.96
0.5s	0.9983	0.9544	0.97	0.93	0.95	0.94	0.98	0.96
1s	0.9993	0.9425	0.96	0.91	0.94	0.92	0.97	0.95
1.5s	1.0	0.9331	0.93	0.92	0.93	0.93	0.94	0.94

Table 20: Results for SVM with Filter Banks (Syma X20P1 and Syma X5UW1 vs Background noise and voices – More Data)

Sample Size	**Accuracy Train**	**Accuracy Test**	**Noise**			**UAV**		
			Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
0.1s	0.9954	0.9733	0.99	0.95	0.97	0.96	0.99	0.97
0.2s	0.9976	0.9815	0.99	0.97	0.98	0.97	0.99	0.98
0.4s	0.9984	0.9800	0.99	0.97	0.98	0.97	0.99	0.98
0.5s	0.9986	0.9753	0.98	0.97	0.97	0.97	0.98	0.98
1s	0.9986	0.9794	0.97	0.99	0.98	0.99	0.97	0.98
1.5s	0.9989	0.9693	0.96	0.98	0.97	0.98	0.96	0.97

Table 21: Results for SVM with STFT (Syma X20P1 and Syma X5UW1 vs Background noise and voices – More Data)

Sample Size	**Accuracy Train**	**Accuracy Test**	**Noise**			**UAV**		
			Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
0.1s	0.7258	0.7257	0.64	1.0	0.78	1.0	0.47	0.64
0.2s	0.7240	0.7304	0.64	1.0	0.78	1.0	0.49	0.65
0.4s	0.7261	0.7161	0.63	1.0	0.77	1.0	0.46	0.63
0.5s	0.7242	0.7210	0.63	1.0	0.77	1.0	0.47	0.64
1s	0.7240	0.7147	0.63	1.0	0.77	1.0	0.45	0.62
1.5s	0.7254	0.7019	0.62	0.99	0.76	0.98	0.44	0.60

Even though the application of SVM marked an improvement on the detection method and it reduced the number false positives, they are still observable in the logs. To solve this issue, a voting method was implemented. The voting method means that instead of providing an only prediction for a sample, many predictions are taken for shorter time sub-samples, and the final prediction is the one that represents a certain percentage of the total. For example, having a sample of 2 seconds, it is possible to get 5 sub-samples of 0.4 seconds, meaning 5 predictions.

If 3 of them are labeled as "UAV" and only 2 as "noise", then under a simple majority criterion (>50%) the whole 2 second sample is labeled as "UAV". This method was implemented at this point, and for that reason, now new models with 0.1, 0.2 and 0.4 second sample sizes were trained (Tables 19, 20 and 21). These shorter time samples showed good performance, with accuracy close or even better than the longer ones. About the purpose of reducing the false positive rate, shorter samples by their own generate more false positives than the long ones since even with better precision (less percentage of false positives), more samples generate more false positives in the total count, but if the shorter samples are combined with the mentioned voting method, results improve considerably. Using this approach, 1 second samples with 0.2 second sub-samples showed 0 false positives in the logs on indoor tests.

Considering that the results for indoor flying UAVs are functional, it is time to test the model with real world data of outdoor flying UAVs.

4.4 Phase 2: Training Data

For outdoor testing, the small Syma UAVs were replaced by a DJI Phantom 4 and an EVO 2 Pro as shown in section 4.2. UAV and environment noise samples were taken at McAllister Park, Lafayette, IN, 47904 (Figure 16).

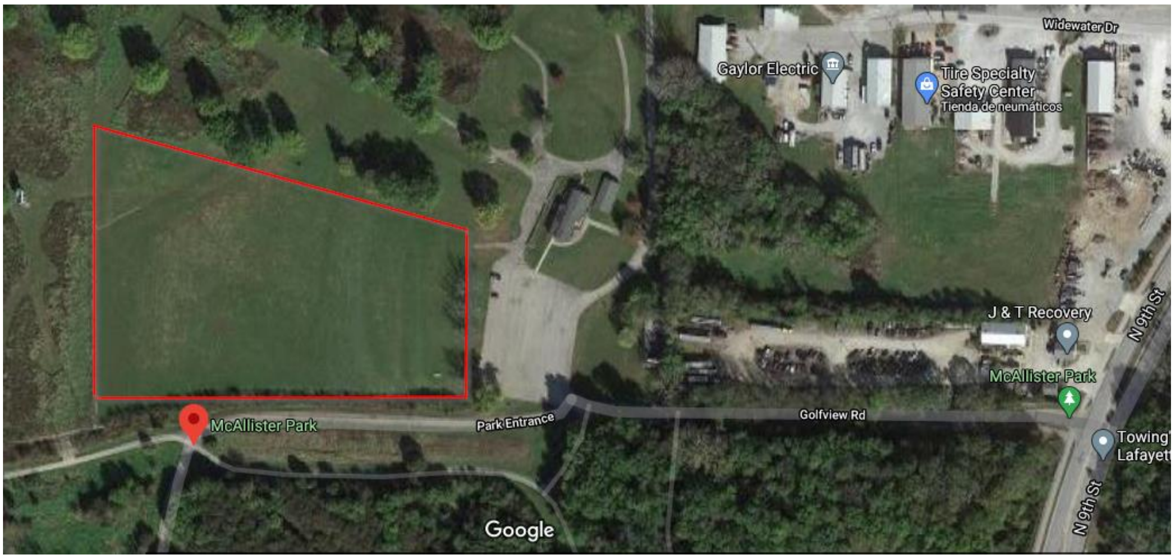

Figure 16: McAllister Park

It is worth mentioning that the samples collected on the first visit to the park had to be discarded because of a strong electromagnetic noise on them. After some research back in the lab, the problem was traced back to the 12V Duracell battery and 500-Watt Energizer Power

Inverter (Figure 17) used to power the Raspberry Pi cards. It was observed that these generate a strong electromagnetic noise on the microphone that does not appear when connected to the wall at 120V as in the lab. To solve the issue, the power source was changed for a generic power bank for charging phones, and two laptops not connected to the Power Inverter, since even connecting the laptop to the Power Inverter and the Raspberry Pi to the laptop, generates this noise.

Figure 17: Battery and Power Inverter used in failed tests.

The final setup of the node including the Raspberry Pi, the microphone and the power bank is shown in Figure 18. It was positioned in a table at around 90cm of height from the floor.

Figure 18: Acoustic Node final setup.

With this new setup, a new visit to McAllister park provided the samples required for training the model. Both mentioned UAV models were recorded flying, with and without a payload to have a wider spectrum of UAV sounds (Figures 19 and 20). The payload attached to the UAVs was a 500ml water bottle which weights around 500 gr.

Figure 19: DJI Phantom 4 flying with payload

Figure 20: EVO 2 Pro flying with payload

Regarding background noise, the environment sound when the data was recorded included a strong bird's noise, voices of the project participants talking, and just a few noises of cars and planes at the distance. These cars and planes noises may not be enough to train the model to ignore them. Wind conditions were low, so no wind other than the wind produced by the UAVs is perceptible in the recordings. Environment temperature was between 7° to 9° Celsius, with no rain.

About details for the system setup, the central server (Figure 21) was the laptop Dell Inspiron 15 mentioned in section 4.2. "Acoustic Sensor 1" (AS1) was powered by the previously mentioned power bank (Figure 18), "Acoustic Sensor 2" (AS2) was powered by the laptop working as a central server (Figure 21), and "Acoustic Sensor 3" (AS3) was powered by a laptop MacBook Pro. The Wi-Fi network was set up using a mobile device (Motorola G7 Power) on Wi-Fi hotspot mode.

Figure 21: Laptop used as the central server connected to Acoustic Sensor 2.

After collecting the samples, they were manually labeled in the same way as with the indoors data. The samples collected on AS1 were very clear, the ones from AS2 had a small electromagnetic noise, and AS3 had a bit more of electromagnetic noise, but all recordings were good enough to work with them. The final number of 10-second samples available for training the model was as described on Table 22.

Table 22: Final Number of samples collected outdoors by sample type.

Sample Type	**Number of Samples**	**Total Time**
Background noise	591	98.50 minutes
Loaded DJI Phantom 4	343	57.16 minutes
Unloaded DJI Phantom 4	302	50.33 minutes
Loaded EVO 2 Pro	297	49.50 minutes
Unloaded EVO 2 Pro	290	48.33 minutes
Total	1823	303.83 minutes

Having the database of samples labeled and cleaned, it is time to train the UAV classification models. From indoor tests it was observed that Filter Banks outperforms the other two feature types analyzed (STFT and MFCC), so from this point on, it was the only feature type used.

GNB (Table 23) and SVM (Table 24) models were trained with different sample sizes, and the first observation that can be made is that accuracy on the test set was reduced considerably from what was achieved during indoor tests for both cases. On GNB, the maximum accuracy went down from 95.72% to 84.09%, while SVM went from 98.15% to 87.61%. The F1-Score of "noise" class was particularly affected, being reduced from values of 0.94 and 0.98, to only between 0.48 to 0.78, meaning that these models find it hard to detect background noise, predicting most of the samples as UAV, which would generate a great number of false positives, one of the main problems throughout the whole project.

Table 23: Results for GNB with Filter Banks (EVO 2 Pro and DJI Phantom 4 vs Background noise)

Sample Size	**Accuracy Train**	**Accuracy Test**	**Noise**			**UAV**		
			Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
0.1s	0.7355	0.7337	0.65	0.38	0.48	0.75	0.90	0.82
0.2s	0.7820	0.7824	0.78	0.46	0.58	0.78	0.94	0.85
0.5s	0.8404	0.8409	0.84	0.63	0.72	0.84	0.94	0.89
1s	0.7942	0.7770	0.60	0.95	0.73	0.97	0.69	0.81
1.5s	0.6673	0.6673	0.49	0.95	0.65	0.96	0.53	0.68

Table 24: Results for SVM with Filter Banks (EVO 2 Pro and DJI Phantom 4 vs Background noise)

Sample Size	**Accuracy Train**	**Accuracy Test**	**Noise**			**UAV**		
			Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
0.1s	0.9737	0.8749	0.89	0.70	0.78	0.87	0.96	0.91
0.2s	0.9860	0.8761	0.94	0.66	0.78	0.86	0.98	0.91
0.5s	0.9844	0.8371	0.94	0.53	0.68	0.81	0.98	0.89
1s	0.9907	0.8187	0.88	0.51	0.65	0.80	0.97	0.88
1.5s	0.9963	0.8277	0.88	0.54	0.67	0.81	0.96	0.88

Since GNB and SVM models did not provide good enough results with outdoor data, a new model was tested. Neural Networks have consistently been depicted as the most promising model for UAV classification [7], [14], [48], [56], so it was the model chosen to continue working. It is not in the scope of the project to find the best neural network model to classify UAV sounds, so the simple Multi-layer Perceptron (MLP) classifier provided by scikit-learn library [74] was used, with “Random State” variable set to 0 and only testing the value of alpha, which as shown in Table 25 finds its ideal value when alpha = 0.1.

MLP takes way less than SVM to train and performs better, with an accuracy of up to 95.38% on the test set and an F1-Score of up to 0.93 for “noise” samples and 0.97 for “UAV”. Table 26 shows that short samples of 0.1 seconds perform better than the longer ones, again maybe the reason is that they provide more amount of data for training. This is the best model found for real world data, so it is the one to proceed with.

Table 25: Results for MLP with Filter Banks on 0.1 second samples (EVO 2 Pro and DJI Phantom 4 vs Background noise)

Alpha	**Accuracy Train**	**Accuracy Test**	**Noise**			**UAV**		
			Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
0.0001	0.9974	0.9472	0.94	0.90	0.92	0.95	0.97	0.96
0.001	0.9961	0.9448	0.92	0.91	0.91	0.96	0.96	0.96
0.01	0.9940	0.9436	0.92	0.90	0.91	0.95	0.96	0.96
0.1	0.9887	0.9538	0.93	0.93	0.93	0.96	0.97	0.97
1	0.9569	0.9346	0.94	0.85	0.89	0.93	0.97	0.95

Table 26: Results for MLP with Filter Banks and alpha = 0.1 (EVO 2 Pro and DJI Phantom 4 vs Background noise)

Alpha	Accuracy Train	Accuracy Test	Noise			UAV		
			Precision	Recall	F1-Score	Precision	Recall	F1-Score
0.1s	0.9887	0.9538	0.93	0.93	0.93	0.96	0.97	0.97
0.2s	0.9857	0.8664	0.79	0.79	0.79	0.90	0.90	0.90
0.5s	1.000	0.8627	0.81	0.75	0.78	0.88	0.92	0.90
1s	0.9998	0.8206	0.70	0.77	0.74	0.89	0.84	0.86
1.5s	0.9978	0.8259	0.71	0.79	0.75	0.89	0.84	0.87

4.5 Phase 3: Performance Data

For the final experiments, four tests were made. The first and second tests were to check the performance of the system on UAV detection, and the false positive ratio (which was a problem throughout the project), while the third and fourth tests were to analyze the performance of the position prediction algorithm, although the data generated by these last two were used as well for the same purposes as the first and second ones.

The conditions explained in section 4.3 were repeated. A DJI Phantom 4 and an EVO 2 Pro were flown at McAllister park, with the same setup previously used. Samples of 1 second were recorded and processed using an MLP Neural Networks model trained with sample size of 0.1 seconds, feature type being Filter Banks, and alpha being 0.1, since it is the best model until this moment. 1-second samples on a 0.1 second model means that for each sample, 10 predictions are obtained. The approval criterion was set at 60% or more, meaning that at least 6 predictions must be "UAV" for the recording to be labeled as a UAV. Even though the system was trained with the UAVs carrying and not carrying payload, for these performance tests the UAVs did not carry any payload.

There are several variables which can affect a real-world test, but it is not viable to consider all of them in this project, that would add too much complexity to the experiment. For simplicity, these variables are documented but were not considered during the experiments:

- Noise Level: during experiments, background noise was similar as described in section 4.3, that means low wind conditions, strong birds' noise in the background, voices of the experiment supervisors talking, and just some noises of planes and cars at a distance.

- UAV Speed: When the speed of the UAV changes, it produces a different sound. The speed of approach of the UAV was not considered nor measured in the experiment. Although it can be mentioned that the maximum speed of both DJI Phantom 4 and EVO 2 Pro is 20 m/s, this speed was never reached. As a reference, it can be estimated that the speed was somewhere between 10 m/s and 20 m/s, but it did not remain constant.
- UAV Height: The UAV flying too low or too high can have an impact on the detection performance. The height of the UAV oscillated between 4 and 5 meters of height during the experiments.
- Acoustic Sensor Height: The height of the acoustic sensor is another variable that can have an impact on the detection range. As the acoustic sensors were kept on a table, they were at around 90cm of height from the floor.

Talking about the details of the experiment layout, the position of each element can be seen in Figure 22. The acoustic sensors (AS) are separated 19.2 meters between each other, with a position marker in the middle of them. This distance is an estimation of the maximum range at which it was observed that the UAV could be recognized, because as mentioned on section 3.3.2.3, the sensors should be positioned at the maximum distance the microphones can cover, since at least two acoustic sensors have to recognize the UAV to predict a position between them.

The system works with geographic coordinates, but to simplify the analysis, the positions were set from 0m (position of the AS1) to 38.4m (position of the AS3), from left to right.

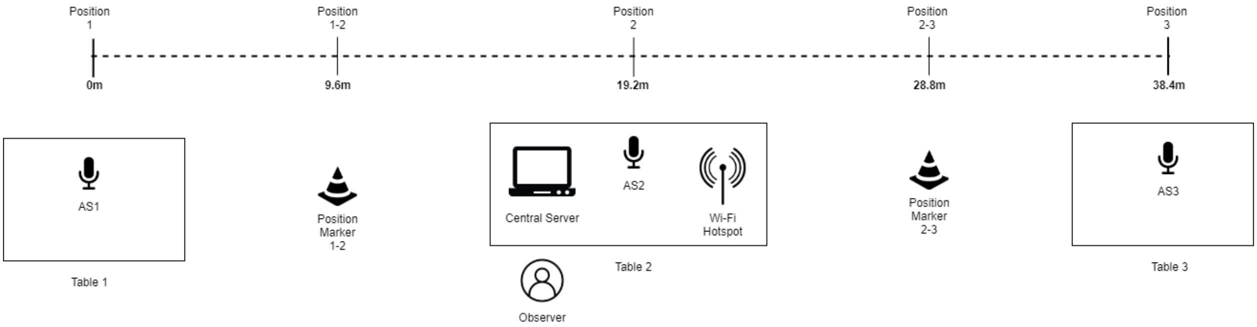

Figure 22: Outdoor tests layout

4.5.1 UAV Detection Performance

For the first two tests, the DJI Phantom 4 was flown around the nodes, with no particular pattern. The observer could see if it was being detected, and if the position shown through the UI was correct or if it was showing too many false positives. Every detection and position prediction were logged for their posterior analysis, and by hearing the recordings saved, it was possible to analyze exactly at what moments the UAV was flying. Since tests 3 and 4 also collected these data, they were considered in the analysis as well. Results for the four tests are shown in table 27.

Table 27: UAV detection performance

	Test 1	**Test 2**	**Test 3**	**Test 4**	**Total**
UAV Type	Phantom 4	Phantom 4	Phantom 4	Evo 2	-
Avg Prediction Time	0.0029s	0.0038s	0.0057s	0.0034s	-
AS1 Samples	180	586	774	434	1974
AS2 Samples	250	673	678	375	1976
AS3 Samples	55	503	586	-	1144
Total Samples	485	1762	2038	809	5094
Total Positives (“UAV”)	263	406	1039	385	2093
Total Negatives (“noise”)	222	1356	999	424	3001
True Positives	242	309	1015	372	1938
False Positives	21	97	24	13	155

These results show that the average prediction time was between 0.0029 seconds and 0.0057 seconds. The variance on the prediction time could have been caused by other processes running in the local server, but what is important to mention is that these values meet the “real time prediction” expectation that was settled.

It can be observed that not all acoustic sensors recorded the same number of samples on each test scenario. There are multiple potential reasons for this to happen, but since AS2 is consistently the one that provides the most samples, this can be attributed either to the power source (Raspberry Pi works slower with lower voltage) or most probably to the proximity to the Wi-Fi hotspot, since AS2 was positioned in the same table as the phone generating the network.

For the way in which the system is designed, it was not possible to determine the number of false negatives and true negatives, which are necessary to calculate the true positive rate and false positive rate, but some conclusions can be made from the proportions. Being the number of false positives a constant problem throughout the project, having only 7.41% of false positives (155 over 2093 total positives) and 92.59% of true positives (1938 over 2093 total positives) is a promising result. Even this 7.41% could be improved since it was observed that most of the false positives happen over a reduced period, for example for test 1, 20 out of the 21 false positives happen over a period of 64 seconds, meaning that something could have happened at that moment, like the sound of a plane or a car passing by, or the system not updating properly because of a bad network connection.

As mentioned, it is not possible to count the number of false negatives and true negatives, the reason for this that one microphone not identifying the UAV can be caused by different reasons, like the UAV being out of range for example, but for the periods the UAV was flying, an alert was sent each 1.1 seconds in average, meaning that the system was alerting about the presence of the UAV most of the time it was there, and the number of false negatives should be low.

4.5.2 Position Prediction performance

As previously mentioned, the third and fourth tests were made for analyzing the performance of the position prediction algorithm. The third test was done using the DJI Phantom 4, while the fourth test was done using the EVO 2 Pro. Each of these tests could be split in two parts based on the flight pattern used: a perpendicular flight scenario and a horizontal flight scenario.

In both the perpendicular and horizontal test scenarios, the flight of the UAV is not a perfect line, it can have a deviation which introduces some error to the sample. To log the time at which the UAV passes over an acoustic sensor or one of the middle points, the observer (positioned in the middle as Figure 22 shows) manually presses a button on the central server to log a timestamp of that exact moment. This approach was chosen because of its simplicity and the lack of resources for a more sophisticated implementation, but it is acknowledged that it introduces some error in the result since it is affected by the time of reaction of the observer and his sight perspective relative to the positions marked.

4.5.2.1 Perpendicular flight tests

The perpendicular flight test scenario consists of the UAV approaching in a direct line to one of the positions marked, passing forth and back on the same line as shown in Figure 23. The UAV starts at around 19.2m away from the target, but this is just an estimation, the distance is not precise in each test (for that reason the UAVs are not aligned in Figure 23).

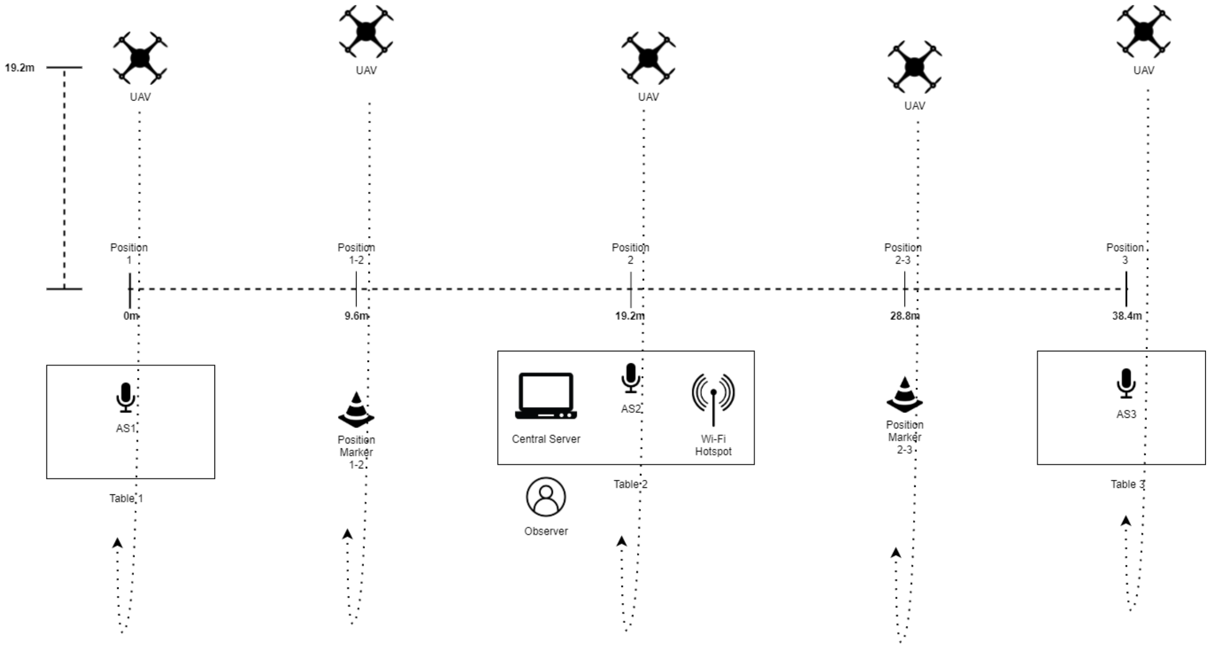

Figure 23: Outdoors perpendicular flight test scenario

In Tables 28 and 30, the results are shown by “Closer Prediction Time”, which means that the log corresponds to the first log observed after the UAV passes over the position marked. Considering samples of 1 second plus the time the system takes to predict the result, which is below 0.01 seconds, the first log observed after the UAV passes over the position marked should be at least 1.01 seconds after, but this is not always the case. The reason for this is that the observer manually logs the time, so an error of ± 1 second can happen. If there was a delay on the network connection, it can affect the time as well.

In Tables 29 and 31, the results are shown by “Closer Position”, meaning that the log corresponds to the one with the closer position to the position marked, under reasonable time conditions (less than 7 seconds). The closer position criterion can improve the accuracy of the results because maybe the system provided a correct prediction, but it did not log it on time for some external factor like the previously mentioned ones.

When running the tests with EVO 2 Pro, one of the nodes ran out of battery, so the AS3 was removed. For that reason, Tables 30 and 31 do not refer to Acoustic Sensor 3.

Table 28: DJI Phantom 4 position prediction results on perpendicular flight by closer prediction time.

Position	Real position	Real time	Closer Prediction Time	Predicted Position	Time difference	Position difference
1	0.00m	16:20:59.15	16:21:00.33	0.00m	1.18s	0.00m
1	0.00m	16:21:24.75	16:21:25.83	0.00m	1.08s	0.00m
1	0.00m	16:22:23.17	16:22:24.04	0.00m	0.88s	0.00m
1	0.00m	16:22:39.51	16:22:41.47	6.18m	1.96s	6.18m
1	0.00m	16:22:53.66	16:22:55.47	0.00m	1.81s	0.00m
1-2	9.60m	16:20:59.15	16:21:00.33	2.25m	1.18s	-7.35m
1-2	9.60m	16:21:24.75	16:21:25.85	1.36m	1.10s	-8.24m
1-2	9.60m	16:22:23.17	16:22:25.91	2.74m	2.74s	-6.86m
1-2	9.60m	16:22:39.51	16:22:41.47	6.18m	1.96s	-3.42m
1-2	9.60m	16:22:53.66	16:22:55.48	2.83m	1.81s	-6.77m
1-2	9.60m	16:23:38.71	16:23:39.82	8.04m	1.12s	-1.56m
2	19.20m	16:24:08.42	16:24:10.68	5.50m	2.26s	-13.70m
2	19.20m	16:24:24.84	16:24:26.63	2.25m	1.79s	-16.95m
2	19.20m	16:24:35.88	16:24:36.58	19.20m	0.70s	0.00m
2	19.20m	16:24:48.66	16:24:50.53	8.11m	1.87s	-11.09m
2	19.20m	16:25:01.61	16:25:02.10	19.20m	0.50s	0.00m
2-3	28.80m	16:25:30.36	16:25:31.99	19.20m	1.63s	-9.60m
2-3	28.80m	16:25:43.05	16:25:45.80	19.20m	2.75s	-9.60m
2-3	28.80m	16:26:00.13	16:26:02.91	19.20m	2.78s	-9.60m
2-3	28.80m	16:26:13.24	16:26:14.23	17.89m	0.98s	-10.91m
2-3	28.80m	16:26:26.76	16:26:27.64	19.62m	0.88s	-9.18m
3	38.40m	16:26:58.44	16:26:59.50	34.50m	1.07s	-3.90m
3	38.40m	16:27:10.63	16:27:12.63	30.68m	2.00s	-7.72m
3	38.40m	16:27:20.02	16:27:22.33	28.72m	2.31s	-9.68m
3	38.40m	16:27:32.38	16:27:33.00	34.22m	1.62s	-4.18m
3	38.40m	16:27:41.81	16:27:43.31	37.44m	1.49s	-0.96m

Table 29: DJI Phantom 4 position prediction results on perpendicular flight by closer position.

Position	Real position	Real time	Closer position	Time at closer position	Time difference	Position difference
1	0.00m	16:20:59.15	0.00m	16:21:00.33	1.18s	0.00m
1	0.00m	16:21:24.75	0.00m	16:21:25.83	1.08s	0.00m
1	0.00m	16:22:23.17	0.00m	16:22:24.04	0.88s	0.00m
1	0.00m	16:22:39.51	0.79m	16:22:43.53	4.01s	0.79m
1	0.00m	16:22:53.66	0.00m	16:22:55.47	1.81s	0.00m
1-2	9.60m	16:20:59.15	2.25m	16:21:00.33	1.18s	-7.35m
1-2	9.60m	16:21:24.75	7.62m	16:21:27.63	2.88s	-1.98m
1-2	9.60m	16:22:23.17	9.29m	16:22:27.77	4.60s	-0.31m
1-2	9.60m	16:22:39.51	6.18m	16:22:41.47	1.96s	-3.42m
1-2	9.60m	16:22:53.66	7.38m	16:22:57.21	3.55s	-2.22m
1-2	9.60m	16:23:38.71	8.04m	16:23:39.82	1.12s	-1.56m
2	19.20m	16:24:08.42	19.20m	16:24:12.40	3.98s	0.00m
2	19.20m	16:24:24.84	10.18m	16:24:28.72	3.87s	-9.02m
2	19.20m	16:24:35.88	19.20m	16:24:36.58	0.70s	0.00m
2	19.20m	16:24:48.66	19.20m	16:24:54.66	6.00s	0.00m
2	19.20m	16:25:01.61	19.20m	16:25:02.10	0.50s	0.00m
2-3	28.80m	16:25:30.36	23.39m	16:25:33.90	3.54s	-5.41m
2-3	28.80m	16:25:43.05	21.28m	16:25:49.82	6.77s	-7.52m
2-3	28.80m	16:26:00.13	19.20m	16:26:02.91	2.78s	-9.60m
2-3	28.80m	16:26:13.24	19.20m	16:26:17.80	4.56s	-9.60m
2-3	28.80m	16:26:26.76	19.62m	16:26:27.64	0.88s	-9.18m
3	38.40m	16:26:58.44	38.40m	16:27:03.39	4.95s	0.00m
3	38.40m	16:27:10.63	30.68m	16:27:12.63	2.00s	-7.72m
3	38.40m	16:27:20.02	28.72m	16:27:22.33	2.31s	-9.68m
3	38.40m	16:27:32.38	34.22m	16:27:33.00	1.62s	-4.18m
3	38.40m	16:27:41.81	37.44m	16:27:43.31	1.49s	-0.96m

Table 30: EVO 2 Pro position prediction results on perpendicular flight by closer prediction time.

Position	**Real position**	**Real time**	**Closer Prediction Time**	**Predicted Position**	**Time difference**	**Position difference**
2	19.20m	16:53:45.99	16:53:46.57	19.20m	0.58s	0.00m
2	19.20m	16:53:56.61	16:53:57.45	18.95m	0.84s	-0.25m
2	19.20m	16:54:16.48	16:54:17.32	19.20m	0.84s	0.00m
2	19.20m	16:54:23.33	16:54:23.48	14.08m	0.14s	-5.12m
2	19.20m	16:54:38.70	16:54:41.35	19.20m	2.65s	0.00m
2	19.20m	16:54:44.82	16:54:45.24	19.20m	0.42s	0.00m
2	19.20m	16:55:00.56	16:55:03.54	19.20m	2.99s	0.00m
2	19.20m	16:55:10.00	16:55:11.90	19.20m	0.91s	0.00m
1-2	9.60m	16:55:40.91	16:55:42.09	6.47m	1.18s	-3.13m
1-2	9.60m	16:55:50.22	16:55:51.13	9.58m	0.91s	-0.02m
1-2	9.60m	16:56:07.78	16:56:11.70	9.42m	3.92s	-0.18m
1-2	9.60m	16:56:15.49	16:56:17.49	12.79m	2.00s	3.19m
1-2	9.60m	16:56:31.30	16:56:32.85	12.37m	1.55s	2.77m
1-2	9.60m	16:56:40.27	16:56:41.24	9.93m	0.96s	0.33m
1-2	9.60m	16:56:50.82	16:56:51.00	0.00m	0.18s	-9.60m
1	0.00m	16:57:10.02	16:57:11.31	0.00m	1.29s	0.00m
1	0.00m	16:57:20.40	16:57:24.03	0.00m	3.63s	0.00m
1	0.00m	16:57:28.55	16:57:30.13	0.00m	1.58s	0.00m
1	0.00m	16:57:37.32	16:57:40.34	0.00m	3.02s	0.00m
1	0.00m	16:57:44.51	16:57:47.83	0.00m	3.33s	0.00m
1	0.00m	16:58:01.44	16:58:02.99	0.00m	1.55s	0.00m
1	0.00m	16:58:11.39	16:58:12.99	0.00m	1.60s	0.00m
1	0.00m	16:58:31.49	16:58:32.08	0.00m	0.59s	0.00m
1	0.00m	16:58:49.01	16:58:50.23	0.00m	1.22s	0.00m
1	0.00m	16:59:06.10	16:59:07.14	0.00m	1.04s	0.00m

Table 31: EVO 2 Pro position prediction results on perpendicular flight by closer position.

Position	Real position	Real time	Closer position	Time at closer position	Time difference	Position difference
2	19.20m	16:53:45.99	19.20m	16:53:46.57	0.58s	0.0m
2	19.20m	16:53:56.61	19.20m	16:53:58.13	1.53s	0.0m
2	19.20m	16:54:16.48	19.20m	16:54:17.32	0.84s	0.0m
2	19.20m	16:54:23.33	19.20m	16:54:27.87	4.53s	0.0m
2	19.20m	16:54:38.70	19.20m	16:54:41.35	2.65s	0.00m
2	19.20m	16:54:44.82	19.20m	16:54:45.24	0.42s	0.00m
2	19.20m	16:55:00.56	19.20m	16:55:03.54	2.99s	0.00m
2	19.20m	16:55:10.00	19.20m	16:55:11.90	0.91s	0.00m
1-2	9.60m	16:55:40.91	10.69m	16:55:43.10	2.19s	1.09m
1-2	9.60m	16:55:50.22	9.58m	16:55:51.13	0.91s	-0.02m
1-2	9.60m	16:56:07.78	9.42m	16:56:11.70	3.92s	-0.18m
1-2	9.60m	16:56:15.49	12.79m	16:56:17.49	2.00s	3.19m
1-2	9.60m	16:56:31.30	12.37m	16:56:32.85	1.55s	2.77m
1-2	9.60m	16:56:40.27	9.93m	16:56:41.24	0.96s	0.33m
1-2	9.60m	16:56:50.82	8.94m	16:56:54.52	3.70s	-0.66m
1	0.00m	16:57:10.02	0.00m	16:57:11.31	1.29s	0.00m
1	0.00m	16:57:20.40	0.00m	16:57:24.03	3.63s	0.00m
1	0.00m	16:57:28.55	0.00m	16:57:30.13	1.58s	0.00m
1	0.00m	16:57:37.32	0.00m	16:57:40.34	3.02s	0.00m
1	0.00m	16:57:44.51	0.00m	16:57:47.83	3.33s	0.00m
1	0.00m	16:58:01.44	0.00m	16:58:02.99	1.55s	0.00m
1	0.00m	16:58:11.39	0.00m	16:58:12.99	1.60s	0.00m
1	0.00m	16:58:31.49	0.00m	16:58:32.08	0.59s	0.00m
1	0.00m	16:58:49.01	0.00m	16:58:50.23	1.22s	0.00m
1	0.00m	16:59:06.10	0.00m	16:59:07.14	1.04s	0.00m

Table 32 shows statistics for all perpendicular flight scenarios. All position errors are calculated based on their absolute value, so they represent the Root Mean Squared Error (RMSE). It can also be observed that a distinction was made between the positions at acoustic sensors and the positions between them (“middles”). The reason for this is that the algorithm works with information of at least two acoustic sensors. If only one acoustic sensor detects a

UAV, then the system has no way to predict the position but to just set the acoustic sensor position as the predicted position. For that reason, the intermediate positions between acoustic sensors are more interesting, because an actual calculation is made.

Talking about specific results for the experiments made, it can be observed that the first prediction happens after 1.59 seconds in average for DJI Phantom 4, and 1.56 seconds for EVO 2 Pro, which are results fast enough to consider them as real time predictions. Regarding precision, for DJI Phantom 4 the average error is 6.06m, but if the prediction time is delayed to 2.70 seconds, the error is reduced almost a half to 3.48m. For EVO 2 Pro, the results are even better, with an average error or 0.98m, and 0.33m if just waiting 0.38 seconds more (1.95s), meaning that EVO 2 Pro produces a more recognizable sound for the system. As mentioned, the positions between sensors are of particular interest, and the error range in those positions goes from 7.55m to just 1.18m. This difference can be attributed to AS3, since it is observed that it is consistently the node with the worst results, and when removed from the tests with the EVO 2 Pro, the error was reduced considerably. It can be assumed that AS3 was affected by some factor like internal or external noise, or network connectivity issues. As a conclusion for these experiments, the position errors show that, even though the method has room for improvement, the response time and the position error are good enough for practical applications.

Table 32: Statistics for Perpendicular flight scenario

	DJI Phantom 4		**EVO 2 Pro**	
	By Closer Prediction Time	**By Closer Position**	**By Closer Prediction Time**	**By Closer Position**
Avg Time Difference	1.59s	2.70s	1.56s	1.94s
Max Time Difference	2.78s	6.77s	3.92s	4.53s
Avg Error	6.06m	3.48m	0.98m	0.33m
Avg Error at AS1	1.24m	0.79m	0.00m	0.00m
Avg Error at 1-2	5.70m	2.81m	2.75m	1.18m
Avg Error at AS2	8.35m	1.80m	0.67m	0.00m
Avg Error at 2-3	9.78m	8.26m	-	-
Avg Error at AS3	5.29m	4.50m	-	-
Avg Error Only AS	5.31m	2.16m	0.30m	0.00m
Avg Error Only middles	7.55m	5.29m	2.75m	1.18m

4.5.2.1 Horizontal flight tests

In the horizontal flight test scenario, the UAV flies over the nodes from left to right, and back from right to left, as shown in Figure 24. The purpose of this is that, since the system only calculates a position between nodes, this flight pattern can be particularly difficult for it to predict in real time, introducing more error than a perpendicular flight.

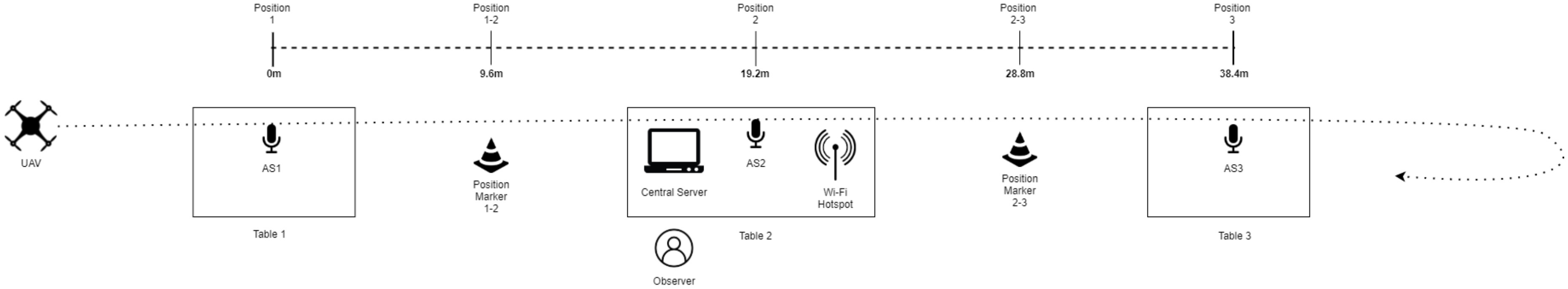

Figure 24: Outdoors horizontal flight test scenario

Results by “Closer Prediction Time” are shown in Tables 33 and 35, and results by “Closer Position Time” are shown in Tables 34 and 36. The criteria and testing conditions for these are the same as explained in section 4.4.2.1. One detail to mention is that in these experiments, only the positions between acoustic sensors were considered, because those are the most relevant ones to evaluate the prediction algorithm.

Table 33: DJI Phantom 4 position prediction results on horizontal flight by closer prediction time.

Position	Real position	Real time	Closer Prediction Time	Predicted Position	Time difference	Position difference
2-3	28.80m	16:33:00.05	16:33:01.21	19.20m	1.16s	-9.60m
1-2	9.60m	16:33:09.80	16:33:10.24	14.85m	0.44s	5.25m
1-2	9.60m	16:33:19.60	16:33:20.84	0.64m	1.24s	-8.96m
2-3	28.80m	16:33:29.12	16:33:31.40	37.97m	2.27s	9.17m
2-3	28.80m	16:33:39.79	16:33:40.69	29.55m	0.89s	0.75m
1-2	9.60m	16:33:49.60	16:33:51.88	19.20m	2.28s	9.60m
1-2	9.60m	16:34:00.44	16:34:01.72	9.85m	1.28s	0.25m
2-3	28.80m	16:34:08.18	16:34:09.67	19.20m	1.49s	-9.60m
2-3	28.80m	16:34:15.49	16:34:17.40	24.70m	1.91s	-4.10m
1-2	9.60m	16:34:23.66	16:34:26.85	16.75m	3.19s	7.15m
1-2	9.60m	16:34:35.82	16:34:36.99	3.12m	1.17s	-6.48m
2-3	28.80m	16:34:43.82	16:34:45.78	31.14m	1.96s	2.34m
2-3	28.80m	16:34:59.28	16:35:00.11	36.71m	0.82s	7.91m
1-2	9.6.m	16:35:12.27	16:35:13.16	11.05m	0.89s	1.45m
1-2	9.60m	16:35:29.01	16:35:31.72	4.71m	2.71s	-4.89m
2-3	28.80m	16:35:36.26	16:35:38.19	35.89m	1.94s	7.09m
2-3	28.80m	16:35:48.21	16:35:49.32	32.95m	1.10s	4.15m
1-2	9.60m	16:35:56.78	16:35:57.95	11.00m	1.17s	1.40m

Table 34: DJI Phantom 4 position prediction results on horizontal flight by closer position.

Position	Real position	Real time	Closer position	Time at closer position	Time difference	Position difference
2-3	28.80m	16:33:00.05	19.2m	16:33:01.21	1.16s	-9.60m
1-2	9.60m	16:33:09.80	8.53m	16:33:11.76	1.96s	-1.07m
1-2	9.60m	16:33:19.60	14.45m	16:33:22.85	3.25s	4.85m
2-3	28.80m	16:33:29.12	37.97m	16:33:31.40	2.27s	9.17m
2-3	28.80m	16:33:39.79	29.55m	16:33:40.69	0.89s	0.75m
1-2	9.60m	16:33:49.60	7.41m	16:33:52.60	3.00s	-2.19m
1-2	9.60m	16:34:00.44	9.85m	16:34:01.72	1.28s	0.25m
2-3	28.80m	16:34:08.18	19.20m	16:34:09.67	1.49s	-9.60m
2-3	28.80m	16:34:15.49	24.70m	16:34:17.40	1.91s	-4.10m
1-2	9.60m	16:34:23.66	4.36m	16:34:27.06	3.40s	-5.24m
1-2	9.60m	16:34:35.82	3.12m	16:34:36.99	1.17s	-6.48m
2-3	28.80m	16:34:43.82	31.14m	16:34:45.78	1.96s	2.34m
2-3	28.80m	16:34:59.28	30.55m	16:35:03.29	4.01s	1.75m
1-2	9.60m	16:35:12.27	11.05m	16:35:13.16	0.89s	1.45m
1-2	9.60m	16:35:29.01	4.71m	16:35:31.72	2.71s	-4.89m
2-3	28.80m	16:35:36.26	35.89m	16:35:38.19	1.94s	7.09m
2-3	28.80m	16:35:48.21	32.95m	16:35:49.32	1.10s	4.15m
1-2	9.60m	16:35:56.78	11.00m	16:35:57.95	1.17s	1.40m

Table 35: EVO 2 Pro position prediction results on horizontal flight by closer prediction time.

Position	Real position	Real time	Closer Prediction Time	Predicted Position	Time difference	Position difference
1-2	9.60m	17:02:13.18	17:02:14.68	7.69m	1.50s	-1.91m
1-2	9.60m	17:02:29.69	17:02:31.65	8.12m	1.96s	-1.48m
1-2	9.60m	17:02:50.01	17:02:51.13	10.1m	1.12s	0.50m
1-2	9.60m	17:03:03.67	17:03:04.02	0.00m	0.35s	-9.60m
1-2	9.60m	17:03:29.25	17:03:30.43	11.36m	1.19s	1.76m
1-2	9.60m	17:03:41.95	17:03:41.88	9.19m	-0.07s	-0.41m
1-2	9.60m	17:04:06.68	17:04:07.48	18.68m	0.80s	9.08m
1-2	9.60m	17:04:20.17	17:04:22.04	12.77m	1.87s	3.17m
1-2	9.60m	17:04:53.01	17:04:54.74	2.11m	1.72s	-7.49m
1-2	9.60m	17:05:01.98	17:05:02.57	0.00m	0.59s	-9.60m
1-2	9.60m	17:05:21.33	17:05:23.58	0.00m	2.25s	-9.60m
1-2	9.60m	17:05:39.10	17:05:40.32	6.94m	1.22s	-2.66m
1-2	9.60m	17:06:01.65	17:06:02.48	17.26m	0.82s	7.66m
1-2	9.60m	17:06:12.15	17:06:14.49	14.47m	2.34s	4.87m
1-2	9.60m	17:06:27.12	17:06:29.53	18.66m	2.41s	9.06m
1-2	9.60m	17:06:36.67	17:06:37.51	0.94m	0.84s	-8.66m

Table 36: EVO 2 Pro position prediction results on horizontal flight by closer position.

Position	Real position	Real time	Closer position	Time at closer position	Time difference	Position difference
1-2	9.60m	17:02:13.18	7.69m	17:02:14.68	1.50s	-1.91m
1-2	9.60m	17:02:29.69	8.12m	17:02:31.65	1.96s	-1.48m
1-2	9.60m	17:02:50.01	10.01m	17:02:51.13	1.12s	0.41m
1-2	9.60m	17:03:03.67	0.00m	17:03:04.02	0.35s	-9.60m
1-2	9.60m	17:03:29.25	11.36m	17:03:30.43	1.19s	1.76m
1-2	9.60m	17:03:41.95	9.19m	17:03:41.88	-0.07s	-0.41m
1-2	9.60m	17:04:06.68	18.68m	17:04:07.48	0.80s	9.08m
1-2	9.60m	17:04:20.17	12.77m	17:04:22.04	1.87s	3.17m
1-2	9.60m	17:04:53.01	2.11m	17:04:54.74	1.72s	-7.49m
1-2	9.60m	17:05:01.98	10.64m	17:05:04.26	2.28s	1.04m
1-2	9.60m	17:05:21.33	0.00m	17:05:23.58	2.25s	-9.60m
1-2	9.60m	17:05:39.10	6.94m	17:05:40.32	1.22s	-2.66m
1-2	9.60m	17:06:01.65	17.26m	17:06:02.48	0.82s	7.66m
1-2	9.60m	17:06:12.15	7.47m	17:06:12.81	0.65s	-2.13m
1-2	9.60m	17:06:27.12	1.25m	17:06:29.59	2.47s	-8.35m
1-2	9.60m	17:06:36.67	11.31m	17:06:39.20	2.53s	1.71m

Statistics for all horizontal flight scenarios are shown in Table 37. All errors are calculated based on their absolute value (RMSE) as explained in section 4.4.2.1.

The results for horizontal flight scenarios are consistent with the ones observed in perpendicular flight scenarios. The average response time is in the range of 1.55 seconds to 1.98 seconds, which is even better than with perpendicular flight. The overall position errors increased, but not as much as expected, which is a good sign. The biggest error difference is observed when flying the EVO 2 Pro at position 1-2, where the error increased from 1.18m-2.75m to 4.28m-5.47m, more than the double of error, but still good enough results for a practical application.

The most remarkable detail to mention from these experiments is that the flight pattern was always clearly visible through the UI and in the logs, demonstrating that the microphones can follow the path of the UAV with this approach.

Table 37: Statistics for horizontal flight scenario

	DJI Phantom 4		EVO 2 Pro	
	By Closer Prediction Time	**By Closer Position**	**By Closer Prediction Time**	**By Closer Position**
Avg Time Difference	1.55s	1.98s	1.31s	1.42s
Max Time Difference	3.19s	4.01s	2.41s	2.53s
Avg Error	5.56m	4.24m	5.47m	4.28m
Avg Error at 1-2	5.04m	3.09m	5.47m	4.28m
Avg Error at 2-3	6.08m	5.39m	-	-

4.6 Summary

In this chapter, the whole experimental process that resulted in the model proposed by this project was explained. This explanation included details about the equipment used, the initial approaches of the project, and the indoor and outdoor experiments performed.

A UAV classification model was proposed after finding the best elements to construct it. This model includes an MLP Neural Networks algorithm with parameter alpha = 0.01, Filter Banks as the feature type to extract from sound samples, and sound samples of 1 second split in sub samples of 0.1 seconds reconciled by a voting system with 60% acceptance criteria. This model showed an accuracy of 95.38% and an F1-Score of 0.93 on a test set conformed by outdoor UAV flying sound samples.

The UAV position location algorithm developed was tested under realistic conditions as well. A DJI Phantom 4 and an EVO 2 Pro were flown with different flight patterns over the acoustic sensors, and the system reached a response time below 1.59 seconds on average, and a maximum average position error of 6.06m, but as good as 0.33m, depending on the case. These results are good enough to consider that the proposed solution meets the expectations of it being a real time UAV detection and localization system that provides practical information to protect a target from an attacking UAV.

CHAPTER 5: CONCLUSIONS, DISCUSSIONS & RECOMMENDATIONS

5.1 Introduction

The problem presented on this project was the need to find a real-world adaptable solution to demonstrate that an array of acoustic sensors can be used to detect and estimate the direction of arrival of potentially harmful UAVs under realistic environmental conditions, with minimal cost and competitive performance. The current chapter will explore if the problem was addressed, and if the research questions presented at the beginning of the project were answered. Conclusions will be made based on the results obtained during experimentation, and based on them, limitations of the project and possible improvements will be presented.

5.2 Conclusions

In the current project, a working model for UAV detection and localization using an interconnected array of acoustic sensors along with machine learning algorithms and sound recognition techniques was presented and tested outdoors, under real world environmental conditions and with different types of commercial UAVs. Based on this, it can be concluded that the problem initially proposed was addressed, but that a solution is delivered does not mean that the solution is effective, so it is time to analyze how effective this proposed solution is, which are the advantages and drawbacks of it, and what other conclusions can be taken from the results.

As initially mentioned, the main criterion for the success of the project is that the cost and response time are minimized without sacrificing performance. To assess this, three elements have to be analyzed: cost, response time and performance.

In the proposed solution, each node consists of a Raspberry Pi (or any single board computer) and a microphone. The total cost per node, using the same equipment as in the experiments performed, is less than 75 USD, but these are not minimal requirements. Equipment with way less power than the ones used could provide results as effective as the ones obtained, and for a company the cost of these components could be significantly reduced. The experiments performed show that two nodes provide a coverage of at least 20m. For each 20m of additional coverage needed, a new node must be added, so the cost increases linearly based on the coverage desired. The central server used was a laptop Dell Inspiron 15 Series 3000, which at the time of this project is already far outdated, meaning that the computational requirements for the central server are low, and the investment needed on this element of the solution is low as well. In conclusion, the hardware requirements of the solution are minimal, a single microphone and a processing unit with network connectivity per node should be enough to implement this solution, so the cost minimization goal was achieved.

About response time, the MLP algorithm proposed for UAV classification provides a prediction in less than 0.0057 seconds on average. The performance results on outdoor tests flying two types of UAVs with different flight patterns show a response time below 1.59 seconds on average for samples of 1 second, meaning the prediction is shown after 0.59 seconds that the recording was taken. These time results are enough to consider the model as a real time solution, so the response time goal was met as well.

The third element to consider is performance. The UAV classification algorithm proposed achieves an accuracy of 95.38% and an F1-Score of 0.93 on a test set conformed by outdoor UAV sound samples. The model has minimal adjustments over the default MLP model provided by scikit-learn library, meaning that results are accurate enough even without a deep model analysis. About the error on direction of arrival prediction, a maximum average position error of 6.06m was obtained, but on certain scenarios it was as good as 0.33m. It was observed that if the waiting time is roughly doubled, the prediction error can be reduced to 3.48m on average, almost a half of the previous value. The solution is designed for direction of arrival prediction, this means the UAV transversally crossing the acoustic sensors barrier, but even with the UAV flying parallel to the acoustic sensors barrier, the results showed an average error between 4.24m and 5.56m depending on the test scenario. The conclusion is that, even though the position prediction results are not the best obtained in the research area, they are good enough for practical implementations, and they were achieved with minimal cost conditions, so the expectations for the project were met.

Besides cost, response time and performance, other advantages of the proposed solution are that it is adaptable to any environment and implementation layout, this means that the nodes can be moved and positioned at will and the solution should still work, it is also an asynchronous solution, so there is no need to synchronize the clocks on each node, and in case that one of them stops working, the others would still provide protection. Finally, the solution is modular, so each component of it could be improved separately.

About the limitations of the current solution, the presence of false positives was a constant problem during the whole project. Even though the number of false positives was reduced to 7.41% of the total positives, it is still a threat to the validity of the solution since they deviate the attention from real positive scenarios. Another limitation of the solution is the position prediction error, which is good enough for the purposes of the project, but it may not be enough for other applications. The current solution is just an alert system, it should be complemented by a counterattack system to stop the UAV threat.

To summarize and conclude, here are the answers the project provided for each one of the research questions initially proposed:

- *RQ1 - How accurate, precise, and cost-effective is the proposed model for locating potentially harmful UAVs in real time?:* The model proposed has an accuracy of 95.38% and a precision of 0.93 for UAV detection. About location precision, the model showed a maximum average position error of 6.06m. Regarding cost effectiveness, the solution proposed minimizes the resources needed, and each node of the ones used for the experiments performed costs less than 75 USD.
- *RQ2 - What error level can be achieved on the identification of position and direction of arrival of a UAV using an array of acoustic sensors?*: Even though the maximum position error obtained was 6.06m, lower errors can be obtained. A 0.33m position prediction error was achieved by waiting 4.53 seconds on average for a prediction.
- *RQ3 - What is the response time that can be achieved on UAV detection using an array of acoustic sensors running machine learning algorithms?:* The system calculates a prediction in a maximum of 0.0057 seconds on average. Including the recording time of 1 second the system achieved to log a result on 1.59 seconds on average on its best test result.
- *RQ4 - What is the minimum cost an acoustic detection and location system can achieve while keeping an acceptable performance?*: The nodes used for experimentation have a cost of less than 75 USD, but the cost can be reduced even more if using cheaper components or buying them in quantity. It is observed that the system could even work properly with components with less computing power.

5.3 Discussion

The project managed to accomplish the goals it had initially proposed, but there is still work to be done, considerations to be taken, and possible improvements to make.

The design of the solution is modular, so improvements could be made on every part of the solution: sound recording, sound transformation, machine learning algorithm, or position prediction algorithm.

Besides using more expensive better quality acoustic sensors, other improvements that could be made include the implementation of some noise removal technique for preprocessing the data, or the use of microphones with directional recording or other pickup patterns specific to the problem. With these changes the coverage could be improved to more than 20 meters.

During sound transformation, only a few parameters were tested, but fine-tuning sound transformation parameters for UAV detection could be the topic of an entire research. On the experiments performed, Filter Banks were the feature type that showed the best results, but it does not mean that MFCC or STFT cannot perform better under the right circumstances and with the right parameters. Even the current Filter Banks solution could be improved by tuning parameters such as the number and shape of filters.

About the machine learning algorithm used, it was out of scope to find the best existing learning algorithm for UAV detection, or tune its parameters to perfection, so a lot of work could be done here. In fact, related works have addressed better accuracy results with more complex models, like other neural network architectures or an ensemble of different machine learning models. Other ways of improving the machine learning algorithm include adding more data to the training database, recording background noise under different environmental conditions, and adding sounds of cars, planes or other elements which could be confused with UAVs. About the false positives problem observed during the whole project, the percentage needed for approval on the voting system implemented could be increased, which may increase the number of false negatives, but also would certainly reduce the number of false positives.

The position prediction algorithm has plenty of room for improvement as well. For practical reasons, RMS amplitude of the signal was used as a basis for sound "intensity" change, but other sound analysis methods could be used to improve precision, such as sound power, sound intensity (meaning the actual definition of "sound intensity", which is the ratio of sound power by area), loudness units relative to full scale (LUFS) [75], and more. Another possible improvement to position prediction could be to add redundant nodes, in that way the system could have more protection against failures and higher accuracy.

Regarding the cost of the system, as previously mentioned, it could be reduced if using cheaper components with less computing power, even the central server could be replaced by another Raspberry Pi (or any alternative single board computer brand).

Finally, one important observation that is worth mentioning is that the results obtained during the experiments are coherent with the literature reviewed, and consistent on the progression of data, which enforces the reliability and validity of them.

5.4 Recommendations

One of the lessons learned from this project and which could help future researchers is that, before any field test, the exact data collection conditions should be replicated at the lab. It happened during this project that a whole batch of samples collected had to be discarded because the power source (a battery and a power inverter) introduced too much noise on the recordings. A noise-free power source is important to consider in any acoustic implementation, as well as removing any additional noise possible from the microphones.

An approach that worked successfully during this project was to split the samples in 10 second samples, which is easy to manage when they need to be cleaned and organized.

Finally, the experiments performed do not allow to identify false negatives and true negatives, so some data analyses could not be made. This means that while thinking about the solution to be implemented, it is important to think in detail how it would be possible to evaluate it, and which metrics need to be taken.

REFERENCES

[1] J. Kim, C. Park, J. Ahn, Y. Ko, J. Park, and J. C. Gallagher, "Real-time UAV sound detection and analysis system," in *2017 IEEE Sensors Applications Symposium (SAS)*, Mar. 2017, pp. 1–5, doi: 10.1109/SAS.2017.7894058.

[2] B. Taha and A. Shoufan, "Machine Learning-Based Drone Detection and Classification: State-of-the-Art in Research," *IEEE Access*, vol. 7, pp. 138669–138682, 2019, doi: 10.1109/ACCESS.2019.2942944.

[3] E. E. Case, A. M. Zelnio, and B. D. Rigling, "Low-Cost Acoustic Array for Small UAV Detection and Tracking," in *2008 IEEE National Aerospace and Electronics Conference*, Dayton, OH, Jul. 2008, pp. 110–113, doi: 10.1109/NAECON.2008.4806528.

[4] B. Yang, E. T. Matson, A. H. Smith, J. E. Dietz, and J. C. Gallagher, "UAV Detection System with Multiple Acoustic Nodes Using Machine Learning Models," in *2019 Third IEEE International Conference on Robotic Computing (IRC)*, Naples, Italy, Feb. 2019, pp. 493–498, doi: 10.1109/IRC.2019.00103.

[5] S. Seo, S. Yeo, H. Han, Y. Ko, K. E. Ho, and E. T. Matson, "Single Node Detection on Direction of Approach," in *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Dubrovnik, Croatia, May 2020, pp. 1–6, doi: 10.1109/I2MTC43012.2020.9129016.

[6] Z. Shi, X. Chang, C. Yang, Z. Wu, and J. Wu, "An Acoustic-Based Surveillance System for Amateur Drones Detection and Localization," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 2731–2739, Mar. 2020, doi: 10.1109/TVT.2020.2964110.

[7] S. Li, H. Kim, S. Lee, J. C. Gallagher, D. Kim, S. Park, and E. T. Matson, "Convolutional Neural Networks for Analyzing Unmanned Aerial Vehicles Sound," in *2018 18th International Conference on Control, Automation and Systems (ICCAS)*, Oct. 2018, pp. 862–866.

[8] "Drone market outlook: industry growth trends, market stats and forecast," *Business Insider*, Mar. 03, 2020.

[9] M. S. Schmidt and M. D. Shear, "A Drone, Too Small for Radar to Detect, Rattles the White House," *The New York Times*, Jan. 26, 2015.

[10] W. Ripley, "Drone found on Japanese Prime Minister's rooftop," *CNN*, Apr. 22, 2015. https://www.cnn.com/2015/04/22/asia/japan-prime-minister-rooftop-drone/index.html (accessed Nov. 06, 2020).

[11] J. Warrick, "Use of weaponized drones by ISIS spurs terrorism fears," *Washington Post*, Feb. 21, 2017.

[12] C. Koettl and B. Marcolini, "A Closer Look at the Drone Attack on Maduro in Venezuela," *The New York Times*, Aug. 10, 2018.

[13] A. Bernardini, F. Mangiatordi, E. Pallotti, and L. Capodiferro, “Drone detection by acoustic signature identification,” *Electron. Imaging*, vol. 2017, no. 10, pp. 60–64, Jan. 2017, doi: 10.2352/ISSN.2470-1173.2017.10.IMAWM-168.

[14] Y. Seo, B. Jang, and S. Im, “Drone Detection Using Convolutional Neural Networks with Acoustic STFT Features,” in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Auckland, New Zealand, Nov. 2018, pp. 1–6, doi: 10.1109/AVSS.2018.8639425.

[15] T. B. Lee, “Watch the Pirate Party fly a drone in front of Germany’s chancellor,” *Washington Post*, Sep. 18, 2013.

[16] “Anti-Drone Market Size to Reach USD 2.315 Billion by 2025 Valuates Reports,” *Valuates Reports*, May 22, 2020.

[17] V. Baron, S. Bouley, M. Muschinowski, J. Mars, and B. Nicolas, “Drone localization and identification using an acoustic array and supervised learning,” in *Artificial Intelligence and Machine Learning in Defense Applications*, Strasbourg, France, Sep. 2019, vol. 11169, p. 111690F, doi: 10.1117/12.2533039.

[18] “Anti-drone Market Size & Share | Global Industry Report, 2019-2026,” May 2019. https://www.grandviewresearch.com/industry-analysis/anti-drone-market (accessed Oct. 12, 2020).

[19] “Anti-drone Market Size Worth $4.5 Billion By 2026 | CAGR: 29.9%.” https://www.grandviewresearch.com/press-release/global-anti-drone-market (accessed Oct. 12, 2020).

[20] “Meaning of UAV in English.” https://dictionary.cambridge.org/dictionary/english/uav (accessed Nov. 09, 2020).

[21] A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow*, 2nd Edition. O’Reilly Media, Inc., 2017.

[22] “Meaning of Payload in English.” https://dictionary.cambridge.org/dictionary/english/payload (accessed Nov. 09, 2020).

[23] “A-Z Databases: Engineering & Technology.” https://guides.lib.purdue.edu/az.php?s=71214 (accessed Nov. 01, 2020).

[24] “Drone market outlook: industry growth trends, market stats and forecast,” *Business Insider*, Mar. 03, 2020. https://www.businessinsider.com/drone-industry-analysis-market-trends-growth-forecasts (accessed Nov. 06, 2020).

[25] P. Paganini, “Thieves are using commercial drones for burglaries,” *Security Affairs*, May 22, 2015. https://securityaffairs.co/wordpress/37050/cyber-crime/thieves-using-commercial-drones.html (accessed Nov. 06, 2020).

[26] “‘Well-organised’ gang flew drones carrying drugs into prisons,” *BBC News*, Aug. 30, 2018. https://www.bbc.com/news/uk-england-45358876 (accessed Nov. 06, 2020).

[27] T. Cozzens, "Report predicts drone threats to infantry units," *GPS World*, Mar. 13, 2018. https://www.gpsworld.com/new-report-predicts-small-drone-threats-to-infantry-units/ (accessed Nov. 06, 2020).

[28] M. I. Skolnik, Ed., *Radar handbook*, 2nd ed. New York: McGraw-Hill, 1990.

[29] İ. Güvenç, O. Ozdemir, Y. Yapici, H. Mehrpouyan, and D. Matolak, "Detection, localization, and tracking of unauthorized UAS and Jammers," in *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, Sep. 2017, pp. 1–10, doi: 10.1109/DASC.2017.8102043.

[30] 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, doi: 10.1109/IRS.2016.7497351.

[31] Y. Liu, X. Wan, H. Tang, J. Yi, Y. Cheng, and X. Zhang, "Digital television based passive bistatic radar system for drone detection," in *2017 IEEE Radar Conference (RadarConf)*, May 2017, pp. 1493–1497, doi: 10.1109/RADAR.2017.7944443.

[32] B. Torvik, K. E. Olsen, and H. Griffiths, "Classification of Birds and UAVs Based on Radar Polarimetry," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 9, pp. 1305–1309, Sep. 2016, doi: 10.1109/LGRS.2016.2582538.

[33] M. Ritchie, F. Fioranelli, H. Borrion, and H. Griffiths, "Classification of loaded/unloaded micro-drones using multistatic radar," *Electron. Lett.*, vol. 51, no. 22, pp. 1813–1815, Oct. 2015, doi: 10.1049/el.2015.3038.

[34] Z. Shi, M. Huang, C. Zhao, L. Huang, X. Du, and Y. Zhao, "Detection of LSSUAV using hash fingerprint based SVDD," in *2017 IEEE International Conference on Communications (ICC)*, Paris, France, May 2017, pp. 1–5, doi: 10.1109/ICC.2017.7996844.

[35] Y. Tian, L. Njilla, A. Raja, J. Yuan, S. Yu, A. Steinbacher, T. Tong, and J. Tinsley, "Cost-Effective NLOS Detection for Privacy Invasion Attacks by Consumer Drones," in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, San Diego, CA, USA, Sep. 2019, pp. 1–7, doi: 10.1109/DASC43569.2019.9081802.

[36] C. Zhao, C. Chen, Z. Cai, M. Shi, X. Du, and M. Guizani, "Classification of Small UAVs Based on Auxiliary Classifier Wasserstein GANs," in *2018 IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 206–212, doi: 10.1109/GLOCOM.2018.8647973.

[37] U. Seidaliyeva, D. Akhmetov, L. Ilipbayeva, and E. Matson, "Real-Time and Accurate Drone Detection in a Video with a Static Background," *Sensors*, vol. 20, no. 14, p. 3856, Jul. 2020, doi: 10.3390/s20143856.

[38] E. Unlu, E. Zenou, N. Riviere, and P.-E. Dupouy, "Deep learning-based strategies for the detection and tracking of drones using several cameras," *IPSJ Trans. Comput. Vis. Appl.*, vol. 11, no. 1, p. 7, Jul. 2019, doi: 10.1186/s41074-019-0059-x.

[39] V. Thai, W. Zhong, T. Pham, S. Alam, and V. Duong, “Detection, Tracking and Classification of Aircraft and Drones in Digital Towers Using Machine Learning on Motion Patterns,” in *2019 Integrated Communications, Navigation and Surveillance Conference (ICNS)*, Herndon, VA, USA, Apr. 2019, pp. 1–8, doi: 10.1109/ICNSURV.2019.8735240.

[40] M. Hammer, M. Hebel, B. Borgmann, M. Laurenzis, and M. Arens, “Potential of lidar sensors for the detection of UAVs,” in *Laser Radar Technology and Applications XXIII*, Orlando, FL, United States, May 2018, vol. 10636, p. 1063605, doi: 10.1117/12.2303949.

[41] M. Salhi and N. Boudriga, “Multi-Array Spherical LIDAR System for Drone Detection,” in *2020 22nd International Conference on Transparent Optical Networks (ICTON)*, Bari, Italy, Jul. 2020, pp. 1–5, doi: 10.1109/ICTON51198.2020.9203381.

[42] B. R. V. Voorst, “Counter drone system,” US20170261613A1, Sep. 14, 2017.

[43] A. Sedunov, H. Salloum, A. Sutin, N. Sedunov, and S. Tsyuryupa, “UAV Passive Acoustic Detection,” in *2018 IEEE International Symposium on Technologies for Homeland Security (HST)*, Woburn, MA, USA, Oct. 2018, pp. 1–6, doi: 10.1109/THS.2018.8574129.

[44] I. Guvenc, F. Koohifar, S. Singh, M. L. Sichitiu, and D. Matolak, “Detection, Tracking, and Interdiction for Amateur Drones,” *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 75–81, Apr. 2018, doi: 10.1109/MCOM.2018.1700455.

[45] M. Wakabayashi, H. G. Okuno, and M. Kumon, “Multiple Sound Source Position Estimation by Drone Audition Based on Data Association Between Sound Source Localization and Identification,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 782–789, Apr. 2020, doi: 10.1109/LRA.2020.2965417.

[46] K. Chang, H. Yujing, and S. Lin, “Method for distinguishing acoustic of drone from e.g. sound of car, involves producing feature vector for combining first and second feature vectors and distinguishing acoustic signal according to unmanned aerial vehicle.”

[47] W. Yoon S., S. Yin, and S. Un, “Method of identifying and neutralizing low-altitude unmanned aerial vehicle, involves comparing sound and shape information included in vehicle target image with prestored sound and shape information of each vehicle type.”

[48] D. Lim, H. Kim, S. Hong, S. Lee, G. Kim, A. Snail, L. Gotwals, and J. C. Gallagher, “Practically Classifying Unmanned Aerial Vehicles Sound Using Convolutional Neural Networks,” in *2018 Second IEEE International Conference on Robotic Computing (IRC)*, Laguna Hills, CA, Feb. 2018, pp. 242–245, doi: 10.1109/IRC.2018.00051.

[49] B. D. V. Veen and K. M. Buckley, “Beamforming: a versatile approach to spatial filtering,” *IEEE ASSP Mag.*, vol. 5, no. 2, pp. 4–24, Apr. 1988, doi: 10.1109/53.665.

[50] J. Franklin and B. Hearing, “Drone detection and classification with compensation for background clutter sources,” US10032464B2, Jul. 24, 2018.

[51] K. Gröchenig, *Foundations of Time-Frequency Analysis*. Birkhäuser Basel, 2001.

[52] S. R. M. Penedo, M. L. Netto, and J. F. Justo, “Designing digital filter banks using wavelets,” *EURASIP J. Adv. Signal Process.*, vol. 2019, no. 1, p. 33, Jul. 2019, doi: 10.1186/s13634-019-0632-6.

[53] S. Sarangi, M. Sahidullah, and G. Saha, “Optimization of data-driven filterbank for automatic speaker verification,” *Digit. Signal Process.*, vol. 104, p. 102795, Sep. 2020, doi: 10.1016/j.dsp.2020.102795.

[54] B. Logan, “Mel Frequency Cepstral Coefficients for Music Modeling,” presented at the 1st Int. Symposium Music Information Retrieval, Plymouth, Massachusetts, Oct. 2000.

[55] V. Oleynikov, O. Zubkov, V. Kartashov, I. Koryttsev, S. Sheiko, and S. Babkin, “Experimental Estimation of Direction Finding to Unmanned Air Vehicles Algorithms Efficiency by Their Acoustic Emission,” in *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S T)*, Kyiv, Ukraine, Oct. 2019, pp. 175–178, doi: 10.1109/PICST47496.2019.9061337.

[56] Y. Seo, B. Jang, and S. Im, “Drone Detection Using Convolutional Neural Networks with Acoustic STFT Features,” in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Auckland, New Zealand, Nov. 2018, pp. 1–6, doi: 10.1109/AVSS.2018.8639425.

[57] T. R. P. Foundation, “Teach, Learn, and Make with Raspberry Pi,” *Raspberry Pi*. https://www.raspberrypi.org/ (accessed Nov. 18, 2020).

[58] P. Podder, T. Zaman Khan, M. Haque Khan, and M. Muktadir Rahman, “Comparative Performance Analysis of Hamming, Hanning and Blackman Window,” *Int. J. Comput. Appl.*, vol. 96, no. 18, pp. 1–7, Jun. 2014, doi: 10.5120/16891-6927.

[59] I. Kavalerov, S. Wisdom, H. Erdogan, B. Patton, K. Wilson, J. Le Roux, and J. R. Hershey, “Universal Sound Separation,” in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct. 2019, pp. 175–179, doi: 10.1109/WASPAA.2019.8937253.

[60] H. Fayek, “Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What’s In-Between,” *Haytham Fayek*, Apr. 21, 2016. https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html (accessed Mar. 10, 2021).

[61] E. H.-L. U. 27 Dec’18 2018-12-27T08:37:12+00:00, “What is Edge Computing: The Network Edge Explained,” *Cloudwards*, Dec. 31, 2018. https://www.cloudwards.net/what-is-edge-computing/ (accessed Mar. 10, 2021).

[62] “sklearn.model_selection.StratifiedShuffleSplit — scikit-learn 0.24.1 documentation.” https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedShuffleSplit.html (accessed Mar. 11, 2021).

[63] “scikit-learn: machine learning in Python — scikit-learn 0.23.2 documentation.” https://scikit-learn.org/stable/ (accessed Dec. 06, 2020).

[64] "pickle — Python object serialization — Python 3.9.2 documentation." https://docs.python.org/3/library/pickle.html (accessed Mar. 11, 2021).

[65] "Leaflet — an open-source JavaScript library for interactive maps." https://leafletjs.com/ (accessed Mar. 22, 2021).

[66] "Maps, geocoding, and navigation APIs & SDKs | Mapbox." https://www.mapbox.com/ (accessed Mar. 22, 2021).

[67] "Apache Kafka," *Apache Kafka*. https://kafka.apache.org/ (accessed Mar. 22, 2021).

[68] B. Yang, "UAV DETECTION SYSTEM WITH MULTIPLE ACOUSTIC NODES USING MACHINE LEARNING MODELS," thesis, Purdue University Graduate School, 2019.

[69] "Welcome to Flask — Flask Documentation (1.1.x)." https://flask.palletsprojects.com/en/1.1.x/ (accessed Mar. 22, 2021).

[70] M. Rehkopf, "What is a Kanban Board?," *Atlassian*. https://www.atlassian.com/agile/kanban/boards (accessed Nov. 19, 2020).

[71] U. Sekaran and R. Bougie, *Research Methods For Business: A Skill Building Approach*, 7th Edition. John Wiley & Sons, 2016.

[72] "Amazon.com: Frambuesa Pi 3 Modelo B Junta: Computers & Accessories." https://www.amazon.com/-/es/4328498196-Frambuesa-Pi-Modelo-Junta/dp/B01LPLPBS8 (accessed Apr. 02, 2021).

[73] "Amazon.com: Micrófono de ordenador, micrófono de PC Plug & Play Home Studio micrófono condensador para escritorio/portátil/portátil, grabación para YouTube, podcasting, juegos, chat en línea, negro..." https://www.amazon.com/-/es/Micr%C3%B3fono-ordenador-condensador-escritorio-podcasting/dp/B07BDFP6XC/ref=sr_1_2?dchild=1&m=A2US6ATHMB6XXW&qid=1617371754&s=merchant-items&sr=1-2 (accessed Apr. 02, 2021).

[74] "sklearn.neural_network.MLPClassifier — scikit-learn 0.24.1 documentation." https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html (accessed Mar. 30, 2021).

[75] "LUFS: How To Measure Your Track's Loudness in Mastering," *EDMProd*, Jun. 02, 2020. https://www.edmprod.com/lufs/ (accessed Apr. 02, 2021).