QATAR UNIVERSITY

COLLEGE OF ENGINEERING

DDI: DRONES DETECTION AND IDENTIFICATION USING DEEP LEARNING TECHNIQUES

BY

SARA ABDULRAZAQ AL-EMADI

A Thesis Submitted to

the College of Engineering

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Computing

January 2021

COMMITTEE PAGE

The members of the Committee approve the Thesis of
Sara Abdulrazaq Al-Emadi defended on 23/11/2020.

Dr. Abdulla Al-Ali
Thesis Supervisor

Dr. Abdulaziz Al-Ali
Thesis Co-Supervisor

Prof. Mohsen Guizani
Thesis Co-Supervisor

Prof. Roberto Di Pietro
Committee Member

Dr. Elias Yaacoub
Committee Member

Dr. Tamer Khattab
Committee Member

Approved:

Khalid Kamal Naji, Dean, College of Engineering

ABSTRACT

Al-Emadi, Sara, A., Masters : January: 2021, Master of Science in Computing

Title: DDI: Drones Detection and Identification using Deep Learning Techniques

Supervisor of Thesis: Dr. Abdulla Al-Ali.

Drones are becoming increasingly popular not only for recreational purposes but in day-to-day applications in engineering, medicine, logistics, security and others. Besides their useful applications, an alarming concern in regards to the physical infrastructure security, safety and privacy arose due to the potential of their use in malicious activities. To address this problem, we work towards the proposed solution by the following twofold contribution, first we propose a novel solution that automates the drone detection and identification processes using drone's acoustic features with different deep learning algorithms. However, the lack of acoustic drone datasets hinders the ability to implement an effective solution. Therefore, we aim to fulfil this gap by introducing a hybrid drone acoustic dataset composed of recorded drone audio clips and artificially generated drone audio clips using a state of the art deep learning model known as the Generative Adversarial Network. Furthermore, we examine the effectiveness of using drone audio with different deep learning algorithms, namely, the Convolutional Neural Network, the Recurrent Neural Network and the Convolutional Recurrent Neural Network in drone detection and identification. Moreover, we investigate the impact our proposed hybrid dataset has on drone detection. The second contribution is laying the foundation for the next step of the anti-drone proposed system which is focused around swarm drones localisation and tracking using data fusion of audio and radio frequency signals using deep learning techniques. This is made possible through the design of a novel swarm of drones simulator. Our findings prove the advantage of using deep learning techniques with acoustic data for drone detection and identification while confirming our hypothesis on the benefits of using the Generative Adversarial Networks to generate real-like drone audio clips with an aim of enhancing the detection of new and unfamiliar drones.

DEDICATION

To my parents

ACKNOWLEDGMENTS

I would like to thank my supervisor Dr.Abdulla Al-Ali for taking on my thesis and for his continued encouragement, guidance and support throughout this research. I also would like to express my great appreciation to my co-supervisor Dr.Abdulaziz Al-Ali who has provided his insights and support that greatly assisted this research. Finally, I wish to extend my special thanks to my co-supervisor Prof.Mohsen Guizani and Dr. Elias Yaacoub for their advice throughout this research.

Furthermore, grateful acknowledgement is made to Qatar University for Internal Grant No. QUCP-CENG-2018/2019-1 which enabled us to carry out the experiments throughout this research.

To conclude, I would like to add my personal thanks to my family and friends for their unconditional support, without which this work would not have been possible.

Sara A. Al-Emadi
November 2020

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AI Artificial Intelligence.

BS Base Station.

CH Cluster head.

CHs Cluster heads.

CNN Convolution Neural Networks.

Conv Convolution Layer.

CRNN Convolutional Recurrent Neural Network.

DL Deep Learning.

DNN Deep Neural Network.

DSP Digital Signal Processing.

FC Fully-Connected.

FCL Fully-Connected Layer.

FN False Negative.

FP False Positive.

FSPL Free Space Pathloss.

GAN Generative Adversarial Network.

GMM Gaussian Mixture Models.

GRU Gated Recurrent Unit.

KNN K-Nearest Neighbor.

LSTM Long Short-Term Memory.

ML Machine Learning.

NN Neural Network.

P_l Pathloss.

P_r Power Received.

P_t Power Transmitted.

PIL Plotted Image Learning.

Pool Pooling Layer.

R_{xs} Receivers.

R_x Receiver.

RF Radio-Frequency.

RFDS RF Drones Simulator.

RNN Recurrent Neural Network.

RPGM Reference Point Group Mobility.

RWMP Random WayPoint Mobility.

SNR Signal-to-Noise Ratio.

SVM Support Vector Machine.

TN True Negative.

TP True Positive.

UAV Unmanned Aerial Vehicle.

CHAPTER 1: INTRODUCTION

In recent years, drones, also known as Unmanned Aerial Vehicles (UAVs), became significantly popular due to the rapid technical enhancements in both, their hardware: by equipping them with cameras and audio recording technologies; as well as their software: by providing the support of autonomous flying and human tracking capabilities. Initially, drones were mainly used for cinematography and recreational purposes, however, their usage has been extended to automate day-to-day operations such as vegetation monitoring [1], various wildfire mapping applications [2], precision agriculture [3] and flying over dangerous and out of reach areas for search and rescue missions [4]. Besides their useful applications, their use in malicious activities to invade privacy, security and safety regulations was alarmingly increasing. In recent events, a number of drone attacks on Gatwick airport led to the closure of the airport for a few days in an attempt to detect and impede manually the drones' malicious missions [5]. The closure had affected thousands of passengers and implied a significantly high financial costs [6]. Another incident was reported in which an explosive equipped drone was hovering over a great crowd in a formal occasion in Venezuela, targeting a high profile personnel and the general public. In this incident, the drone dropped a number of attached explosives randomly which, consequently, injured civilians on scene [7]. Furthermore, UAV attacks could potentially have a negative global impact such as the recent UAVs attacks on the Khurais oilfield and the processing plant at Abqaiq, both operated by Aramco of Saudi Arabia, causing large fires that halted their operation. This attack led to a decrease of 5.7 million barrels in crude oil production which contributed to an increase of 15% in the price of crude oil globally [8][9].

In addition to the safety issues associated with malicious drones' activities mentioned above, drones are also being utilized to violate security measures. Such a violation has been witnessed in an incident where smugglers flew drones with illegal drugs and cell phones over prison facilities [10]. Moreover, the drone violations extends to participate in disrupting sports events by flying illegally over football stadiums [11].

Similarly, privacy concerns arose with the malicious use of drones as it was reported in multiple incidents, where drones were used to spy and record videos and audio clips of people in their private properties [12][13][14].

Hence, in order to secure physical premises against malicious drone attacks, a typical approach is to design an anti-drone system that is composed of multiple stages as illustrated in Fig.1.

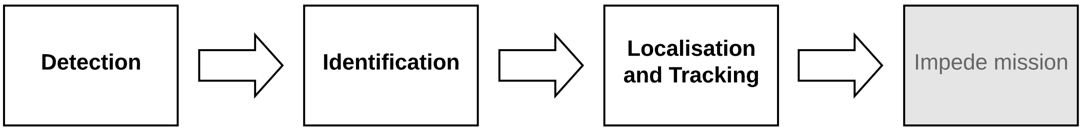


Figure 1. Anti-drone system

In the first stage, the presence of a drone within a restricted area is detected. Next, the system identifies whether the drone is authorised or unauthorised through analysing its characteristics using parameters such as the drone's type or model. Then, the system should be able localise and track the drone. In the final process, the system impedes the drone's mission using different conventional mechanisms such as shooting drones using guns [15], nets [16], a laser beam [17], disrupting the drone's localization system [18] or interfering with transmission signals between the controller to hijack the drone and land it safely [19]. However, the traditional techniques used in implementing an anti-drone systems are mainly designed around the final stage of impeding the drone's mission while being heavily dependent on manual human resources to detect and identify drones. This increases the operational cost and leaves room for human errors. Therefore, this work provides a novel solution to automatically detect and identify drones using acoustic fingerprints and expand the solution by laying the foundation for the next step of the anti-drone proposed system which is focused around swarm drones localisation and tracking using data fusion of audio and radio frequency signals through a novel design of a novel swarm of drones simulator. Therefore, this solution overcomes the current limitations of the conventional anti-drone systems.

In literature, various techniques exists to detect the presences of drones, such as:

- **Visual Analysis**: this approach uses videos or image recognition techniques to detect drones. Although these methods have proven their effectiveness in an ideal environment scenarios, their performance are heavily affected by different external factors, such as weather conditions, dust, fog or rain, as well as by other flying objects that might look like a drone, e.g. birds. Besides their susceptibility to the issue of occlusion[20].

- **Radar**: approaches like a GSM passive coherent location system [21] and a digital TV based bi-static radar [22] are used to detect drones using Radar systems. Although radars are highly effective for detecting large flying bodies, they are not useful for detecting drones. This is due to the drones' feature of having low radar cross section. In addition to flying at low altitudes with low speed in comparison to larger aircrafts [20]. Moreover, since radar systems operate at a high electromagnetic energy continuously, they might be unsuitable and illegal to operate in urban areas [20]. Also, radar systems are considered expensive to deploy [23].

- **Radio-Frequency (RF)** [24] [25]: this technique requires a live communication of RF signals between the drone and it's controller in order to detect the presence of the drone accurately. However, in scenarios where autonomous drones (preprogrammed and doesn't require an on-going communication) are being used in malicious activities, the RF based system will fail to detect the presence of the drone. Furthermore, in some areas, implementing an RF system might not be applicable such as in military areas and airports. Additionally, this detection approach is subject to high RF noise omitted from other devices present in an area [20] which contributes in decreasing the Signal-to-Noise Ratio (SNR) [23]. Hence, leading to a significant deterioration of the performance of the RF based detection system. Nevertheless, the RF technique have several advantages such as being cheap and accurate where deployed [26]. Therefore, we put forward the hypothesis that through data fusion mechanism, the RF signals gathered through a passive multi-receiver system coupled with other drone features such as acoustics, we believe that the performance of the RF based solution can be further enhanced and useful for detection, localisation and tracking of large number of drones. This solution is further discussed in Chapter 7.

1.1. Motivation

To address the current limitation of the drone detection systems discussed above, this study seeks to overcome the constraints of the drone detection techniques by introducing an autonomous system that, in addition to *detecting*, is able to *identify* drones based on their acoustic signatures using different Deep Learning (DL) techniques, namely the

Convolution Neural Networks (CNN), the Recurrent Neural Network (RNN) and the Convolutional Recurrent Neural Network (CRNN), such that no human intervention is needed. However, the following two challenges are faced by researchers in the field of drone audio analysis:

1. Lack of large acoustic drone datasets which are needed to train the DL algorithms effectively.
2. Most drone datasets only cover a few types of drones. Hence, not covering all types and models of drones available weakens the detection process and makes it more vulnerable to unfamiliar drone types.

To overcome these obstacles, we utilize the Generative Adversarial Network (GAN) [27], a state-of-art DL technique for artificial data generation to generate a large artificial drone acoustic dataset with the aim of improving the detection of drone presence.

Furthermore, we aim to extend this study by establishing the foundations of a new methodology of localization and tracking of unauthorized swarm of drones within a restricted area. The approach would provide an end to end solution by combining the data acquired from different sources such as acoustic features through the drones' sound waves and the RF signals capture by a grid of receivers. To achieve this aim, we propose a novel design of swarm of drones simulator that would mimic the swarm of drone's behaviour in terms of mobility and physical layer characteristics.

1.2. Research Questions

This thesis seeks to answer the following key questions:

1. How do deep learning algorithms perform in detecting the presence of a drone using it's acoustic features?

2. How effective deep learning algorithms are in differentiating and identifying between different types of drones using their acoustic features?

3. How feasible is the integration of an artificially generated dataset with recorded drone dataset in enhancing the drone detection performance?

4. Is it possible to create a swarms of drones simulator that would simulate the physical layer communication of a large number of simultaneously moving drones.

1.3. Contribution

To answer the research questions mentioned above, we aim through this thesis to:

- Evaluate the effectiveness of the selected DL algorithms in drone detection and identification based on specific evaluation metrics such as accuracy, F1 score, precision and recall, while providing the computational time required to train and test the models proposed.

- Examine the validity and efficacy of combining an artificially generated datasets with a recorded drone audio dataset in enhancing the drone detection process through a comparison with a only recorded drone dataset.

- Provide an open-source drone audio dataset with recorded and artificial drone audio to be further utilized by the research community in order to fulfill the shortage of drone training dataset for DL models.

- Provide an open-source swarms of drones simulator that is able to generate RF based dataset of dynamic swarm mobility patterns while accommodating various physical layer communication models to be used in training ML/DL and can be enhanced by the research community.

1.4. Document Overview

The rest of the thesis is organized as follows: Chapter 2 introduces the fundamental concepts and the background information which the solution proposed is based on. Followed by Chapter 3 which explores the literature and the state-of-art solutions. A description of the proposed framework, datasets and the neural networks architectures are presented in Chapter 4. Chapter 5 discusses the setup of the different experiments carried out through this work. Whereas, in Chapter 6 the experimental results of the drone detection and identification approaches are presented and analyzed. Prior concluding, Chapter 7 provides an overview of the simulator architecture, design and functionality. Finally, the thesis closes with a conclusion in Chapter 8.

CHAPTER 2: BACKGROUND

This chapter presents an overview of the fundamental concepts that are being utilised in the formulation of the proposed solution. Starting with Section 2.1 that introduces the spectrograms. Followed by Section 2.2 which provides a general introduction of DL. Moving on to discussing each of the different deep learning techniques selected in subsections 2.2.1 to 2.2.4. Finally, closing this chapter with an explanation of the performance evaluation criteria selected for the solution proposed in Section 2.3.

2.1. Spectrograms

Spectrograms offers a way of visualising a spectrum of frequencies with respect to time through a 2-Dimensional (2D) plot in which the frequency is defined in the y-axis, the time element through the x-axis and the amplitude of the signal is represented in the intensity of the colour in a heat map like fashion. An example is shown in Fig. 9 in Section 4.3.1.2.

Moving forward, spectrograms were the choice of dataset representation used in the experiments for classification carried out through this thesis as described in Section 4.3.1.2. Whereas, audio waves are used to train GAN.

2.2. Neural Networks

The traditional Machine Learning (ML) approaches in image recognition domain depend on hand-designed feature extraction methodology in which the information is selected based on the relevance. On a later stage, a classifier is used to categorise the resulting feature vectors into classes [28]. Although traditional ML could provide reasonable outcomes, it is considered time-consuming and not very accurate in more complex scenarios. Therefore, multilayer Neural Network (NN) became an attractive solution in image recognition domain due their ability of learning high dimensional, extremely complex and nonlinear mapping from collections of data [28]. Although this could be achieved through the conventional Fully-Connected (FC) feedforward NN, there are several drawbacks to this type of NNs:

1. Given that in an image classification problem, an image usually has a very large number of pixels, to be used as input variables, which implies that in case of FC network, the first layer would have a very large number of parameters to train and the growth in the number of parameters would also occur in the next layers of the network leading to an increase in the capacity of the system such that in order for the model to have an acceptable performance and avoid fast overfitting, a very large training set would be required.

2. As a consequence of the previous point, a large memory requirement to store all of the parameters generated in (1) and computational power would be needed [28]. Hence, it might be infeasible to train such network on normal hardware machines.

3. Another drawback of using FC networks for image or speech recognition applications is that it does not accommodate for invariance with respect to translations of local distortions of the input data [28]. That is, if a pattern emerges in one part of the image and showed up in another part in a later example, the FC network will not be able to recognise the pattern. Instead it has to re-learn the entire image or time-frequency representation of audio [29].

Hence, in this work, we have selected other types NNs, namely: CNN, RNN and CRNN that were designed to overcome the drawbacks of the conventional FC network.

2.2.1. Convolutional Neural Network

Originally, CNN was designed to provide an alternative solution that offers a higher performance than the conventional FC NNs specifically for image recognition applications. The two well-known characteristics of CNNs that contributes in achieving such outstanding performance in comparison to the conventional FC networks are:

1. The patterns learnt by CNN are said to be translation invariant[29]. Typically, FC NNs are only capable of learning global patterns from the input feature spaces, that is, the pattern learnt involves all the pixels of an image. Furthermore, learning the global features means that if a pattern located in specific position of an image is learned by the model and then appeared in another location of a later image the whole image must be re-learned by the model. To overcome this limitation, CNNs were designed to, specifically, learn local patterns. In this case, if a pattern located in specific position of an input image was learned by the model, CNN will be able to recognise this pattern even if it appears in another location of a later image. Therefore, CNN is considered data efficient given that it requires fewer training examples to generalise well [29]. An example of local features could be illustrated as corners and edges in an image.

2. CNN learn the spatial hierarchies of pattern[29]. This means that in the first layer, the model would learn the small details in an image, for example all horizontal lines within an image. Then, in the second layer, it would learn the larger patterns made up of features of the previous layer, and so on. Hence, one can conclude that CNN learns efficiently and gradually, starting from the simple features to more complex ones.

Fig.2 illustrates a typical CNN architecture. Usually a CNN model is built using a combination of different types of layers, including:

- **Convolution Layer (Conv)**: This layer extracts features from its input through the convolution operation.
- **Pooling Layer (Pool)**: This layer is responsible for subsampling the input with the aim of reducing the computational requirements, memory usage through reducing the number of parameters. Hence, reducing the possibility of overfitting. Unlike Conv, Pool does not effect the weights of the neurons, it rather aggregate the input using an aggregation function such as the maximum or mean, commonly known as MaxPooling or AveragePooling respectively.
- **Fully-Connected Layer (FCL)**: Conventionally, the last few layers of the CNN are fully connected. As the name implies, all neurons in the consequent FCLs are connected such that it calculates the probability of the output being in a certain class based on all the computations from the previous layers. Thus, producing an output from a global perspective, that is, taking the whole input into consideration in the classification process.

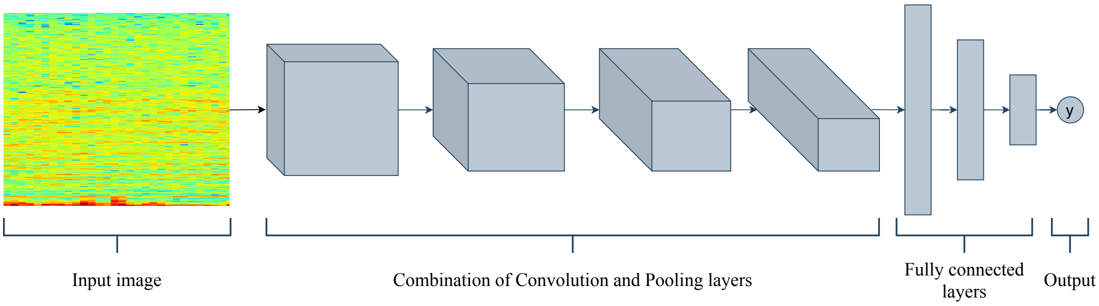


Figure 2. Representation of a general CNN architecture

2.2.2. Recurrent Neural Network

The input in FC and CNN networks is traditionally processed independently of the past and future inputs and therefore they are considered a field of NNs with no memory capabilities. However, in audio and speech recognition applications, there is a distinct relation between the current input and the proceeding events. To make use of this relation, the development of RNN came into play. RNN is a well-known Deep Neural Network (DNN) for detecting and classifying sequential and temporal data. Major applications of RNN lie in speech recognition [30] and video activity recognition domain [31] where the internal memory characteristic of RNN enables it to recall the features. A typical representation of RNN's architecture is illustrated in Fig. 3 where x is the input at every time instant, a is the activation parameter and y being the output.

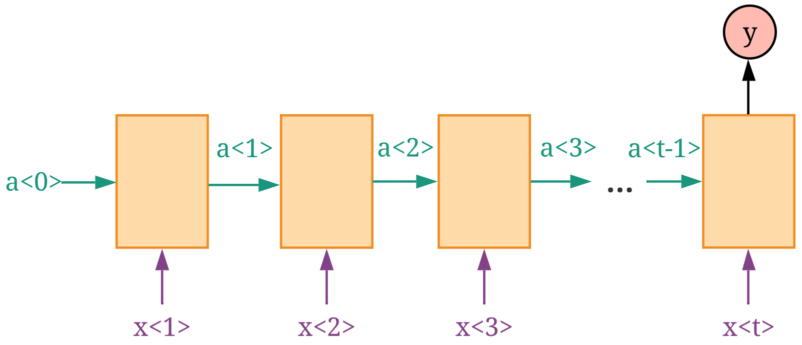


Figure 3. Representation of the RNN architecture

A simple RNN architecture suffers from the concept of vanishing gradient [29][32], which means it becomes harder for the model to remember and maintain information over long time sequences. To solve this issue, an enhanced RNN architecture called Long Short-Term Memory (LSTM) was designed by the authors in [33] to store information through extended time sequences. This is made possible through introducing memory cells, which mainly comprises of: the input gate, the forget gate and the output gate [34]. These parameters are used to calculate the memory cell state. Another advantage of the LSTM is having better performance, faster convergence and ability to detect long-term dependencies in the data [29] in comparison to the simple RNN. Fig. 4 illustrated a simple LSTM example where *x* is the input at every time instant, *a* is the activation parameter, memory cell state is denoted as *c* and *y* being the output. Initially, a<0> is set to zeros or very small random variables. Then, the output of LSTM cell *t-1*, *a<t-1>, c<t-1>*, are fed as an input to LSTM cell at *t* along with input *x<t>*. Hence, the output uses the previous events, which can go further back to *c<0>*, in the prediction process.

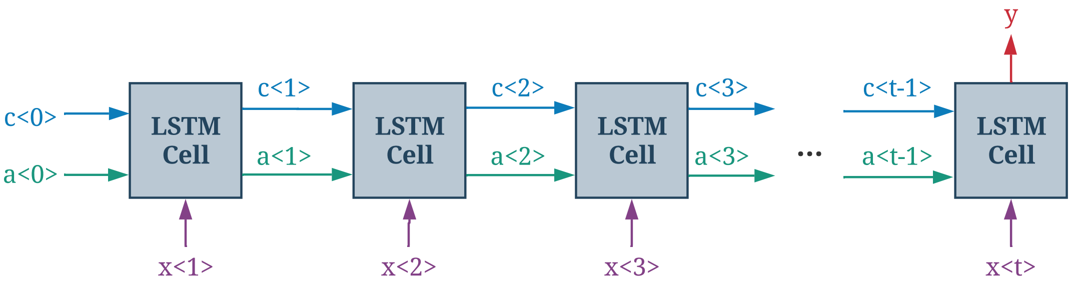


Figure 4. Representation of the RNN-LSTM architecture

Taking the advantage of this unique characteristic of LSTM, in this thesis, the performance of RNN-LSTM is investigated in detecting whether capturing time-based dependencies improves classification performance.

2.2.3. Convolutional Recurrent Neural Network

CRNN is a hybrid architecture made up of CNN and RNN layers [35]. One of the main features of this DL architecture is that it combines the unique characteristics of

CNN, through the utilization of the local temporal or spatial association using the CNN layers, as well as takes advantages of RNN characteristic of finding the global temporal dependencies between the different features [36]. In this thesis, we explore the feasibility of the CRNN architecture proposed in [36] for drone detection and identification. Fig. 5 illustrates the general components of CRNN, starting with a single CNN layer, followed by a series of RNN layers with Gated Recurrent Unit (GRU) as the base cell instead of LSTM unit and a FC layer which is implemented to obtain the output.

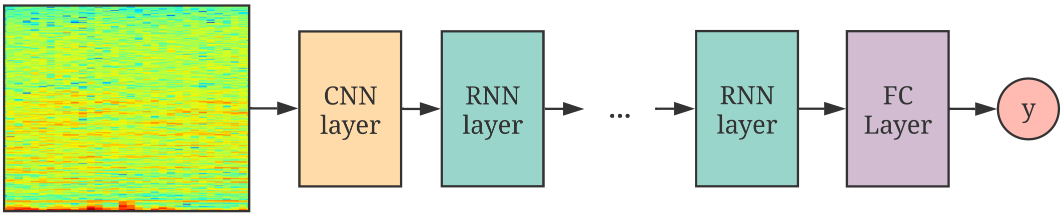


Figure 5. Representation of the CRNN architecture

2.2.4. Generative Adversarial Network

The concept of GANs was first introduced by the authors in [27], where they proposed an unsupervised model build using two different types of neural networks: the Generative model *(G)* and the Discriminative model *(D)*. *(D)* can be any NN classifier such as FC, CNN, RNN, etc. Whereas, *(G)* is a specifically designed NN model for generating new set of synthetic data based on the training dataset fed to it. Fig. 6 demonstrates the procedure undertaken by GAN to generate real-like dataset. Both of these models work to fine-tune their parameters using backpropagation in order for the output of the generative model to sound or look more realistic. The basic idea is that *G* will generate fake data in attempts to fool D into classifying them as real data. When D classifies the fake data as fake, it penalises G, this is achieved through a signal which is redirected to G from D using the backpropagation.

When it comes to training GAN, the first step is feeding a known dataset of pure drone audio and random noise as an initial input to D, in which it achieves reasonable classification performance in differentiating between the drone audio samples and the noise samples. Then, G starts by generating data which are initially random. As the training progresses for both models, the performance of both models improves. G generates better data samples based on the successful attempts of fooling D.

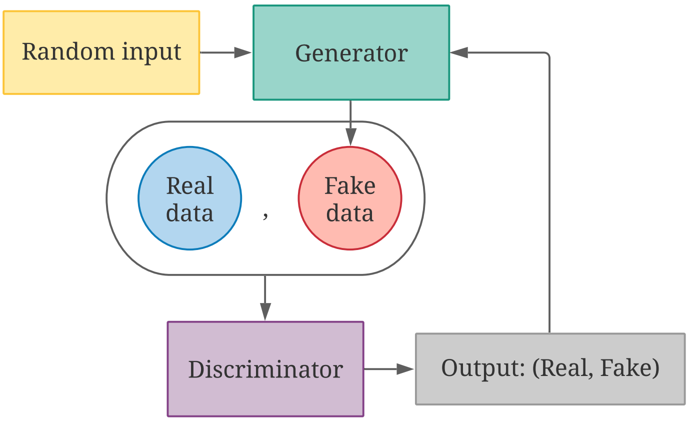


Figure 6. Representation of the GAN procedure

2.3. Performance Evaluation Criteria

Four conventional evaluation metrics, namely: accuracy, F1_score, precision and recall, were selected to evaluate the performance of NN models discussed above. Each of these metrics are calculated using the values of True Positive (TP) , True Negative

(TN), False Positive (FP) and False Negative (FN) which were calculated during the testing phase and represented using the confusion matrix. Table 1 illustrates the general confusion matrix for a binary classification problem.

Table 1. Binary Classification Confusion Matrix

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Each of the selected metrics would provide certain insights on performance of the model which will enhance the evaluation procedure. A short description of each is indicated below:

- **Accuracy:** is the measurement of the ratio of correct predictions to total number of predictions. In a binary classification this can be calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

- **Precision:** is the ratio of the correctly predicted examples to the overall positively predicted examples. This means a model with high precision is able to identify majority of the predicted examples correctly. This relation is illustrated mathematically in Equation 2.2 below:

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

- **Recall:** this metric provides an overview on the sensitivity of the model. That is, the ratio of the positive examples which was correctly identified as positive to the overall positive examples. Equation 2.3 describes this relation further:

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

- **F1 score:** Using precision and recall, the fourth evaluation metric is calculated using Equation 2.4:

$$F1\ score = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision} \tag{2.4}$$

  F1_score is used to show the overall performance of the model in terms of both precision and recall. The advantage of using the F1_score is that it is capable of measuring the overall performance of a model. Moreover, F1 score takes into consideration the distribution of data and the scenario of uneven classes.

CHAPTER 3: RELATED WORK

This chapter presents an overview of previous research on drone detection and identification using audio features in Sections 3.1 and 3.2 respectively. Furthermore, Section 3.3 reviews the latest literature on utilizing GAN in synthetic data generation. In each of these sections, a thorough explanation of the advantages and limitations of each of the solutions described in the literature is provided. Furthermore, a brief explanation is presented on how the solution proposed in this thesis is expected to overcome each of the shortcomings.

3.1. Drone Detection

Several researchers focused their studies on drone detection using audio characteristics. A research was undertaken by the authors in [37], in which a new methodology through using Digital Signal Processing (DSP) to detect the presence of drones in an area was proposed. Similarly, in the study conducted by the authors in [38], a new technique of drone detection was implemented by combining DSP with ML algorithms such as the Support Vector Machine (SVM) algorithm. It was reported by the authors the effectiveness of using SVM in drone detection which have yielded high accuracy, yet, the research was limited to explicit background sounds. Moreover, SVM requires manual extraction and optimization of hand-crafted features to fine tune the model, which is an additional step to the classification solution. From this perspective, DL models have the capability to surpass these shortcomings and eliminate the additional steps required in the conventional ML algorithms by providing an end to end training of the model autonomously [39]. Following this vein, an approach was brought forward in [40] to target drone detection using DSP along with two MLalgorithms, the Plotted Image Learning (PIL) and the K-Nearest Neighbor (KNN). While the algorithms demonstrated their effectiveness and detection ability, yet, the overall accuracy of KNN algorithm reported was remarkably low. The authors argued that this is due to the limitation imposed by the design of the proposed solution and the fact that KNN lacks the ability of building hierarchies of internal representation which could aid in classifying of similar targets . Another shortcoming is derived from the fact that PIL requires a large amount of pre-stored images datasets with a consistently varying background noises to avoid biases and overfitting the noise, thus deploying such a solution in real environment is challenging.

Lately, the effectiveness of implementing solutions based on the DL algorithms has been observed in audio applications, a famous example is the speech recognition [41][42]. However, at the time of writing this thesis, little is known about the utilization of DL techniques in drone detection using drone's acoustic features, in fact, to our best knowledge, the only study found in this field was in [43]. The authors have opted to using the Gaussian Mixture Models (GMM), RNN and CNN for this application. The authors have designed and examined the different machine and DL models to come to a conclusion that RNN has outperformed, in terms of F1 score, the other two algorithms.

3.2. Drone Identification

In regards to the identification aspect of the anti-drone systems, the authors in [44] conducted a remarkable research in which they have utilized DL techniques for non-verbal audio identification. In their research, they studied and examined the implementation of using DL techniques for bird species identification which showed that such mechanism when used to identify bird species based on their acoustic signatures would yield promising results. Being inspired by their work, in this work, we aim to design a drone detection and identification solution using DL techniques, explicitly: CNN, RNN and CRNN, using recorded drone datasets. We extend our study to investigate what role, if any, the introduction and usage of an artificial datasets generated through GAN plays on improving the overall performance of the DL models as well as to verify if the artificially generated data will be good enough to fulfil the shortage gap of drone audio datasets.

3.3. Data Generation using GANs

Recently, GANs have been used extensively in generating new images and photos of people, these images resemble a combination of features extracted from a variety of real human photos and in some cases, these photos have been modified by the GAN algorithm, through changing the hair colour or adding accessories to the human photo for example, to produce new real-like human photos [45]. Similarly, in [46], the authors introduced a new method of generating drums and piano-like audio clips using GAN models through two methods, WaveGAN and SpecGAN. In the former, audio examples are fed as an input to the GAN model, while the latter converts the audio to spectrograms and feed the generated images as inputs to the GAN model. Qualitatively, they evaluated the output of both experiments through human experts, in which the listeners preferred the WaveGAN audio clips over those generated through SpecGAN. To the best of our knowledge, there has been no prior literature that implements GAN architecture to generate drone like audio to enhance detection of drones which we introduce in our work.

CHAPTER 4: PROPOSED FRAMEWORK

In this chapter, we discuss our proposed solution, starting with a thorough explanation of the designed research framework in Section 4.1, followed by the description of the DL algorithms used throughout this work in Section 4.2, and finally we breakdown our proposed drone audio datasets in Section 4.3 which we are releasing to the public and can be found in [47] and [48].

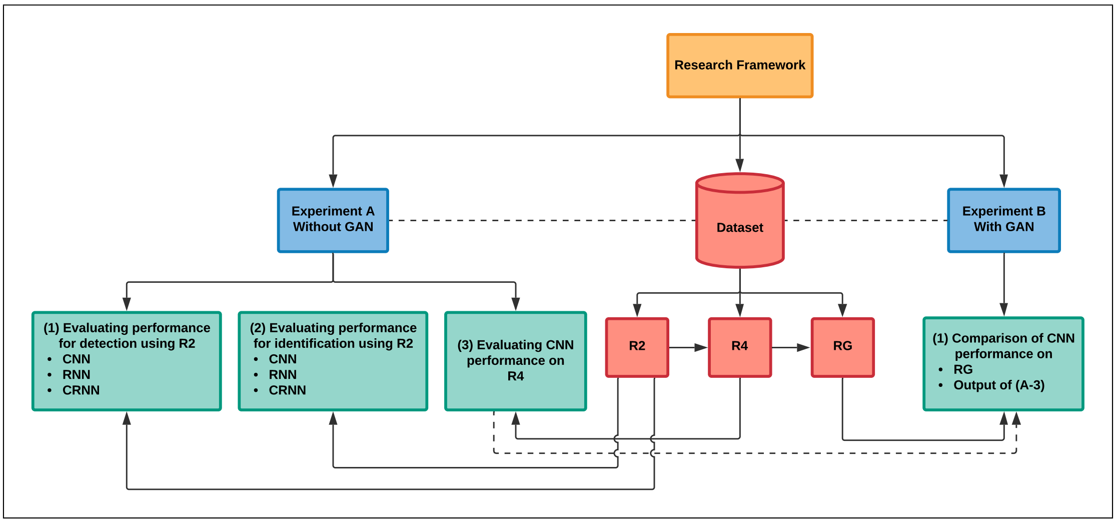

Research Framework
Experiment A Without GAN
Dataset
Experiment B With GAN
(1) Evaluating performance for detection using R2
• CNN
• RNN
• CRNN
(2) Evaluating performance for identification using R2
• CNN
• RNN
• CRNN
(3) Evaluating CNN performance on R4
R2
R4
RG
(1) Comparison of CNN performance on
• RG
• Output of (A-3)

Figure 7. High level design of the proposed framework. Where R2 dataset consists of the recorded audio of two drones, R4 dataset consists of the recorded audio of 5 drones, one of which is reserved for testing purposes, and RG dataset is a hybrid dataset of GAN generated dataset and R4.

4.1. Research Framework

Fig.7 below illustrates the design of the experiments that will be carried out throughout this research. In order to implement the proposed solution using DL techniques, large amount of drone audio data were required. However, due to various reasons such as privacy, there were no public drone audio dataset available for this application in literature as of the time of writing this thesis. Hence, in experiments **A.1** and **A.2**, we have created our own drone audio dataset by acquiring, through audio recording using a microphone, more than 1300 audio clips of drone sounds. These clips can be found in [47]. Moreover, to mimic real life scenarios, we have used the publicly available noise datasets [49] and [50] to artificially augment the drone audio clips with noise. The main purpose of the artificial augmentation is to measure the feasibility of the system in a noisy environment. In addition to training the DL algorithms, CNN, RNN and CRNN, on the augmented sound clips, we have dedicated a portion of the dataset to include pure noise, silence and pure drone audio clips in order to ensure that the system will be able to detect and identify the drone's sound from similar noises in an environment. Throughout this work, we will be referring to this dataset as **R2** as it consist of audio clips of *two* drones.

We further expand, through this study, our dataset to incorporate other types of drones with an aim of consequently increasing the diversification of the dataset. The new drone audio clips were collected from a variety of open-source YouTube [51] drone videos [52]–[57]. We cleaned and preprocessed the acquired audio clips using similar techniques as those used for R2, to produce an enhanced drone audio dataset that incorporates five distinct drones, *four* of which are used in the training of the DL classifier and the remaining one drone is reserved for the testing phase. Hence, this dataset is referred to as **R4**. Then, by conducting experiment **A.3**, we evaluate the performance of CNN on the enhanced dataset.

By expanding the R2 dataset to R4 we aim to increase the diversification of the dataset. We speculate that the added drone types will ensure that the artificial dataset generated from the GAN model through experiment **B** is less biased towards a specific drone type. The dataset generated through the GAN model using R4 is referred to **RG** given that it is made up of recorded and GAN generated drone audio clips.

In experiment **B.1**, we evaluate CNN performance on the RG dataset in order to establish whether adding an artificial dataset to recorded drone audio dataset would improve the overall performance of the model by comparing the outcomes of this experiment to those found in experiment **A.3**.

4.2. Deep Learning Algorithms

To implement the drone detection and identification solution, we have used the open-source code available in [36] to build our RNN, CNN and CRNN models. This code is an enhanced version of TensorFlow's open-source tutorial [41]. In our implementation, the default values were used for the models' architectures and hyperparameters per the original authors' setup. However, it is important to note that we have modified the code to suit our application by incorporating the validation termination condition (further discussed in Section 5). Furthermore, given that an essential part of designing the drone detection is to bridge the gap of the shortage of drone acoustic dataset, we will attempt, in this work, to build a system based on GANs to generate new artificial drone audio clips using a slightly modified version of the WaveGAN code in [46]. The selection of WaveGAN over SpecGAN in our solution was based on the recommendation and comparative study outcome provided in [46].

4.3. Dataset

In this section, we will explore the different types of datasets created as part of the solution proposed through this work. Starting with the R2 dataset breakdown in Section

4.3.1, followed by the R4 dataset in Section 4.3.2 and concluding this section with RG dataset 4.3.3.

4.3.1. R2: Recorded Drone Audio Dataset

4.3.1.1. Data Acquisition

To acquire the drone's audio, we have recorded, using a microphone embedded within a smart-phone, the sound generated by the drone's propellers while flying and hovering in a quiet indoor environment. This enabled us to publish the dataset publicly without breaching any privacy regulations. Furthermore, we acquired a balanced number of audio clips per drone with equivalent time intervals to ensure that the audio clips will be equivalently random when fed to the algorithm to avoid any biases. This process yielded in a total audio clip of 11 minutes and 6 seconds per drone formatted in MPEG-4 audio format (m4a) with a sampling rate of 44.1KHz and bitrate of 64Kbps.

4.3.1.2. Data Preprocessing

In order to prepare the audio files for the selected DNNs, firstly, we reformatted the output audio clips produced from the microphone's recording and the background noise clips by converting audio file type to WAVE, sampling rate to 16KHz, bitrate to 16Kbps and the channel to mono to ensure consistency.

Secondly, we divided the formatted audio files into multiple short (one second) segments by specifying the time intervals at which the audio clip will be segmented, this will enable the DL algorithm to optimize the training of the model for real-time deployment in which the time required for the detection and identification is critical. Hence, to investigate whether the size of the audio segment affects the overall performance of the classifier, we have experimented with multiple segment sizes such as one, two and five second segments. Based on our heuristic observations, we deduced that the one second segmentation was sufficient.

One possible way to train ML or DL algorithm on audio input is by converting the audio clips into spectrograms [41]. Various features are then extracted from the generated spectrograms by the algorithm to train the DL models. To illustrate the outcome of this process, Fig.8 represents a one second example of a drone's audio. Whereas, Fig.9 represents an audio clip of a random noise such as a person typing.

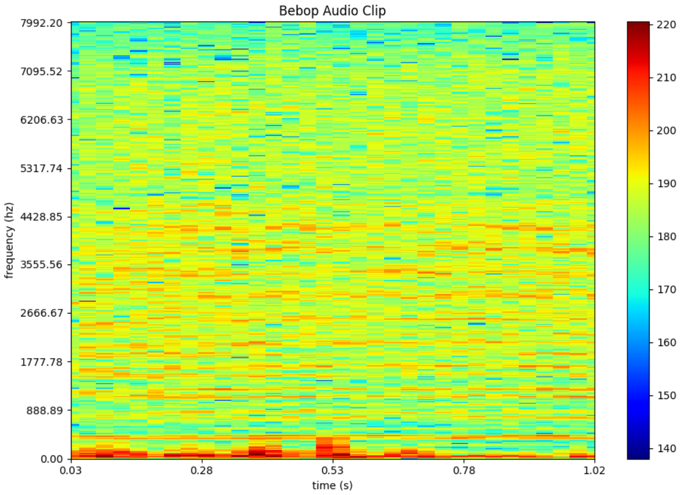


Figure 8. Example of drone noise in spectrogram representation

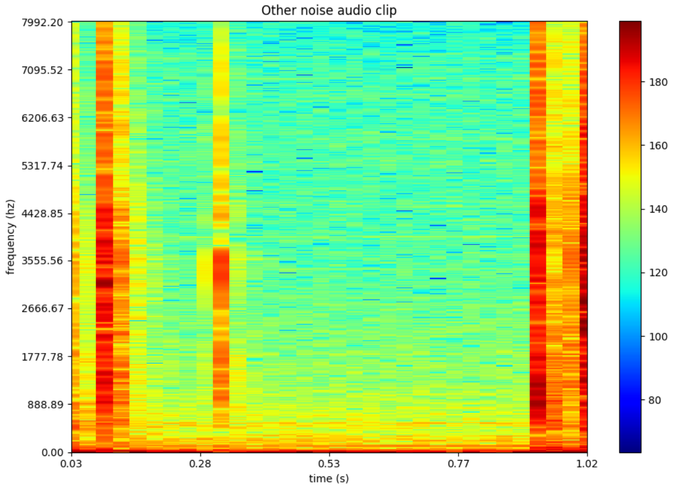


Figure 9. Example of other noise in spectrogram representation

4.3.1.3. Data Augmentation

Since the application of drone detection and identification could be deployed in areas with a variety of background noises, we have approached the problem by introducing a method of augmentation, in which a real-life background noise is overlapped with the drone audio without any modification on the actual audio features, such as the amplitude or the frequency of the audio clip. Particularly, we have used the background noise from the publicly available dataset [49] [50]. However, it was rather important to reformat the audio clips acquired from these datasets as discussed in Section 4.3.1.2 to ensure the consistency of the audio files. Using this mechanism enabled us to mimic real-life scenarios.

4.3.1.4. Data Labeling

We have collected our drone acoustic data for two commercially available drones, **Bebop** and **Mambo**, manufactured by Parrot. This leaves us with **R2** dataset, which represent those *two* drones. For the identification problem, we have labelled our dataset, [47], simply as **Unknown** for other noises in an environment, **Bebop** as the first drone and **Mambo** representing the second drone. The distribution of audio clips acquired per label is represented in Table 2.

Table 2. Data per label

Type of Drone	**Records**		
	Original	***Augmented***	***Total***
Bebop	331	335	666
Mambo	331	335	666

Similarly, for the detection aspect of this system, we have combined the data collected for both Bebop and Mambo drones as one entity and labelled it as **drone** and any other audio clip was labelled as **not a drone**.

4.3.2. R4: Enhanced Recorded Drone Audio Dataset

As discussed in the introduction of Chapter 4.1, we expanded our drone dataset which initially consisted of two drones to incorporate other types of drones from a variety of manufacturers to be be used in the drone detection experiment. The drone audio clips were collected from a variety of open-source YouTube drone videos [51] in both indoor and outdoor environments [52]–[57]. The additional drones selected are:

- **3DR Solo**
- **DJI Phantom 4**
- **AR Drone**

The selection of acquiring drone audio in two different locations, indoor and outdoor environment, was considered mainly to avoid the data augmentation process mentioned in Section 4.3.1 above. We have manipulated the raw videos by, firstly, converting them into audio files. Secondly, we selected the relevant sections from the entire audio for this application. Finally, we divided, cleaned and preprocessed the collected drone audio clips using the same techniques as those used for R2 in Section 4.3.1 to produce the final **R4** dataset, which stands for four recorded drone audio dataset. The addition of these drones contributes in increasing the diversity of the R2 dataset in order to be used later on in the production of a hybrid version of the dataset using GANs.

To better understand the behaviour of the classification models on different combinations of drones, where some might be more difficult to detect than others due to their audio features' nature, we have divided our recorded drone audio dataset into five different groups referred to as ***D*** experiments, each with a different combination of drones illustrated in Table 3. In every experiment, a single *unseen* drone was not used in the training phase of the classifiers. The audio clips of the *unseen* drone is left for an exclusive testing in order to observe if the models can generalise well beyond the four drones seen during the training phase.

Table 3. Enhanced Drone Audio Dataset

Drone Type	**Experiment**				
	D1	***D2***	***D3***	***D4***	***D5***
Bebop	Unseen	✓	✓	✓	✓
DJI Phantom 4	✓	Unseen	✓	✓	✓
3DR Solo	✓	✓	Unseen	✓	✓
Mambo	✓	✓	✓	Unseen	✓
AR Drone	✓	✓	✓	✓	Unseen

In each of the *D* experiments, the seen drones were grouped together and labelled as **drone** and the same noise audio clips mentioned in Section 4.3.1 above were used and labelled as **not a drone**.

4.3.3. RG: Hybrid Drone Audio Dataset

To generate the artificial drone audio dataset we have implemented a GAN model based on WaveGAN architecture described in [46]. We fed the algorithm with long pure drone audio clips which were, explicitly, recorded in an indoor environment for each of

D experiments mentioned in Section 4.3.2 above. It is important to note that in every *D* experiment, the *unseen* drone was not exposed to the training of the GAN model nor in the training phase of the classifiers. After training, the GAN model generated 200 artificial drone audio clips with a duration of one second each. Figures 10 to 13 shows the training of each GAN model in each of the D partitions. Moreover, we have used the loss function as calculated in [46]. The termination of the training was based on heuristic observation at the instance where both the discriminative model and the generative model loss converges.

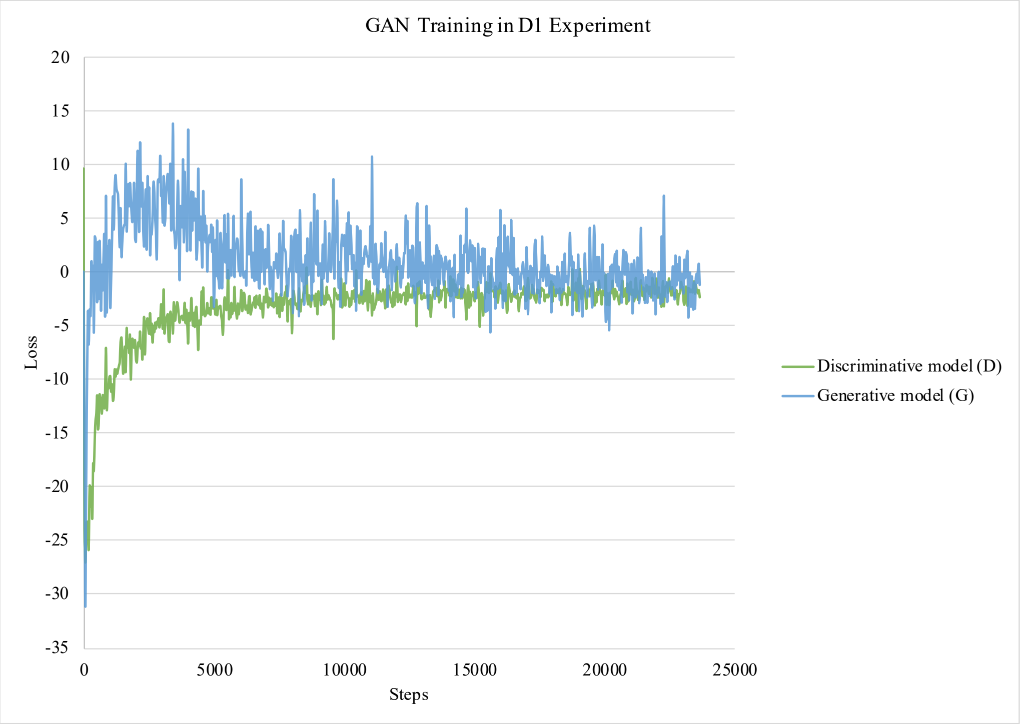


Figure 10. GAN training graph for D1 experiment

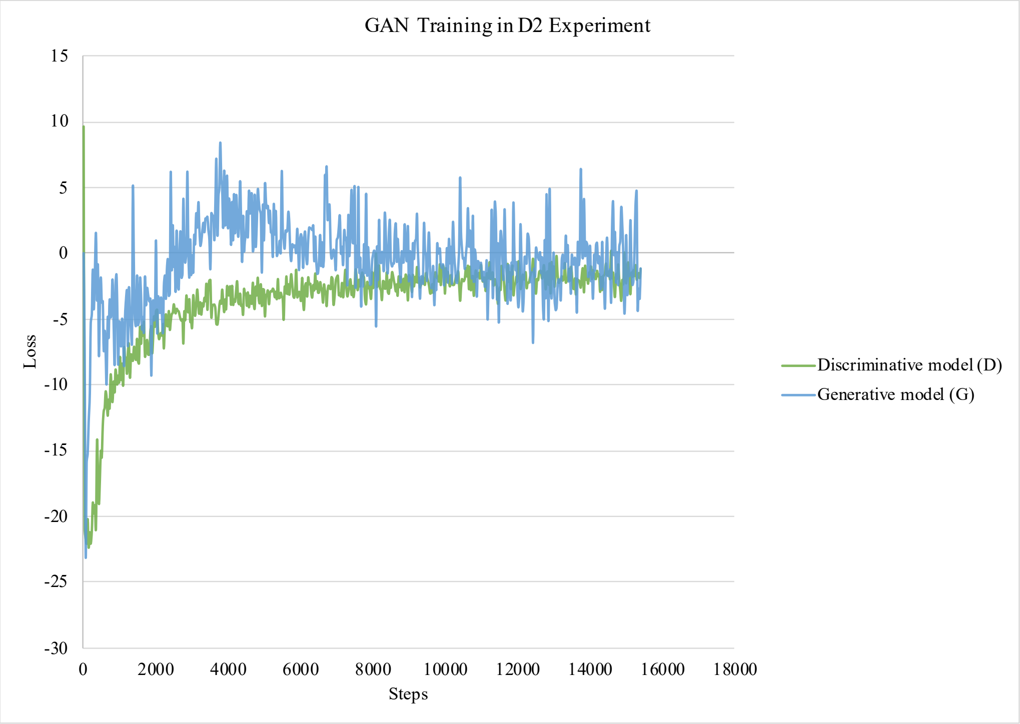


Figure 11. GAN training graph for D2 experiment

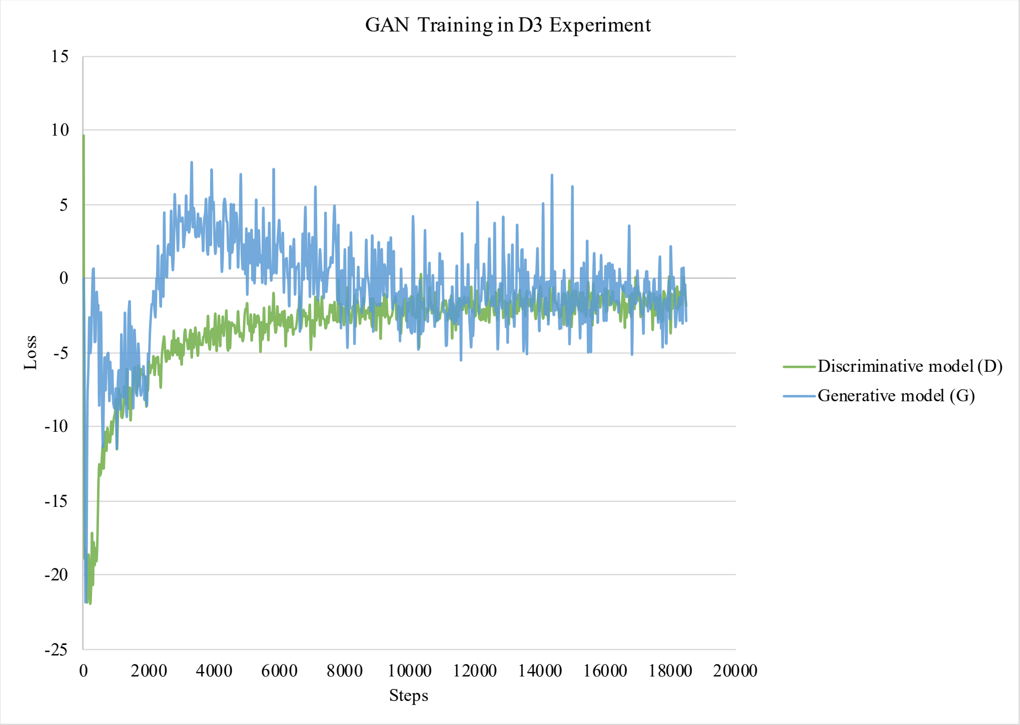


Figure 12. GAN training graph for D3 experiment

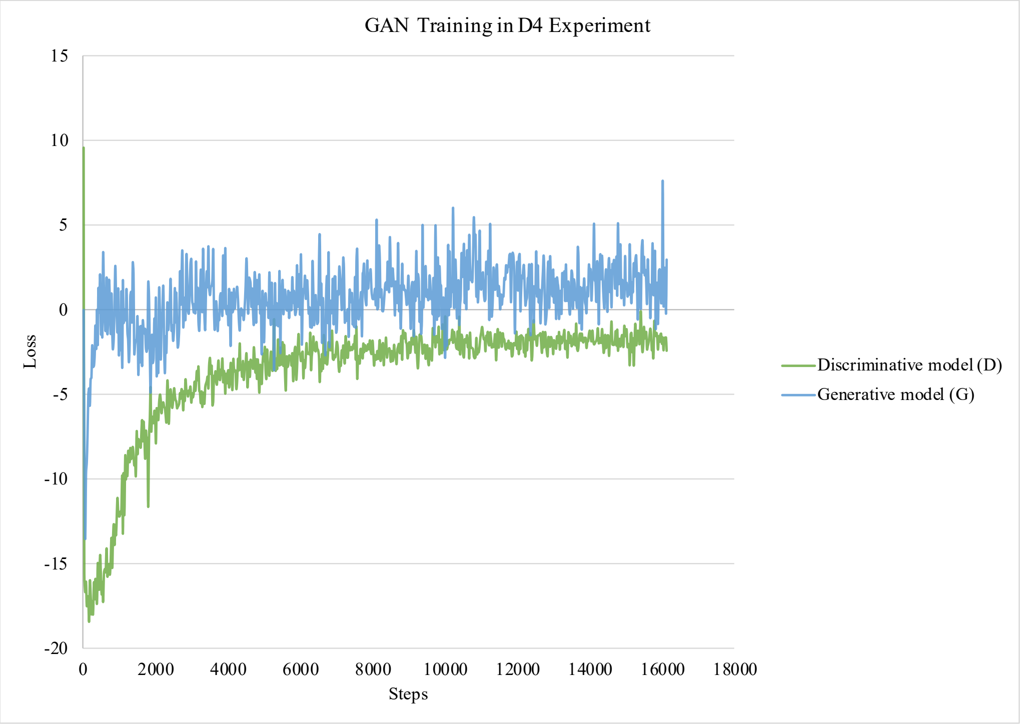


Figure 13. GAN training graph for D4 experiment

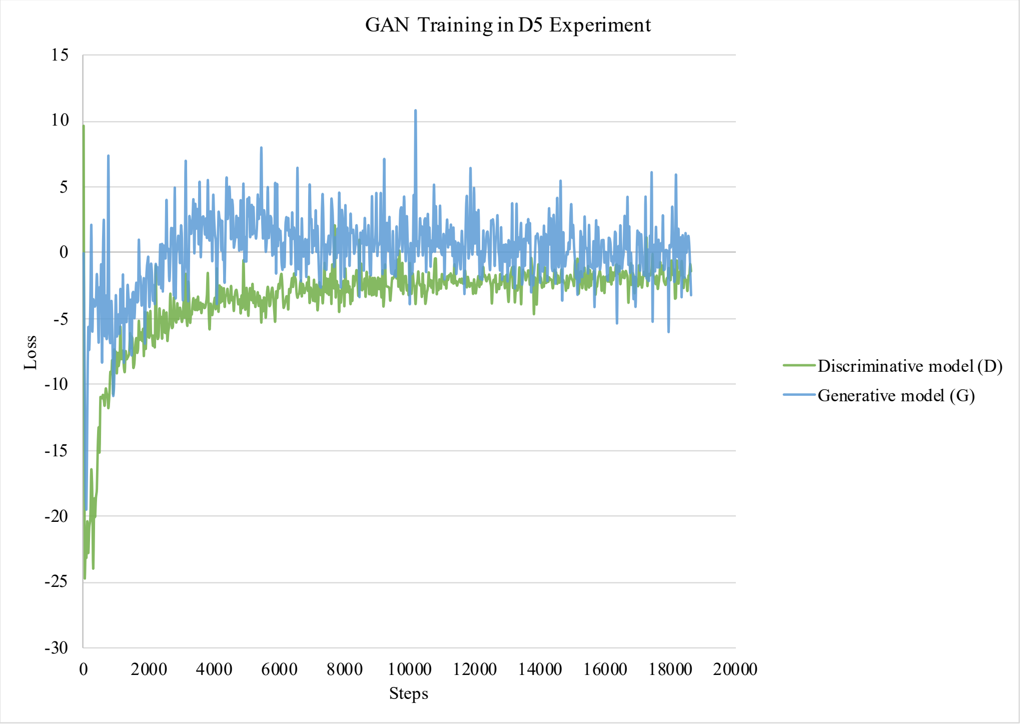
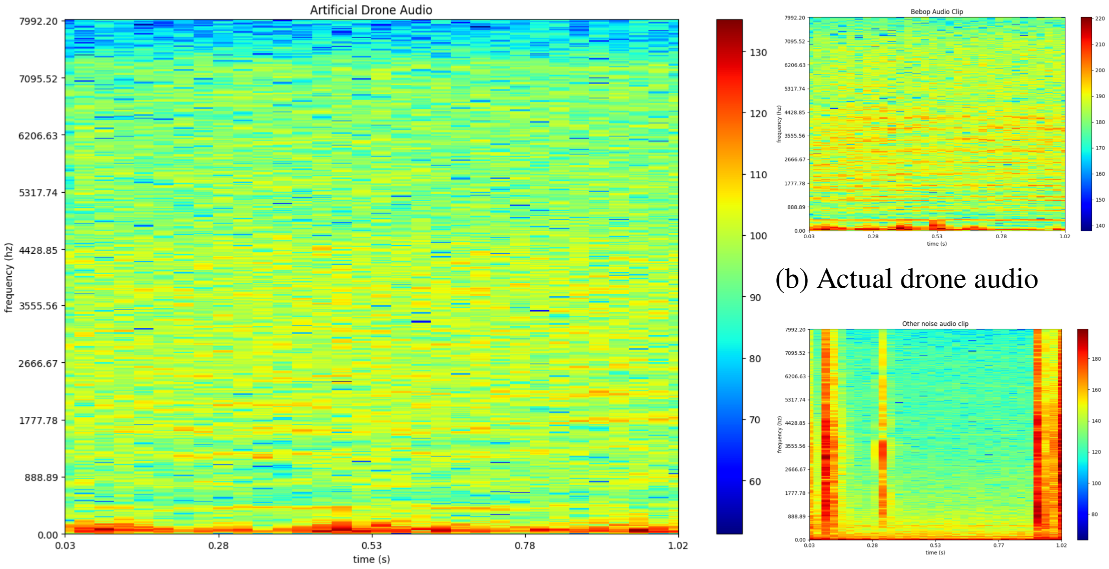


Figure 14. GAN training graph for D5 experiment

A sample output of the artificial drone audio generated through the GAN is illustrated in Fig.15a. As it can be observed, the synthetic audio's spectrogram looks very similar in terms of features to the one of the recorded drone audio shown in 15b. Whereas, it is distinctly different to the other noise audio clip illustrated in 15c.



(b) Actual drone audio

(c) Other noise

(a) Example of GAN generated drone audio in D2

Figure 15. Audio clips comparison in spectrogram representation

Furthermore, we have carried out human-hearing tests with a number of volunteers to test how different the GAN generated audio clips are in each partition. It was concluded that a distinguishable difference was recognised in the sound generated for each of the *D* partition. Those artificially generated audio clips were then combined with R4 drone dataset resulting in what we refer to as RG, which stands for recorded and GAN drone dataset. Table 4 shows the proportion of each in the RG dataset.

Table 4. RG Drone Audio Dataset

Audio Type	**Experiment**				
	D1	***D2***	***D3***	***D4***	***D5***
Recorded Drone Clips	868	1331	1248	868	1288
GAN Drone Clips	200	200	200	200	200
Total	1068	1531	1448	1068	1488

Moreover, each set of the recorded drone clips and GAN drone clips for every *D* experiment illustrated in Table 4 were grouped together and labelled as **drone** and the same noise audio clips mention in Section 4.3.1 above were used and labelled as **not a drone**.

CHAPTER 5: EXPERIMENTAL SETUP

This chapter presents an explanation of the experiments setup carried throughout this thesis. It begins by describing the experimental setup of drone detection and identification using R2 dataset experiments, also known as **A.1** and **A.2** in Fig.7 of Section 5.1. This is followed by Section 5.2, which provides a description of the experimental setup of drone detection using R4 and RG datasets experiments, indicated as **A.3** and **B.1** in Fig.7.

5.1. Experiments **A.1-2**: Drone Detection and Identification using R2 Dataset

As already noted in Fig.7, we started our experiments by investigating the performance of the DL models in drone detection and identification as shown in **A.1** and **A.2** parts of the diagram in Fig.7. This means that our initial experiment was divided into two categories, the first, **A.1**, being the binary classification experiment in which we assess the DL algorithms in their ability to detect whether a drone is present or not. Hence, we have defined this experiment to handle two use-cases, which are either (a) a drone was detected or (b) no drone in the area.

The second category, **A.2**, is the multi-class classification experiment, where we measure the performance of the DL algorithms to identify which type of drone was detected. In this experiment, there exists three distinct labels, namely **Bebop**, **Mambo** and other **Unknown** noises, to identify drones based on their type as mentioned in Section 4.3.1.4.

The details of the environment setup at which we deployed the algorithms, trained the models and carried out the experiments are indicated in Table 5.

Table 5. Experiments A.1-2 Environment Setup Details

Operating system	Ubuntu 18.04-Linux
CPU	Intel(R) Xeon(R) x86_64 CPU E5-2695 v4 @ 2.10GHz
Number of CPU	36
Framework/APIs	Python 2.7 and Google TensorFlow API

It is important to emphasise that the three main objectives we are attempting to achieve through these experiments are, firstly, we are interested in investigating the performance of the binary classification aspect of the problem and comparing the results with the literature. Secondly, we aim at observing and evaluating the outcome of the multi-class identification classification aspect of the problem and show the best algorithm out of the three algorithms implemented, namely; CNN, RNN and CRNN, using the different evaluation metrics. Finally, we aim to answer the question of which algorithm has the lowest training and evaluation time.

In order to accomplish these objectives, we have chosen to evaluate and compare the different algorithms based on their accuracy, F1_score, precision and recall metrics as mentioned in Section 2.3. Additionally, we have also considered the computational time (CPU time) required to train and test the model as an attribute in evaluating the performance of the models.

Furthermore, we have experimented with several combinations for the ratio of training to testing datasets and we have deduced from those experiments that the variance of this ratio had minor effect on the overall performance of the model. Therefore, given that the difference is negligible, we opted to use the typical combination of 70:30.

Moreover, we have defined the distribution of the labelled data for the binary classification as well as the multi-class identification problem as presented in Table 6. The values of these parameters are defined as a result of handful of experiments in which we have observed the model's output by looking at the F1_Score and the accuracy in the validation phase and systematically tuned these parameters to find the optimal distribution of the dataset. Also, the optimal learning rate found was 0.01.

Table 6. Details on Data Distribution

Criteria	**Parameter**
Unknown audio files	50%
Binary Classification Problem	
Drone audio files	50%
Multi-class Classification Problem	
Drone 1 - Bebop	25%
Drone 2 - Mambo	25%

5.2. Experiment **A.3** and **B.1**: Drone Detection using R4 and RG Datasets

In this experiment we aim to determine whether adding artificially generated drone audio-like data to our recorded drone data has an effect on the detection performance. More specifically, would the integration of GAN generated drone audio dataset improve the performance of a DL classifier. Our hypothesis is that the hybrid dataset would add a generalization element which will, consequently, have a positive impact on the overall performance of the classifier.

To train the CNN classifier on the R4 dataset as mentioned in experiment **A.3** of Fig.7 for the *seen* drones experiment, while excluding the *unseen* drone, the dataset distribution in Table 7 was used.

Table 7. R4 Drone Audio Dataset

Data Type	Training	Validation	Testing
Percentage	80%		20%
	70%	30%	

Table 8 shows the individual proportions of the RG dataset for each of the *D* experiments conducted in **B.1** experiment in an *unseen* drone scenario.

Table 8. RG Drone Audio Dataset

Experiment	Training		Testing
	Training	**Validation**	
	70%	30%	
D1	83%		17%
D2	82%		18%
D3	81%		19%
D4	83%		17%
D5	82%		18%

In order to implement the experiments proposed in this section, an environment setup as shown in Table 9 was used in the training, validation and testing phases of the DL models.

Table 9. Experiments A.3 and B.1 Environment Setup Details

Operating System	Ubuntu 18.04-Linux
GPU	Nvidia Titan V
CPU	Intel(R) Xeon(R) x8664 CPU E5-2695 v4 @2.10GHz
Number of CPU	36
Framework/APIs	Python 3.7 and Google TensorFlow APIs

CHAPTER 6: PERFORMANCE EVALUATION AND DISCUSSION

Using the experimental setup discussed in the previous chapter, in this chapter, the experiments are carried out and a detailed discussion on the performance of the proposed solutions are presented for drone detection and identification using R2 dataset in Section 6.1, drone detection solution using R4 Dataset in Section 6.2 and the performance of the drone detection solution using R4 Vs. RG dataset in Section 6.3.

6.1. Experiments **A.1-2**: Drone Detection and Identification using R2 Dataset

In order to ensure that every algorithm is performing at its optimum, we have carefully chosen the steps below to define the termination condition of the training phase:

1. Executing the algorithm with a very large number of training steps.
2. At an interval of 100 steps, the trained model is tested on the validation-set and the accuracy is calculated and recorded.
3. We compare the new accuracy of the validation-set with the best accuracy achieved so far.
4. If the accuracy did not improve over three successive validation tests, we test the trained model on the testing-set and report the observed results.

An example of training and validation of CRNN in a single run for the binary classification are illustrated in Fig.16. It can be observed from the graphs that the termination condition selects the model at the best validation accuracy. Hence, the training can terminate before the model overfits the data. The same can be observed in the example of the multi-class classification training and validation phases for and CRNN in Fig.17.

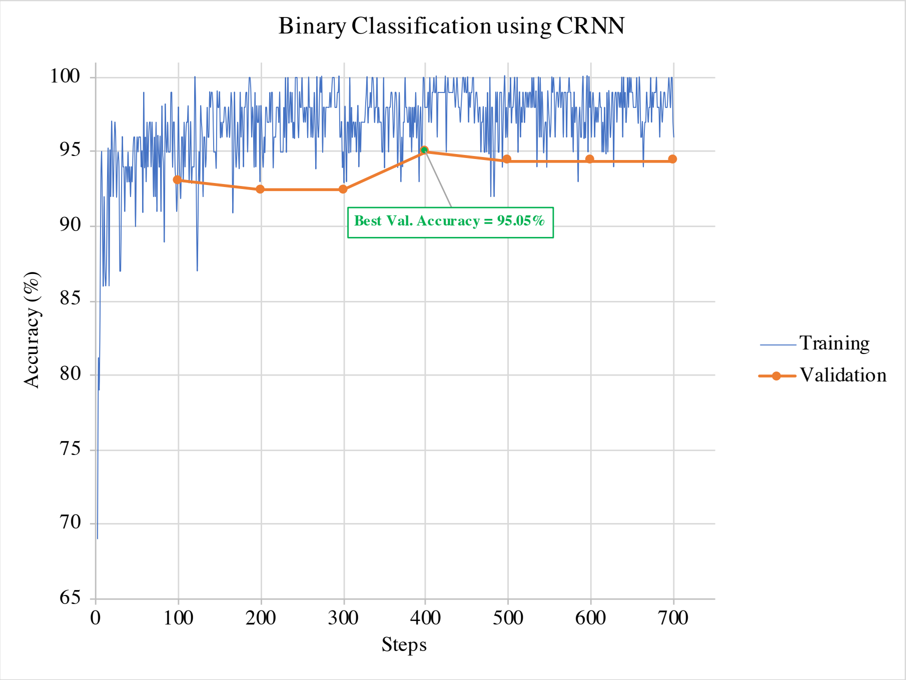


Figure 16. Example of the training and validation phases of CRNN for binary classification in a single run

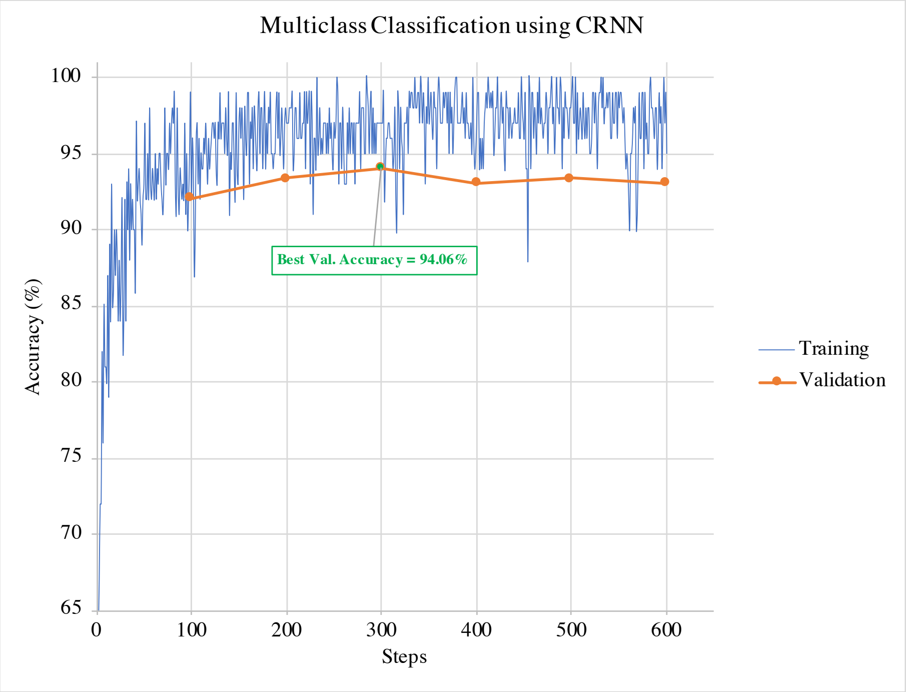


Figure 17. Example of the training and validation phases of CRNN for multi-class classification in a single run

Given that the training, validation and testing datasets were shuffled randomly at the start of every execution of learning, we repeated each experiment ten times. Hence, the values discussed in Section 6.1.1 and Section 6.1.2 represent the average results of the ten runs.

*6.1.1. **A.1**: Drone detection: Binary classification results*

In this experiment, we have examined the effectiveness of our proposed system in detecting drones using their acoustic signatures. We have calculated the evaluation metrics for the three different models in addition to the corresponding standard deviation values for the 10 runs as illustrated in Table 10 below.

Table 10. Detection Results

Evaluation Metric	**RNN**	**CNN**	**CRNN**
CPU-Time (s)	333.45±60.90	957.33±320.01	487.53±178.75
Accuracy (%)	75.00±6.60	96.38±0.69	94.72±1.36
Precision (%)	75.92±10.30	96.24±0.81	95.02±1.14
Recall (%)	68.01±7.59	95.60±0.84	93.08±1.98
F1-score (%)	68.38±8.16	95.90±0.78	93.93±1.61

It can be deduced from Table 10 that CNN have outperformed RNN with a relative improvement of 21.38% in accuracy, 20.32% in F1 score, 20.32% in precision and 27.59% recall. However, the average overall training time required for CNN to yield such precise results was much higher in comparison to RNN. Whereas, it was observed that RNN had the lowest training time and overall performance among the three model. Additionally, the performance of CRNN in all evaluation criteria was better than RNN. It is important to take into consideration that the nature of RNN algorithm is best suited for sequential data. Even though, CRNN did not perform better than CNN, the difference between the performance of both models was negligible, in which CNN have shown an improvement of 1.66% in accuracy, 1.98% in F1 score, 1.21% in precision and 1.98% in recall, yet, CRNN was noticeably faster than CNN by 49.07%. This is an interesting finding because it can guide practitioners to consider the model with a lower training time without sacrificing the performance of the model.

In addition to evaluating the performance of the three different models we proposed to detect drones, we aimed to compare the output of the system with similar implementations from the literature. As of the time of writing this thesis, only one source [43] was found that targets the same problem using sound *detection* approach.

The results found in detecting drones using our approach contrasts with the results found in the literature by the authors in [43]. The authors of [43] noted that RNN have achieved the best performance in comparison to CNN. Whereas our results do not support their observation. In fact, we have deduced from our experimental results that CNN have outperformed RNN remarkably. There are a number of factors which might have contributed to the difference of the outcomes between the two approaches such as tuning the algorithm parameters by the authors on the testing-set directly rather than using a validation-set to serve this purpose. Moreover, the discrepancies in our findings can be attributed to the difference of the models' architecture and design parameters such as the number of the convolutional layers used in the CNN algorithms in both applications. Due to the lack of availability of their training and testing datasets, we were not able to perform a direct comparison between the results of both approaches.

Although the results yielded from our proposed experiment do not align with those found by the authors [43], it can nevertheless be concluded that both approaches agreed on the great effectiveness of using DL in drone detection using acoustic features.

6.1.2. ***A.2****: Drone identification: multi-class classification results*

The main goal of this experiment is to examine the effectiveness of the DL methods in identifying drones based on their acoustic signatures. We have used the evaluation metrics mentioned in Section 2.3 to examine the performance of the three models in the multi-class problem. Moreover, it is worth mentioning that the final results were calculated by taking the macro-average over all the classes in the experiment. The overall results of the evaluation metrics are presented in Table 11.

Table 11. Identification Results

Evaluation Metric	**RNN**	**CNN**	**CRNN**
CPU-Time (s)	389.02±73.18	807.10±278.09	605.67±252.83
Accuracy (%)	57.16±11.33	92.94±11.89	92.22±1.03
Precision (%)	59.64±13.56	92.75±1.26	92.54±0.95
Recall (%)	57.16±11.27	92.63±1.32	92.23±1.03
F1-score (%)	55.62±13.53	92.63±1.32	92.25±1.01

Results that emerge from this experiment have shown that the results of both CNN and CRNN are outstanding with accuracy, precision, recall and F1 score all above 90%. Moreover, we have observed that CNN have outstandingly outperformed RNN by an improvement of 35.78% in accuracy, 37.01% in F1 score, 33.11% in precision and 35.48% in recall. However, although RNN have shown the worst performance, it converged faster than CNN by 51.80% and than CRNN by 35.77%. In addition, it can be observed from the standard deviation values in Table 11 that RNN was the fastest to converge regardless of the difficulty of the dataset. Furthermore, it is suspected that the weak performance of RNN algorithm was due to the nature of algorithm since it is mainly based on time-dependent trend which is not the case in this experiment as the audio clips used are of a short length in which they have a constant distance with less variation over time.

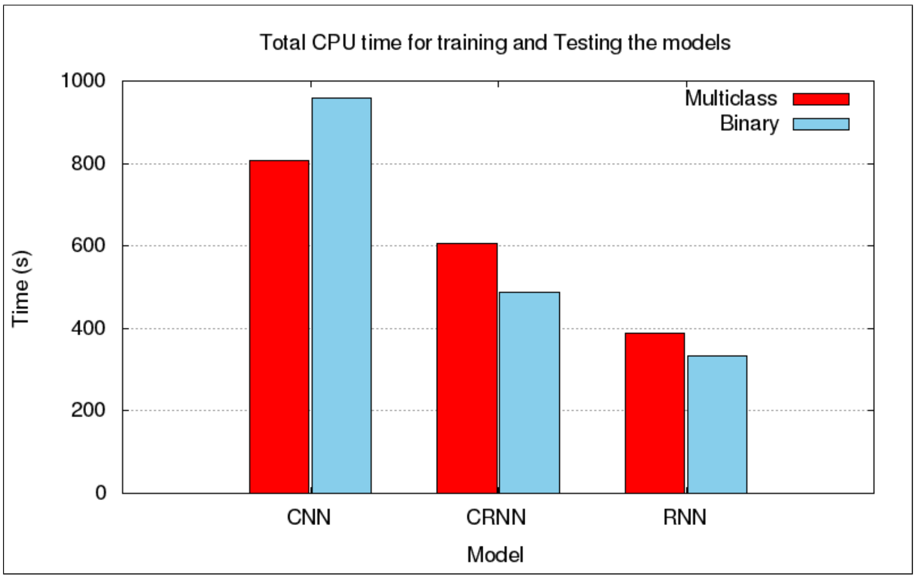


Figure 18. CPU time Results

Moving on to the comparison between the performance of CNN and CRNN, we have observed that CNN have also performed better than CRNN by 0.72% in accuracy, 0.39% in F1 score, 0.21% in precision and 0.40% in recall. Although CNN have shown some improvement in the performance, one can deduce from the standard deviation values reported in Table 11 that the performance of CRNN is more robust in comparison to the other algorithms regardless of the data fed to the algorithm. Moreover, CRNN was significantly faster by 24.96% in execution time than CNN. This finding, as illustrated in Fig.18, provides a conclusive support for the results found in Section 6.1.1, since in both detection and identification aspects of the problem, it had been observed that practitioners can still utilize a model with significantly fast computational time without jeopardizing the overall performance of the model.

In addition to the results presented in Sections 6.1.1 and 6.1.2, we have observed that the system was able to identify different drones and other noises while maintaining the precision in the evaluation metrics per label. Table 12 summarizes the average performance of the 10 runs for each label in terms of F1 score for the CRNN model. The results presented below suggest that the proposed method has the ability to adjust its identification feature to accommodate more labels based on its application without sacrificing or degrading the performance per label.

Table 12. F1 scores per label for CRNN

Label	Unknown	Bebop	Mambo
F1 Score	92.766%	93.78%	90.192%

Based on the findings in this experiment, where CNN have outperformed while being the most stable algorithm among the other two DL algorithms in drone detection using acoustic features, we will proceed with CNN as our selected DL algorithm for experiments **A.3** and **B.1**.

6.2. Experiment **A.3**: Drone Detection using R4 Dataset

In this experiment, ten CNN models were trained on R4 dataset then tested in every *D* partition using the testing dataset described in Section 5.2. The first testing dataset included, exclusively, drones types that the model has *seen* during the training phase.

Whereas, the second scenario consisted of the remaining previously *unseen* drone types as the testing dataset. This type of testing was necessary in order to better understand how the model performs when faced with a completely new drone which it was not exposed to during the training phase. For this reason, we have extended our experiments to study such behaviour. Furthermore, in order to ensure the optimal performance of each CNN model, we have followed the same four steps mentioned in Section 6.1 that defines the termination condition of training the model. The outcome of this series of experiments is illustrated in Fig.19.

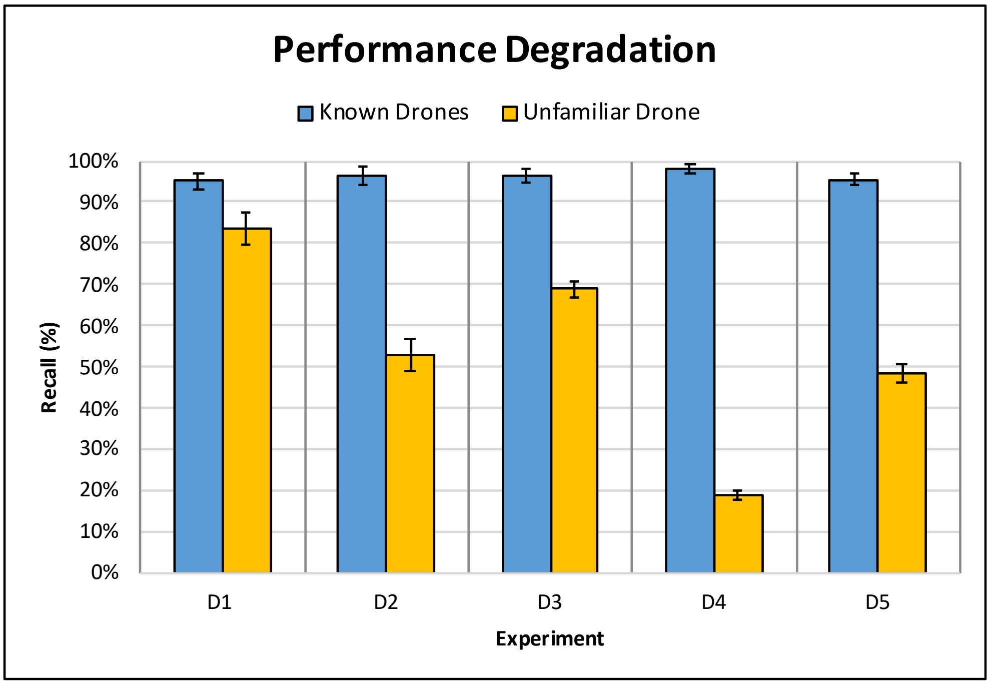


Figure 19. The performance,in terms of *recall*, of the average CNN models trained on the R4 drone dataset and tested on known (recorded) drone types (which the model has seen during the training phase). Whereas, the yellow bars are when tested on new and unfamiliar types of drones.

The results presented in Fig.19 clearly indicate that there is a negative performance hit when the model is used to detect the presence of a drone it has never seen before (not included in the training set). Hence, it had been deduced from this experiment that the findings are consistent with our initial problem statement. Therefore, in the next experiment, **B**, we attempt to improve the performance of the CNN model on *unseen* drone by using the hybrid dataset, RG.

6.3. Experiment **B**: Drone Detection using R4 Vs. RG Dataset

As observed from experiment **A.3** previously, there is a noticeable degradation of the CNN model performance when faced with an unseen drone, hence, what we aim for by conducting this experiment is to investigate and understand whether a hybrid dataset such as RG which consist of GAN generated drone-like audio and an actual recorded drone would have an positive impact, if any, on the overall performance of the DL models.

In order to guarantee the optimal performance of the CNN models, we have followed the steps mentioned in Section 6.1 to terminate the training phase of the model. It is important to note that, as of the time of writing this thesis, no work was found on using GAN generated data to improve the performance of DL models in audio applications, hence, we put forward a novel concept to explore.

Following our initial hypothesis in which we assume that the hybrid dataset RG would improve a generalisation of our classifier, hence, it would improve the overall performance of the classifier, we divided this experiment into two sections which enable us to compare the performance of the CNN models trained on drone audio dataset without GAN, using the R4 dataset, and with GAN generated audio dataset using RG dataset.

In experiment **A.3**, we carried out the performance evaluation experiment of the

CNN models on R4 dataset to acquire the performance of the CNN model for drone detection through training and testing the proposed solution on R4 drone audio testing dataset. The outcome of this experiment will be used in evaluating the performance of the proposed hybrid RG dataset through a quantitative comparison.

Moreover, we have designed two scenarios for evaluating the CNN models on both R4 and RG datasets; the first is where the drone detected is one of the *seen* drones in which the performance of the model was examined on the same types of drones it was exposed to throughout the training phase. Whereas, the second scenario is the detection of an *unseen* drone; the drone type which was never used during the training phase. In testing the performance on an *unseen* drone, we assessed the significance of using the CNN models trained on the RG drone audio dataset to detect the new *unseen* drone described in Table 3 in comparison to the CNN models trained on the R4.

To achieve this aim, we have carried out the experiment as demonstrated in Fig.20 where the following steps were carefully selected for each of the five *D* partitions:

1. Train a CNN model on the R4 dataset of the *D* partition

2. Train a CNN model on the RG dataset (composed of the selected R4 of the same *D* partition and GAN generated from the same R4 dataset of the *D* partition)

3. Test the models trained in (1) on the *seen drones* testing set from the selected *D* partition

4. Test the models trained in (1) using the *unseen drone* testing set in the selected *D* partition

5. Repeat (3) and (4) for the models trained in (2)

The above experiment is repeated for each of the *D* partitions ten times.

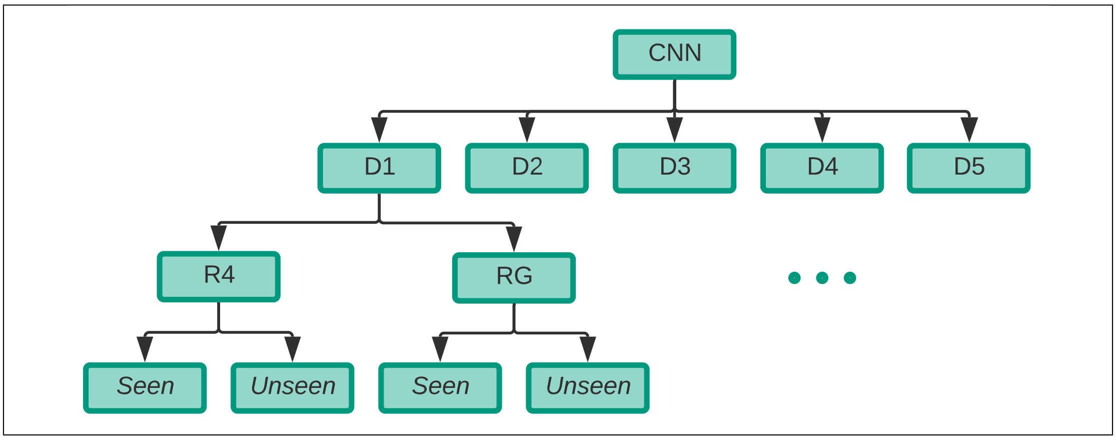


Figure 20. Breakdown of Experiment 2

Additionally, it is important to note that for drone detection application, the most crucial evaluation metric is *recall*. As in typical intrusion detection scenario, false positive predictions are tolerated more than false negatives; where drones pass by undetected.

*6.3.1. **B.1**: Drone Detection using RG Dataset*

6.3.1.1. Testing on seen Drones

To asses whether the performance of CNN model would be improve when trained using RG dataset and tested on the *seen* drones, we have conducted ten experiments for each of the *D* as mentioned in the introduction of this section. The results in Table 13 show that the CNN model trained on the RG dataset have outperformed, in terms of precision, the model trained on the R4 dataset with an increase of 0.49% in D1, 0.15% in D2, 0.10% in D3, 0.27% in D4 and 0.64% in D5. The bold values in Table 13 illustrate that, in addition to the fact that the model had better performance in terms of precision, the standard deviation is, also, lower.

Table 13. Seen Drones Experiment

Experiment		**Performance of the CNN classifier**			
		Precision	**Recall**	**F1 Score**	**Accuracy**
D1	R4	0.9773±0.0100	**0.9509±0.0189**	**0.9638±0.0113**	**0.8630±0.0171**
	RG	**0.9821±0.0058**	0.9111±0.0401	0.9448±0.0220	0.8269±0.0364
D2	R4	0.9883±0.0045	**0.9636±0.0202**	**0.9756±0.0094**	**0.8740±0.0183**
	RG	**0.9898 ±0.0040**	0.9575±0.0378	0.9730 ±0.0194	0.8686 ±0.0343
D3	R4	0.9859 ±0.0050	**0.9633±0.0171**	**0.9744±0.0085**	**0.8735±0.0155**
	RG	**0.9869±0.0040**	0.9450±0.0182	0.9654±0.0098	0.8569±0.0165
D4	R4	0.9884±0.0037	**0.9815±0.0097**	**0.9849±0.0044**	**0.8908±0.0088**
	RG	**0.9911±0.0025**	0.9782±0.0102	0.9846±0.0051	0.8878±0.0092
D5	R4	0.9853±0.0044	**0.9553±0.0130**	**0.9700±0.0058**	**0.8665±0.0118**
	RG	**0.9916±0.0036**	0.9401±0.0236	0.9649±0.0115	0.8527±0.0214

However, it can be deduced from this experiment that RG is not useful in the application where the model was already exposed to the different drone types during training given that it didn't show any improvement in the other metrics, specifically recall. In fact, the models trained on RG had worse performance, in terms of recall, compared to those trained on R4 as illustrated in Fig.21. A more plausible explanation for such behaviour would be that although the performance of the CNN classifier was worst when trained on RG dataset and tested on seen drones, it is worth noting that the performance degradation was very minor.

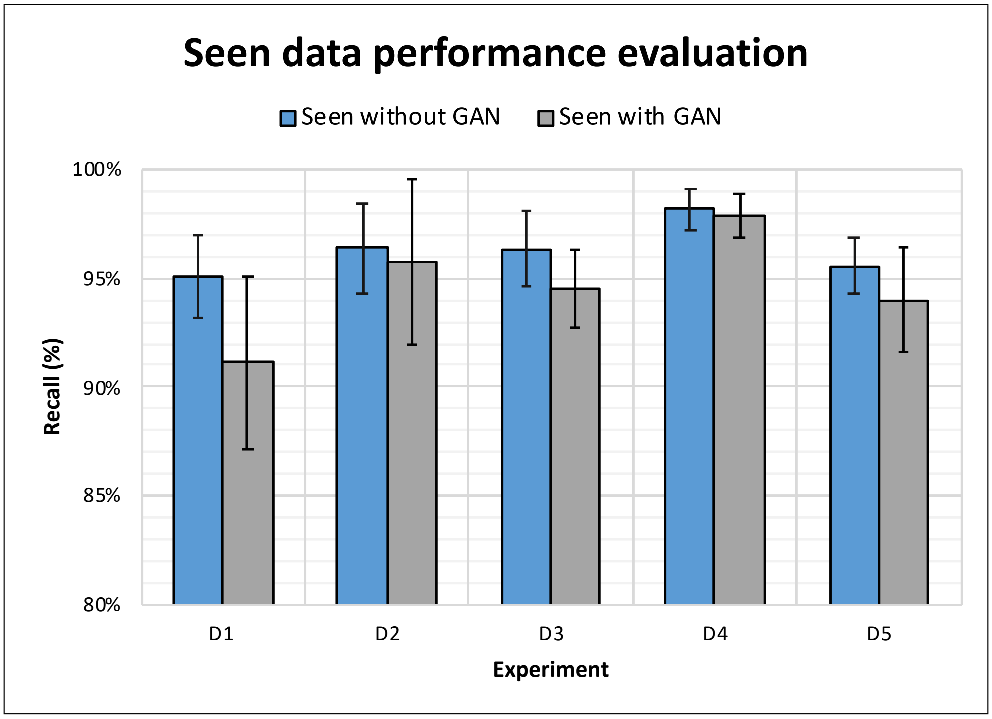


Figure 21. The performance of the average CNN models trained R4 Vs. RG drone dataset and tested on *seen* drones of in terms of *recall*.

6.3.1.2. Testing on unseen Drone

We aim through this experiment to improve the performance degradation of the classifier when it is met with an *unseen* drone which was observed in Section 6.2 by training the CNN models on RG dataset. To examine the model's performance, we have tested the CNN models trained on the RG dataset and R4 dataset separately using the *unseen* drone. This is done to evaluate the generalization capabilities of the CNN models and whether the integration of artificial drone acoustic data would have any positive effect on the overall performance of the model in comparison to the observations of *seen* drone experiment in Section 6.3.1.1. The results yielded from this experiment are further illustrated in Table 14.

Table 14. Unseen Drone Experiment

Experiment		Performance of the CNN classifier			
		Precision	Recall	F1 Score	Accuracy
D1	R4	0.9826±0.0055	0.8365±0.1097	0.8996±0.0685	0.7600 ±0.0996
	RG	**0.9861±0.0025**	0.8455 ±0.1150	0.9059±0.0722	0.7682±0.1045
D2	R4	0.9790±0.0252	0.5287±0.2239	0.6602±0.1962	0.4792±0.2030
	RG	**0.9892 ±0.0124**	**0.6609 ±0.1807**	**0.7767±0.1462**	**0.5990±0.1638**
D3	R4	0.9774±0.0106	0.6890±0.1347	0.8011±0.0915	0.6237±0.1219
	RG	**0.9836±0.0082**	**0.7047 ±0.0987**	**0.8172±0.0701**	**0.6379 ±0.0893**
D4	R4	0.9363±0.0128	0.1860±0.0747	0.3045±0.0960	0.1690±0.0679
	RG	0.9552±0.0198	0.2574±0.1824	0.3791±0.1912	0.2338±0.1657
D5	R4	0.9752±0.0128	0.4836±0.1421	0.6358±0.1128	0.4396±0.1292
	RG	**0.9882±0.0048**	0.5521±0.1442	0.6980±0.1131	0.5019±0.1311

Our study reveals that in the situation where the drone is completely new to the classifier, the average performance of the CNN model trained on the RG dataset has outperformed, in all evaluation metrics, the average performance of the model that was trained only on the R4 dataset as illustrated in Table 14. The shaded cells represent the occurrences where the model trained on the RG drone dataset has higher performance in comparison to the model trained on the R4 dataset in all five *D* experiments, whereas, the bold text shows the improvement, if any, in the standard deviation and performance for all four evaluation metrics.

In a similar vein, Fig.22 demonstrates the comparison between the average performance of CNN models trained on RG drone audio dataset versus the average performance of CNN model trained on R4 drone audio dataset, in terms of recall, when met with an *unseen* drone. It can be observed from the graph that there was an noticeable improvement in recall by 1.08% in D1, 25% in D2, 2.28% in D3, 38.39% in D4 and 14.16% in D5. Furthermore, this suggests that the adding GAN generated dataset to the training of a model further enhances the performance of the model in comparison to the one trained on the R4 drone dataset, particularly in recall due to the generalisation that GAN data adds to the training. Also, this addition led to having a more diverse training dataset which improved the generalisation. Hence, this is a clear demonstration that using GAN in the hybrid dataset, RG, adds a significant improvement in detection of *unseen* drones in comparison to *unseen* drone detection using the recorded drone data, R4. Thus, the benefits of training CNN model on RG and using it in the *unseen* scenario outweigh the costs in the seen scenario.

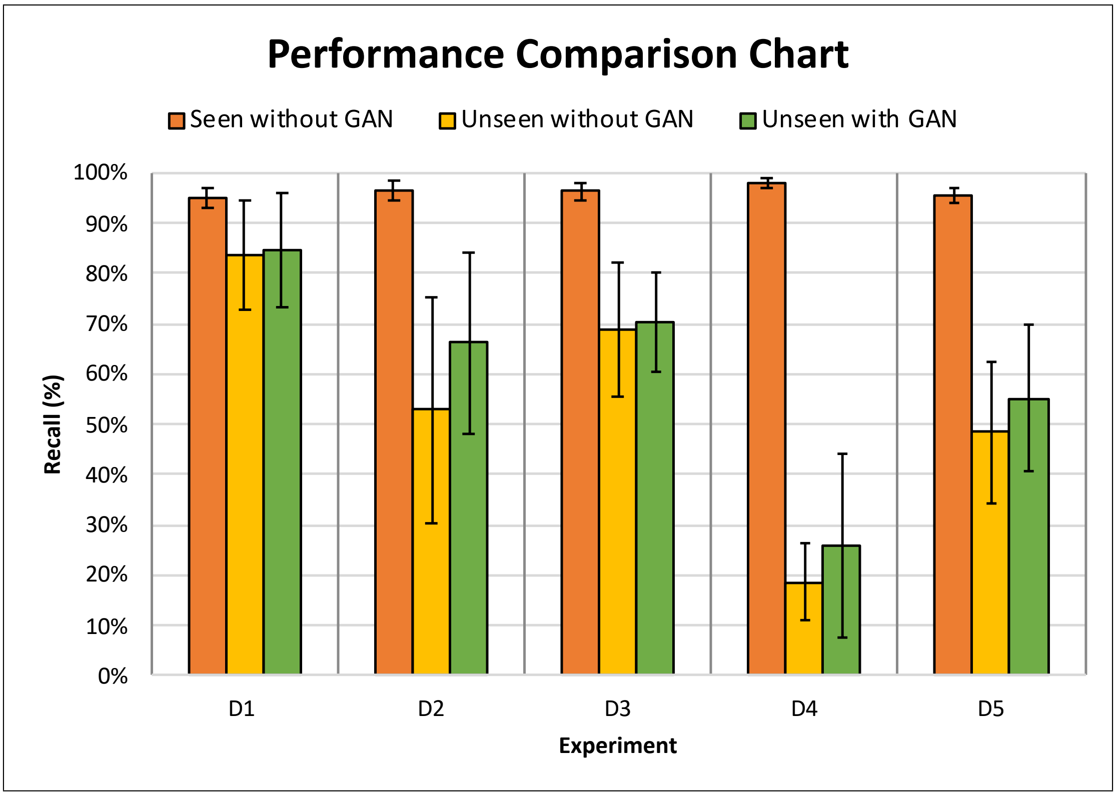


Figure 22. The performance of the average CNN models trained R4 Vs. RG drone dataset and tested on *unseen* drones of in terms of *recall*.

This interesting finding appears to confirm our hypothesis that the integration of the artificially generated dataset through GAN with an actual drone audio dataset does not only fulfil the gap of drone audio shortage but it also boosts the generalization of the trained classifier for the cases of detecting completely new and *unseen* drones.

From Table 14, it is worth noting that the model trained on **D1** distribution had the best performance of around 90% in F1 score among the other experiments. This suggests that if the model was trained on DJI Phantom 4, 3DR Solo, Mambo and AR Drone, testing it on **Bebop** becomes a simple classification problem to the CNN classifier. A further explanation to this performance is that the **Bebop** drone is of a similar physical size to the majority of the drones used in the training phase. The inverse of this performance was observed in **D4** experiment, where **Mambo** drone was used in testing the performance of the model. The experiment revealed that the CNN model had the weakest performance in terms of recall, F1 score and accuracy. To understand this behaviour, we have conducted human-hearing tests with a number of volunteers. It was particularly noticeable that there was a significant difference in the sound generated from the propellers of the **Mambo** drone, being the smallest in size, in comparison to the other drones used in the training phase, to the human-ears. Hence, one can conclude that the influence of the size of the drone on the performance of the CNN model is indisputable and a variety of drone recordings from various drone sizes is needed to further enhance the GAN model.

Concluding this chapter, we can say that in applications were it is expected to detect explicit types of drones that are available to train the model on, using a recorded dataset with those types of drones without GAN would be sufficient. However, in applications where detection of any type of drone is required, a hybrid dataset with GAN would be highly effective.

CHAPTER 7: SWARM OF DRONES LOCALISATION AND TRACKING USING A SIMULATOR GENERATED DATASET

Future research on drone detection and identification application using deep learning might extend the proposed solution to address the problem associated with swarm of drones attacks as swarm of drone attacks having recently been raising and are is expected to increase dramatically with the development of their technology in near future [58]. Another interesting extension to this work could be the incorporation of other features, such as RF, in addition to acoustic features in targeting more complex scenarios, such as swarm of drones. Furthermore, this could provide a solution that is, in addition to detecting and identifying unauthorised drones, is capable of real time tracking and localising swarm of drones. Therefore, to move forward with this solution, we started the first step by designing an RF based simulator, which we refer to as RF Drones Simulator (RFDS), with the aim of generating a large RF based dataset that could be later used in training the DL models later on.

An illustration of a scenario where swarm of drones are flying over a restricted area is provided in Fig. 23. In order to better understand the behaviour of drones in a swarm, we introduce the concept of drones' clusters. In many different attacks or applications, the swarm of drones are usually separated into clusters to extend the range of the swarm or perform different mission simultaneously as observed in various Flying Ad-hoc Networks applications [59]. An example of such scenario is when a cluster of drones is assigned to launch an attack, whereas, another cluster would record, film and stream the attack.

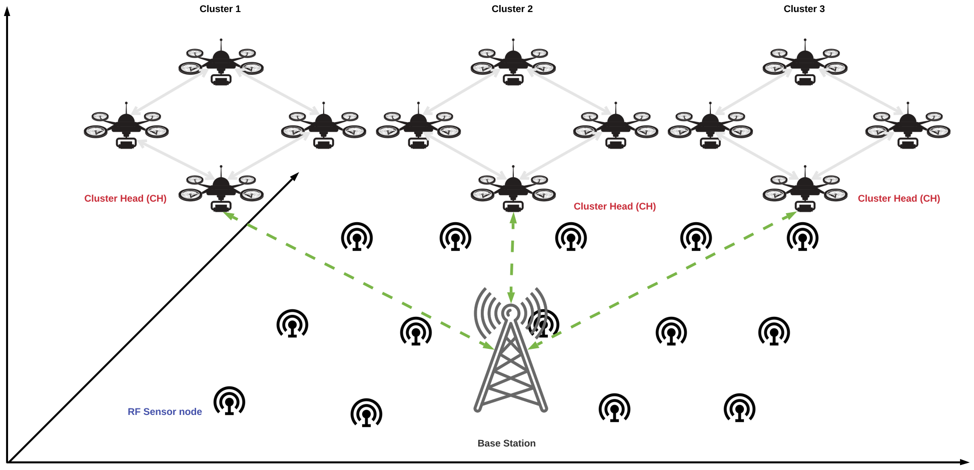


Figure 23. Swarm of drones' attack scenario in RFDS

To design the RFDS, we started with the assumption that each Cluster head (CH) would initiate a communication with the Base Station (BS) through which the information gathered from the cluster could be shared. Next, using passive RF Receiver (R_x) nodes which are deployed within the premises of the restricted area, the communication signals are received at each R_x and shared with a centralised unit for further analysis.

Moving on to the architecture of the simulator, Fig. 24 shows the high level system architecture of the proposed RFDS.

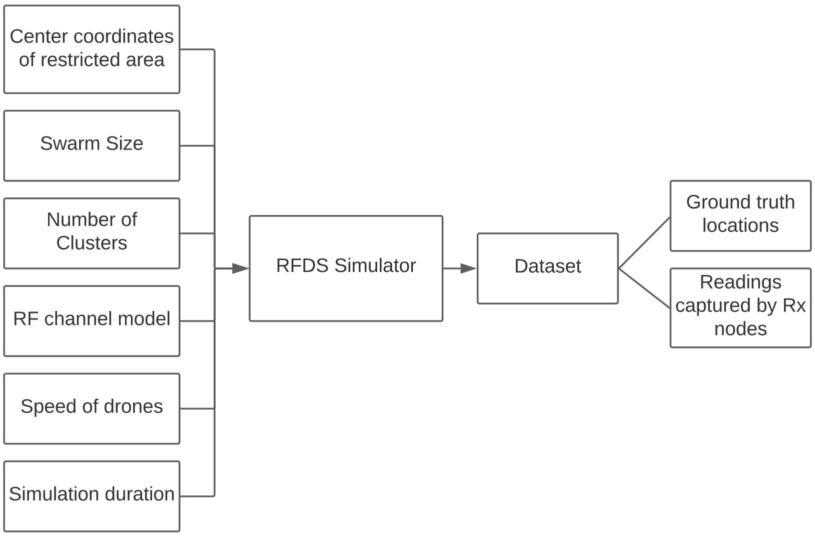


Figure 24. High level design of the RFDS

Initially, a number of parameters will set by the user and fed into the simulator. These parameters can be selected based on the application requirements and can be defined as:

- **Center coordinates of restricted area**: initial latitude and longitude coordinates of the center of a restricted area are required to specify the area which needs to be secured from the swarm attack.
- **Swarm Size**: this parameters defines the total number of drones in the simulation
- **Number of Clusters**: Through this parameter, the total number of clusters in the swarm mission is established
- **RF channel model**: Using this parameter, the wireless channel model can be selected. E.g. The free space propagation model.
- **Speed of drones**: User can define average speed at which the drones are flying using this parameter.
- **Simulation duration**: As the name suggests, this parameter is used to specify the total duration of the swarm of drones mission simulation.

In the second stage, the simulator starts by generating the location of the CHs. Followed by the generation of the location of all other drones locations within each cluster with respect to its CH and the adjacent cluster. Afterwards, the simulator starts to generate the mobility pattern for each drone in the simulation over the total duration of the simulation (further details on the mobility pattern can be found in Section 7.1). During the simulation and while the CHs are communicating with the BS, each of the deployed R_x nodes passively listens to the communication and captures the Power Received (P_r) from each communication at its end (a detailed description about the physical layer characteristic and assumptions can be found in 7.2). Then, the information is collected at a center node where it is used to calculate the aggregate of the P_r at each of R_x.

Finally, two sets of data as shown in Fig. 24 are produced through RFDS. A short description of each is mentioned below.

1. Ground truth locations in longitude, latitude and altitude for each drone during the mission. This can be used later on to verify the performance of the localisation and tracking system.

2. The total Power Received (P_r) value capture at each R_x along the corresponding latitude and longitude coordinates of the R_x. This dataset will be used to train the DL algorithms in a later stage to aid in tracking and localising the drones.

Two examples demonstrating the RFDS in action are shown in Figs. 25 and 26. The first scenario consist of a single drone in each cluster and the second is representing three clusters each made up of six drones.

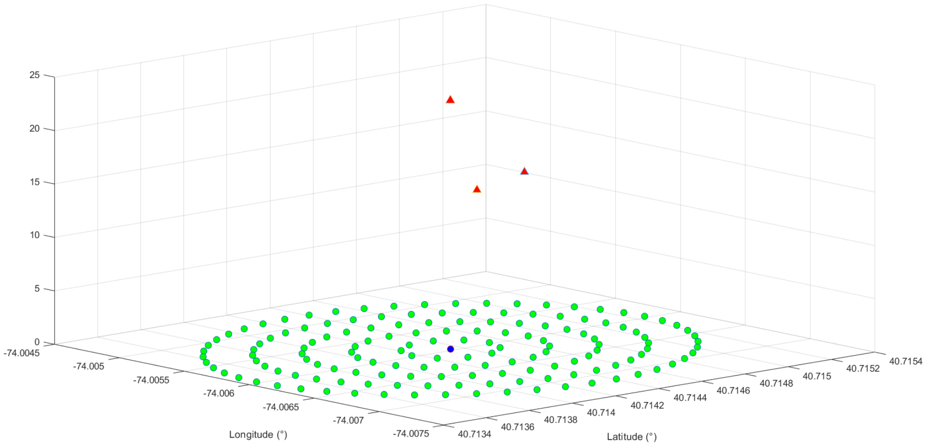


Figure 25. RFDS simulating three drones each at a different cluster

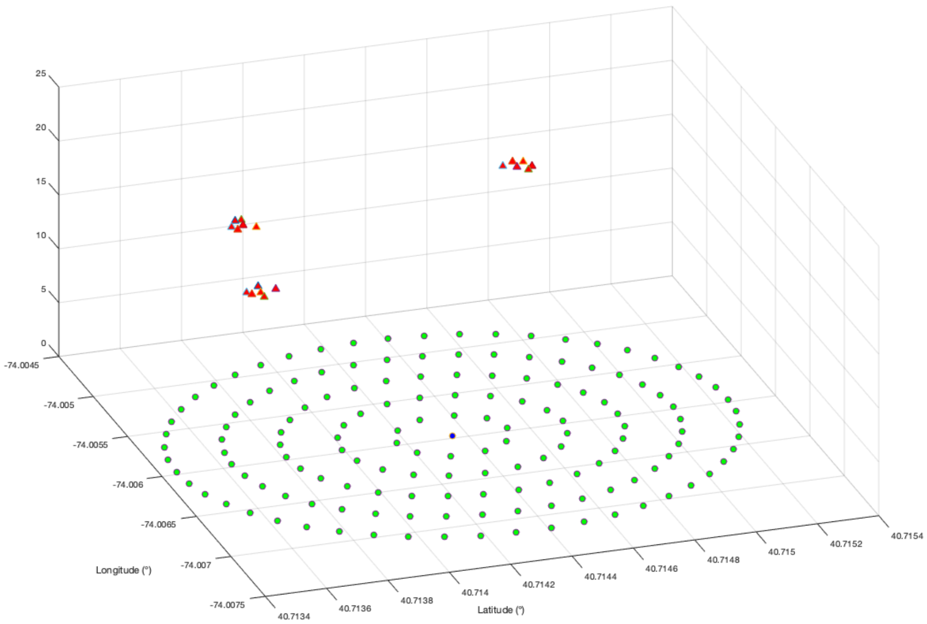


Figure 26. RFDS simulating three clusters consisting of six drones each

7.1. Drone Swarms Mobility Pattern

The mobility pattern of the drones in RFDS is inspired by the Reference Point Group Mobility (RPGM) model proposed by the authors in [60]. The authors have defined the mobility pattern such that the targets move with respect to a reference node in Random WayPoint Mobility (RWMP) pattern. However, the three main limitations of the RPGM model if deployed for swarm of drones application are: (1)The proposed system functions in 2-Dimensional plane, whereas, drones fly in a various altitudes. (2) The RPGM model lacks the location aspect as it is not built based on any coordinate system. (3)The RPGM model does not implement any collision detection or avoidance mechanisms between different clusters nor between targets within the same cluster. Hence, in order to overcome the limitations of the RPGM model, we built RFDS to take into account the main characteristics of swarm of drones by incorporating the following features:

- We build a 3-Dimensional (3D) implementation of the system including the x direction for the latitude, y direction for the longitude and z direction for the altitude.

- We introduce CHs in each cluster and link the drones within the same cluster formation to it.

- We include a collision detection and avoidance technique between drones within the same cluster. That is, any drone, regardless of its altitude, would maintain a minimum d distance to all other points in the x-y plane. Furthermore, at time instance t, all new locations calculated will be checked against the location of other drones in order to ensure that the random new location of a specific drone is adhering to the collision avoidance criteria mentioned above. If a collision between two drones has been detected, the coordinates of one of these drones will be recalculated.

- We design the simulator such that each cluster would fly on a different altitude to other cluster with a separation distance of n meters between them.

- Although a similar mobility model to the Random WayPoint Mobility (RWMP) model is used for the underlying movement of the drones, the randomness percentage is significantly decreased and bounded such that it reflects real drone movement. For example, instead of a drone moving from point at location (1,1) at time t=1 and suddenly to location (2,4) at t=2, then, (-5,-10) at t=3, the movement of the drone would happen gradually through a number of steps over a duration of time in order to simulate real drone movement.

7.2. Drone Swarms Physical layer design

In order to implement the physical layer communication between drones, we assume that all drones have the same power transmission values $\mathrm{P_t}$ and are equipped with isotropic antennas. From this perspective, RFDS attempts to calculate the $\mathrm{P_l}$ values using the FSPL model. Next, using the $\mathrm{P_l}$ value calculated above and the pre-knowledge of the $\mathrm{P_t}$ value, RFDS would calculate the $\mathrm{P_r}$ value at each $\mathrm{R_x}$ as formulated in Equation 7.1.

$$P_r(dBm) = P_t(dBm) - P_l(dB) \tag{7.1}$$

Then, the total $\mathrm{P_r}$ values will be summed up for all drones at a specific time (t) in each $\mathrm{R_x}$, leaving us with the aggregate $\mathrm{P_r}$ value, referred to as $\mathrm{P_{r_{Total}}}$ in Equation 7.2.

$$P_{r_{Total}}(dBm) = \sum_{i=1}^{d}(P_t(dBm) - P_{l(i)}(dB)) \tag{7.2}$$

Where *d* is the total number of drones and *i* resembles a single drone instance.

A sample output of the simulation in Fig. 25 for 5 sensors out of 150 deployed is shown in Table 15. The latitude, longitude and $\mathrm{P_r}$ values at each of sensor with respect to time (t) is shown in each row.

Table 15. Sample of the dataset generated through RFDS simulator

Latitude	Longitude	t=1	t=2	t=3
40.714533	-74.00596881	-151.8464222	-153.1457417	-154.2472369
40.7144987	-74.00582933	-152.6936543	-153.6408505	-154.4193323
40.7144087	-74.00574313	-156.5094747	-157.143287	-157.6084363
40.7142976	-74.00574313	-160.9921942	-161.5058458	-161.8584282

CHAPTER 8: CONCLUSION AND FUTURE WORK

In this thesis, we address the issue of illegal use of drones in malicious activities by proposing a novel approach that automates the drone detection and identification processes using the drone's acoustic features with different DL algorithms. However, the lack of acoustic drone datasets restricts the ability to implement an effective solution using DL algorithms. Therefore, our work targets this gap by introducing a hybrid drone acoustic dataset, RG, composed of recorded drone audio clips and artificially generated drone audio clips using the Generative Adversarial Network (GAN). From the experiments conducted throughout this work, it was found that CNN has outperformed both RNN and CRNN in detecting and identifying drones of familiar, seen during training, types of drones.

Furthermore, when presented with seen drones, the CNN classifier trained on the recorded drone acoustic dataset, R4, outperformed the CNN classifier trained on RG dataset. However, when met with completely new drone types, the classifier was less effective and the classifier trained on RG dataset was outstandingly better. Thus, the benefits of RG dataset in the unseen scenario outweigh the costs in the seen scenario.

The proposed approach of using GANs to generate real-like drone audio clips illustrates a promising way to fulfil the gap imposed by the lack of drone acoustic dataset while also contributing to an improvement in classifier's performance. These findings are aimed to help the research community use GAN generated drone audio clips along with recorded drone audio dataset, which we are releasing publicly, in various DL applications for further analysis.

An interesting extension to this work could be the incorporation of other features, such as RF, in addition to acoustic features in targeting more complex scenarios, such as swarm of drones. Furthermore, this could provide a solution that is, in addition to detecting and identifying unauthorised drones, is capable of real time tracking and localising a swarm of drones. As for the RFDS proposed in this work, we would investigate the feasibility of using it for data fusion with the aim of tracking and localising malicious swarm of drones.

PUBLICATIONS

Parts of this thesis has been presented and published in IEEE International Wireless Communications and Mobile Computing Conference in 2019 (IWCMC) [61]. Furthermore, an earlier version of this thesis has been submitted for a publication in [62].

REFERENCES

[1] J. A. J. Berni, P. J. Zarco-Tejada, L. Suarez, and E. Fereres, “Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle”, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, pp. 722–738, Mar. 2009, ISSN: 0196-2892. DOI: 10.1109/TGRS.2008.2010457.

[2] V. Ciullo, L. Rossi, T. Toulouse, and A. Pieri, “Fire geometrical characteristics estimation using a visible stereovision system carried by unmanned aerial vehicle”, in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Nov. 2018, pp. 1216–1221. DOI: 10.1109/ICARCV.2018.8581167.

[3] P. Katsigiannis, L. Misopolinos, V. Liakopoulos, T. K. Alexandridis, and G. Zalidis, “An autonomous multi-sensor uav system for reduced-input precision agriculture applications”, in *2016 24th Mediterranean Conference on Control and Automation (MED)*, Jun. 2016, pp. 60–64. DOI: 10.1109/MED.2016.7535938.

[4] R. Tariq, M. Rahim, N. Aslam, N. Bawany, and U. Faseeha, “Dronaid : A smart human detection drone for rescue”, in *2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT IoT (HONET-ICT)*, Oct. 2018, pp. 33–37. DOI: 10.1109/HONET.2018.8551326.

[5] *’sustained’ drone attack closed gatwick, airport says*, 2019. [Online]. Available: https://www.bbc.com/news/business-47302902.

[6] *Gatwick drone policing costs ’shocking’*, 2019. [Online]. Available: https://www.bbc.com/news/uk-england-47696499?intlink_from_url=

https://www.bbc.com/news/topics/cnx1xjxwp51t/gatwick-drone-shutdown&link_location=live-reporting-story.

[7] J. P. Daniels, *Venezuela's nicolas maduro survives apparent assassination attempt*, Aug. 2018. [Online]. Available: https://www.theguardian.com/world/2018/aug/04/nicolas-maduros-speech-cut-short-while-soldiers-scatter.

[8] *Saudi arabia oil facilities ablaze after drone strikes*, 2019. [Online]. Available: https://www.bbc.com/news/world-middle-east-49699429.

[9] F. Gardner, *Saudi oil facility attacks: Race on to restore supplies*, 2019. [Online]. Available: https://www.bbc.com/news/world-middle-east-49775849.

[10] *Well-organised gang used drones to deliver drugs to inmates, court told.* [Online]. Available: http://www.itv.com/news/2018-08-30/well-organised-gang-used-drones-to-deliver-drugs-to-inmates-court-told/.

[11] *Man fined after flying drones over premier league stadiums*, 2019. [Online]. Available: https://www.bbc.com/news/uk-england-nottinghamshire-34256680.

[12] M. Chen, *San carlos woman says drone hovered near bedroom - and wouldn't go away*, 2018. [Online]. Available: https://www.10news.com/news/san-carlos-woman-says-drone-hovered-near-bedroom-wouldnt-go-away.

[13] B. Steffen, *Drone spotted hovering outside bedroom window*, 2018. [Online]. Available: https://www.10news.com/news/drone-spotted-hovering-outside-bedroom-window.

[14] *Drone flies over macron's holiday home in wake of maduro 'attack'*, 2018. [Online]. Available: https://www.thelocal.fr/20180807/drone-flies-over-macrons-holiday-home-in-wake-of-maduro-attack.

[15] E. Limer, *How to shoot down a drone*, Apr. 2018. [Online]. Available: http://tinyurl.com/p5zaso3.

[16] D. Sathyamoorthy, "A review of security threats of unmanned aerial vehicles and mitigation steps", *The Journal of Defence and Security (In press)*, vol. 6, no. 2, 2015.

[17] R. Vander Schaaf, "What technologies or integrating concepts are needed for the us military to counter future missile threats looking out to 2040?", Ph.D. dissertation, AIR WAR COLLEGE - AIR UNIVERSITY, 2014.

[18] T. E. Humphreys, "Statement on the security threat posed by unmanned aerial systems and possible countermeasures", 2015.

[19] J.-S. Pleban, R. Band, and R. Creutzburg, "Hacking and securing the ar. drone 2.0 quadcopter: Investigations for improving the security of a toy", in *Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications 2014*, International Society for Optics and Photonics, vol. 9030, 2014, p. 90300L.

[20] X. Shi, C. Yang, W. Xie, C. Liang, Z. Shi, and J. Chen, "Anti-drone system with multiple surveillance technologies: Architecture, implementation, and challenges", *IEEE Communications Magazine*, vol. 56, no. 4, pp. 68–74, Apr. 2018, ISSN: 1558-1896. DOI: 10.1109/MCOM.2018.1700430.

[21] B. Knoedler, R. Zemmari, and W. Koch, “On the detection of small uav using a gsm passive coherent location system”, in *Radar Symposium (IRS), 2016 17th International*, IEEE, 2016, pp. 1–4.

[22] Y. Liu, X. Wan, H. Tang, J. Yi, Y. Cheng, and X. Zhang, “Digital television based passive bistatic radar system for drone detection”, in *Radar Conference (RadarConf)*, IEEE, 2017, pp. 1493–1497.

[23] M. M. Azari, H. Sallouha, A. Chiumento, S. Rajendran, E. Vinogradov, and S. Pollin, “Key technologies and system trade-offs for detection and localization of amateur drones”, *IEEE Communications Magazine*, vol. 56, no. 1, pp. 51–57, Jan. 2018, ISSN: 1558-1896. DOI: 10.1109/MCOM.2017.1700442.

[24] P. Nguyen, M. Ravindranatha, A. Nguyen, R. Han, and T. Vu, “Investigating cost-effective rf-based detection of drones”, in *Proceedings of the 2Nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, ser. DroNet ’16, Singapore, Singapore: ACM, 2016, pp. 17–22, ISBN: 978-1-4503-4405-0. DOI: 10.1145/2935620.2935632. [Online]. Available: http://doi.acm.org/10.1145/2935620.2935632.

[25] P. Nguyen, H. Truong, M. Ravindranathan, A. Nguyen, R. Han, and T. Vu, “Matthan: Drone presence detection by identifying physical signatures in the drone’s rf communication”, in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, ACM, 2017, pp. 211–224.

[26] S. Al-Emadi and F. Al-Senaid, “Drone detection approach based on radio-frequency using convolutional neural network”, in *2020 IEEE International Con-*

ference on Informatics, IoT, and Enabling Technologies (ICIoT), IEEE, 2020, pp. 29–34.

[27] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets", in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: `http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf`.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[29] F. Chollet, *Deep Learning with Python*, 1st. USA: Manning Publications Co., 2017, ISBN: 1617294438.

[30] A. Graves, A.-r. Mohamed, and G. Hinton, *Speech recognition with deep recurrent neural networks*, 2013. arXiv: `1303.5778 [cs.NE]`.

[31] F. M. Noori, B. Wallace, M. Z. Uddin, and J. Torresen, "A robust human activity recognition approach using openpose, motion features, and deep recurrent neural network", in *Scandinavian Conference on Image Analysis*, Springer, 2019, pp. 299–310.

[32] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult", *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994. DOI: `10.1109/72.279181`.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] S. Raschka and V. Mirjalili, *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd, 2019.

[35] S. O. Arik, M. Kliegl, R. Child, J. Hestness, A. Gibiansky, C. Fougner, R. Prenger, and A. Coates, *Convolutional recurrent neural networks for small-footprint keyword spotting*, 2017. arXiv: `1703.05390 [cs.CL]`.

[36] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello edge: Keyword spotting on microcontrollers", *CoRR*, vol. abs/1711.07128, 2017. arXiv: `1711.07128`. [Online]. Available: `http://arxiv.org/abs/1711.07128`.

[37] J. Mezei, V. Fiaska, and A. Molnar, "Drone sound detection", in *2015 16th IEEE International Symposium on Computational Intelligence and Informatics (CINTI)*, Nov. 2015, pp. 333–338. DOI: `10.1109/CINTI.2015.7382945`.

[38] A. Bernardini, F. Mangiatordi, E. Pallotti, and L. Capodiferro, "Drone detection by acoustic signature identification", *electronic imaging*, vol. 2017, pp. 60–64, 2017.

[39] N. Takahashi, M. Gygli, and L. V. Gool, "Aenet: Learning deep audio features for video analysis", *CoRR*, vol. abs/1701.00599, 2017. arXiv: `1701.00599`. [Online]. Available: `http://arxiv.org/abs/1701.00599`.

[40] 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`.

[41] 2018. [Online]. Available: https://www.tensorflow.org/tutorials/sequences/audio_recognition.

[42] X. L. xjli and Z. Z. zixuan, “Speech command recognition with convolutional neural network”, 2017.

[43] S. Jeon, J.-W. Shin, Y.-J. Lee, W.-H. Kim, Y. Kwon, and H.-Y. Yang, “Empirical study of drone sound detection in real-life environment with deep neural networks”, *CoRR*, vol. abs/1701.05779, 2017. arXiv: 1701.05779. [Online]. Available: http://arxiv.org/abs/1701.05779.

[44] E. Sprengel, M. Jaggi, Y. Kilcher, and T. Hofmann, “Audio based bird species identification using deep learning techniques”, in *CLEF*, 2016.

[45] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks”, *CoRR*, vol. abs/1812.04948, 2018. arXiv: 1812.04948. [Online]. Available: http://arxiv.org/abs/1812.04948.

[46] C. Donahue, J. McAuley, and M. Puckette, “Adversarial audio synthesis”, in *ICLR*, 2019.

[47] S. Al-Emadi, *Saraalemadi/droneaudiodataset*, 2018. [Online]. Available: https://github.com/saraalemadi/DroneAudioDataset.

[48] ——, *Saraalemadi/droneaudiodataset_r4rg*, 2020. [Online]. Available: https://github.com/saraalemadi/DroneAudioDataset_R4RG.

[49] K. J. Piczak, “ESC: Dataset for Environmental Sound Classification”, in *Proceedings of the 23rd Annual ACM Conference on Multimedia*, Brisbane, Australia: ACM Press, Oct. 13, 2015, pp. 1015–1018, ISBN: 978-1-4503-3459-4. DOI:

10.1145/2733373.2806390. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2733373.2806390.

[50] P. Warden, “Speech commands: A dataset for limited-vocabulary speech recognition”, *CoRR*, vol. abs/1804.03209, 2018. arXiv: 1804.03209. [Online]. Available: http://arxiv.org/abs/1804.03209.

[51] (2019), YouTube, [Online]. Available: https://www.youtube.com.

[52] *3dr solo fly manual mode test*. [Online]. Available: https://www.youtube.com/watch?v=gHlF_EjbfvI.

[53] *3dr solo - unboxing, set up, first flight*. [Online]. Available: https://www.youtube.com/watch?v=xRqwq-zDiXs&list=PLKrTJOW1plxiARpra-Nn6cpF2S-SjwXaR&index=6&t=907s.

[54] *How loud are they? mavic pro vs phantom 4 pro*. [Online]. Available: https://www.youtube.com/watch?v=C5abCs5VHsE&t=72s+(3).

[55] *Dji phantom 4 indoor (living room) flight test p mode*. [Online]. Available: https://www.youtube.com/watch?v=_P-8fI4m1Dc&list=PLKrTJOW1plxiARpra-Nn6cpF2S-SjwXaR&index=3&t=0s%20(2).

[56] *Ar.drone test flight*. [Online]. Available: https://www.youtube.com/watch?v=Uecpd7LRJRY&list=PLKrTJOW1plxiARpra-Nn6cpF2S-SjwXaR&index=10&t=0s.

[57] *Ar drone 2.0 indoor christmas flight*. [Online]. Available: https://www.youtube.com/watch?v=mD6N_vknSnk&list=PLKrTJOW1plxiARpra-Nn6cpF2S-SjwXaR&index=10&t=6s.

[58] M. Safi, *Are drone swarms the future of aerial warfare?*, 2020. [Online]. Available: https://www.theguardian.com/news/2019/dec/04/are-drone-swarms-the-future-of-aerial-warfare.

[59] S. Al-Emadi and A. Al-Mohannadi, "Towards enhancement of network communication architectures and routing protocols for fanets: A survey", in *2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet)*, 2020, pp. 1–10.

[60] M. Khan, K. Heurtefeux, A. Mohamed, K. A. Harras, and M. M. Hassan, "Mobile target coverage and tracking on drone-be-gone uav cyber-physical testbed", *IEEE Systems Journal*, vol. 12, no. 4, pp. 3485–3496, 2018.

[61] S. A. Al-Emadi, A. K. Al-Ali, A. Al-Ali, and A. Mohamed, "Audio based drone detection and identification using deep learning", in *IWCMC 2019 Vehicular Symposium (IWCMC-VehicularCom 2019)*, Tangier, Morocco, Jun. 2019.

[62] S. Al-Emadi, A. Al-Ali, and A. Al-Ali, "Audio based drone detection and identification using deep learning techniques with dataset enhancement through generative adversarial networks", *Submitted*, 2020.