University of Louisville

ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

8-2021

Signal fingerprinting and machine learning framework for UAV detection and identification.

Olusiji Oloruntobi Medaiyese
University of Louisville

Follow this and additional works at: https://ir.library.louisville.edu/etd

Part of the Artificial Intelligence and Robotics Commons, Data Science Commons, Hardware Systems Commons, and the Signal Processing Commons

Recommended Citation

Medaiyese, Olusiji Oloruntobi, "Signal fingerprinting and machine learning framework for UAV detection and identification." (2021). *Electronic Theses and Dissertations.* Paper 3687.
https://doi.org/10.18297/etd/3687

This Doctoral Dissertation is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

SIGNAL FINGERPRINTING AND MACHINE LEARNING FRAMEWORK FOR UAV DETECTION AND IDENTIFICATION

By

Olusiji Oloruntobi Medaiyese
B.Eng., M.Sc.

A Dissertation
Submitted to the Faculty of the
J.B. Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy in Computer Science and Engineering

Department of Computer Engineering and Computer Science
University of Louisville
Louisville, Kentucky

August 2021

Copyright 2021 by Olusiji Oloruntobi Medaiyese

All rights reserved

SIGNAL FINGERPRINTING AND MACHINE LEARNING FRAMEWORK FOR UAV DETECTION AND IDENTIFICATION

By

Olusiji Oloruntobi Medaiyese
B.Eng., M.Sc.

A Dissertation Approved On

July 15, 2021

by the following Dissertation Committee:

Dr. Adrian P. Lauf, Dissertation Director

Dr. Mehmed Kantardzic

Dr. Roman V. Yampolskiy

Dr. Hongxiang Li

Dr. Jeffrey Hieb

ACKNOWLEDGEMENTS

Firstly, I would like to sincerely appreciate my advisor, Dr. Adrian Lauf, for his guidance and mentorship throughout my Ph.D. program. Your style of mentorship brought about a unique experience.

I want to thank the members of the dissertation committee, Dr. Mehmed Kantardzic, Dr. Roman V. Yampolskiy, Dr. Hongxiang Li, and Dr. Jeffrey Hieb, for their service and for providing different dimensions of constructive feedback to my work, especially during the formative period of this research.

I would also like to thank the Electrical and Computer Engineering department at the North Carolina State University, especially Dr. Ismail Guvenc and Mr. Martin Ezuma, for providing free access to their laboratory equipment and space. Their assistance significantly facilitated the data acquisition aspect of this research.

My deepest appreciation goes to my family. To my beloved dad, I thank you very much for all your care and for believing in me up until your last breath. I would like to appreciate my beloved mother for her unwavering support, love, and prayers from my birth till this very moment of my life. My appreciation goes to my brothers and sisters especially Dr and Mrs Ayorinde Medaiyese, Mr and Mrs Oluwaseun Medaiyese, Oluwatosin Medaiyese and Ayodeji Medaiyese.

I appreciate all my friends Fiyin Lasisi, Ayodeji Adeniran, Felix Olabode, Kayode Omotoye, Babajide Ayinde, Mounirat Mahmoud, Temitope Ijiyode, Olamide Opadokun, Olabisi Ali, Tomi Fatoba, and many more for their support and prayers.

Finally, I would like to thank God for his provision, grace and inspiration. I would not have done this work without God's strength. I am so thankful.

ABSTRACT

SIGNAL FINGERPRINTING AND MACHINE LEARNING FRAMEWORK FOR UAV DETECTION AND IDENTIFICATION

Olusiji Oloruntobi Medaiyese

July 15, 2021

Advancement in technology has led to creative and innovative inventions. One such invention includes unmanned aerial vehicles (UAVs). UAVs (also known as drones) are now an intrinsic part of our society because their application is becoming ubiquitous in every industry ranging from transportation and logistics to environmental monitoring among others. With the numerous benign applications of UAVs, their emergence has added a new dimension to privacy and security issues. There are little or no strict regulations on the people that can purchase or own a UAV. For this reason, nefarious actors can take advantage of these aircraft to intrude into restricted or private areas. A UAV detection and identification system is one of the ways of detecting and identifying the presence of a UAV in an area. UAV detection and identification systems employ different sensing techniques such as radio frequency (RF) signals, video, sounds, and thermal imaging for detecting an intruding UAV. Because of the passive nature (stealth) of RF sensing techniques, the ability to exploit RF sensing for identification of UAV flight mode (i.e., flying, hovering, videoing, etc.), and the capability to detect a UAV at beyond visual line-of-sight (BVLOS) or marginal line-of-sight makes RF sensing techniques promising for UAV detection and identification. More so, there is constant communication between a UAV and its ground station (i.e., flight controller). The RF signals emitting from a UAV or UAV flight controller can be exploited for UAV detection and identification. Hence, in this work, an RF-based UAV detection and identification system is proposed and investigated. In RF signal fingerprinting research, the transient and steady state of the RF signals can be used to extract a unique signature. The first part of this work is to use two different wavelet analytic transforms (i.e., continuous wavelet transform and wavelet scattering transform) to investigate and analyze the characteristics or impacts of using either state for UAV detection and identification. Coefficient-based and image-based signatures are proposed for each of the wavelet analysis transforms to detect and identify a UAV. One of the challenges of using RF sensing is that a UAV's communication links operate at the industrial, scientific, and medical (ISM) band. Several devices such as Bluetooth and WiFi operate at the ISM band as well, so discriminating UAVs from other ISM devices is not a trivial task. A semi-supervised anomaly detection approach is explored and proposed in this research to differentiate UAVs from Bluetooth and WiFi devices. Both time-frequency analytical approaches and unsupervised deep neural network techniques (i.e., denoising autoencoder) are used differently for feature extraction. Finally, a hierarchical classification framework for UAV identification is proposed for the identification of the type of unmanned aerial system signal (UAV or UAV controller signal), the UAV model, and the operational mode of the UAV. This is a shift from a flat classification approach. The hierarchical learning approach provides a level-by-level classification that can be useful for identifying an intruding UAV. The proposed frameworks described here can be extended to the detection of rogue RF devices in an environment.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
LIST OF TABLES xi
LIST OF FIGURES xiii
LIST OF ALGORITHMS xx

CHAPTER

1 INTRODUCTION AND DISSERTATION OVERVIEW 1
  1.1 Motivation 1
  1.2 Challenges 3
    1.2.1 Size of a UAV 3
    1.2.2 Modality of sensing 4
    1.2.3 Detection range 4
    1.2.4 Weather effect 4
    1.2.5 System evaluation 4
  1.3 Overview on state of the art 5
    1.3.1 Detection 5
    1.3.2 Identification 9
    1.3.3 Neutralization 9
  1.4 Design of UAV Detection System 9
  1.5 Contribution 10
  1.6 Organisation 11

2 BACKGROUND AND LITERATURE REVIEW 12
  2.1 Introduction 12

2.2 Anatomy of Unmanned Aerial System 12
2.3 RF Fingerprinting 15
    2.3.1 RF Fingerprint Generation 15
    2.3.2 RF Fingerprint classification 17
2.4 Analysis of Signal in Time-Frequency Domain 18
    2.4.1 Fourier Transform 18
    2.4.2 Short Time Fourier Transform 19
    2.4.3 Wavelet Transform 19
2.5 Roles of Machine Learning and Deep Learning Techniques in UAV Detection Systems 23
2.6 Supervised Learning 23
    2.6.1 Logistic Regression 24
    2.6.2 Support Vector Machine (SVM) 24
    2.6.3 K-Nearest Neighbor 25
    2.6.4 Decision Tree 27
    2.6.5 Random Forest 28
    2.6.6 Deep Neural Network 29
2.7 Semi-supervised Learning 30
2.8 Criteria of Algorithm Selection 31
2.9 Literature Survey on RF-based UAV detection and identification . . . 32
2.10 Conclusion 37

3 CARDINAL RADIO FREQUENCY DATASET 38
2.10 Conclusion
3.1 Introduction 38
3.2 Review of Publicly Available Dataset 38
3.3 System Modeling 39
3.4 CardRF Data Acquisition 40
    3.4.1 Meta Data 41
    3.4.2 RF signal sensing and capturing system 42

3.4.3 Calibration of oscilloscope based on the absence of RF signal . 44
3.4.4 Data acquisition category 44
3.4.5 Type of signal captured 45
3.5 Data Description 46
3.6 Data Labeling 47
3.7 Data Format 49
3.8 Data Size and Usage 49
3.9 Conclusion 50

4 WAVELET TRANSFORM ANALYTICS FOR UAV DETECTION AND IDENTIFICATION 51
4.1 Introduction 51
4.2 Dataset Description 52
4.3 Preprocessing RF signal 52
4.4 Feature Extraction 53
4.4.1 Feature Extraction based on Continuous Wavelet Transform . . 53
4.4.2 Feature Extraction based on Wavelet Scattering Transform . . 55
4.5 UAV Classification Algorithm 59
4.5.1 Transfer Learning and SqueezeNet 60
4.6 Experimental Results and Discussions 63
4.6.1 Model Performance based on Classification Metrics 64
4.6.2 Performance at Difference SNR 69
4.6.3 Computational Complexity and Actual Time Cost 72
4.7 Conclusion 74

5 SEMI-SUPERVISED LEARNING FRAMEWORK FOR UAV DETECTION 76
5.1 Introduction 76
5.2 Anomaly Detection 76
5.3 Problem Definition using a Semi-supervised Approach 77

5.4 Dataset Description 78
5.5 Feature Extraction Using Wavelet Packet Transform 78
5.6 UAV Detection Algorithm 79
  5.6.1 Local Outlier Factor 80
  5.6.2 Training Process of Local Outlier Factor 83
5.7 Experimental Results and Discussions 83
  5.7.1 Selection of k (number of neighbors) 83
  5.7.2 Model Performance based on Classification Metrics 84
  5.7.3 Performance at Difference SNR 85
5.8 Conclusion 86

6 HIERARCHICAL LEARNING FRAMEWORK FOR UAV DETECTION AND IDENTIFICATION 87
6.1 Introduction 87
6.2 Dataset Description 87
6.3 Stacked Denoising Autoencoder 87
  6.3.1 Proposed SDAE Architecture 92
6.4 Local Outlier Factor 93
6.5 Training the SDAE-LOF Model 93
6.6 Hierarchical Classification 95
  6.6.1 Local Hierarchical Classification 96
  6.6.2 Hierarchical Classification Algorithm Definition 97
  6.6.3 Proposed Hierarchical Classification Algorithm 97
6.7 Feature Extraction 98
  6.7.1 Hilbert Huang Transform 98
  6.7.2 Wavelet Packet Transform 101
6.8 Experimental Results and Discussions 102
  6.8.1 Evaluation of SDAE 102
  6.8.2 Evaluation of LOF 103

6.8.3 Evaluation of each model in the Hierarchical Classifier 108
6.8.4 Evaluation of the Hierarchical Classifier 108
6.9 Conclusion 110

7 CONCLUSION AND FUTURE WORK 111
7.1 Conclusion 111
7.2 Future Work 113

REFERENCES 114

CURRICULUM VITAE 123

LIST OF TABLES

TABLE Page

1 Comparison of pros/cons for different UAV detection and tracking techniques. 8

2 RF bands use in wireless applications. 14

3 Related work on RF based UAV Detection and Identification. 32

4 Comparison of four open-source dataset for RF-based UAV detection and identification research. 38

5 Catalog of RF devices used in the experiment for RF fingerprints acquisition. All of these devices operate at 2.4 GHz frequency band. 40

6 Summary of the number of signals captured at LOS for each device in CardRF dataset. 49

7 Comparison of the three forms of transfer learning approaches (adapted from [1]). 61

8 Average classification accuracy (%) of the ML models and SqueezeNet based on the feature extraction methods and the state of RF signal use for fingerprinting at 30 dB SNR. 64

9 Average classification recall (%) of the ML models and SqueezeNet based on the feature extraction methods and the state of RF signal use for fingerprinting at 30 dB SNR. 65

10 Average classification precision (%) of the ML models and SqueezeNet based on the feature extraction methods and the state of RF signal use for fingerprinting at 30 dB SNR. 66

11 Average F_1-score (%) of the ML models and SqueezeNet based on the feature extraction methods and the state of RF signal use for fingerprinting at 30 dB SNR. 66

12 Computational time cost for classifier's inference given in seconds. 74

13 Statistical features definition from wavelet packet transform. 79

14 Average accuracy of LOF model based on the validation and test set when increasing the number of neighbors for local density estimation at 30 dB SNR. 84

15 Definition of statistical features used for feature extraction. 101

16 Performance measure of each local classifier in the hierarchy. 108

17 Performance measure of the local hierarchical classifier at different levels in the hierarchy. 109

18 Comparison of pros/cons using supervised and semi-supervised learning approach for UAV detection and identification. 112

LIST OF FIGURES

FIGURE Page

1 UAV threat mitigation steps. 5

2 RF signal from DJI Phantom 4 controller with various parts (i.e., transient and steady state) of the signal labeled. 16

3 Example of wavelets used in wavelet transform: (a) Haar, (b) Morlet (c) Symlet, (d) Mexican hat. 20

4 Illustration of a wavelet scaling operation. 22

5 Illustration of a wavelet shifting operation. 22

6 SVC explained. 25

7 kNN explained. 26

8 Decision tree algorithm explained. 27

9 Random forest algorithm explained. Four decision trees are connected in parallel to contribute in giving a prediction by using a voting system that is based om majority rule. 28

10 A typical neuron. 29

11 DNN Propagation (adapted from [2]). 29

12 A scenario of the collection of RF signals from various devices (i.e UAV and ISM devices such as WiFi and Bluetooth) to aid the development of RF-based UAV detection and identification system for an infrastructure under surveillance. Both the UAV controller signal and UAV signal can be exploited for UAV detection and identification. 40

13 UAVs from (a) DJI Matrice 600 Pro, (b) DJI Inspire 1 Pro, (c) DJI Phantom 4 Pro, (d) DJI Mavic Pro 1, (e) Beebeerun. The UAVs considered include medium, small and mini size. 41

14 UAV controllers from: (a) DJI Inspire 1 Pro, (b) Beebeerun (c) 3DR Iris, (d) DJI Phantom 4 Pro, (e)DJI Matrice 600 Pro. 41

15 The satellite view from Google map showing the Lake Wheeler site where the experiment was carried out. The indicated rectangle is the exact area. The site is located at 4191 Mid Pines road, Raleigh, North Carolina, USA. 42

16 Electrical circuit block diagram RF signal sensing and capturing system. . . 42

17 The setup of the RF signal sensing and capturing system at the Lake wheeler site where the experiment is carried out. 43

18 The RFSSCS setup for visual line of sight capturing of signals from : (a) a UAV controller, (b) a UAV (DJI Matrice 600). 44

19 The RFSSCS setup for beyond-visual-line-of-sight capturing of signals from UAV. 45

20 Captured RF signal from: (a) DJI Matrice 600 controller, (b) DJI Matrice 600 UAV (c) Beebeerun controller, (d) Beebeerun UAV (e) DJI Inspire controller (f) DJI Inspire UAV. 46

21 Captured RF signal at non-line-of-signal from: (a) DJI Matrice 600 UAV (b) DJI Inspire UAV. 47

22 Directory tree for data label of the Cardinal RF dataset. 48

23 Directory tree for the LOS UAV signal labels showing the UAV models and UAV flight modes as sub-directories and sub-directories of sub-directories, respectively. 48

24 Directory tree for the NLOS UAV signal labels showing the UAV models and UAV flight modes as sub-directories and sub-directories of sub-directories, respectively. 49

25 Single level Haar wavelet decomposition for raw RF signal prepossessing. The raw captured signal from RF devices is denoted as $y[n]$. The low pass and high pass filter are $g[n]$ and $h[n]$ respectively. The approximate and detail coefficients are denoted as $a[n]$ and $d[n]$. Approximate coefficient, $a[n]$, is used for feature extraction. 52

26 Divisions of the four proposed feature extraction methods based on CWT and WST. The decomposed signal, $a[n]$, is the approximation coefficients from the single HWD. 54

27 Examples of scalogram extracted from the transient state of the captured RF signals: (a) IPhone 6S, (b) DJI Inspire, and (c) TPLink (WiFi device). . . 55

28 Examples of scalograms extracted from the steady state of the captured RF signals: (a) IPhone 6S, (b) DJI Inspire, and (c) TPLink (WiFi device). . . 55

29 Wavelet scattering transform process flow with the three successive steps in transforming a signal. 56

30 Tree of wavelet scattering transform algorithm for signal decomposition with mathematical expressions for computing the scalogram and scattering coefficients at each scattering levels. 56

31 Filter bank for the WST framework: (a) First filter bank, and (b) Second filter bank. 58

32 Examples of scattergrams extracted from the transient state the captured RF signals: (a) IPhone 6S, (b) DJI Mavic Pro 1, and (c) TPLink (WiFi device). 59

33 Examples of scattergrams extracted from the steady state of the captured RF signals: (a) IPhone 6S, (b) DJI Mavic Pro 1, and (c) TPLink (WiFi device). 59

34 A flowchart of the machine learning model. 60

35 A typical example of a fire module where the squeeze layer has two 1 × 1 filters and the expand layer has a mix of three 1 × 1 and three 3 × 3 filters. 62

36 Squeeze architecture showing the connections of the two convolutional layers and eight fire modules. 62

37 Confusion matrices: (a) Group-device classification using the transient state, CWT framework and kNN, (b) Specific-device classification using transient state, WST framework and SqueezeNet (c) Group-device classification using the steady-state, WST framework and SqueezeNet, and (d) Specific-device classification using transient state, WST framework, and Ensemble + PCA. 68

38 SNR results: (a) Group-device classification using transient state and CWT framework, (b) Group-device classification using steady state and CWT framework, (c) Specific-device classification using transient state and CWT framework, and (d) Specific-device classification using steady state and CWT framework. 70

39 SNR results: (a) Group-device classification using transient state and WST framework, (b) Group-device classification using steady state and WST framework, (c) Specific-device classification using transient state and WST framework, and (d) Specific-device classification using steady state and WST framework. 71

40 Running time for the feature extraction methods. 73

41 A two-level wavelet packet decomposition tree used to sub-band captured RF signals into packets for feature extraction. 78

42 Illustration of local and global outliers in a dataset with three clusters. The red datapoint is a global outlier and the green data points are the local outliers. 79

43 Illustration of the reachability distance for different data points p from point o assuming that k is equal to 5. 81

44 Confusion matrix of the LOF model ($k = 100$) when evaluating the model with the test set at 30 dB. 84

45 Classification accuracy under varying SNR and number of neighbors for the LOF model. 85

46 A neural network architecture of a typical autoencoder with one hidden layer and two nodes at the hidden layer. The hidden layer represents the code layer. 89

47 Common examples of activation functions: (a) ReLU, (b) Leaky ReLU (c) Sigmoid, (d) Tanh. 90

48 The schematic representation of a denoising autoencoder. An example x is corrupted by a stochastic process N to give an input x'. Input x' is passed into an autoencoder to map it to latent representation y through the encoder function f_θ. The decoder function $g_{\theta'}$ is used to reconstruct the input x' as output z. The reconstruction error is measured through the loss function $L(x, z)$. 91

49 The architecture of the proposed stacked denoising autoencoder for signal compression and extraction of latent representation. The SDAE's input and output size are 1024. The input layer has no activation function but the output layer uses a sigmoid as the activation function. The SDAE has five hidden layers with size 512, 128, 32, 128 and 512, respectively. Each of the hidden layers uses a ReLu as the activation function. The dimension of the latent representation is 32. 92

50 Stacked denoising autoencoder-local outlier factor (SDAE-LOF) training process for UAV Detection. The training process is a dual level process. The first level trains the SDAE and the second level trains the LOF. After the training of SDAE, the encoder is detached from the network and used in the second stage to compress the uncorrupted training data. The compressed training data is used to train the LOF model. Similarly, the testing data is first compressed using the encoder before passing to the trained LOF for UAV detection. 93

51 The two types of hierarchical structure: (a) tree-based structure, (b) DAG based structure. 95

52 Tree based structure for UAS identification using hierarchical classification. The nodes in yellow are nodes with more than one child node. A classifier is employed at every node with more than one child node. For this reason, the hierarchical classifier is made up of six classifiers. 98

53 The empirical mode decomposition showing the first two IMFs and the residual of the captured RF signals (i.e., using the steady state) from: (a) DJI Inspire, (b) Beebeerun. 100

54 The Hilbert spectrum showing the time-frequency-energy characteristics from the steady state of the captured RF signals: (a) DJI Inspire, (b) Beebeerun. 101

55 The learning curve of SDAE. This shows learning performance of the SDAE at every epoch. The loss function of the training set and validation set drop steadily per epoch. 103

56 Average accuracy of the LOF model based on the validation and test set when increasing the number of neighbors for local density estimation at 30 dB SNR. 104

57 Confusion matrix of the LOF model ($k = 20$) when evaluating the model with the test set at 30 dB. 105

58 (a) The relationship between training set size and accuracy for the LOF model, (b) The relationship between training set size and inference time for the LOF model. Using either Manhattan or Euclidean distance has no significant differences on the LOF model. The Manhattan and Euclidean distance outperforms the Cosine distance. 105

59 The classification accuracy of LOF model (distance measure is Manhattan) under varying SNR and the number of nearest neighbors. The number of nearest neighbors is increased at the step of 40 starting from 20 to 180. . . 106

60 Performance of the LOF model (the number of nearest neighbors is 20) under varying SNR and distance measure. The Manhattan and Euclidean distance have similar characteristics when varying the SNR of the signal. Both Manhattan and Euclidean distance outperform the Cosine distance. 107

LIST OF ALGORITHMS

ALGORITHM Page

1 Sifting Procedure of Empirical Mode Decomposition [3] 99

CHAPTER 1

INTRODUCTION AND DISSERTATION OVERVIEW

In this chapter, the motivation for this work is discussed. After listing why drone detection and identification (DDI) systems are important, current challenges in developing a DDI system are highlighted. A brief overview of state-of-the-art approaches is discussed. In the final sections, the contributions of this work are highlighted.

1.1 Motivation

Every new technology comes with both positive and negative impacts on our society. We cannot negate or nullify the downside of such technologies. Unmanned aerial vehicles (UAVs) (commonly known as drones) and their technologies are drastically spurring an evolution in our world as their application is significantly broad. These systems are no longer restricted to military usage, as civilians are rapidly adopting it for both commercial and noncommercial use. Over 1.7 million UAVs are now registered in the United States of America, with over 70% of registered UAVs being utilized for recreational reasons and the remaining percentage for commercial reasons [4].

The immensity of UAV applications in different domains such as healthcare [5], logistics, remote sensing [6], data acquisition [5], precision agriculture [6], environmental and disaster management [7] has resulted in the emergence of several new business prospects. More so, the bid to integrate civilian UAVs into the national low altitude airspace by the Federal Aviation Administration's (FAA) Next Generation Air Transportation System (NextGen) and the National Aeronautics and Space Administration's (NASA) Unmanned Aircraft Systems Traffic Management project is already in progress [8, 9]. On the other hand, the growing use of UAVs is raising both security and privacy concerns. UAVs are known to have been used for cybercrimes, terrorism, drug smuggling, invasion of privacy, and other malicious intents. Some of these instances in include:

- On the 28th of March 2014, it was unveiled that UAVs can be used for information harvesting. A malicious system can be mounted on a UAV to collect or steal personal information or track an individual using the wireless signal from their smartphone without the user's knowledge [10].
- On the 26th of January 2016, an unauthorized UAV crash-landed in the White House lawn [11].
- On the 9th of April 2015, a dissident in a protest against the nuclear energy policy in Japan landed a UAV with a radioactive payload on the roof of the Japanese Prime Minister's house [12].
- On the 24th of June 2018, it was reported in the news that smugglers are using UAVs to inspect the vulnerable access points along the US-Mexico border [13]

Aside from these malicious acts, there are little or no strict regulations on who can own or purchase a UAV. So, airspace contravention is another challenge of UAVs. For example, a hobbyist without the knowledge of airspace restriction or regulation can fly a UAV in restricted airspace, violating the airspace regulation.

More importantly, every month, the FAA receives over 100 cases or incidents of UAVs sighted in an unauthorized area in the United States [14]. These sightings are from pilots, citizens, and law enforcement, which involve human efforts as UAV detection mechanisms are still not prevalent.

UAV manufacturers have started enabling UAVs to have the geofencing capabilities. Geofencing is restricting the ability of a UAV to fly into or take off within some predefined areas (e.g., areas marked as No-Fly Zones) based on GPS information. For instance, DJI has a system called GEO (Geospatial Environment Online), which enables geofencing in their products [15] and Parrot's newly released FreeFlight 6 application also has a geofencing capability [16]. However, there is a need for a UAV detection system in geofencing-free areas that are sensitive to security and privacy.

While there are radar or velocity-speed guns to detect when a driver is speeding on highways by law enforcement agents, UAV detectors are not currently available for airspace and infrastructure monitoring. There is no automatic system in place for law enforcement agencies to detect airspace contravention caused by UAVs and solutions are still underdeveloped, or in the infant stage of development [17–19]. As a matter of fact, the radar system in the White House designed to detect flying objects was unable to detect the intruding UAV that crash-landed in 2016 [11]. Having this solution will enable law enforcement to tackle crimes involving UAVs. Hence, the ever-increasing application of UAVs has made UAV detection and identification research gain momentum in the research domain of UAVs and unmanned aerial systems (UAS) because of privacy and security issues [20, 21].

UAV detection and identification falls under the field of identification of non-biological entities called artimetrics. Artimetrics is a field synonymous with biometrics that involves identification, classification, and authentication of non-biological entities such as robots, software, avatars, and so on [22]. Biometric research has passed the infant stage of development as it has been in existence for more than three decades [23]. Emulating the successes of biometrics in the identification of non-biological entries (specifically electronic systems such as UAVs) is essential as our world gets more connected. This will enhance network security, forensic auditing, seamless authentication, and so on.

1.2 Challenges

Several factors make UAV detection difficult. These factors include:

1.2.1 Size of a UAV

The size of a UAV is one of the most important factors in designing a DDI system, and that is why RADAR systems, which are commonly used for detection of flying objects such as airplanes, has not been effective in DDI.

1.2.2 Modality of sensing

Sensing modality significantly determines operational limitations of detecting UAVs. From the literature, four different sensing approaches have been explored for DDI. These include audio, RF signal, video, and thermal sensing. While there is no one particular sensing technique that works perfectly, there are trade-offs to leverage when selecting a sensing mechanism. In a later section, detailed information will be provided for each sensing mechanism.

1.2.3 Detection range

A DDI system should have a long-range detection capability to enable a timely detection. Timely detection and identification of incoming UAVs are critical to public safety [24] because it allows countermeasures to be effected if a rogue UAV is detected from afar. However, the detection range depends on the countermeasure's response time and the stakeholder's requirements. This makes it difficult to define and qualify a good detection range.

1.2.4 Weather effect

Environmental effects could have an adverse impact on sensing techniques in DDI. For instance, in a high-temperature weather condition, the thermal sensing may not be effective. There is a need for DDI systems in which performance is not degraded by elements of weather or time of the day.

1.2.5 System evaluation

The evaluation of the DDI system expands beyond functional requirements. Meeting non-functional requirements (i.e., the criteria that characterised the operation of a system) such as availability, reliability, maintainability, detection accuracy and precision, and other performance metrics (e.g., response time, early detection, low false alarm, stealth, etc.) are central to the system design of DDI. Making a balanced trade-off among these requirements without any skew to one or more particular requirements is a difficult challenge to overcome.

1.3 Overview on state of the art

In [18], the mitigating steps for curbing security or safety-threatening UAVs are divided into three steps. The sequence of the steps is as shown in Fig. 1.

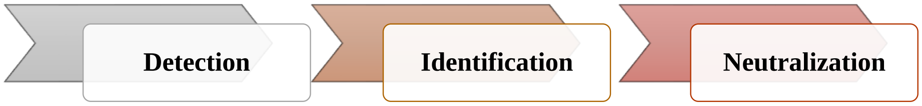


Figure 1: UAV threat mitigation steps.

1.3.1 Detection

This involves using sensors to capture necessary information that shows some attributes of the presence of UAV in the vicinity. Several sensing techniques have been exploited in the literature. Below are the most commonly-used sensing techniques:

1.3.1.1 Audio Based Technique

Generally, UAVs generate sounds (i.e., a humming sound) when in operation. The sound waves propagated from a given UAV when in operation are adopted as the audio fingerprint or signature of the UAV. Acoustic sensors such as a microphone can be utilized for the detection of sound pulses. The use of audio signals has been used for the DDI system in [25–28]. In [25], the author put forward a UAV identification system using the hidden Markov model (HMM) and the frequency analysis of audio signals emitted from UAVs. By using the sound produced by UAVs in different flight modes, the performance of their classification algorithm was improved.

A correlation analysis was adopted for UAV detection in [26, 27]. Here, the audio signature was transformed from the time domain to frequency using Fourier transform, and the resulting component was stored in a database. During classification, correlation analysis was used to analyze the frequency component of the unclassified sound signal with components in the database. Any correlated component in the signature database labels the unknown signal. From their results, it shows that correlation gives a high detection rate.

Furthermore, a distributed wireless acoustic detection system was proposed in [28] where software-defined radio (SDR) was also employed for the detection and localization of unauthorized UAVs.

1.3.1.2 Visual Imaging Technique

This involves using video surveillance cameras to monitor a restricted area. Human operators traditionally monitor the area by watching the real-time video or using data mining techniques (i.e., object detection algorithms) to extract anomaly scenes from the real-time video feeds. Video has been used to also detect UAVs by using cameras and data mining techniques [29–31].

1.3.1.3 Thermal Sensing Technique

This is an infrared thermography approach. It is a standard norm that any object with a temperature greater than absolute zero can radiate or emit an infrared spectrum. The rotor or motor of UAVs emits heat, and the thermogram of this heat energy can be used as a thermal signature for the UAV. In [32], a UAV detection system that uses thermal signatures, which is only for nighttime surveillance, was presented.

1.3.1.4 RADAR Technique

RADAR is the most commonly-used mechanism for detecting flying objects and is found at every airport worldwide. It uses radio pulses to detect aircraft, guided missiles, and other airborne moving-objects. A radar system has both transmitter and receiver. The transmitter propagates radio pulses toward an object so that the object can reflect the radio waves. The receiver receives the reflected waves, and the temporal properties of the reflected radio waves can be used to determine or identify the object.

In [33,34], radar systems for detection and classification of UAVs were proposed. In [33], a millimeter-wave (35 GHz frequency modulated continuous wave (FMCW)) radar was used to detect two different sized (small and medium) UAVs at a detection range between 30 - 90m. Using a radar system, Doppler signatures of UAVs and their respective spectral correlation functions (SCFs) were used to classify three UAVs. The SCFs of signatures were used to train a DBN classifier with an accuracy of 90% [34].

1.3.1.5 RF Technique

Most UAVs use a radio-controlled (RC) communication system. The radio pulses of the communication between the UAV and its flight controller are intercepted and collected as an RF signature for the given UAV. From the literature and studies, it has been observed that the RF signals emanating between UAV-flight controller communication are unique among UAVs. Using these RF signals as RF signatures is based on the premise that each UAV-flight controller communication has unique features that are not necessarily based on the modulation types or propagating frequencies but maybe as a result of imperfection in the communication circuitry in the devices [35]. RF signatures or fingerprints have been used for DDI systems [36–41]. According to [38], RF signals can be grouped into two major headings:

- RF physical layer features based.
- RF link layer (medium access control (MAC) layer) feature-based techniques.

The RF physical layer relies on the properties of the physical layer (open systems interconnection (OSI) model) in terms of the RF propagation or transmission of the UAV radio. Such properties include the amplitude envelope or the spectrum of the RF signal. The RF MAC involves using packet statistics for the detection and classification of WiFi-controlled UAVs. It is also called a WiFi fingerprinting technique. This approach uses a WiFi packet sniffer that intercepts the packet traffic between the UAV and its flight controller. In [39], the authors proposed a machine learning framework for the identification of UAV over encrypted WiFi traffic where packet size and inter-arrival time were extracted for training a classifier.

1.3.1.6 Multimodality signature

This approach involves combining one or more sensing techniques (i.e., sound, video, RF, and thermal) discussed above. In [42], ADS-ZJU was proposed in which hybrid (audio, video, and RF) passive surveillance systems were used for UAV detection, localization, and radio frequency jamming. In [19], using the system-of-system (SoS) approach, a multimodality UAV detection system for airports was proposed where a collection of independent sensors were integrated to extend the limitation of a single sensor.

TABLE 1: Comparison of pros/cons for different UAV detection and tracking techniques.

Detection Technique	**Pros**	**Cons**
Audio (e.g., [25–28])	It is cost-effective in implementation [42], its operation is passive [18].	Low detection accuracy, difficult to estimate the maximum detection range [18], performance susceptible to ambient noise, high complexity in acquiring and maintaining an audio signature database, it cannot be used for UAVs with noise cancellation [20].
Visual Imaging (e.g., [29–31]	Commercial-off-the-shelf solutions that can be rapidly utilized, it can be used for detecting autonomous UAVs.	Illumination of the environment or property under surveillance can degrade the performance, vulnerability to weather conditions, ineffectiveness for a crowded or cluttered area (i.e., line-of-sight (LOS) between the surveillance camera and the target is essential), a high-quality lens with an ultra-high-resolution may be needed to detect UAV at a long-range, it has limited coverage for a large environment as a camera can only focus on one direction per unit time.
Thermal sensing (e.g., [32])	Less susceptible to weather fluctuation, and background clutter [18]	UAVs have a low thermal signature, and it has limited coverage when a UAV is at non-line-of-sight (NLOS).
Radar (e.g., [33, 34])	Can be used for detecting autonomously controlled UAVs, and it is not susceptible to weather fluctuation.	It induces interference on other RF bands, especially in a crowded environment [20], it is not effective for UAV detection because the radar cross-section (RCS) depends on the absolute size of a flying object, making radar not effective, high cost of deployment, it requires line-of-sight to target for effective and efficient operation, and high powered radars are not recommended for areas with crowded human habitats because of their high active electromagnetic energy [42].
RF (e.g., [38, 41, 43, 44])	It is relatively inexpensive. Its operation is stealthy, and the size of the UAV does not affect its effectiveness. It works for both line-of-sight and non-line-of-sight. It is not dependent on protocol standardization [41], and beyond using it for detecting the presence of a UAV, it can be exploited for the identification of operation modes of UAVs [36, 39].	It is not a trivial task to use RF-based detection because UAV radios operate at the industrial, scientific, and medical (ISM) band and because there are several other devices (e.g., WiFi and Bluetooth) that operate at this same band, making it challenging to capture the UAV's RF signature.
Multi-modality signature	it increases the diversity of detection [45].	Fusing of signatures from spectrum of sensors together could increase redundant features and the computational time complexity.

Table 1 shows the comparison of these detection techniques where the advantages and disadvantages are summarized.

1.3.2 Identification

This involves identifying the target based on the data provided at the detection stage. One widespread mechanism integrated to identify an unauthorized UAV automatically and intelligently is data mining techniques. Data mining is the process of discovering knowledge or underlying patterns in data [46]. The frequently adopted data mining techniques in literature for DDI systems are machine learning, and deep learning (also known as deep neural network (DNN)) approaches.

1.3.3 Neutralization

This phase relies on the output of the identification phase. If the identification phase identifies a target as UAV, then an alarm is raised with a counteracting measure to bring down the UAV if necessary. Counteracting measures may include jamming the RF communication between the UAV-flight controller, shooting down the UAV, and other methods. However, this phase is not part of the focus of this work.

1.4 Design of UAV Detection System

While many different DDI systems have been proposed, there are certain design trade-offs to consider in developing a robust DDI system. These trade-offs are listed below:

- Stealth (low probability of intercept)
- Accuracy
- Cost
- High detection range
- Low false alarm rate

- Early detection - Timely detection and identification of incoming UAVs are critical to public safety [20]. The computational time complexity for detection must be minimal to allow early detection such that unnecessary computation from extracting signatures and classification must be pruned to avoid delays.

- Easy maintenance – Maintaining the signature database can be problematic in terms of space complexity, so a system for saving compressed signature might be considered.

1.5 Contribution

An efficient detection system is needed for UAV localization during the neutralization step. Hence, this work mainly focuses on detection and classification steps of curbing an intruding UAV in an environment. Therefore, UAV neutralization is out of the scope of this research. This work exploits the physical layer of the communication link between a UAV and its flight controller for UAV detection and identification. It is not a trivial task to use the RF sensing technique because UAV radios operate at the industrial, scientific and medical (ISM) radio band. Several other devices (e.g., WiFi and Bluetooth) operate at this same band, making it challenging to capture the UAV RF signature. With the challenges previously discussed, this work addresses some of these challenges and proposes an ML framework for an RF-based UAV detection system. The author's contribution includes:

1. Curate RF signals dataset from UAVs, UAV controllers, Bluetooth, and WiFi devices through an experimental approach. The RF signals dataset is called the Cardinal RF dataset with the acronym CardRF. The CardRF is publicly available to foster new development in the UAV detection and identification research community.

2. Analyze the impact of using the transient and steady-state of RF signals for UAV detection and identification using two wavelet analytic approaches (i.e., continuous wavelet transform and wavelet scattering transform). The impact of using either a coefficient-based or image-based signature from these two wavelet analytic approaches is considered.

3. Introduce the semi-supervised anomaly detection approach to UAV detection and identification. This approach is based on detecting outliers locally instead of globally. Here, two different approaches for extracting features from signals are proposed. One relies on using a wavelet packet transform, which is a time-frequency domain approach, and the other exploits unsupervised deep neural network technique (i.e., autoencoder, which is also called an auto-associator) for the compression of signals into a latent representation. The latent representation is used as a feature set or signature to detect UAVs.

4. Lastly, the author introduces a hierarchical classification framework for UAV identification. Beyond detecting the presence of UAVs, the framework also identifies the operation mode of UAVs. This framework exploits Hilbert Huang transforms and wavelet packet transforms for signal decomposition. Some statistical features are extracted from the decomposition products, which acted as signals' signatures.

1.6 Organisation

The dissertation is organized into seven chapters. Chapter II outlines the background knowledge and discusses the literature reviews. Chapter III introduces the Cardinal RF dataset and describes the experiment on how the data was curated. Chapter IV presents the wavelet transform analytics framework for RF-based UAV detection and identification. Here, the benefit of using either the transient or steady-state of RF signals for UAV detection is analyzed. Chapter V introduces a semi-supervised anomaly detection approach to UAV detection and identification by decomposing signals using wavelet packet transform. Chapter VI presents an unsupervised deep learning approach for compressing signals into a higher representation. This a shift from using time-frequency methods to extract features from signals. Subsequently, the author introduces a hierarchical classification framework for UAV identification where both Hilbert Huang transform and wavelet packet transform are exploited for feature extraction. Chapter VII discusses the overall conclusions and future direction of this research work.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

This chapter provides an overview of the anatomy of unmanned aerial system (UAS) and RF fingerprinting. Also, the fundamentals of using time-frequency domain functions for signal analysis are examined. More so, the role of machine learning (ML) and deep learning (DL) in UAV detection and identification is discussed. Finally, a review of the past works or papers on RF-based UAV detection and identification is carried out.

2.2 Anatomy of Unmanned Aerial System

The first UAV was developed due to a United States Navy research project led by Elmer Sperry. Originally, UAVs are used for an operation that is defined as dangerous, dirty, and dull (i.e., the three'Ds operation) [47]. However, the application of UAVs has extended to several civilian uses.

The UAS architecture consists of three parts: the unmanned aircraft (UAV), the control station, and the communication data link [47]. The unmanned aircraft is the airborne robot that flies. The control station is the infrastructure that controls and monitors the airborne robot during flight. The communication link is a wireless communication system that facilitates the flow or transmission of information (e.g., control commands and telemetry data) between the control station and the unmanned aircraft.

The UAV control system can be categorized into three possible ways. These include [8]:

- Remote Pilot Control: This is also known as human-in-the-loop. The control mechanisms are implemented remotely by a human operator.

- Remote Supervised Control: This is also referred to as human-on-the-loop. Here, the control measures are carried out in part by the UAV and a remote human operator. This type of control helps the drone to operate independently of human commands while allowing for human intervention.

- Full Autonomous Control: This is also called human-out-of-the-loop. This involves the UAV making all of the requisite decisions for a mission to be completed successfully.

The type of control system in civilian drones used by hobbyists is generally human-in-the-loop, and pilots remotely operate them. A Human-in-the-loop control system is usually a small hand-held controller/transmitter. Military UAVs can be controlled using any of the three control types, and the control system can be a large infrastructure with workstations [47].

Furthermore, drone missions can be classified into Visual Line-of-Sight (VLOS) missions and Beyond Visual Line-of-Sight (BVLOS) missions depending on the distance between the UAV and ground control system (GCS) [47]. In a VLOS mission, control signals can be transmitted and received directly by radio waves. Conversely, in the BVLOS mission, the UAV is controlled via satellite communications, or through a relay infrastructure [47].

Radio Controlled (RC) and VLOS mission UAVs use physical and data links. The physical and data link layer of the UAV communication link must comply with Federal Communications Commission (FCC) Title 47 part 15 (RF devices), and FCC Title 47 part 97 (amateur radio services) in the USA [48]. The UAV-to-flight controller communication link is synonymous with transmitter to receiver communication. The flight controller transmits RF signals which are mostly control signals to the UAV. The UAV can send back telemetry signals or secondary data (such as images or videos) to the control center (flight controller). The telemetry signals carry and provide the basic information about the UAV's status (e.g., the estimated flight time, flight direction, position, speed, battery level, etc.) to the flight operator. The secondary data signals usually transmit or carry the payload information. For instance, a camera is a common payload in UAVs that provide either image or video data; this information is transmitted using the link.

For these signals to be transmitted, different modulation techniques are used by different UAV manufacturers. Direct Sequence Spread Spectrum (DSSS), Frequency Hopping Spread Spectrum (FHSS), and Orthogonal Frequency Division Multiplexing (OFDM) are some of the most commonly used modulations in UAV-flight controller communication [49, 50]. Most UAV radios available on the market use the unlicensed spectrum (ISM band), which are listed below [49]:

1. 433 MHz to 434 MHz
2. 902 MHz to 928 MHz
3. 2.4 GHz
4. 5.8 GHz

Coincidentally, several wireless devices use the unlicensed instrument, scientific and medical (ISM) RF band as the operating frequency [51] and examples of applications that operate at this frequency band are listed in Table 2. Examples of devices that use this band include alarm signals, car remote keys, WLAN, wireless microphones, cordless phones, RC drones, and so on.

TABLE 2

RF bands use in wireless applications.

Frequency	Application
13/27 MHz	Near Field Communication
915/868 MHz	ZigBee
2.4 GHz	IEEE 802.11b, g, Bluetooth, ZigBee
5.8 GHz	IEEE 802.11a

One way to distinguish devices operating at the same frequency band is to exploit their physical layer characteristics through the RF signal propagated from the devices. This approach is called RF fingerprinting, and it is extended to UAV detection and identification. RF fingerprinting is exploited in differentiating one UAV from another and other ISM devices like WiFi and Bluetooth.

2.3 RF Fingerprinting

Radio frequency fingerprinting or RF fingerprinting is the process of using RF signals broadcast by a transmitter to distinguish it among other transmitters [52]. A signal is generally defined as a time-dependent or time-varying voltage or current that conveys information. Also, according to the authors in [53], a signal is an electromagnetic wave that embeds information in its propagation. Beyond conveying a message or information, a fundamental assumption is that the fingerprint or signature of a device (i.e., transmitter) is embedded in their RF signal. The uniqueness of a device's RF fingerprint is intrinsic to the device's electronic circuitry (e.g., diode, capacitors, resistors, filters, printed circuit board, etc.), manufacturing processes, and environmental factors [54–57]. Radio fingerprints differ from human fingerprints in that radio fingerprints can change over time due to system aging, channel effect, replacement of electronic components in the system, and so on. However, human fingerprints are immutable even with time [52].

Device detection and identification using RF fingerprinting has a wide range of use in military and/ or civilian communications. RF fingerprinting can be used for forensic analysis, authentication [55], military surveillance, network security [58], protocol identification, device identification (e.g., UAV, mobile phone), spectrum utilization, cognitive decision making, and so on. In exploiting RF fingerprinting for device detection and identification, there are two important steps involved, which are RF fingerprint generation and RF fingerprint classification [58].

2.3.1 RF Fingerprint Generation

Fingerprint generation involves the collection or/and extraction of unique features from a signal so as to enable the distinguishing of devices. Selecting the representative features of a signal and measuring the discrimination between two signals are the challenges of using signals for device identification, especially when the signals are propagating at the same frequency [59] [60]. The direct use of a signal without feature extraction may have a detrimental effect on computational and memory efficiency of a detection and identification system [58]. Feature extraction is a critical process that must be carried out meticulously. Hence, an extracted feature should have the following attributes [58]:

- To reduce or achieve optimal time and space complexity, the feature set must be kept minimal.
- Intra-device repeatability and stability must be enhanced.
- Inter-device uniqueness must also be enhanced through the extracted feature set.

The transient and the non-transient state (steady state) are the two regions where the unique features can be extracted from a signal. Fig. 2 shows a typical example of a signal from the DJI Phantom 4 (UAV) controller with the labeling of the transient and steady region. The transient state is short-lived, and it happens as systems or devices are turned on and off [61]. According to the authors in [52], the three reasons that make using the transient state of a signal for real-time identification of device challenging are:

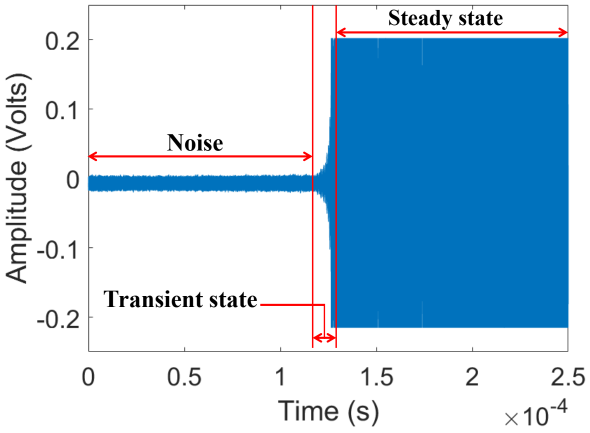


Figure 2: RF signal from DJI Phantom 4 controller with various parts (i.e., transient and steady state) of the signal labeled.

1. Difficulty in classical frequency analysis due to the short duration of the transient state.
2. The non-stationary nature of the signal.
3. Broader variation in the structure and integration of components that generated transients.

On the other hand, the steady region is a state that does not flicker. It has a longer duration than transient state and is the state that occurs at the start and end of transient state's interval [55].

2.3.2 RF Fingerprint classification

RF fingerprint classification can also be called signal classification. It involves the process of comparing, identifying, and distinguishing between two or more RF fingerprints. Because a signal is a form of time-domain data (i.e., time series), data mining principles for analyzing or classifying time series data can be introduced to signal classification. From the standpoint of data mining, signal classification can be accomplished in two ways. These are instance-based classification and feature-based classification [59].

2.3.2.1 Instance-based classification

Instance-based classification is a distance metric approach which is a commonly used and straightforward approach for comparing or classifying time series data [59]. It involves computing the distances between the time series data by directly using their sequential values (i.e., the raw values). The distance measure must be clearly defined and calibrated to understand and identify the (dis)similarity in a time series data [60]. This method is computationally demanding and requires high storage cost because of the high dimensionality of time series [60,62]. Examples of instance-based algorithms in the literature are Euclidean distance [63], dynamic time warping [64,65], longest common subsequence [66] and so on.

2.3.2.2 Feature based classification

Feature-based classification is also called the representation method [60]. It involves comparing sets of extracted features from the signals or a time series data. This approach aim at reducing the dimensionality of a time series while retaining the fundamental characteristics of the underlying data [60]. The benefit of dimensionality reduction includes [67]:

1. Reduction of computational cost.

2. Improved classification accuracy through noise reduction in data.
3. Features that can be more easily interpreted, which can help in identifying characteristics of a target or class.

Examples of the algorithm used for extracting features from time-series data are discrete Fourier transforms, discrete wavelet transforms, discrete Cosine transforms, single value decomposition, and so on.

2.4 Analysis of Signal in Time-Frequency Domain

The following are the three most commonly used algorithms for analyzing a signal in the time-frequency domain:

2.4.1 Fourier Transform

Time-frequency domain analysis is an essential step in extracting unique attributes (signatures) from a signal. Just as Ohm's law is significant in understanding and analyzing electrical circuits, the Fourier transform (FT) is the most applied mathematical function for analyzing the frequency content of a signal in signal processing. It essentially represents a signal by series of sines and cosines in the frequency domain, revealing several important features of the signal that were hidden or unknown in the time domain. Given a time-variant variable or vector (signal) $f(t)$, the FT of the signal is defined as:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}\, dt \tag{1}$$

where j is the unity imaginary number which is the square root of -1, (i.e., $\sqrt{-1}$), and ω is the range of angular frequencies within the signal. It defines the frequency content of the signal. Angular frequency is defined as $2\pi f$. The above expression simply compares the signal and the exponential function, which is an orthogonal function of sine and cosine. The larger the similarity between the two, the higher the amplitude of the transform that results. While the FT defined above is for a variable with an infinite set (continuous or analog signal), it can be used for a variable that has a finite set (discrete or digital). The

Discrete Fourier Transform (DFT) is the Fourier transform for analyzing the frequency content of a digital signal. Given a time-variant variable $f(t)$, which has a finite set or sequence $f(0), f(1), f(2), \ldots, f(N-1)$ which are equally spaced at time T with N number of points, the DFT of the variable $f(t)$ is defined as:

$$F(\omega) = \int_{0}^{(N-1)T} f(t)e^{-j\omega t}\,dt \tag{2}$$

If each point in the finite sequence of $f(t)$ is treated as a unit impulse function, then the integral of impulse is given as 1. Also, the integral of e^x is e^x. Hence, $F(w)$ can be expressed as:

$$F(\omega) = \sum_{k=0}^{N-1} f(k)e^{-j\omega kT} \tag{3}$$

2.4.2 Short Time Fourier Transform

While the FT shows the frequency components of a signal, it does not give the frequency variation with time, and it is of limited application, particularly with signals of varying frequencies such as non-stationary signals. The Short-Time Fourier Transform (STFT) overcomes this limitation by sliding a window through the signal $f(t)$ at short interval/time, along the time axis, and performing FT on the data within that window.

$$STFT(\omega, \tau) = \sum_{k=-\infty}^{\infty} f(k)h(k-\tau)e^{-j\omega kT} \tag{4}$$

where τ is the time shift and k is the frequency component index and h is the window function. The effectiveness of the STFT depends on the choice of window function used. The challenge with STFT is finding a suitable window size that balances time resolution and frequency resolution. Choosing a window size that gives higher frequency resolution gives a lower time resolution and vice versa. For instance, a wide window would give a better frequency resolution but a poor time resolution.

2.4.3 Wavelet Transform

The wavelet transform (WT) seeks to mitigate the challenges of STFT. Unlike the STFT, which slides a fixed window through the signal, the wavelet transform utilizes a variable window function [68]. WT is the convolution of a wavelet function (i.e., wavelike function) with the signal under analysis [69]. The WT of signal $f(t)$ is given as [70]

$$w(s,\tau) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} f(t)\psi(\frac{t-\tau}{s})dt \tag{5}$$

where $s > 0$ is scaling factor, $\psi(\frac{t-\tau}{s})$ is the template function (i.e., base wavelet) and τ is the time shifting factor. The WT decomposes a signal into a set of basis functions that allows us to analyze hidden qualities or features that characterize the signal [71]. These basis functions are obtained using a single template function (base wavelet) to perform both scaling and shifting operations along the time axis. Some examples of base wavelets that are commonly used for WT are shown in Fig. 3.

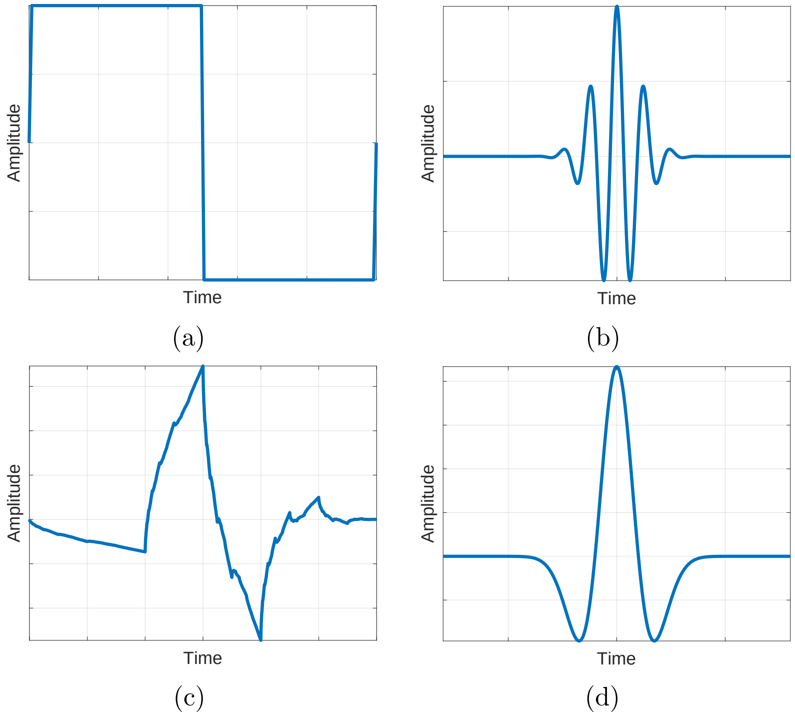


Figure 3: Example of wavelets used in wavelet transform: (a) Haar, (b) Morlet (c) Symlet, (d) Mexican hat.

A wavelet must satisfy three mathematical requirements which are [69]:

1. A wavelet must have a finite energy:

$$E = \int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty \tag{6}$$

where E is the energy of the wavelet and $|\psi(t)|$ is the magnitude of the wavelet. If $\psi(t)$ is a complex function then the magnitude of the wavelet must be computed using the real and imaginary parts.

2. A wavelet has no zero frequency component:
   Given that the Fourier transform of wavelet $F(\Phi)$ is:

$$F(\Phi) = \int_{-\infty}^{\infty} \psi(t) e^{-j\omega t}\, dt \tag{7}$$

   then the condition below must be satisfied.

$$C_g = \int_{0}^{\infty} \frac{|F(\Phi)|^2}{\Phi}\, dt < \infty \tag{8}$$

   This implies that $F(0) = 0$ or the wavelet $\psi(t)$ must have a zero mean [69]. C_g is called the admissibility constant, and its value depends on the selected wavelet.

3. For a complex wavelet, the Fourier transform of the wavelet must be real and vanish for negative frequencies.

Scaling is the process of stretching or shrinking the wavelet to match the feature of interest. The scaling factor, s, is used to categorize this process. The higher the scaling factor, the higher the stretch; the lower the scaling factor, the higher the shrink. The scaling factor is inversely proportional to the frequency of the signal [71]. Fig. 4 depicts a schematic of the scaling operation of WT when analyzing a signal. The wavelet is initially shrunk and later stretched based on the scaling factor.

The shifting factor, τ, is used to move the base wavelet along the time axis. This is illustrated in Fig 5. It looks for the similarities between the signal and the base wavelet like other signal transforms such as FT and STFT. The underlying features or attributes in the signal can be extracted during the operations. These similarities are in the form of wavelet coefficients [70]. When there is a similarity in shape between the base wavelet and signal at a given scale and time shift (i.e., location), a higher value is returned as the coefficient in that particular instance. On the other hand, dissimilarity between the wavelet and signal will return a low value coefficient.

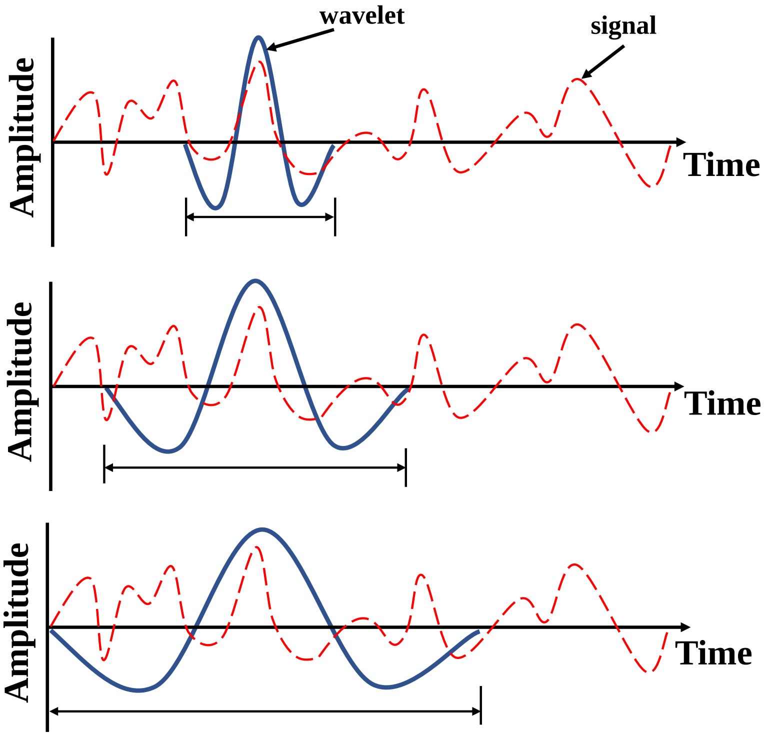


Figure 4: Illustration of a wavelet scaling operation.

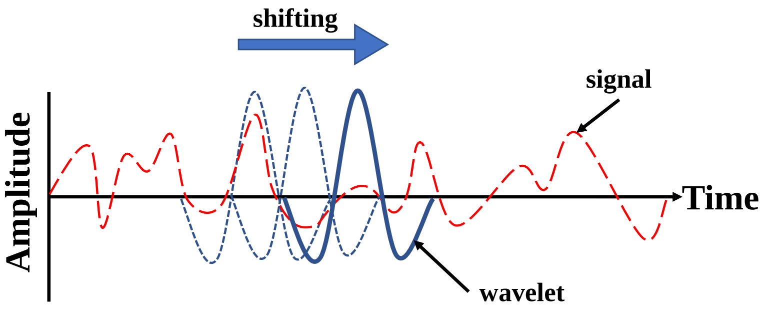


Figure 5: Illustration of a wavelet shifting operation.

During the transform, the various shifting factors and scaling factors are used which fill up the transform plane [69]. When factoring is done in a continuous approach then this type of WT is referred to as continuous wavelet transform. Conversely, when the factoring is in a discrete step then the WT is called discrete wavelet transform.

2.5 Roles of Machine Learning and Deep Learning Techniques in UAV Detection Systems

Most of the UAV detection and identification systems in the literature use data mining techniques to make data-driven detection or identification. Advancement in the semiconductor industry has enabled computers to break both computing power and memory limitations. This fosters the application of data mining techniques in system design, making systems more data-driven and intelligent. Recently, the most frequently-used data mining techniques for pattern recognition in literature is machine learning (ML). ML is a subset of artificial intelligence (AI) and it has brought significant impact to so many fields (e.g., healthcare, cybersecurity, etc.).

ML can be sub-divided into supervised learning, un-supervised learning, semi-supervised learning and reinforcement learning. Supervised learning is learning from labeled data and it is a type of inductive learning [46]. Unsupervised learning is learning from unlabeled data. Semi-supervised learning is a learning paradigm that exploits both supervised and unsupervised learning. It involves learning from labeled and unlabeled data [72]. Reinforcement learning is another learning paradigm where an agent observes and acts in an environment to maximize reward while minimizing penalty [73].

Supervised learning has been widely applied in the majority of literature on UAV detection and identification. However, no attention has been paid to applying other learning approaches. For the essence of this work, supervised learning and semi-supervised learning are discussed in the later sections.

2.6 Supervised Learning

The primary goal of supervised learning is to learn the mapping of feature x to target or label y provided there is a training set that consist of pairs (x_i, y_i) [74]. Supervised learning can be either a regression or a classification problem. When the target or label is a continuous set then the problem is treated as a regression. Conversely, when the target is a discrete set then the problem is termed as a classification. The problem of UAV detection and identification using RF sensing and ML techniques has been treated as a supervising learning problem (i.e., classification) where the RF signature of a UAV is used for detecting or identifying the UAV. Some of the supervised learning algorithms that have been utilized are random forest, support vector machine (SVM), k-nearest neighbor (kNN), decision tree, logistic regression, support vector domain description (SVDD) [75], and deep neural networks (DNN).

2.6.1 Logistic Regression

Logistic regression is also called logit regression. It is a classification algorithm for estimating the likelihood that an instance belongs to a specific class using a linear function of a set of predictor variables [46, 73]. It uses a sigmoid function that gives an output value between 0 and 1. The sigmoid function is mathematically defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{9}$$

The logistic regression model prediction is expressed as

$$\widehat{y} = \begin{cases} 0 \; if \; p \; < \; 0.5 \\ 1 \; if \; p \; \geq \; 0.5 \end{cases} \tag{10}$$

2.6.2 Support Vector Machine (SVM)

The principle of an SVM as a classifier (support vector classification (SVC)) is based on structural risk minimization (SRM), which involves maximizing the width of a margin between different classes. SVC operates on the premise of fitting (linear or nonlinear) margins or hyperplanes to define decision boundaries between classes. To keep it simple, let us consider two-dimensional (x_1 and x_2) input features, which are attributes that define whether a drone is present or not in an environment. In Fig. 6, SVC fits a line (hyperplane) as boundaries that separate the two classes based on the training data where the region with a blue circle represents the presence of a drone and the region with black circles denotes “no drone is present.” The data points where the broken lines pass are called the support vectors. This concept serves as the basis for the model to perform classification. Any unclassified sample (input features) passed into the model can be classified accordingly by checking the region in which the sample point falls.

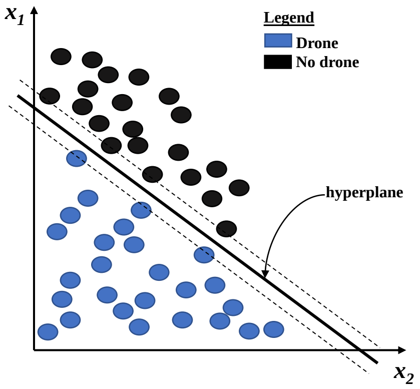


Figure 6: SVC explained.

The benefit of SVM is that it requires a relatively small amount of training data, and it is not sensitive to the dimensionality of the training set [46]. SVM works for both linear and nonlinear models [73].

2.6.3 K-Nearest Neighbor

kNN is based on feature similarity where decision boundaries are determined locally [46]. The determination of the nearest neighbors is based on a distance measure. The distance measure is one of the important factors that affect the performance of this algorithm because there are several ways to measure distances. These include Manhattan, Euclidean, Cosine, Hamming, Mahalanobis, Minkowski, Spearman, Jaccard, and so on. However, it is common to use Euclidean distance if the value of k is large as it does not bias the prediction [46]. The training of a kNN algorithm is to determine the value of k that will give a minimal loss function. The kNN classification process can be summarized by the five listed steps below [46]:

1. Determine of the value k nearest neighbors, which gives a minimal loss function.
2. Compute the distance between the unlabeled sample and all the training samples

(labeled samples).

3. Sort the training samples based on the calculated distance and select the k nearest neighbors of the unlabeled sample.
4. Determine the class of each k nearest neighbors.
5. Find the modal class using the majority rule.

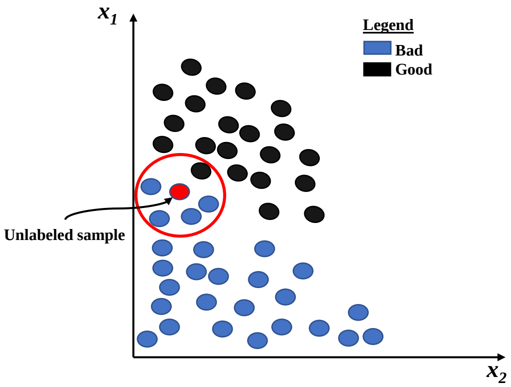


Figure 7: kNN explained.

Because classification is based on the majority rule, the value of k needs to be an odd integer. Let us consider Fig. 7 where kNN is used for detecting whether a drone is present or not based on two-dimensional input features (x_1 and x_2). The black and blue circles represent the training data which are called labeled data. The red circle is the unlabeled sample that needs to be classified. Assuming the k parameter is set to 5, this implies that the algorithm will pick any five labeled data points closest to the red circle to determine the class of the unlabeled sample. In the five nearest neighbors to the red circle, there are four data points representing the presence of a drone and one data point representing the absence of a drone. Based on the majority rule, the unlabeled data point is classified as a detected drone. Some of the merits of kNN are: i) it has a low training cost; ii) it is simple to implement and interpret; iii) it has few hyperparameters (i.e., the distance metric and k). On the other hand, the setbacks of kNN are: i) it has high inference cost especially when dealing with high dimensional dataset; and ii) it is sensitive to unbalanced datasets.

2.6.4 Decision Tree

This is a logic-based approach in learning a system function where decision rules for classifying or predicting a value are established from the training data [46]. These rules form a tree-like representation. This implies that there is a root, node, and leaf. Each node consists of one or more logical constraints for splitting the data until a leaf is attained. The algorithm searches for the best feature in the feature sets of the data, which could be used to establish the first decision rule, forming the root point for the tree. The first decision rule would split the data into subsets such that data with similar attributes are grouped. This process is recursively executed on every subset until a leaf is achieved that defines the labeling of samples. For instance, given that training set T has a two-dimensional feature set x_1, x_2 which determines that a drone is present in an environment or not, if x_1 is the attribute that optimally sub-divides T into T_1 and T_2 based on certain constraints or conditions then this serves as the root of the tree. This is illustrated in Fig. 8 For each subset of T (i.e., T_1 and T_2), the decision tree algorithm searches another rule to split the subsets into a leaf node.

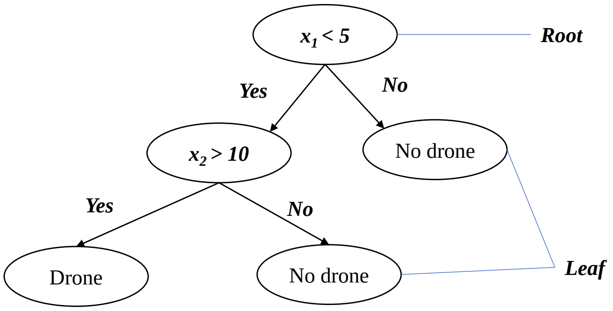


Figure 8: Decision tree algorithm explained.

There are tree-growing algorithms which are used for constructing a decision tree. These include classification and regression trees (CART), C4.5, ID3 and so on [46,76]. The some of the advantages of decision trees are: i) the generation of the model is fast and interpretable [46]; ii) feature scaling is not necessary in the data preprocessing stage [73]; and iii) it requires small amount of training data. On the other hand, the setbacks of decision trees are: i) it is sensitive to new samples in training data. Anytime new samples are added to the training set, it changes the whole structure of the model; ii) It is prone to overfitting [73]; iii) as the complexity of the data or problem increases, the complexity of the algorithm also increases [46].

2.6.5 Random Forest

A decision tree algorithm is a fundamental unit of the random forest algorithm, and the random forest is an ensemble of decision trees [73]. Here, it randomly selects a subset of the training set and trains two or more decision trees that are connected in parallel. For classification purposes, random forest output is based on the majority rule, where the vote from each decision tree is aggregated. This is shown in Fig. 10 where four decision-tree models are connected to vote for the final output.

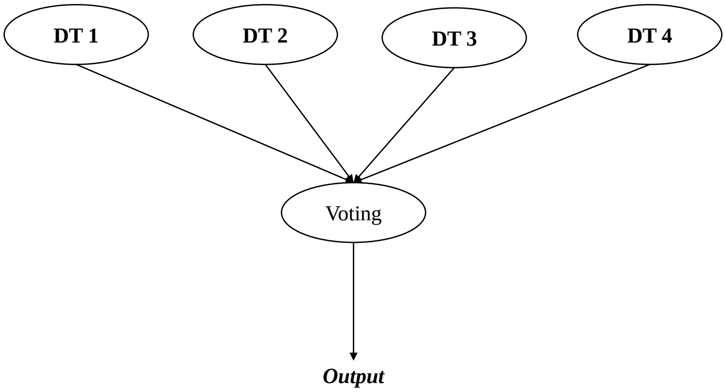


Figure 9: Random forest algorithm explained. Four decision trees are connected in parallel to contribute in giving a prediction by using a voting system that is based om majority rule.

Random forest has better performance when compared to decision tree and is efficient when dealing with large datasets. Conversely, random forest increases the complexity of the learning process. This implies more computational resources will be required during the learning process. More so, the training cost is higher than using a decision tree algorithm.

2.6.6 Deep Neural Network

Deep neural networks (DNNs) consist of processing units called neurons which are interconnected by links in layers. There are three essential parts of a neuron [46], and these include input links, the summation, and activation function. The input links connect the inputs $(x_1, x_2, \ldots, x_n)$ to the node of a neuron, and each link has a weighted value. The internal node has two sequential parts. Firstly, the summation function sums up the products of a given input with their corresponding weight (i.e., the weighted sum of inputs) and bias constant. The output S of the summation is given by equation 11

$$S = x_1w_1 + x_2w_2 + \ldots + x_{n-1}w_{n-1} + x_nw_n + b \tag{11}$$

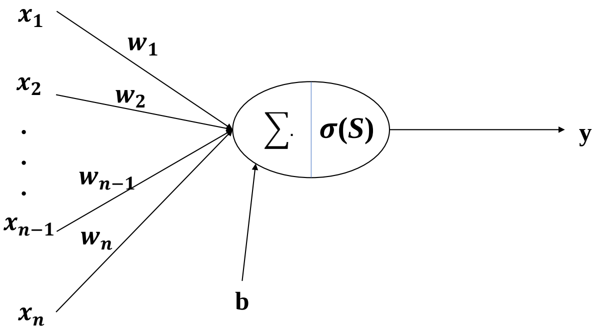


Figure 10: A typical neuron.

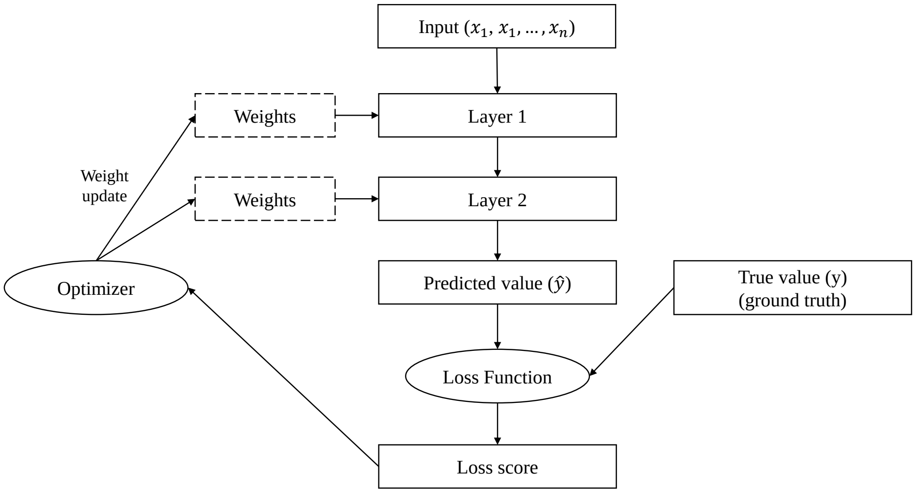

Input $(x_1, x_1, \ldots, x_n)$
Weights
Layer 1
Weight update
Weights
Layer 2
Optimizer
Predicted value $(\hat{y})$
True value (y) (ground truth)
Loss Function
Loss score

Figure 11: DNN Propagation (adapted from [2]).

The bias b is a constant which is synonymous to the intercept in a linear equation, and it is used to adjust the output S before applying the activation function by either increasing or decreasing depending on if the constant is a positive or negative value [46]. The output S serves as an input to the activation function. The activation function is like a switch that determines or transforms the output of a neuron. Examples of activation functions are sigmoid, linear function, rectified linear unit (ReLU), saturated linear function (SeLU), softmax, and so on [46,73].

Each layer of a DNN holds the weights that are used for mapping the input values to the target value (label). The training process of a DNN involves finding the set of values appropriate for each layer's weights. At the initial stage of the learning process, the weights of each layer are initialized with certain values (this can be random values); these values are used to compute the expected target value (this expected target value is also called the predicted value) using the input values. In order to foster learning, the algorithm uses a control mechanism called the loss function to guide the learning process. The loss function, also called the objective function, compares the predicted value and the ground truth; the difference of the comparison is called error or the loss score. Based on the loss score, the optimizer uses backward propagation to adjust the weights of each layer to reduce the loss score of the next forward propagation. This process is repeated until the loss score is minimized. DNN does not require feature extraction or selection like other classical ML algorithm. Also, a trained DNN can be reused for another task using the principle of transfer learning. On the other hand, the setbacks of DNN are: i) it requires a large amount of data for the learning process; ii) the training process is computationally demanding; iii) it cannot be interpreted; and iv) the deeper the network, the higher the complexity of the algorithm.

2.7 Semi-supervised Learning

One important and common application of semi-supervised learning is label propagation. The goal of label propagation is to use labeled data to propagate the labeling of unlabeled data. There are several semi-supervised methods for label propagation. These include the graph-based method, expectation maximization with generative mixture models, co-training, self training and so on [72]. However, there are other forms of semi-supervised learning such as positive and unlabeled data learning (i.e., PU learning) [72,77]. PU learning is similar to binary classification. The training set in PU learning consists of only one (i.e. positive) class. The positive class is solely used to train a model, and any unlabeled data that falls outside of the positive class's distribution is assigned to the negative class [77]. PU learning is sometimes referred to as one-class classification [78].

Semi-supervised learning can be a transductive or inductive learning [72]. Transductive learning deals with inference from particular case to particular case without producing a generic hypothesis or model (i.e., a generalization function) for predicting the labels of unseen data [79]. Given a training set T that consists both labeled data (X_l, Y_l) and unlabeled data X_u, where $(X_l, Y_l) = \{(x_{1:l}, y_{1:l})\}$ and $X_u = \{(x_{l+1:k})\}$, transductive learning tries to find the labels of the k points in the unlabeled data. On the other hand, inductive learning observes a training set with labeled data only to generate a general rule(s) or mathematical function (i.e., predictive model) for predicting labels of unseen data.

2.8 Criteria of Algorithm Selection

In literature, a classifier's performance metrics (such as recall, f1-score, precision, and accuracy) have been the key methods for evaluating or measuring the performance of the algorithms for UAV detection and identification systems [36–39, 44]. However, other important factors to consider are training cost, interpretability of the algorithm, inference time [39], and the number of features. The UAV detection and identification system needs a prompt reaction time to detect the presence of drones so that necessary neutralization steps can be taken in a timely manner. In this case, the inference time and other classifier metrics are the most imperative factors that will be considered in selecting the appropriate algorithm.

2.9 Literature Survey on RF-based UAV detection and identification

Exploiting RF fingerprinting for UAV detection and identification is actively being investigated in the literature. Several frameworks for UAV detection and identification have been proposed. It is difficult to quantitatively evaluate all the frameworks on the same scale because of several design or experimental factors. However, a few of the factors include:

1. Environmental factors: The surveillance environment (e.g., background noise) can cause channel effects that can lead to variation in RF fingerprints. For instance, an experiment can be conducted indoors, outdoor, or even in an anechoic chamber.
2. Type of UAVs used for the study or experiment.
3. Type of UAV signal utilized: The detection of UAV using RF sensing can either exploits the signal emitting directly from the UAV itself, UAV controller, or a combination of both.
4. System for capturing the RF signals.

TABLE 3: Related work on RF based UAV Detection and Identification.

Literature	Feature extraction method	State of the signal		Algorithms	Accuracy	Interference	Inference time
		Transient	Steady				
[36]	DFT	N/A	N/A	DNN	99.7%	None	No
[38]	DWT, statistical features	Yes	No	ML algorithms	98.13%	Bluetooth, WiFi	Yes
[41]	PCA	N/A	N/A	AC-WGANs	95%	WiFi, unknown signal	No
[43]	CSI	N/A	N/A	rule-based algorithm	86.6%	WiFi	No
[44]	DWT, statistical features	Yes	No	ML algorithms	96.3%	None	No
[80]	power spectral density	N/A	N/A	logistic regression	91%	None	No
[81]	DFT and CNN	N/A	N/A	DNN	94.6%	None	No
[82]	spectrogram	Yes	N/A	CNN	99.5%	Bluetooth, WiFi	No

The techniques or approaches proposed in the literature can be qualitatively assessed. Table 3 gives the summary of the frameworks that have been proposed in the literature. It shows the feature extraction method, the signal region for extracting RF fingerprints, algorithms, classification accuracy, consideration for interference signals and classifier's inference time.

In [36], three deep neural networks (DNN) classifiers are proposed for detecting the presence of UAVs, the type of UAV, and the flight mode of the UAV, respectively. The RF signals from three UAVs are collected over the air in an indoor experiment. The signals are converted into the frequency domain by using discrete Fourier transform (DFT). The frequency component of the signals was used to train a deep neural network (DNN) model. An average accuracy of 99.7%, 84.5%, and 46.8% is achieved when detecting and identifying the presence of UAV, the type of UAV, and the flight mode of the UAV, respectively. However, the authors did not consider other ISM devices that operate at the same frequency band (i.e., 2.4 GHz) as UAV-flight controller communication. Other ISM devices (like Bluetooth or WiFi devices) can interfere in the data collection stage or during the operation of the classifier. More so, the impact of channel effect on the three classifiers is not examined or assessed. However, the authors made available the first open-source UAV dataset for UAV detection and classification research. This open-source dataset is can DroneRF.

In [38], UAV detection and identification using RF signatures under the presence of wireless inference (WiFi and Bluetooth) was studied. The author proposed a multistage detector to separate the RF signal of the drone radio from both background noise and other wireless inference signals. In the first stage, a two-state Markov model-based Naïve Bayes is used to assess if the captured signal is an RF signal or a noise. Once verified that the captured signal is an RF signal, the second stage identifies the source of the RF signal (whether the signal is a WiFi, Bluetooth, or a UAV controller signal) by assessing the bandwidth and modulation properties of the signal. In the last stage, any detected UAV RF signal is classified using an ML algorithm. The transient state of the UAV controller signal is used as the basis for the RF signatures. Fifteen statistical features are extracted from the RF energy transient time-frequency domain, which are statistical estimations such as mean, absolute mean, standard deviation, skewness, kurtosis, variance, entropy, root, peak value, peak to peak value, shape factor, crest factor, impulse factor, RMS and clearance factor. The author demonstrated that the energy transient in the UAV controller signal is unique to different UAVs, so this feature vector from the energy transient of the RF signal is then used for the classification of fifteen different drones. Using five ML algorithms (Random forest, KNN, discriminant analysis, NN, and SVM) as classifiers, it was observed that random forest outperforms other classifiers with an accuracy of 98.53%, followed by KNN with an accuracy of 97.30%. Because there is a possibility for the fifteen features to be correlated among each other and to lower the computational cost of the classifier, neighbor component analysis (NCA) was used to select the three most significant features, which are shape factor, kurtosis, and variance. The three most important features were then used to train the five classifiers. It was observed that the accuracy of the random forest drops to 97.73%, but KNN outperforms the random forest classifier with an accuracy of 98.13%. Although the computational time of the classifiers drops, random forest performs better than KNN in terms of computational efficiency. The setback of this approach is that the capture window for the RF signal to properly contain the embedded energy transient needs to be high, else the transient state can be missed, and the feature set from noisy transient can be extracted, thereby misleading the classifiers. Ambient noise can inhibit the capturing of the transient state of the RF signal. According to the authors in [55], it was noted that transient feature extraction is only efficient and good for classification performance when the beginning and the end of the transient state can be steadily detected and identified.

In [41], a discriminator that is based on an Auxiliary Classifier Wasserstein Generative Adversarial Networks (AC-WGANs) was proposed for the detection and classification of UAVs. The motivation for this approach is to use a small amount of data compared to the amount of data utilized by classical ML algorithms or DNN. Here, the UAV signal was captured, and the amplitude envelope of the signal is extracted. To reduce both time and space complexity, the dimensionality of the amplitude envelope is reduced by using principal component analysis (PCA). The reduced features are fed into AC-WGANs for UAV classification. Both indoor and outdoor experiments were conducted using four different UAVs and a WiFi signal as interference. An accuracy of 95% and above 80% was obtained in indoor and outdoor, respectively. Also, the behavior of the classifier was assessed under varying SNR.

In [43], a rule-based algorithm was proposed for UAV detection. The effect of mobility, spatiality, and vibration on channel state information (CSI) of a UAV signal is quantified as a feature set to detect a UAV. The mobility effect was measured by monitoring the signal fluctuation. Both amplitude and phase information of the CSI is used to compute the eigenvalues of amplitude and phase correlation matrices. The maximum and second maximum of the matrices' eigenvalues are used to determine the fluctuation in a signal. More so, the spatiality effect was measured by calculating the elevation angle between a UAV and the signal receiver using a super-resolution angle-of-arrival (AoA) estimation method. Similarly, the vibration effect was measured by carrying a frequency domain analysis on a filtered and smoothened CSI data to determine the UAV's vibration frequency. The quantification of these three effects is used to create rules to determine the presence of a UAV in an environment. The accuracy of the rule-based model was 86.6% in detecting a UAV.

In [80], the authors proposed a logistic regression discriminator for UAV flight mode classification. RF signals from UAVs are captured and transformed into image-based representations, which are power spectral density (PSD), spectrogram, histogram, and raw IQ (in-phase/quadrature) constellation. A pre-trained CNN model called ResNet50 is used to extract features from each image. The extracted features are then used to train to different logistic models for comparison. An average accuracy of 91% was achieved for flight mode identification of UAV using the PSD of UAV signals as the UAV signature or RF fingerprint. More so, PSD outperforms the other three image-based representations.

In [81], the author proposed a multi-channel 1-dimensional CNN framework for UAV detection and identification. The RF signal from the 13 channels within the 2.4 GHz band (i.e., 80 MHz) is captured and re-channeled into eight channels without overlapping. Each channel's bandwidth is 10 MHz, and CNN is used to extract features from each channel. The extracted features are fed into a deep learning classifier. An average accuracy of 100%, to 94.6% and 87.4%, is achieved when detecting and identifying the presence of UAV, the type of UAV, and the flight mode of the UAV, respectively. However, the behavior of the classifiers is assessed under varying SNR. Similarly, other ISM devices' signals propagating in the environment could have been captured with the UAV signals because all the 13 channels of the 2.4 GHz band (i.e., 80 MHz) are monitored.

The authors in [83] proposed a classifier for classifying identical UAVs (i.e., same manufacturer and model) that are in hovering flight mode. Seven identical DJI M100 UAVs are used in their experiment, and the data collection was done in an RF anechoic chamber. An average accuracy of 91% was achieved using multiple neural networks (i.e., AlexNet) as the classifier. However, the authors made no attempt to consider inference mitigation strategy for other ISM devices operating at 2.4 GHz in a real-world setting. More so, the use of multiple neural networks (NN) as classifiers increases the complexity of their proposed algorithm, and the only flight mode considered is hovering.

In [84], a deep residual network classifier is proposed for UAV detection and identification using a spectrogram matrix of signals as an RF fingerprint. The experiment was conducted in an anechoic chamber using nine different UAVs and one WiFi device for data acquisition. An average classification accuracy of 99% was obtained at 0 dB SNR. However, before extracting the spectrogram matrix, AWGN was added to the collected RF signals. This could have possibly introduced some biases in their results.

In [85], the authors proposed a framework for detecting and classifying multiple UAVs by exploiting the transient state of the control and video signals from UAV. The STFT of the signals is extracted as RF signature to determine the number of of UAVs present in an environment. A classification accuracy that is over 98% was achieved using KNN, DNN, CNN, SVM, or decision tree. However, the authors did not consider other ISM devices operating at the same frequency as the UAV communication link that could act as wireless interference. The impact of channel noise on their model's performance was not assessed.

Inn [86], the authors proposed a system that detects the presence of UAV in an environment by exploiting UAV controller signals and video signals emanating from the UAV. The signals are decomposed using empirical mode decomposition to obtain the intrinsic mode functions (IMFs). The STFT of the IMFs is obtained as features that are used to train machine learning algorithm (i.e., KNN, DNN, CNN, SVM, or decision tree). The classification of UAV types (i.e., manufacturer's model) was not considered. More so, the behavior of their model under channel noise was not examined. The author as assumed that there is not major RF interference from the environment.

2.10 Conclusion

This chapter presents the basis for this work, and it influences every aspect of this research. The survey of recent papers shows the trend of using supervised learning for UAV detection and identification system through feature-based classification of RF signals. In the subsequent chapters, it is assumed that readers have the necessary background knowledge.

CHAPTER 3

CARDINAL RADIO FREQUENCY DATASET

3.1 Introduction

This chapter introduces and describes the dataset used for this research. The dataset was collected through an experimental approach. The experiment was conducted in an outdoor setting. The experimental setup, the procedure for data acquisition, and the catalog of devices are discussed.

The goal of the dataset is to demonstrate the feasibility of the proposed techniques in this work for UAV detection and identification problem.

3.2 Review of Publicly Available Dataset

To the best of the author's knowledge, there are only four publicly available datasets for RF-based UAV detection and identification research. Table 4 provides the summary of the datasets' comparison.

TABLE 4: Comparison of four open-source dataset for RF-based UAV detection and identification research.

Reference	**Environment**	**Number of UAVs**	**Same OEM percentage**	**Type of signal**	**Operation mode**	**Interference devices**	**Sampling device**
[36]	indoor	3	Approx. 67% Parrot UAVs	Not clearly stated	Yes	No	USRP
[87]	indoor	17	24% Spektrum UAVs and 29% DJI UAVs	Only UAV controller signal	No	No	Oscilloscope
[84]	indoor (anechoic chamber)	9	20% Spektrum UAVs	Mostly UAV controller signals and two UAV signals	No	1 WiFi	USRP
[83]	indoor (anechoic chamber)	7	100% DJI M100 UAVs	Not clearly stated	No	No	USRP
This work	outdoor	6	67% DJI UAVs	Both UAV and UAV controller signals	Yes	2 WiFi and 5 Bluetooth devices	Oscilloscope

The DroneRF dataset was proposed in [36], and it has been actively used in the UAV detection and identification research space or community. The data collection process was conducted indoors (in a Lab) using a universal software radio peripheral (USRP) and three

UAVs. However, it is assumed that for a given UAV, any signal from the UAV or UAV controller is captured. This is because the author did not clearly state the type of signals captured. More so, there is limited information about the indoor environment, which is a factor that can influence RF signature significantly. More so, there is no consideration for other ISM devices.

The experimenters in [87] curated 17 UAV controller signals in an indoor setting. However, signals from UAV themselves are not included in their repository or considered. This repository cannot be used to investigate the classification of UAV's flight modes. More so, no RF signals from other ISM devices like Bluetooth or WiFi devices are included.

The authors in [84] proposed the DroneSignal dataset, which includes UAV control and video signals, and one WiFi signal. However, a large portion of the signals is control signals. More so, the data collection is done in an anechoic chamber which is a deviation from reality.

Lastly, the authors in [83] curated RF signals from seven identical (i.e., same make and model) UAVs in an anechoic chamber. The only flight mode considered is the hovering mode.

In this work, I curated an open-source dataset called the Cardinal RF dataset, abbreviated as the CardRF dataset. Here, both the UAV and UAV controller signals from six UAVs, WiFi signals from two WiFi devices, and Bluetooth signals from five devices are used to acquire the CardRF dataset in an outdoor environment.

3.3 System Modeling

One of the assumptions that guide the collection of UAV signals, UAV controller signals, WiFi signals, and Bluetooth signals is illustrated in Fig. 12. Given an infrastructure that is under surveillance for the presence of UAVs, because of the ubiquity of WiFi and Bluetooth devices, it is essential to consider such devices operating at the same frequency band as UAVs.

More so, it is assumed that all UAVs, WiFi devices, and Bluetooth devices operate at 2.4 GHz. The detection system, which consists of an antenna, oscilloscope, and other components, is used to intercept the signals and store the signals in a repository.

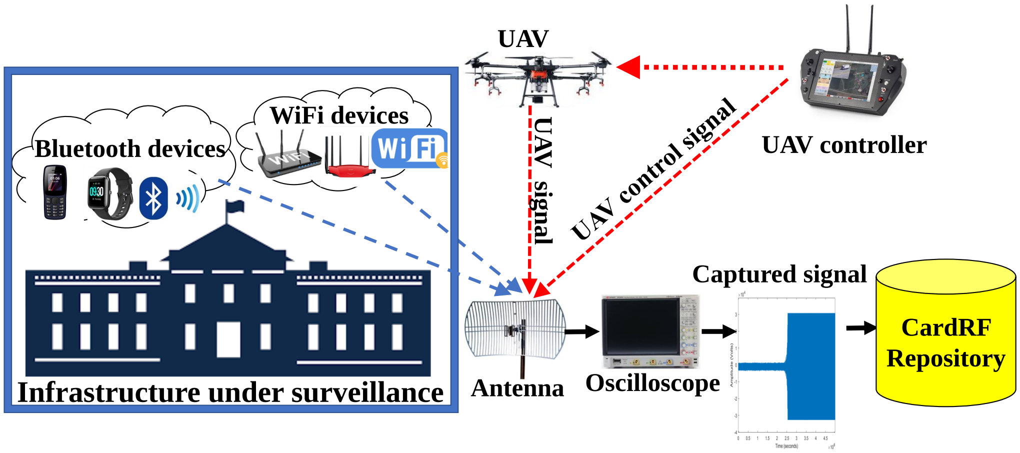


Figure 12: A scenario of the collection of RF signals from various devices (i.e UAV and ISM devices such as WiFi and Bluetooth) to aid the development of RF-based UAV detection and identification system for an infrastructure under surveillance. Both the UAV controller signal and UAV signal can be exploited for UAV detection and identification.

3.4 CardRF Data Acquisition

TABLE 5

Catalog of RF devices used in the experiment for RF fingerprints acquisition. All of these devices operate at 2.4 GHz frequency band.

Device	Make	Model
UAS	DJI	Phantom 4
		Inspire
		Matrice 600
		Mavic Pro 1
	Beebeerun	FPV RC drone mini quadcopter
	3DR	Iris FS-TH9x
Bluetooth	Apple	iPhone 6S
		iPhone 7
		iPad 3
	FitBit	Charge3 smartwatch
	Motorola	E5 Cruise
WiFI	Cisco	Linksys E3200
	TP-link	TL-WR940N

The catalog of devices used for the data acquisition is listed in Table 5. Similarly, the images of the UAVs and UAV controllers are shown in Fig 13 and Fig 14, respectively. It should be noted that four out of the six UAS are from the same manufacturer (i.e., DJI). The selection of the UAS is based on two premises which are:

- Inter-UAV uniqueness (i.e., UAVs from different manufacturers) simplifies the problem of identifying UAVs from different manufacturers, but makes it difficult to identify UAVs of the same manufacturer or model.
- DJI has the largest market share among UAV manufacturers.

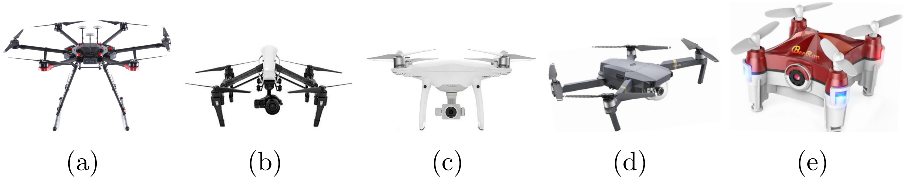

(a) (b) (c) (d) (e)

Figure 13: UAVs from (a) DJI Matrice 600 Pro, (b) DJI Inspire 1 Pro, (c) DJI Phantom 4 Pro, (d) DJI Mavic Pro 1, (e) Beebeerun. The UAVs considered include medium, small and mini size.

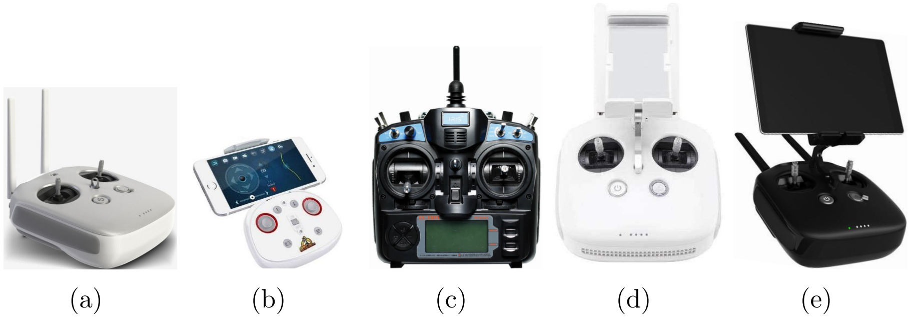

(a) (b) (c) (d) (e)

Figure 14: UAV controllers from: (a) DJI Inspire 1 Pro, (b) Beebeerun (c) 3DR Iris, (d) DJI Phantom 4 Pro, (e)DJI Matrice 600 Pro.

3.4.1 Meta Data

The experiment was conducted during the Summer of 2020 from the 17th - 21st August 2020 between the hours of 12 noon - 5 pm daily using the AERPAW (Aerial Experimentation Research Platform for Advanced Wireless) facility [88] which is called the Lake

Wheeler site. The site is located at 4191 Mid Pines Road, Raleigh, North Carolina, USA. The satellite view of the location is marked in Fig 15. The area is several hectares of free space with two buildings and vegetation-covered areas.

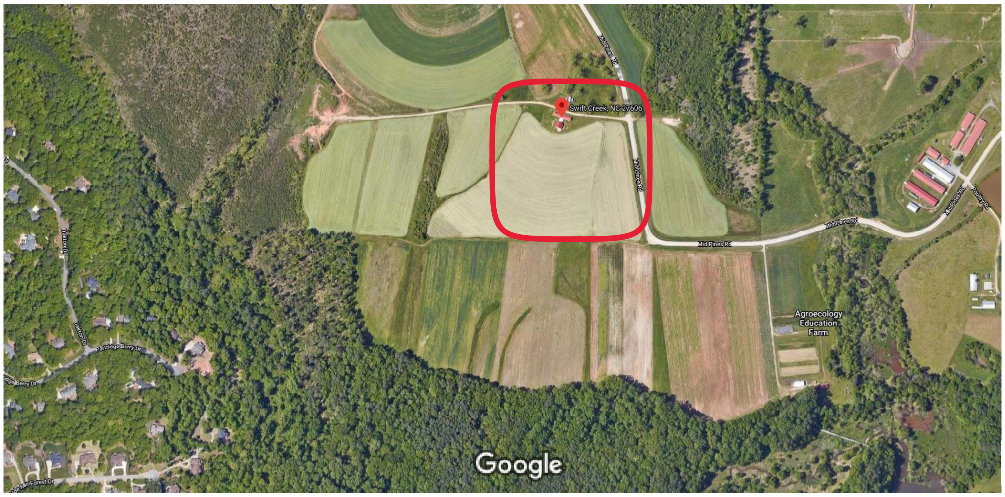


Figure 15: The satellite view from Google map showing the Lake Wheeler site where the experiment was carried out. The indicated rectangle is the exact area. The site is located at 4191 Mid Pines road, Raleigh, North Carolina, USA.

The flying of UAVs is performed under the guidelines of FAA regulation and coordinated by two FAA-certified pilots. Similarly, COVID-19 restrictions are put into consideration. To reduce operator error, signal acquisitions with time, date, and other essential metadata are automatically appended to data files [52].

3.4.2 RF signal sensing and capturing system

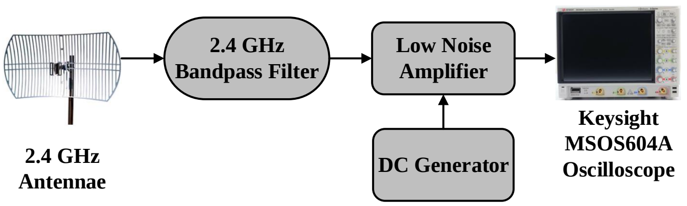


Figure 16: Electrical circuit block diagram RF signal sensing and capturing system.

The RF signal sensing and capturing system (RFSSCS) is a crucial aspect of the data collection. The electrical circuit or building block of the signal capturing system is shown in Fig. 16 which is made up of five electrical and electronics components. These components include an antenna, bandpass filter, low noise amplifier, DC generator, and oscilloscope.

The antenna is a 24 dBi 2.4 GHz parabolic grid and a directional antenna. It is used to intercept RF signals as they propagated from the RF devices (i.e., UAVs, UAV controllers, Bluetooth, and WiFi devices). The intercepted signal is passed through an Airvu 2.4GHz bandpass filter, which guarantees the acquisition of a 2.4 GHz frequency band. A low noise amplifier (LNA), FMAM63007, which operates from 2 GHz to 2.6 GHz with 30 dB gain, is used to amplify the bandpass signal. A DC (direct current) generator (Siglent SPD3303X-E power supply unit) is utilized to power the low noise amplifier attached to the bandpass filter. A 6 GHz bandwidth Keysight MSOS604A oscilloscope, which has a sampling frequency of 20 GSa/s is utilized for collecting and storing the captured RF signals from the devices. The setup of the signal capturing system on the field is shown in Fig 17.

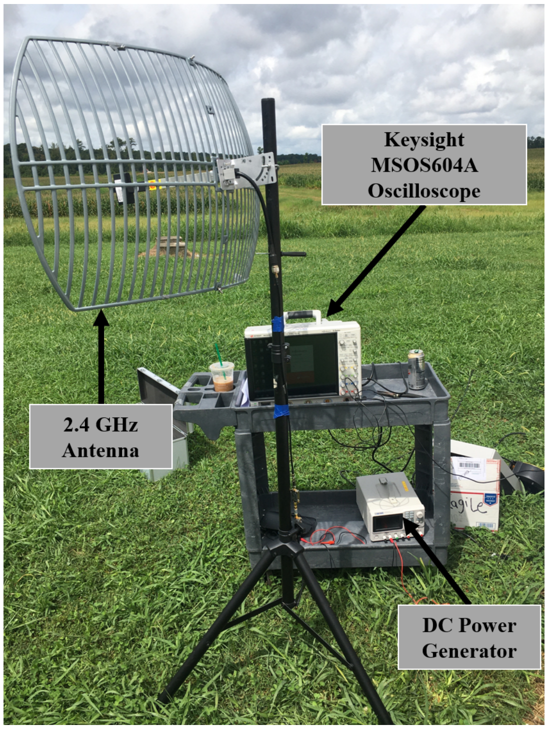


Figure 17: The setup of the RF signal sensing and capturing system at the Lake wheeler site where the experiment is carried out.

3.4.3 Calibration of oscilloscope based on the absence of RF signal

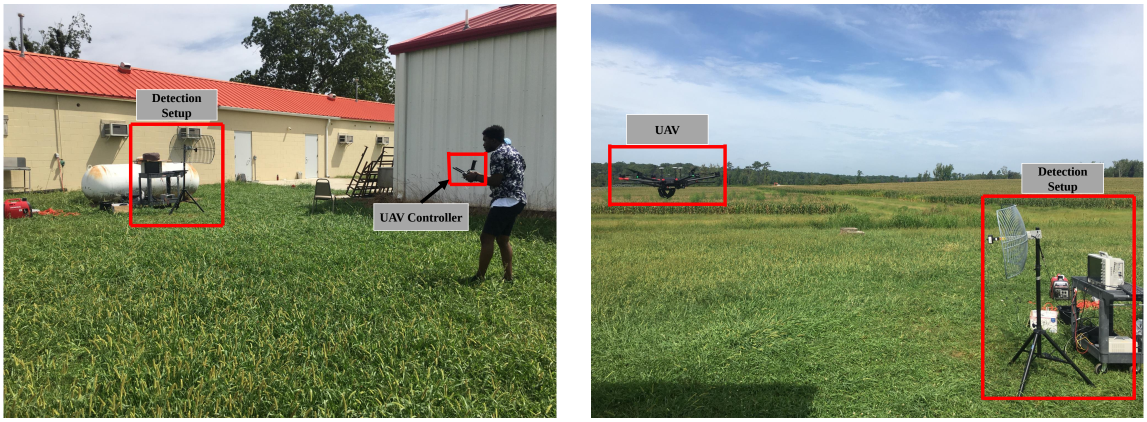
The oscilloscope has a threshold trigger for signal detection. First, the oscilloscope is used to measure the environment's background noise level and calibrate the threshold above that level. When a signal is present, the energy level of the signal exceeds the threshold, causing the oscilloscope to capture and store the detected signal. The captured data becomes the raw RF signals that are stored in the CardRF repository.

3.4.4 Data acquisition category

The data acquisition is categorized into visual line-of-sight (VLOS) and beyond-visual-line-of-sight (BVLOS) data collection.

3.4.4.1 Visual line of sight data

The VLOS synonymous to light-of-sight (LOS). During the VLOS signal capturing, there is no obstruction between the RFSSCS and the device (i.e., UAV, UAV controller, or other devices). Fig 18(a) and Fig 18(b) show the VLOS capturing of a UAV controller and UAV (DJI Matrice 600), respectively. Signals are captured at a distance range of 8 - 12 meters between the devices and RFSSCS.



(a) (b)

Figure 18: The RFSSCS setup for visual line of sight capturing of signals from : (a) a UAV controller, (b) a UAV (DJI Matrice 600).

3.4.4.2 Beyond-visual-line-of-sight

BVLOS is synonymous to non-line-of-sight (NLOS). In BVLOS data capturing, there is an obstruction between the RFSSCS and the devices. This is to analyze the impact of multi-path fading or other channel effects on the RF signature. Fig 19 shows the BVLOS UAV signal capturing where the UAV is flying at the adjacent of a building. Only three UAVs' signals (i.e., DJI Inspire, DJI Matrice 600, and DJI Phantom) are captured for this collection.

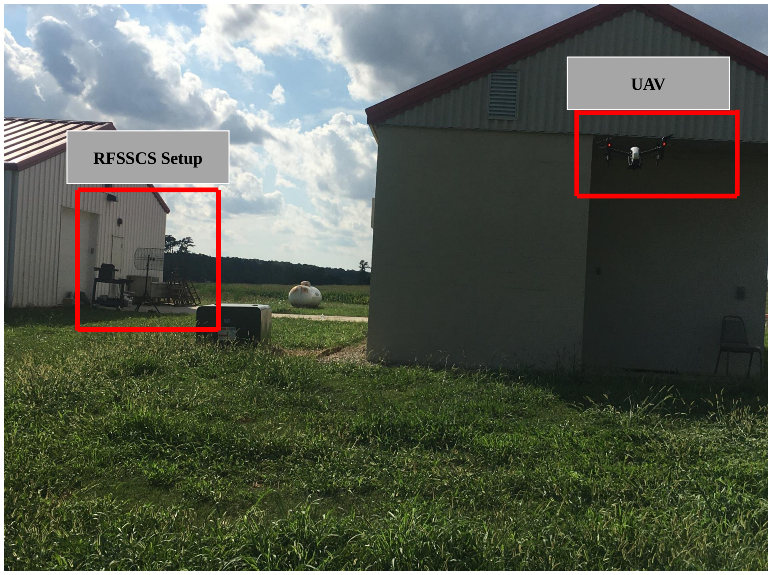
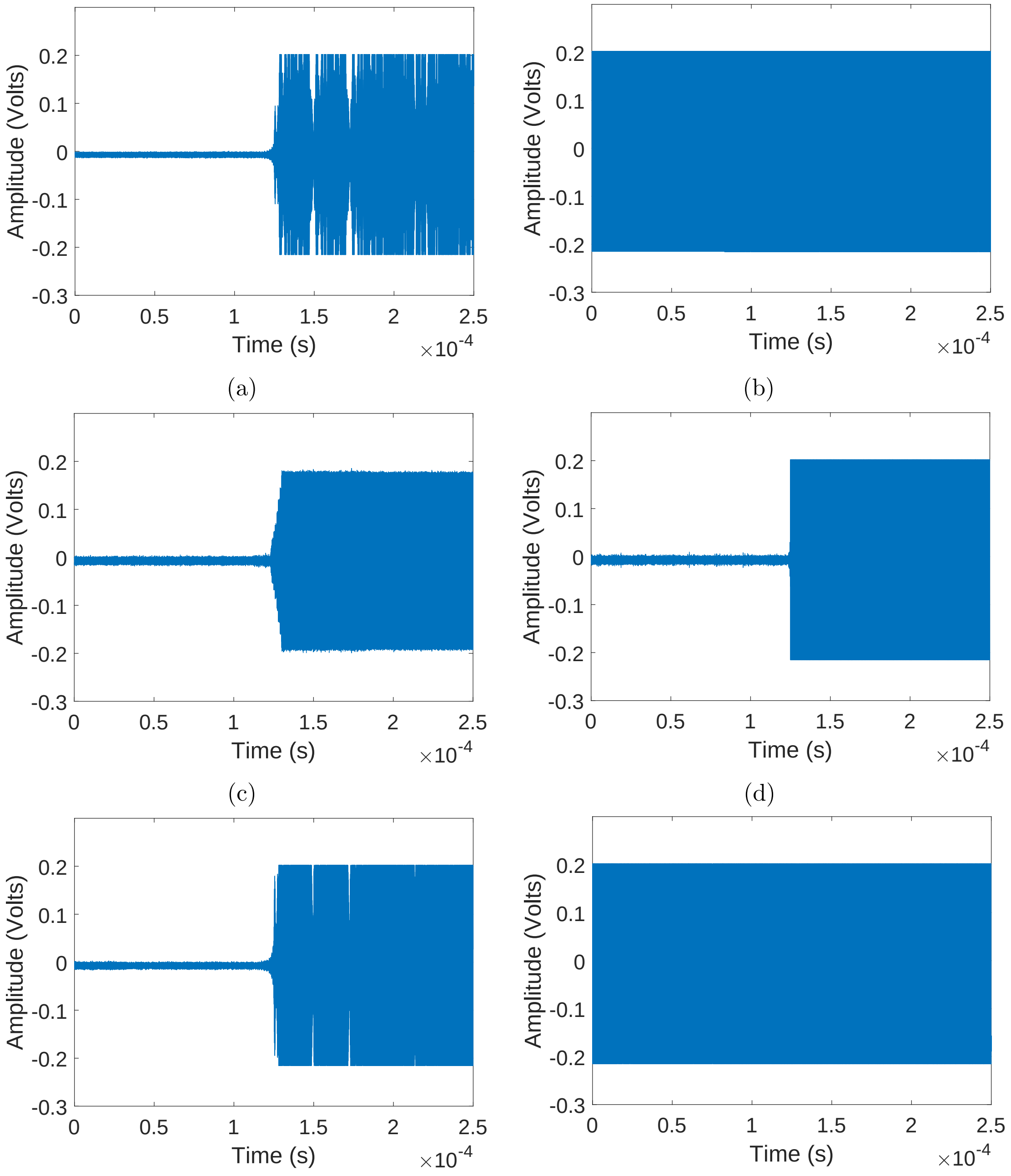


Figure 19: The RFSSCS setup for beyond-visual-line-of-sight capturing of signals from UAV.

3.4.5 Type of signal captured

Three categories of ISM devices (UAS, Bluetooth, and WiFI) are used in this experiment, yielding three signal types. These are the UAS, Bluetooth, and WiFI signals. However, the UAS is made up of both the UAV and UAV flight controllers. The signals from these two components are collected, so UAS signals are divided into UAV signals and UAV controller signals.

3.5 Data Description

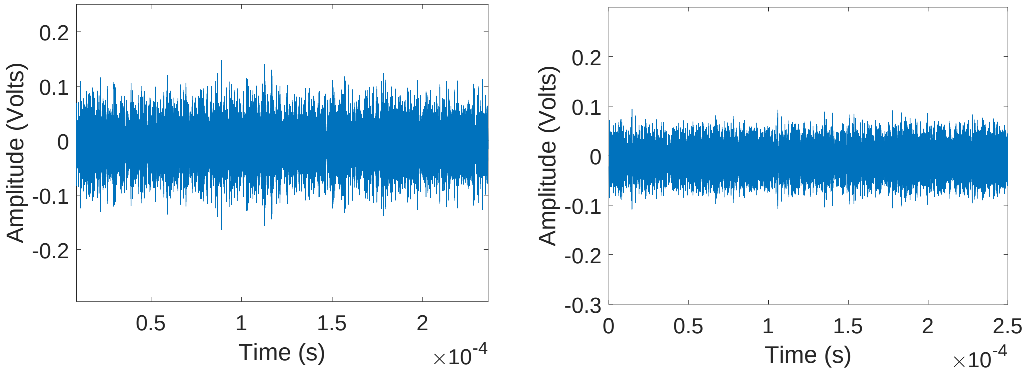


(a) (b) (c) (d) (e) (f)

Figure 20: Captured RF signal from: (a) DJI Matrice 600 controller, (b) DJI Matrice 600 UAV (c) Beebeerun controller, (d) Beebeerun UAV (e) DJI Inspire controller (f) DJI Inspire UAV.

Some examples of the signal captured at VLOS for UAVs and UAV controllers are shown in Fig. 20. One important observation from the data collection is that no visible transient region is seen in some of the captured DJI UAV signals. For instance, Fig. 20 (a) and Fig. 20(b) show examples of signal captured from DJI Matrice 600 controller and DJI Matrice 600 UAV, respectively. The transient state is present in the controller signal and absent in the UAV signal. Conversely, the transient region is visibly present in both the Beebeerun's controller and UAV signal. This presents another supporting reason for the use of a steady state for signature extraction.

In addition, Fig. 21 shows the captured signals from DJI Matrice 600 UAV and DJI Inspire UAV at BVLOS. Due to the channel effect (i.e., building or other obstructions in the environment), the signal's amplitude decreased significantly.



(a) (b)

Figure 21: Captured RF signal at non-line-of-signal from: (a) DJI Matrice 600 UAV (b) DJI Inspire UAV.

3.6 Data Labeling

The data labeling is done by a manual process using a directory and sub-directories. Fig. 22 shows the directory tree for the labeling of the RF dataset. The root directory is the name of the dataset repository, which is called CardRF. Because signals are captured at LOS and NLOS, two sub-directories are created for LOS and NLOS. Four categories of signals (i.e., UAV, UAV controller, Bluetooth, and WiFi) are collected for the LOS data collection. So for each category of signals, a sub-directory is created under the LOS directory to store the signals. Similarly, only UAV signals are captured at NLOS. So the NLOS directory only contains the UAV sub-directory.

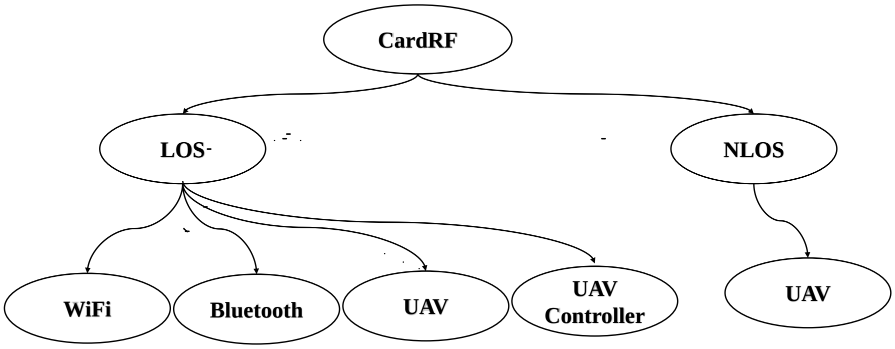


Figure 22: Directory tree for data label of the Cardinal RF dataset.

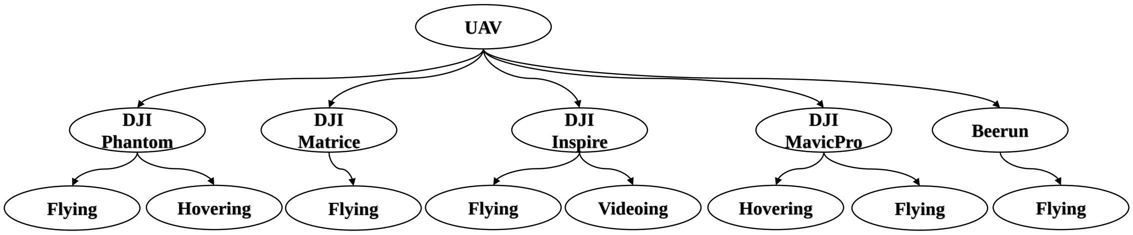


Figure 23: Directory tree for the LOS UAV signal labels showing the UAV models and UAV flight modes as sub-directories and sub-directories of sub-directories, respectively.

For WiFi directory contains the sub-directories for the two WiFi devices (i.e., Cisco Linksys E3200 and TP-link TL-WR940N). The Bluetooth directory contains sub-directories for the five Bluetooth devices listed in Table 5. More so, the UAV controller directory comprises six sub-directories for the UAV controllers listed in 5. The UAV directory takes a different form because signals from UAVs are collected under different flight modes for a given UAV. Fig. 23 depicts the UAV directory under LOS where the UAV models and UAV flight modes are sub-directories and sub-directories of sub-directories, respectively. Similarly, only three UAVs are utilized for UAV signals captured under NLOS, and the flight mode is ”Flying”. Sub-directories under the UAV directory at NLOS are shown in Fig. 24.

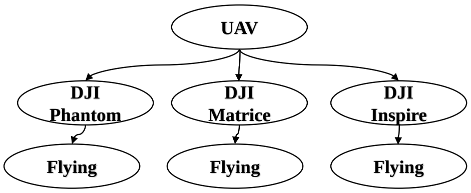


Figure 24: Directory tree for the NLOS UAV signal labels showing the UAV models and UAV flight modes as sub-directories and sub-directories of sub-directories, respectively.

3.7 Data Format

Each signal captured is stored as a file in *".MAT"* format. The *".MAT"* format is a binary data container in MATLAB. The signals are stored as a structured datatype in the binary data container.

3.8 Data Size and Usage

TABLE 6

Summary of the number of signals captured at LOS for each device in CardRF dataset.

Device	Model	Total captured signal	Train set (70%)	Test set (30%)
UAV	DJI Phantom 4	500	350	150
	DJI Inspire	500	350	150
	DJI Matrice 600	500	350	150
	DJI MavicPro 1	500	350	150
	Beebeerun FPV RC drone	500	350	150
UAV Controller	DJI Phantom 4	500	350	150
	DJI Inspire	500	350	150
	DJI Matrice 600	500	350	150
	DJI MavicPro 1	500	350	150
	Beebeerun FPV RC drone	500	350	150
	3DR Iris	500	350	150
Bluetooth	Apple iPhone 6S	500	350	150
	Apple iPhone 7	500	350	150
	Apple iPad 3	500	350	150
	FitBit Charge 3	500	350	150
	Motorola E5 Cruise 3	350	245	105
WiFI	Cisco Linksys E3200	500	350	150
	TP-link TL-WR940N	500	350	150

The total size of the CardRF dataset in memory is 66.4 GB. The LOS data is split into train and test sets in the ratio of 70% to 30%. The splitting process is done by randomly selecting the signals under each category. Table 6 gives the summary of the CardRF dataset for signals captured at LOS. For each device in the catalog, 500 signals are collected except for a Bluetooth device (i.e., Motorola E5 Cruise), where 350 signals are captured. For UAV signals captured from DJI Phantom, DJI Inspire, and DJI MavicPro, where signals are acquired under two flight modes, 250 signals are captured for each flight mode. Similarly, for the NLOS, 300 signals are captured for each device.

3.9 Conclusion

In this chapter, the experimental design and set-up for the acquisition of the Cardinal RF dataset is discussed. This dataset is acquired in an outdoor experiment by using five Bluetooth devices (mobile phones and smartwatch), two WiFi device (i.e., WiFi routers) and six UAVs. The total size of the CardRF dataset in storage memory is 66.4GB. One of the merits of this dataset is the collection of signals from Bluetooth devices, UAV and UAV controller. The CardRF dataset is used for the experiment and analysis in subsequent chapters.

CHAPTER 4

WAVELET TRANSFORM ANALYTICS FOR UAV DETECTION AND IDENTIFICATION

4.1 Introduction

This chapter introduced four new methods for extracting features or unique signatures from RF signals for UAV detection and identification using wavelet transform analytics (i.e., continuous wavelet transform and wavelet scattering transform). These methods can be extended to other RF device classifications. The key aspects of this chapter are summarized as follows:

1. To exploit the RF signals emanating or propagating from UAV flight controllers to detect and identify UAV under wireless interference (i.e., WiFi and Bluetooth).
2. To examine and compare the importance of extracting RF fingerprints from the transient and steady-state of the RF signals for detection and identification of UAVs using both wavelet analytics and machine learning.
3. To utilize coefficient-based and image-based signatures from continuous wavelet transforms and wavelet scattering transforms as features for training machine learning algorithms and convolutional neural networks.
4. Finally, to evaluate the implication of linear dimensionality reduction (i.e., PCA) on the coefficient-based features and the performance of trained models under varying signal-to-noise ratio.

4.2 Dataset Description

For this study, a subset of the CardRF dataset was used. RF signals from two Bluetooth devices (a smartphone and smart wristwatch), two WiFi routers, and six UAV controllers (four DJI UAVs, one BeeBeerun UAV, and one 3DR UAV) are utilized. A total of 3000 RF signals (i.e., 300 RF signals from each device) are collected from the ten devices, which take 20.1 GB of storage. For the training purposes, 200 RF signals from each device are used for the training set and the remaining 100 RF signals are used for the test set. This implies that a total of 2000 RF signals are used for training purposes, and 1000 RF signals are utilized for testing.

4.3 Preprocessing RF signal

A statistical changepoint detection algorithm [89] that determines a changepoint based on the standard deviation is used to find abrupt changes in a captured signal for the essence of detecting the starting point of a signal. This prevents the noisy part of the raw signal from corrupting the extraction of RF fingerprints. A single level Haar wavelet decomposition (HWD) is adopted for preprocessing based on two reasons:

1. To remove high frequency components of the captured signal;
2. To improve the computational efficiency of extracting features;

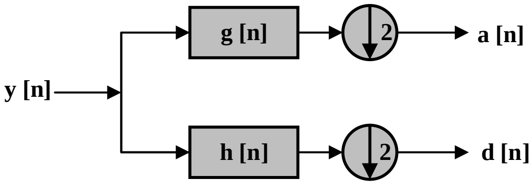


Figure 25: Single level Haar wavelet decomposition for raw RF signal prepossessing. The raw captured signal from RF devices is denoted as $y[n]$. The low pass and high pass filter are $g[n]$ and $h[n]$ respectively. The approximate and detail coefficients are denoted as $a[n]$ and $d[n]$. Approximate coefficient, $a[n]$, is used for feature extraction.

A signal level HWD is a multi-resolution function with two parallel-connected filters which are low pass filter ($g[n]$) and high pass filter ($h[n]$), respectively, that resolved a signal into low and high-frequency components. This is shown in Fig. 25. The captured signal $y[n]$ is passed into the parallel filters and down-sampled. The down-sampled outputs from the low pass filter, $a[n]$, are called the approximation coefficients, and they represent the low-frequency components of the raw signal,$y[n]$. Conversely, the down-sampled outputs from the high pass filter, $d[n]$, are called the detail coefficients, which are the high-frequency components of the signal. The transient and steady states of the approximate coefficients $a[n]$ are acquired for feature extraction.

4.4 Feature Extraction

The transient and steady states of the approximate coefficients, $a[n]$, are acquired for the extraction of features. Four different methods based on wavelet transform analytics are proposed for the extraction of features from RF signals. Fig. 26 illustrates the categories of the extraction methods. These methods can be categorized into continuous wavelet transform (CWT) and wavelet scattering transform (WST). Both coefficient-based and image-based signatures are extracted using each of the wavelet transform (i.e., CWT and WST).

4.4.1 Feature Extraction based on Continuous Wavelet Transform

The continuous wavelet transform (CWT) is defined as in (5), which is the convolution of a signal and a mother wavelet. The CWT returns resultant convolutional coefficients, which is a two-dimensional matrix. The matrix rows represent the scaling, and the column size is equal to the length of the signal. The wavelet coefficients of a signal $x(t)$ can serve as the signature for the signal. Morlet, Mexican hat, Gaussian, frequency B-Spline, harmonics, and Shannon wavelets are examples of base wavelets commonly used for CWT [69, 71]. In CWT, both the scale parameter, s, and translation parameter are continuously varied and this introduces redundant information that may not be of value to the specific application [71].

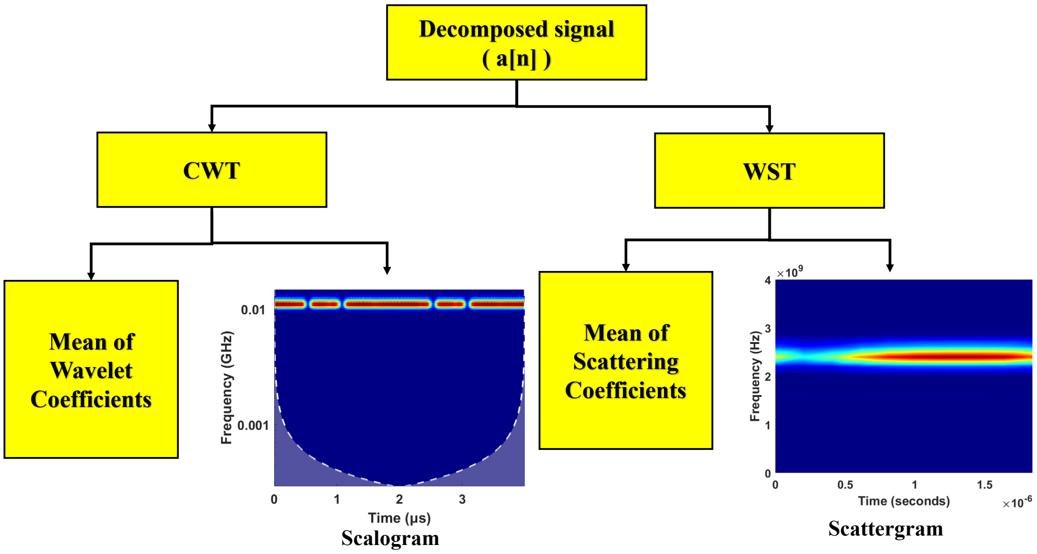


Figure 26: Divisions of the four proposed feature extraction methods based on CWT and WST. The decomposed signal, $a[n]$, is the approximation coefficients from the single HWD.

4.4.1.1 CWT coefficient-based feature

The CWT of signals is done in MATLAB [89]. It returns a 2-D matrix, and each element in the matrix is a complex number. In order to exploit these coefficients as features, the complex magnitude of each element in the matrix is computed. The row-wise mean of the magnitudes is calculated and transposed as the feature set. A set of 114 features is acquired as the signature for each signal, and this feature set is used to train ML algorithms.

4.4.1.2 CWT Image based feature

Similar to STFT, where a spectrogram represents the frequency spectrum and the energy distribution in the time-frequency domain, CWT gives the signal distribution in the signal in the time-scale domain. The wavelet spectrogram, also known as scalogram, is the squared modulus of the CWT or a plot of the energy density $E(s, \tau)$ [70].

$$E(s, \tau) = |w(s, \tau)|^2. \tag{12}$$

By using CWT, the scalogram of a signal is generated as the signature, which is then used to train a convolutional neural network (CNN)-based algorithm (SqueezeNet). Fig. 27 and

Fig. 28 show the examples of some scalograms generated from a UAV controller (i.e., DJI Inspire), WiFi and Bluetooth devices with respect to the transient and steady states of the RF signal.

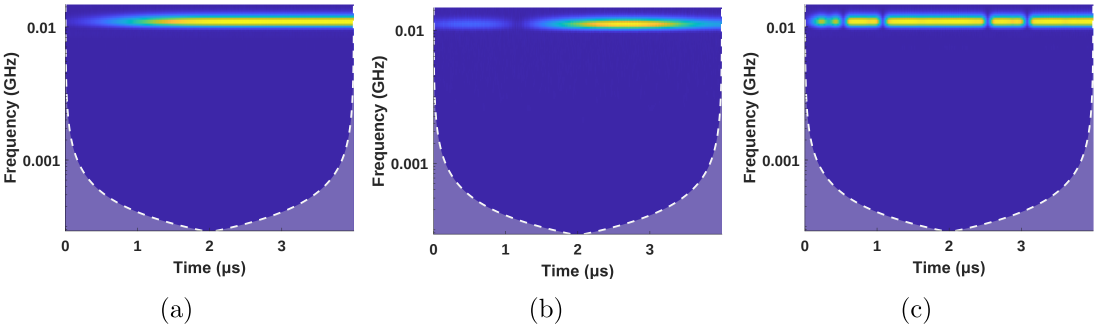


Figure 27: Examples of scalogram extracted from the transient state of the captured RF signals: (a) IPhone 6S, (b) DJI Inspire, and (c) TPLink (WiFi device).

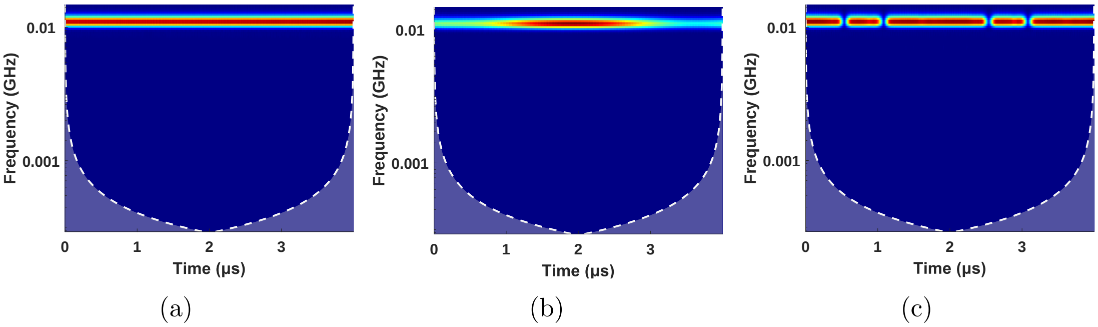


Figure 28: Examples of scalograms extracted from the steady state of the captured RF signals: (a) IPhone 6S, (b) DJI Inspire, and (c) TPLink (WiFi device).

4.4.2 Feature Extraction based on Wavelet Scattering Transform

Another variant of the wavelet transform is wavelet scattering transform (WST). WST is an improved time-frequency analytical technique for signals which are based on the wavelet transform. It allows the derivation of low-variance features from signals which a classifier can use to discriminate signals [90–93]. The three major factors that makes WST advantageous over other wavelet transform are [90]:

- invariance or insensitivity to translation of signals,

- stability to signal deformation,
- discriminative signals (i.e., informative feature representation).

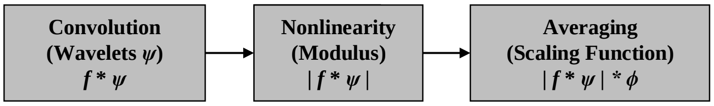


Figure 29: Wavelet scattering transform process flow with the three successive steps in transforming a signal.

WST exploits averaging and modulus operations to offer an improved time-frequency domain analysis. The moving average of the signal enhances invariance and stability. Similarly, the modulus of a wavelet transform is stable and discriminative [90]. The WST framework works like the CNN architecture for feature extraction, but the computational process does not involve the learning of parameters [90]. WST involves three successive processes to decompose a signal as illustrated in Fig. 29. These processes are the convolution of a wavelet function with the signal, modulus operation of the wavelet transformed signal, which is a non-linearity process, and low pass filtering averaging using a scaling function.

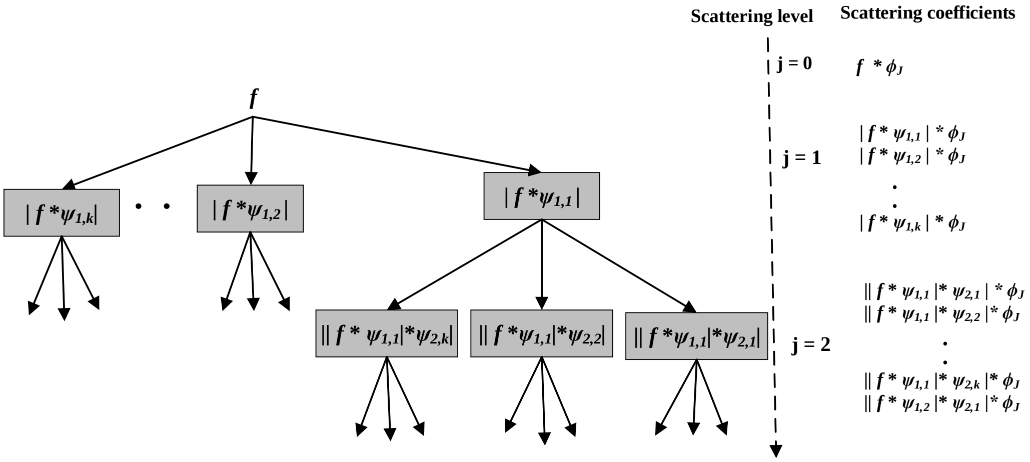


Figure 30: Tree of wavelet scattering transform algorithm for signal decomposition with mathematical expressions for computing the scalogram and scattering coefficients at each scattering levels.

The wavelet function in the wavelet scattering transform is a dilated mother wavelet

ψ with a scaling factor 2^{-j}, where j varies from 1 to J (i.e., maximum scattering level) as follows:

$$\psi_{j,k}(u) = 2^{-2j}\psi(2^{-j}u). \tag{13}$$

Fig. 30 shows the tree algorithm for the WST framework. The tree node contains the scalogram coefficients, while the scattering coefficients are the convolution of the scalogram coefficient and the scaling function. The first step in WST is to get the translation invariance coefficients by averaging the input signal f with a low pass filter (scaling function) ϕ_J. These translation invariance coefficients are the zeroth-order scattering coefficients, and it is denoted as S_0:

$$S_0 = f * \phi_J. \tag{14}$$

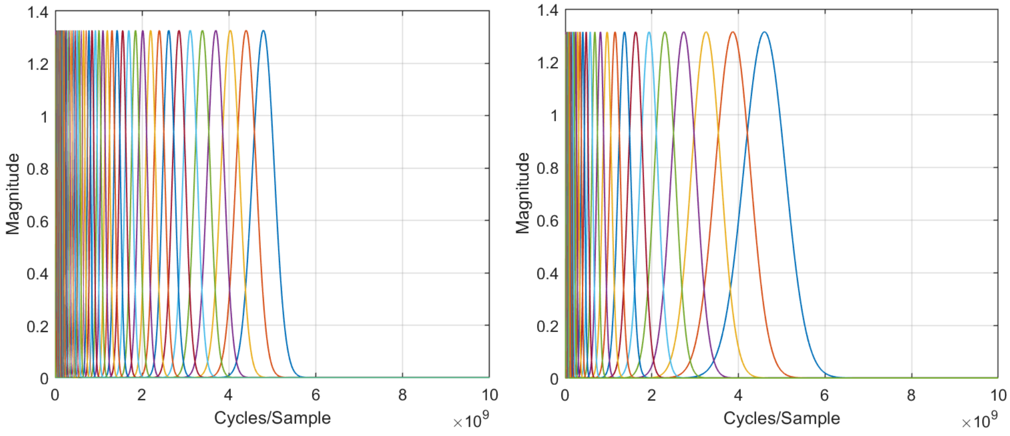
The subsequent steps involve the convolution of the input signal with each wavelet filter k in the first filter bank, followed by the non-linearity operation. This is accomplished by taking the modulus of each of the filtered outputs. This results in the nodes at the first level (i.e., $j = 1$), which are the scalogram for the first level, U_1, as:

$$U_1 = |f * \psi_{1,k}|. \tag{15}$$

The averaging of each modulus with the scaling function will yield the first-order scattering coefficients, S_1, as follows:

$$S_1 = |f * \psi_{1,k}| * \phi_J. \tag{16}$$

This iterative process is carried out on the next jth level using each wavelet filter k in the jth filter bank. The WST framework in MATLAB R2020b [89] is adopted for this work, and it uses the Gabor (analytic Morlet) wavelet. It has been observed in the literature that the Gabor wavelets give an optimal resolution in both time and frequency domains which optimally extract local features in signals or images [91, 94]. Energy dissipation occurs at every scattering level, and energy dissipates with an increase in the number of levels, so two scattering levels are appropriate to decompose a signal. This is because experimental results in literature have shown that the third level scattering coefficients can have energy below one percent [91]. Hence, a two-wavelet filter bank is proposed in this work for feature extraction from signals. Fig. 31(a) and Fig. 31(b) show the Gabor wavelets in the first and the second filter bank, respectively, which are used to compute the scattering coefficients at each level. The quality factors for the first and the second filter bank are eight and four, respectively.



(a) (b)

Figure 31: Filter bank for the WST framework: (a) First filter bank, and (b) Second filter bank.

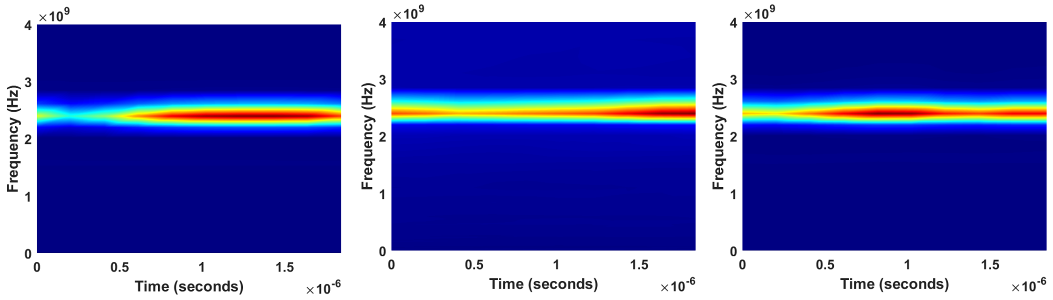
4.4.2.1 Wavelet Scattering coefficient-based feature

Decomposing the signal using the WST framework produces a matrix consisting of scattering coefficients. A row represents the scattering coefficients at a certain level in the tree. A statistical summary is done row-wise by estimating the average values and transposing the average as the feature set to represent each row as a feature. This resolved to a total of 1376 features set for each signal. This feature set is used to train the ML algorithm for classification purposes.

4.4.2.2 Wavelet Scattering Image-based feature

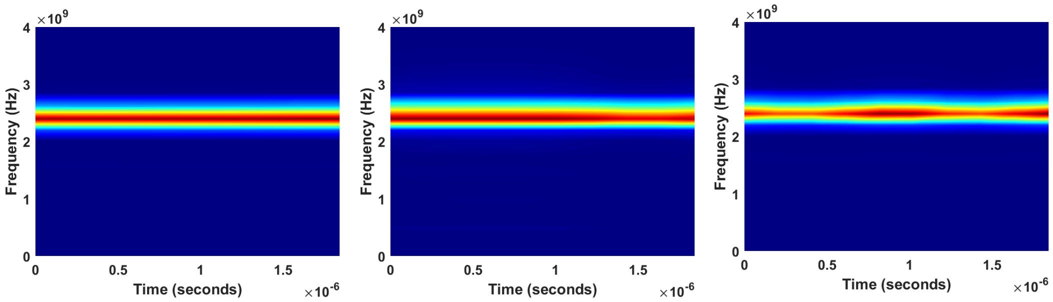
Similar to CWT, a scattergram shows the visual representation of the energy distribution for scattering coefficients that can be generated. Here, the scattergram of the scattering coefficients from the first filter bank is extracted as a signature for a signal.

Fig. 32 and Fig. 33 depict the examples of some scattergrams generated from a UAV controller (i.e., DJI Mavic Pro 1), WiFi and Bluetooth devices based on the state of the RF signal used for feature extraction.



(a) (b) (c)

Figure 32: Examples of scattergrams extracted from the transient state the captured RF signals: (a) IPhone 6S, (b) DJI Mavic Pro 1, and (c) TPLink (WiFi device).



(a) (b) (c)

Figure 33: Examples of scattergrams extracted from the steady state of the captured RF signals: (a) IPhone 6S, (b) DJI Mavic Pro 1, and (c) TPLink (WiFi device).

4.5 UAV Classification Algorithm

ML algorithms are widely used for pattern recognition. In this work, classical ML algorithms such as kNN, SVM, and Ensemble are used as classifiers for UAV detection or classification. The coefficient-based feature sets (wavelet and scattering coefficients) are used to train the ML algorithms. Similarly, the concept of transfer learning is introduced to UAV detection and identification by using a pretrained-CNN model called SqueezeNet.

The ML modeling aspect of this work follows the flowchart in Fig 34. The training data (RF signal collected) is first decomposed using a single level HWD as described earlier, and the approximation coefficient $a[n]$ is utilized for the extraction of the signal's RF signature or feature sets. The extracted features are then used to train an ML algorithm. Similarly, the raw test data is decomposed to compute the approximation coefficients. The feature set is extracted and passed into a trained model for signal type classification.

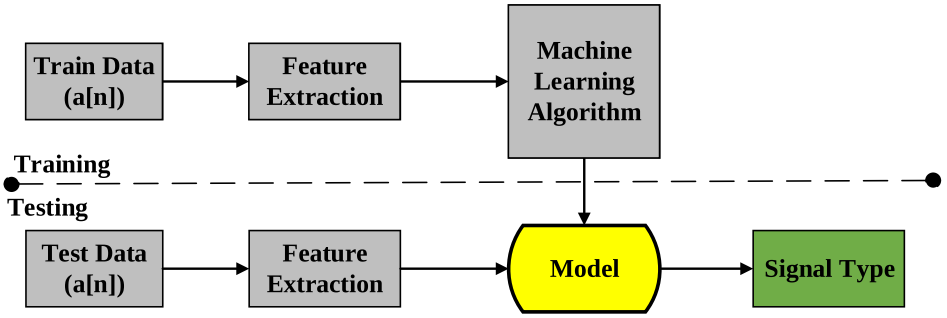


Figure 34: A flowchart of the machine learning model.

4.5.1 Transfer Learning and SqueezeNet

Transfer learning is a learning framework that involves the transferring of knowledge gained in one domain to another domain. The overarching goal of transfer learning is to improve learning performance in a new domain when a base knowledge has been established from another domain. According to authors in [1], domain and task are the two keywords that help in explaining the formal unified definition of transfer learning.

A domain D is defined by two parameters (i.e., $D = \{\Upsilon, P(X)\}$) which are the feature space Υ and the marginal probability distribution $P(X)$ where $X = \{x_1, x_2, ..., x_n\} \in \Upsilon$ and x_i is a feature in X. Similarly, task T is also defined by two parameters (i.e., $T = \{Y, f(.)\}$) which are the label space Y and the model function $f(.)$. The model function $f(.)$ is a mathematical or rule-based predictive function derived from a training set $\{x_i, y_i\}$, where $x_i \in X$ and $y_i \in Y$. Given x_i, the model function $f(.)$ predicts the corresponding label y_i [1]. Hence, the formal definition of transfer learning given as [1]:

"*Given a source domain D_S and learning task T_S, a target domain D_T and learning task T_T, transfer learning aims to help improve the learning of the target predictive function*

$F_T(.)$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$ or $T_S \neq T_T$".

Transfer learning can be divided into three categories which are [1]:

1. Inductive transfer learning: In this approach, the target task is different from the source task. However, the source and target domains can either be same or different (but related).
2. Transductive transfer learning: The source and target domains are different but related. However, the source and target tasks are the same.
3. Unsupervised transfer learning: It is similar to inductive learning. However, it deals with unlabeled data (i.e,. the source and target domains does not have labeled data).

The differences between these learning approaches are shown in Table 7.

TABLE 7: Comparison of the three forms of transfer learning approaches (adapted from [1]).

Type of approach	**Source and target domains relationship**	**Source and target tasks relationship**
Inductive transfer learning	same / different but related	different but related
Transductive transfer learning	different but related	the same
unsupervised transfer learning	same / different but related	different but related

Instead of training a model from scratch, the concept of transfer learning (i.e., inductive transfer learning) was adopted by using SqueezeNet because the network has learned diverse feature representations from over 1000 images (i.e., images such as animals, pencil, mouse and so on). SqueezeNet is a pre-trained model with a small CNN architecture designed to have fewer parameters without accuracy trade-off [95]. This implies that SqueezeNet offers a balance between model size and accuracy. The main objectives of SqueezeNet are i) to enhance distributed training, ii) to reduce the overhead of transferring large or heavy models, and iii) to enhance the portability of deploying models on edge devices or field-programmable gate arrays (FPGA). Three strategies are leveraged in the design of SqueezeNet, which are: [95]

1. By utilizing a lot of 1 × 1 filters in the CNN architecture.
2. Decreasing the number of input channels to 3 × 3 filters by using squeeze layers.

3. Delayed downsampling in the network to enable large activation maps (i.e., stride> 1).

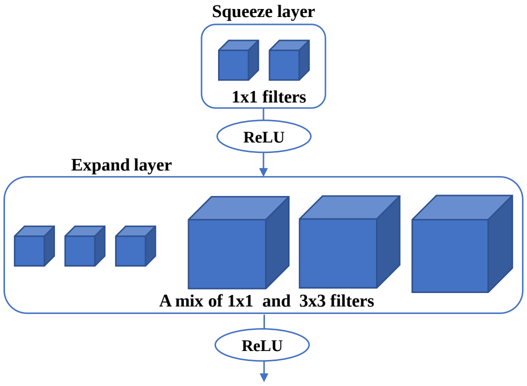


Figure 35: A typical example of a fire module where the squeeze layer has two 1 × 1 filters and the expand layer has a mix of three 1 × 1 and three 3 × 3 filters.

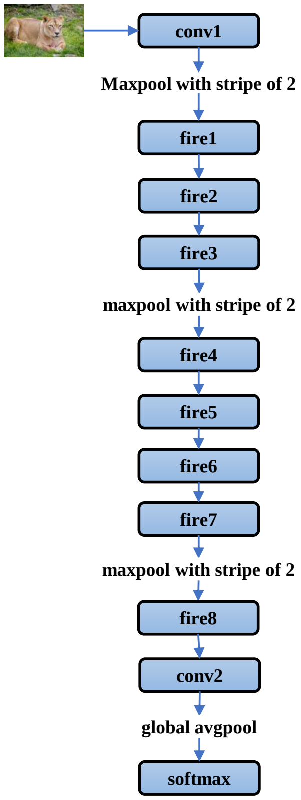


Figure 36: Squeeze architecture showing the connections of the two convolutional layers and eight fire modules.

Strategies 1 and 2 facilitate the reduction in the number of parameters. Similarly, strategy 3 ensures the maximization of accuracy. According to the authors in [95], a fire module was introduced as the new building block for the CNN architecture. As shown in Fig. 35, the fire module is made of a squeeze convolution layer that is fed into an expanded layer (i.e., a layer which comprises a mix of both 1×1 and 3×3 filters). Fig 36 shows the SqueezeNet architecture that consists of two normal or native convolution layers and eight fire modules. Max-pooling with a stride of 2 is done after conv1, fire3, and fire7 layer, respectively. At conv2 average pooling is performed.

4.6 Experimental Results and Discussions

The trained models are evaluated using the test set. The primary focus of the results is in seven directions which are:

1. the state of the RF signal used for feature extraction;
2. the behavior of the model when the classification is a three-class problem (i.e., classifying the signals to Bluetooth, WiFi, and UAV signal) and when the model is a ten class problem (i.e., identifying each device);
3. the use of coefficient based signatures over an image-based signatures for UAV classification;
4. the impact of linear dimensionality reduction on coefficient based signatures as related to model performance;
5. model performance under varying SNR;
6. computational time complexity of feature extraction methods and the classifier's inference time;
7. Lastly, the performance of the type of waveform transforms used for feature extraction.

In a classification problem, accuracy, precision, recall, and F_1-score are widely used as performance evaluation metrics. However, using only accuracy as a performance metric can be misleading, especially when the test data is skewed. Hence, accuracy, precision, recall, and F_1-score are utilized to assess the classifiers' performance. In addition to these metrics, a confusion matrix that provides a visual representation of a classifier performance through a contingency table is also provided for some models. It should be noted that the four metrics previously mentioned altogether give the summary of a confusion matrix. Below is the definition of the four metrics:

$$\text{Accuracy} = \frac{T_\mathrm{P} + T_\mathrm{N}}{T_\mathrm{P} + T_\mathrm{N} + F_\mathrm{P} + F_\mathrm{N}}, \tag{17}$$

$$\text{Precision} = \frac{T_\mathrm{P}}{T_\mathrm{P} + F_\mathrm{P}}, \tag{18}$$

$$\text{Recall} = \frac{T_P}{T_P + F_N}, \tag{19}$$

$$F_1 \text{ score} = 2\left(\frac{Precision \times Recall}{Precision + Recall}\right), \tag{20}$$

where T_P, T_N, F_P and F_N represent true positive, true negative, false positive and false negative, respectively.

4.6.1 Model Performance based on Classification Metrics

TABLE 8: Average classification accuracy (%) of the ML models and SqueezeNet based on the feature extraction methods and the state of RF signal use for fingerprinting at 30 dB SNR.

s/n	Algorithm	Continuous Wavelet Transform				Wavelet Scattering Transform			
		3 Classes		10 Classes		3 Classes		10 Classes	
		Transient	Steady	Transient	Steady	Transient	Steady	Transient	Steady
1	KNN	99.8	99.7	83.8	87.0	99.8	99.9	87.9	87.3
2	SVM	99.5	99.1	81.8	80.8	99.6	99.9	88.2	87.4
3	Ensemble	99.5	99.4	85.2	84.9	99.9	99.9	90.9	85.0
4	KNN + PCA	92.5	79.0	68.8	57.1	99.9	99.8	88.3	87.2
5	SVM + PCA	92.9	79.3	69.6	57.6	99.7	99.7	89.3	89.4
6	Ensemble + PCA	92.9	79.1	70.0	60.7	98.9	99.7	87.4	85.9
7	SqueezeNet	99.0	99.4	88.4	77.4	99.2	99.5	88.5	76.1

TABLE 9: Average classification recall (%) of the ML models and SqueezeNet based on the feature extraction methods and the state of RF signal use for fingerprinting at 30 dB SNR.

s/n	Algorithm	Continuous Wavelet Transform				Wavelet Scattering Transform			
		3 Classes		10 Classes		3 Classes		10 Classes	
		Transient	Steady	Transient	Steady	Transient	Steady	Transient	Steady
1	KNN	99.7	99.5	84.3	87.3	99.8	99.8	88.1	87.7
2	SVM	99.3	98.8	84.2	81.6	99.4	99.8	88.6	88.7
3	Ensemble	99.3	99.4	86.8	86.3	99.9	99.8	91.5	86.5
4	KNN + PCA	89.5	72.5	71.6	55.9	99.9	99.8	88.9	88.0
5	SVM + PCA	90.1	73.0	72.1	57.1	99.5	99.6	89.8	89.9
6	Ensemble + PCA	89.9	72.6	73.1	60.3	98.9	99.7	88.1	86.8
7	SqueezeNet	98.5	99.0	88.8	76.5	99.0	99.2	88.7	77.2

In Table 8, the average accuracy of the classifiers is shown for the RF state (i.e., transient and steady states) utilized for extracting features and the four proposed feature extraction methods at 30 dB SNR. Similarly, recall, precision, and F_1-score values for the classifiers are provided in Table 9, Table 10, and Table 11, respectively. For clarity in analyzing the performance of the models, classification accuracy is used as the key metric, and when necessary, the author refer to other metrics. For any given classifier, the four performance metrics are within the same range such that there is no situation where one metric gives a high value and others return a very low score.

From Table 8, The ML algorithms and SqueezeNet yield an average classification accuracy of over 99% when using the wavelet coefficients (CWT) of the signal, regardless of signal states, in determining the three types of device (i.e., UAV controller, Bluetooth, and WiFi). With an average classification accuracy of 99.8%, kNN outperforms other ML algorithms (such as SVM and Ensemble) and SqueezeNet. This demonstrates that either state of the signal (i.e., transient or steady-state) possesses sufficient information to identify the device type. Because the row-wise average of the CWT coefficients for each signal resulted in a 114 feature set, data overloading due to high dimensionality may limit a model's accuracy [46]. Thus, in an effort to exclude unnecessary features and prevent model overfitting, the author use PCA to reduce the dimensionality of the feature set using a limit of 95% explained variance. This resulted in a reduction of the feature set from 114 to one. By leveraging the transient state of the signal for feature extraction and reducing the feature space through PCA (i.e., one principal component), the average accuracy of the ML models drops to 92.5%, 92.9% and 92.9% for *k*NN, SVM, and Ensemble, respectively. Similarly, using the PCA on the CWT coefficients, a substantial reduction in accuracy was observed when extracting features from the signal's steady state. *k*NN, SVM, and Ensemble have an average accuracy of 79%, 79.3%, and 79.1%, respectively. While *k*NN outperforms other ML algorithms before PCA, it ended up with the lowest average accuracy after dimensionality reduction when exploiting either transient or steady state of the signal for feature extraction. Squeezenet, which employs an image-based signature (scalogram), achieves an accuracy of 99% for transients and 99.4% for steady states, respectively.

TABLE 10: Average classification precision (%) of the ML models and SqueezeNet based on the feature extraction methods and the state of RF signal use for fingerprinting at 30 dB SNR.

s/n	Algorithm	Continuous Wavelet Transform				Wavelet Scattering Transform			
		3 Classes		10 Classes		3 Classes		10 Classes	
		Transient	Steady	Transient	Steady	Transient	Steady	Transient	Steady
1	KNN	99.9	99.8	83.8	87	99.8	99.9	87.9	87.3
2	SVM	99.5	99.3	81.8	80.8	99.6	99.9	88.2	87.4
3	Ensemble	99.4	99.1	85.2	84.9	99.8	99.9	90.9	85.0
4	KNN + PCA	90.9	74.1	68.8	57.1	99.8	99.8	88.3	87.2
5	SVM + PCA	90.8	76.8	69.6	57.6	99.7	99.7	88.3	89.4
6	Ensemble + PCA	91.5	75.4	70.0	60.7	98.6	99.6	87.4	85.9
7	SqueezeNet	99.3	99.7	88.4	77.4	99.0	99.7	88.5	76.1

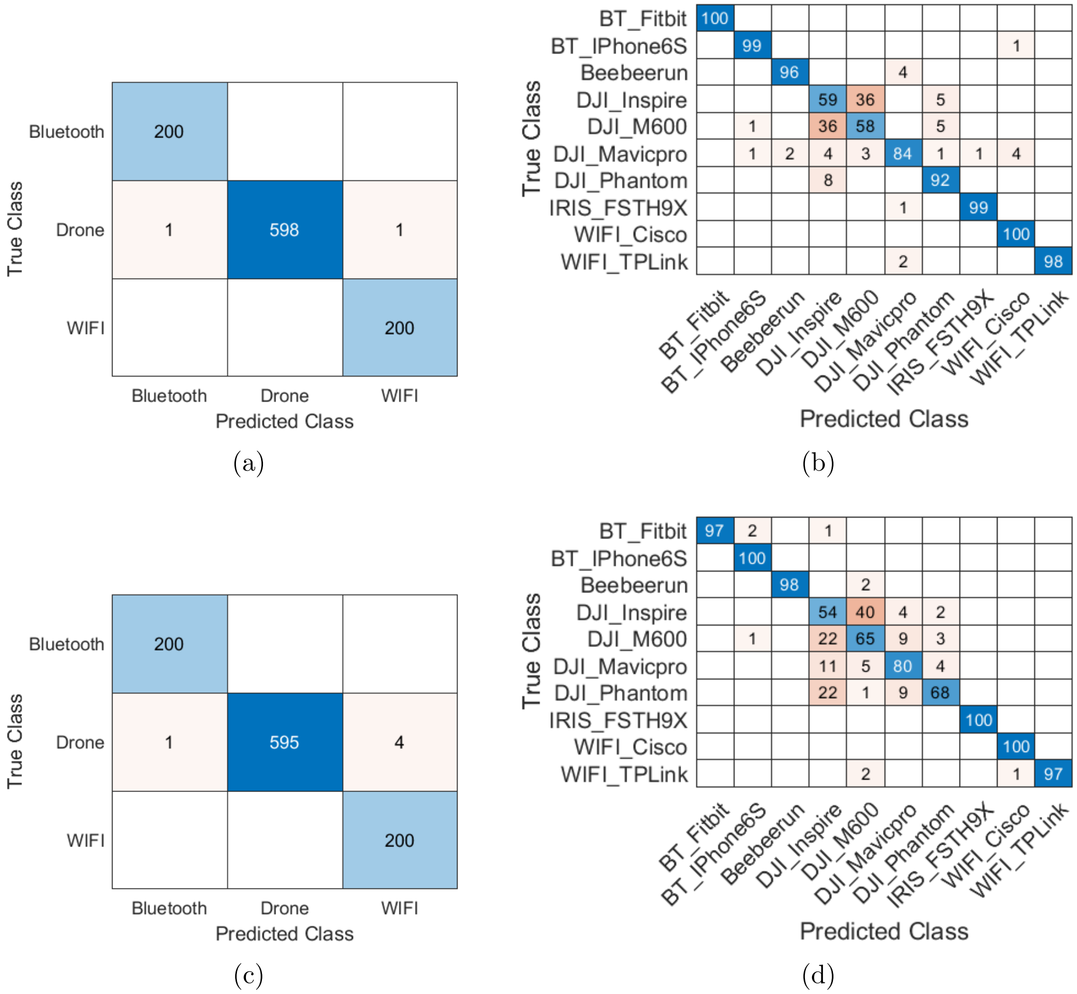
TABLE 11: Average F_1-score (%) of the ML models and SqueezeNet based on the feature extraction methods and the state of RF signal use for fingerprinting at 30 dB SNR.

s/n	Algorithm	Continuous Wavelet Transform				Wavelet Scattering Transform			
		3 Classes		10 Classes		3 Classes		10 Classes	
		Transient	Steady	Transient	Steady	Transient	Steady	Transient	Steady
1	KNN	99.8	99.7	83.6	86.9	99.8	99.9	88.0	87.4
2	SVM	99.4	99.0	82.5	81.1	99.5	99.9	88.4	87.8
3	Ensemble	99.3	99.3	85.5	85.3	99.9	99.9	91.0	85.3
4	KNN + PCA	90.1	72.7	68.3	55.3	99.9	99.8	88.5	87.3
5	SVM + PCA	90.5	74.4	69.7	55.7	99.6	99.7	89.5	89.6
6	Ensemble + PCA	90.5	73.6	69.8	59.5	98.8	99.7	87.7	86.2
7	SqueezeNet	98.9	99.3	88.5	76.3	99.0	99.4	88.6	76.0

When CWT is used to extract features from the steady-state of the signals in order to classify each device (i.e., the ten classes problem), the accuracies of the models are reduced when compared with using the transient state for feature extraction, except for kNN. kNN outperforms other ML algorithms with an accuracy of 87%. The average accuracy of kNN was significantly reduced to 57.1% when the feature space (i.e., CWT coefficients) was reduced using PCA. Regardless of the signal state or number of groups, the output of the ML models degrades significantly when PCA is applied to the CWT coefficients. In comparison, the SqueezeNet achieves an average accuracy of 88.4% for transients and 77.4% for steady-state.

Averaging each set of scattering coefficients produced from the signal transform yielded a 1376 feature set for the WST-based feature extraction process. The ML algorithm is trained using these feature sets. The author use PCA to minimize feature dimensionality with 95% explained variance, similar to CWT. The number of principal components (PCs) that account for 95% of the variance explained varies depending on the signal's state (i.e., transient or steady-state). When the transient state of the signal is transformed with WST and the feature sets are reduced with PCA, the feature set is reduced from 1376 to 131. On the other hand, by employing a steady-state for feature extraction, the feature set was reduced from 1376 to 71.

Comparing the CWT-based feature extraction method to the WST-based feature extraction method, the performance of classifiers is enhanced using the WST-based feature extraction. When extracting features from the transient state of the signal using WST, the accuracy of models trained with WST-based coefficients (i.e., scattergram coefficients) for the three-class problem (i.e., group-device classification) is 99.8%, 99.5%, and 99.9% for kNN, SVM, and Ensemble, respectively. When PCA is applied to the scattering coefficients of the transient signals, it has no detrimental effect on the ML classifiers as opposed to the CWT coefficients. kNN and SVM classification accuracies increased by 0.1% after the PCA reduction of the scattering coefficients when exploiting the transient of the signals. On the other hand, the ensemble classifier's average accuracy is reduced by 1.1%. Based on the average classification accuracy of the ML algorithms and SqueezeNet, the scattering coefficients of the steady-state outperform the CWT coefficients.



(a)

	Bluetooth	Drone	WIFI
Bluetooth	200		
Drone	1	598	1
WIFI			200

(b)

	BT_Fitbit	BT_IPhone6S	Beebeerun	DJI_Inspire	DJI_M600	DJI_Mavicpro	DJI_Phantom	IRIS_FSTH9X	WIFI_Cisco	WIFI_TPLink
BT_Fitbit	100									
BT_IPhone6S		99							1	
Beebeerun			96			4				
DJI_Inspire				59	36		5			
DJI_M600		1		36	58		5			
DJI_Mavicpro		1	2	4	3	84	1	1	4	
DJI_Phantom				8			92			
IRIS_FSTH9X						1		99		
WIFI_Cisco									100	
WIFI_TPLink						2				98

(c)

	Bluetooth	Drone	WIFI
Bluetooth	200		
Drone	1	595	4
WIFI			200

(d)

	BT_Fitbit	BT_IPhone6S	Beebeerun	DJI_Inspire	DJI_M600	DJI_Mavicpro	DJI_Phantom	IRIS_FSTH9X	WIFI_Cisco	WIFI_TPLink
BT_Fitbit	97	2		1						
BT_IPhone6S		100								
Beebeerun			98		2					
DJI_Inspire				54	40	4	2			
DJI_M600		1		22	65	9	3			
DJI_Mavicpro				11	5	80	4			
DJI_Phantom				22	1	9	68			
IRIS_FSTH9X								100		
WIFI_Cisco									100	
WIFI_TPLink					2				1	97

Figure 37: Confusion matrices: (a) Group-device classification using the transient state, CWT framework and kNN, (b) Specific-device classification using transient state, WST framework and SqueezeNet (c) Group-device classification using the steady-state, WST framework and SqueezeNet, and (d) Specific-device classification using transient state, WST framework, and Ensemble + PCA.

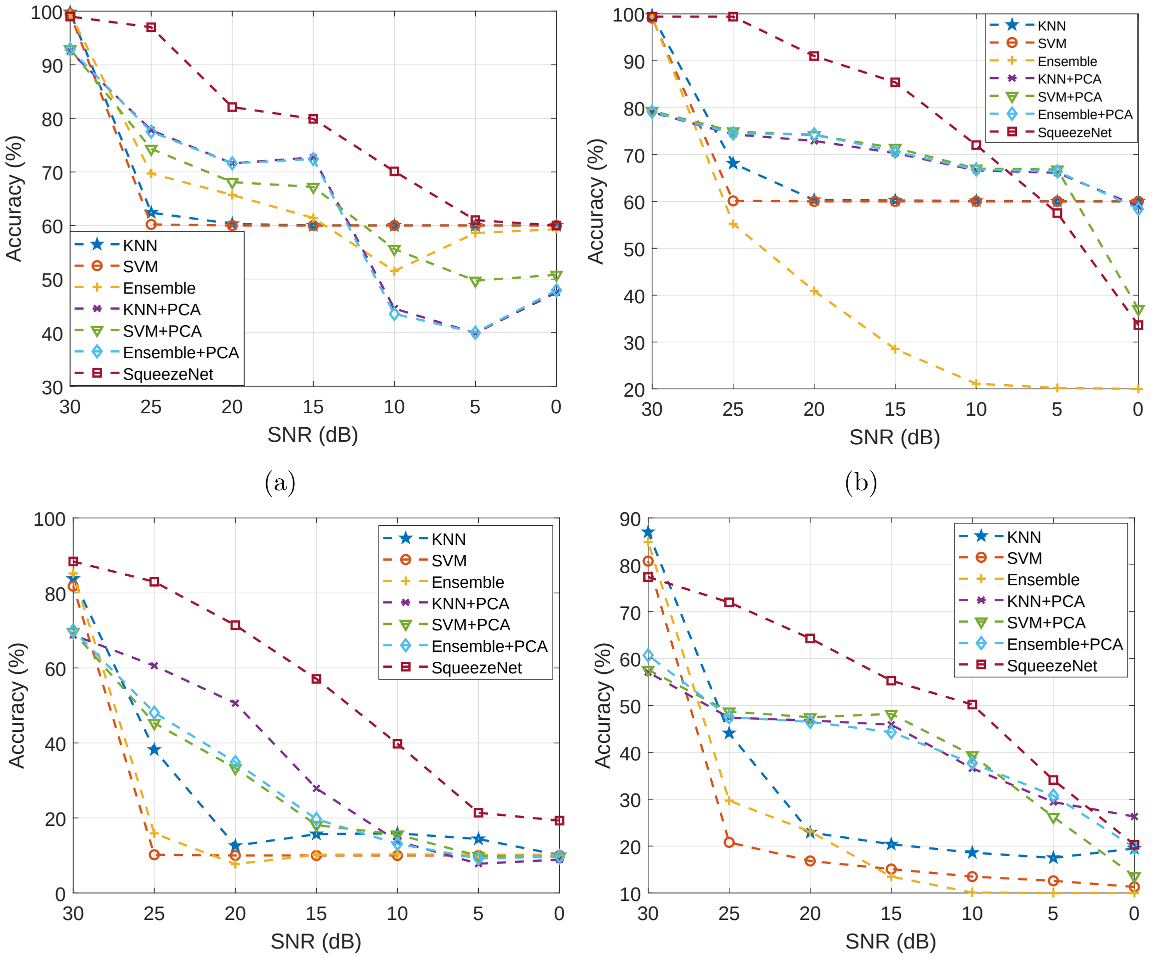
Furthermore, the performance of the ML algorithms and SqueezeNet for both transient and steady-state outperforms CWT-based features when using WST-based features for specific-devices classification (i.e., ten classes). When using the transient of the signals, for example, the average accuracy for$k$NN, SVM, Ensemble, and SqueezeNet is 88%, 88.4%, 91%, and 88.6%, respectively. Furthermore, when PCA is used to minimize the dimensionality of scattergram coefficients, there is no substantial negative effect on the efficiency of ML classifiers. The output of kNN and SVM classifiers, for example, improves after feature reduction to 88.5% and 89.5%, respectively, when using the transient state.

It was observed that the classification accuracy of group-device classifiers is higher than the device-specific classifiers. This is because the misclassification rate is higher among UAVs from the same manufacturer (i.e., the DJI UAVs). Fig. 37(b) and Fig. 37(d) show examples of two confusion matrices for device-specific classifiers (i.e., SqueezeNet classifier where the transient state and WST framework are used and Ensemble + PCA classifier where the transient state and WST framework are utilized). In Fig. 37(b), the SqueezeNet classifier misclassifies 36% of DJI Inspire signals as DJI Matrice 600 (i.e., DJI M600) and vice-versa. On the other hand, the Beebeerun and 3DR Iris FSTH9X, which are UAVs from different manufacturers, are 96% and 99% correctly classified, respectively. Similarly, in Fig. 37(d), it can be seen that the errors cluster around the DJI UAVs.

4.6.2 Performance at Difference SNR

The accuracy or other performance metrics of a classifier are insufficient in evaluating the proposed classifiers due to the nature of RF signals. The wireless channel affects the quality of the signal causing SNR variation. Although classifiers are trained with signals captured at 30 dB SNR, evaluating their performance at different SNR levels is critical. Additive white Gaussian noise (AWGN) applied to signals is a frequently used technique for varying the signal's SNR in a simulated environment. The behavior of the classifiers is investigated by varying the signal's SNR using AWGN. Fig. 38 and Fig. 39 show the behavior of our classifiers under varying SNR based on the feature extraction methods (i.e., CWT and WST), the state of signal used for the extraction of features, and the number of classes, respectively.

Fig. 38(a) shows the performance of the group-device classifiers when CWT is used for extracting features from the transient state of the RF signals. One important factor to note about the test data for group-device classifiers is that it is unbalanced data. That is, 60% of the test data includes UAV signals, while Bluetooth and WiFi signals constitute 20% each. In some cases, when the SNR is below 30 dB, and the average accuracy of a

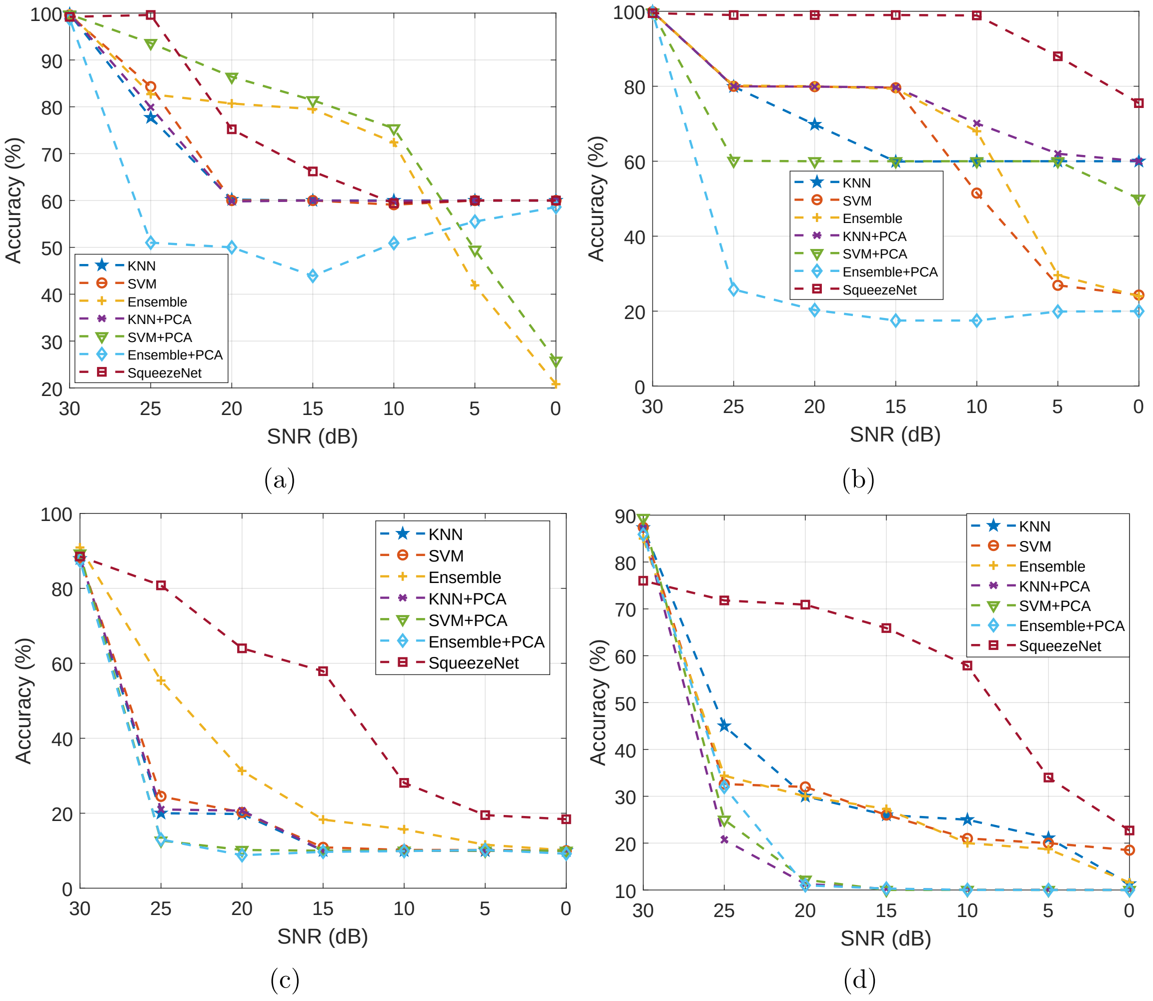


(a) (b) (c) (d)

Figure 38: SNR results: (a) Group-device classification using transient state and CWT framework, (b) Group-device classification using steady state and CWT framework, (c) Specific-device classification using transient state and CWT framework, and (d) Specific-device classification using steady state and CWT framework.

group-device classifier is 60%, the model classifies all signals as UAV controller signals. In this instance, the precision and recall of the model are 33.3% and 20%, respectively. For example, in Fig. 38(a), SVM gives an accuracy of 60% from 25 dB to 0 dB. In this instance, any signal that goes through the model is classified as a UAV controller signal. Despite that, the accuracy of the model is 99.5% at the 30 dB SNR. When this case occurs, the author call it a random classification. All the classifiers with coefficient based signatures are equal to or below the threshold of 60% accuracy from an SNR 5 dB and below, as shown in Fig. 38(a).

On the other hand, the behavior of the classifiers when utilizing the steady-state of RF signals through CWT for feature extraction in the context of group-device classification under varying SNR is shown in Fig. 38(b). Here, ensemble + PCA, kNN + PCA, and SVM + PCA perform non-random classification between 30 dB and 5 dB, with a gradual decline in accuracy as the SNR decreases. Similarly, Squeezenet, which employs scalograms, joins the progression of non-random classification at 10 dB, and its performance degrades as the SNR level decreases. However, SqueezeNet outperforms other classifiers at 30 dB to 10 dB SNR as shown in Fig. 38(a) and Fig. 38(b).



(a) (b)

(c) (d)

Figure 39: SNR results: (a) Group-device classification using transient state and WST framework, (b) Group-device classification using steady state and WST framework, (c) Specific-device classification using transient state and WST framework, and (d) Specific-device classification using steady state and WST framework.

Fig. 38(c) shows the behavior of the specific-device classifiers under varied SNR when CWT is used for signature extraction from the transient state of the signals. In terms of responsiveness to changing the SNR, Squeezenet outperforms other ML algorithms. From 0 dB SNR and above, SqueezeNet has higher accuracy compared to other ML algorithms. Similar performance is observed for SqueezeNet from 25 dB to 5 dB when using CWT on the steady-state signal for specific-device classifiers shown in Fig. 38(d) Also, in Fig. 38(c) and Fig. 38(d), as the SNR decreases, the accuracies of the ML algorithms and SqueezeNet decrease.

Fig. 39(b) shows the performance of the group-device classifiers where WST is used for extracting features from the steady-state RF signals. SqueezeNet has an accuracy of more than 60% at an SNR of 0 dB, and this continues to improve as the SNR increases. The accuracy is as high as 98.9% at 10 dB. This outperforms the case when WST is used on the transient state of the signal or CWT for feature extraction as shown in Fig. 38(a) and Fig. 38(b) for group device classification.

Fig. 39(c) and Fig. 39(d) depict a device-specific classifier where WST is used for feature extraction on the transient and steady-state signals, respectively. SqueezeNet also outperforms the ML algorithms from 25 dB to 0 dB.

4.6.3 Computational Complexity and Actual Time Cost

It is not sufficient to have a classifier with high accuracy and resilient SNR variation. The speed of responsiveness of the classifier is also important. For this reason, the computational running time for each feature extraction method and the inference time for each classifier is considered and assessed. Fig. 40 shows the average running time for extracting each category of features proposed in this work. As opposed to coefficient-based features, extracting image-based features (scalogram and scattergram) from RF signals takes longer. This is shown in Fig. 40 that the two image-based features (i.e., scalogram and scattergram) have the highest time cost. The extraction of scalograms from the CWT framework takes the longest. Extracting scattergrams has the second-highest running time, preceded by the

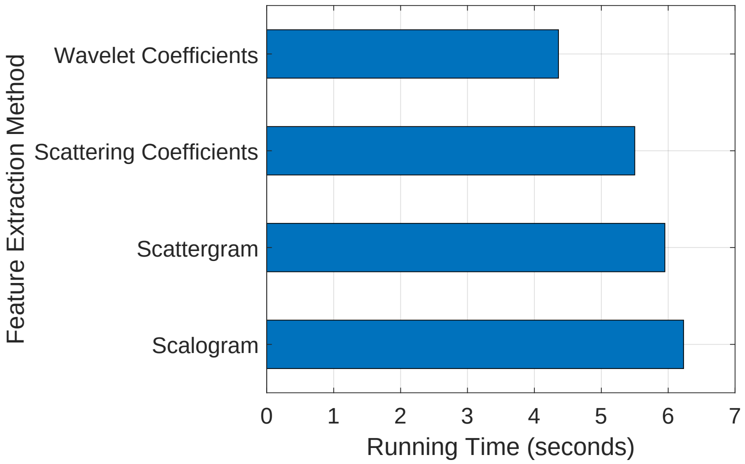


Figure 40: Running time for the feature extraction methods.

scattering coefficient-based method. On the other hand, the wavelet coefficients from the CWT framework have the lowest running time.

Depending on the type of feature used by a classifier, the end-to-end computational time is the sum of run time for extracting features and inference time from the point a signal is captured to inference. Table 12 shows the inference time of the classifiers. From Table 12, the average inference time of the scattering coefficient-based (i.e., using the coefficient of WST as input features) classifiers is higher than the classifiers that take in CWT coefficients as input features. For instance, average inference time for kNN, SVM, and Ensemble classifiers under group-device classification using the transient state (i.e., three classes where CWT coefficients are the input feature) are 108 milliseconds, 55.3 milliseconds, and 56.8 milliseconds, respectively. On the other hand, using scattering coefficient-based features from the transient state, the average inference time for kNN, SVM, and Ensemble classifiers under group-device classification are 822 milliseconds, 155 milliseconds, and 129 milliseconds, respectively. This is primarily due to the higher dimensionality of the scattering coefficient-based features (i.e., 1376 feature set) as compared to wavelet coefficient-based features (i.e., 114 feature set).

Furthermore, the average inference time of SqueezeNet when using scalograms is higher than when using scattergrams. For example, when the scalogram from the transient state of an RF signal is used for group-device classification, the inference time is 200 milliseconds. On the other hand, with scattergram, it takes 150 milliseconds.

Overall, exploiting wavelet coefficients for an RF-based UAV classification has a lower time cost over scattering coefficients. Also, exploiting scalograms has a higher time cost over scattergram.

TABLE 12: Computational time cost for classifier's inference given in seconds.

s/n	Algorithm	Continuous Wavelet Transform				Wavelet Scattering Transform			
		3 Classes		10 Classes		3 Classes		10 Classes	
		Transient	Steady	Transient	Steady	Transient	Steady	Transient	Steady
1	kNN	0.108	0.08	0.0873	0.075	0.822	0.771	0.726	0.81
2	SVM	0.0553	0.0243	0.109	0.083	0.155	0.117	0.423	0.405
3	Ensemble	0.0568	0.357	0.379	0.094	0.129	0.101	1.48	0.108
4	kNN + PCA	0.061	0.107	0.0277	0.221	0.168	0.097	0.188	0.097
5	SVM + PCA	0.081	0.0263	0.11	0.061	0.167	0.075	1.027	0.082
6	Ensemble + PCA	0.187	0.7324	0.046	0.063	0.778	0.124	1.18	0.289
7	SqueezeNet	0.2	0.18	0.192	0.19	0.15	0.151	0.16	0.159

4.7 Conclusion

This chapter introduces wavelet transform analytics (i.e., continuous wavelet transform and wavelet scattering transform) for extracting both coefficients and image-based signatures from RF signals to detect and identify UAV in an environment. Four feature extraction methods are proposed and analyzed by exploiting either the transient or steady-state of RF signals. The performance of the four methods is compared and contrasted based on the ability of the classifiers (i.e., ML algorithm and SqueezeNet) to learn the discriminative knowledge from the extracted features. The results show that using an image-based signature to train a pre-trained CNN-based model (SqueezeNet) outperforms coefficient-based classifiers where classical ML algorithms are utilized in terms of classification accuracy and robustness to variation in SNR. More so, both the transient and steady-state RF signals have unique properties that can be used to identify or classify UAVs. When the efficiency of the classifiers is evaluated under AWGN conditions, I discovered that using wavelet transform analytics for the extraction of RF fingerprints on the steady-state RF signals can handle SNR variation better than using transient states.

CHAPTER 5

SEMI-SUPERVISED LEARNING FRAMEWORK FOR UAV DETECTION

5.1 Introduction

Several research works in literature have exploited the supervised learning approach for UAV detection and identification with little or no attention to unsupervised or semi-supervised learning frameworks. In this chapter, I introduce the concept of semi-supervised learning and anomaly detection for UAV detection. UAV detection is modeled as an anomaly detection problem using a unary classification approach.

5.2 Anomaly Detection

Novelties, anomalies, or outliers are used interchangeably to describe anomalous samples in literature. According to Hawkins' definition of an outlier, an outlier is *"an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism"* [96]. Novelty detection is an approach of classifying a data point as a normal data point (inlier) when there is a relationship with a reference data and anomalous data point when there is no relationship with the reference data [97]. When there is a relationship between a data point and a reference data, it is assumed that the data point and the reference data are having the same generative distribution/process. If there is no relationship between a data point and a reference data, it is assumed that they have a different generative distribution/process. From an ML perspective, anomaly detection can be formulated as a supervised, semi-supervised, or unsupervised learning problem.

In supervised learning, the training data consist of labeled data that are normal and anomalous samples. Classical supervised ML algorithms (i.e., SVM, KNN, DNN, etc.) can be used to learn the discriminating properties in the data. The critical setback for using supervised learning for outlier detection is the problem of imbalanced data. Generally, outliers are less common than the normal data points in a dataset [98].

In semi-supervised learning, the training set contains only the normal (or only the anomalous) samples [99], so an ML algorithm can be trained to learn the underlying generative distribution properties or parameters of the normal samples. When a sample outside of the normal distribution is tested on the trained ML model, it classifies the sample as anomalous. Because the normal samples is a labeled class, this can be considered as one class problem. When a novelty detection is constrained as a one-class problem (i.e., unary classification), then it is referred to as a semi-supervised problem [99]. A one-class problem is when the learning algorithm is trained with only normal data (i.e., without the anomaly samples). A common ML algorithm for semi-supervised learning is a one-class SVM [100].

Unsupervised learning involves detecting anomalous samples from unlabeled data. Here, there is no training or testing set. It is assumed that the unlabeled data consist of both normal and anomalous samples. An ML algorithm is tasked to group or separate the normal samples from the anomalous samples [99]. The fundamental concept of this approach is to score each data point in the dataset based on the properties of the dataset.

5.3 Problem Definition using a Semi-supervised Approach

The ubiquity of RF signals such as Bluetooth and WiFi cannot be denied because of their wide applications in different domains. In this case, Bluetooth and WiFi signals are used as the commonplace signals to formulate this problem. Given that WiFi signals w and Bluetooth signals b are the RF signals S that are allowed to propagate in the environment of infrastructure under surveillance (IuS), but UAV controller signals u are restricted from the environment (i.e., $w, b \in S$ and $u \notin S$), then detecting u in the IuS can be considered a novelty detection problem if members of S are viewed as normal signals. By using an ML algorithm to model just the generative distribution or properties of members S such that the model can detect any deviation (i.e., non-members of S) can be treated as a one-class problem.

5.4 Dataset Description

For this study, a subset of the CardRF dataset was used. RF signals from two Bluetooth devices (a smartphone and smart wristwatch), two WiFi routers, and six UAV controllers (four DJI UAVs, one Beebeerun UAV, and one 3DR UAV) are utilized. A total of 3000 RF signals (i.e., 300 RF signals from each device) are collected from the ten devices, which consume 20.1 GB of storage.

5.5 Feature Extraction Using Wavelet Packet Transform

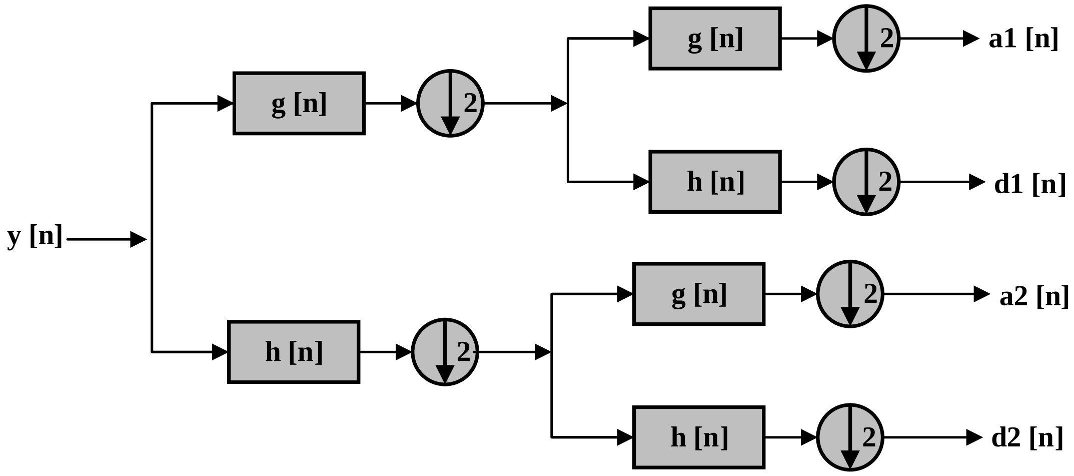


Figure 41: A two-level wavelet packet decomposition tree used to sub-band captured RF signals into packets for feature extraction.

A statistical changepoint detection, which is based on standard deviation, is used to detect the beginning of a transient state of RF signals. The transient state of the RF signals is captured and decomposed using a two-level wavelet packet transform. Fig. 41 shows the configuration of the two-level wavelet packet decomposition. The decomposition produces four packets, and each of the packets is statistically summarized using the eleven parameters in Table 13. The statistical summary of the packets gives a total of 44 features.

To speed up the learning process and avoid overfitting the model, a ranking algorithm based on statistical variance is adopted to pick the top four features from the 44 feature sets. The second-moment statistics of the packets are ranked. These four features are used to train an ML algorithm.

TABLE 13

Statistical features definition from wavelet packet transform.

Features	Formula	Measures
Mean (μ)	$\frac{1}{N}\sum_{i=1}^{N} x_i$	Central tendency
Absolute mean ($\bar{x}$)	$\frac{1}{N}\sum_{i=1}^{N} \lvert x_i \rvert$	Central tendency
Standard deviation(σ_T)	$\left[\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})^2\right]^{\frac{1}{2}}$	Dispersion
Skewness (γ)	$\frac{\sum_{i=1}^{N}(x_i-\bar{x})^3}{(N-1)\sigma_T^3}$	Asymmetry/shape descriptor
Variance (σ)	$\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2$	Dispersion
Kurtosis (k)	$\frac{\sum_{i=1}^{N}(x_i-\bar{x})^4}{(N-1)\sigma_T^4}$	Tail/shape descriptor
Root mean square (x_{rms})	$\left[\frac{1}{N}\sum_{i=1}^{N} x_i^2\right]^{\frac{1}{2}}$	Magnitude/Average power
Peak value (x_{pv})	$\max(\mathrm{x_i})$	Amplitude
Absolute peak value ($\bar{x}_{\mathrm{pv}}$)	$\max(\lvert\mathrm{x_i}\rvert)$	Amplitude
Peak to peak (x_{ppv})	$\max(\mathrm{x_i})-\min(\mathrm{x_i})$	Waveform amplitude
Entropy (H)	$-\sum_{i=1}^{N} x_i \log_2 x_i$	Uncertainty

5.6 UAV Detection Algorithm

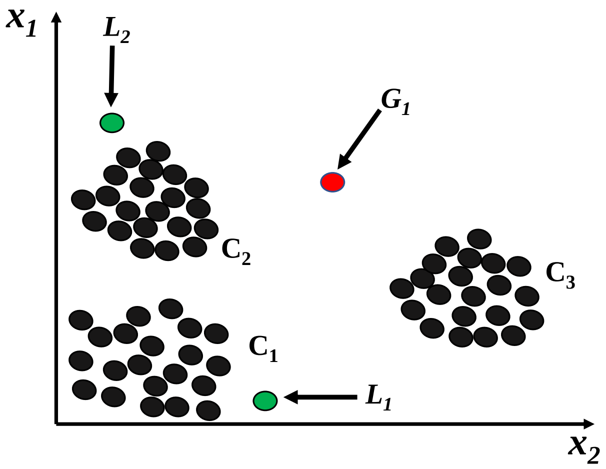


Figure 42: Illustration of local and global outliers in a dataset with three clusters. The red datapoint is a global outlier and the green data points are the local outliers.

Various algorithms have been adopted for anomaly detection. These include one-class support vector machines, local outlier factor, and so on. Generally, an outlier can be categorized as local or global. A sample is said to be a global outlier in a dataset when it is located or found at a far distance or outside of the entirety of the dataset. On the other hand, a local outlier is within the entirety of the dataset but at a distance from the surrounding data points or cluster [98].

Fig. 42 shows the scatter plot of a two dimensional dataset with three clusters (i.e., C_1, C_2 and C_3) and outliers (i.e., L_1, L_2 and G_1). L_1 and L_2 are local outliers to cluster C_1, and C_2, respectively. Contrarily, G_1 is a global outlier to the three clusters. By using a global outlier detection algorithm for a dataset similar to the one depicted in Fig. 42, detecting L_1 and L_2 as outliers might be difficult or even missed. However, a local outlier detection algorithm will be able to detect L_1, L_2 and G_1, respectively. In essence, using a global approach for outlier detection is insufficient when a dataset has clusters with varying densities [101]. The local outlier factor is the best-known approach for detecting local outliers [98].

5.6.1 Local Outlier Factor

Local outlier factor (LOF) is a density-based approach to determining an outlier. LOF uses distance measurement and nearest neighbor principles to estimate local density. LOF is an approach of assigning each data point in a dataset a degree of being an outlier by measuring the local distance from its surrounding neighborhood [101].

Assume o, p, q are data points in a dataset D (i.e., $o, p, q \in D$) and the distance between two data points (i.e., p and o) is denoted as $d(p, o)$. The formal definition of LOF according to [101] is defined as follows:

Definition 1: k-distance of data point p.

The k-distance of data point p can be denoted as $k\text{-}distance(p)$. Given k is any positive integer, $k\text{-}distance(p)$ is the distance between p and the farthest neighbor o such that:

1. for at least k data points $o' \in D \setminus \{p\}$ it holds that $d(p, o') \leq d(p, o)$ and
2. for at most $k - 1$ data points $o' \in D \setminus \{p\}$ it holds that $d(p, o') < d(p, o)$.

Definition 2: k-distance neighborhood of data point p.

The k-distance neighborhood of p contains data points q (i.e., k-nearest neighbors) that have its distance from p to be less than the k-distance of data point p.

$$N_{k-distance(p)}(p) = \{q \in D \backslash \{p\} \mid d(p,q) \leq k\text{-}distance(p)\} \tag{21}$$

Definition 3: reachability distance of data point p with respect to datapoint o.

Given k is a positive integer, the reachability distance of data point p with respect to datapoint o is defined as

$$reach\text{-}dist_k(p,o) = max\{k\text{-}distance(o), d(p,o)\}. \tag{22}$$

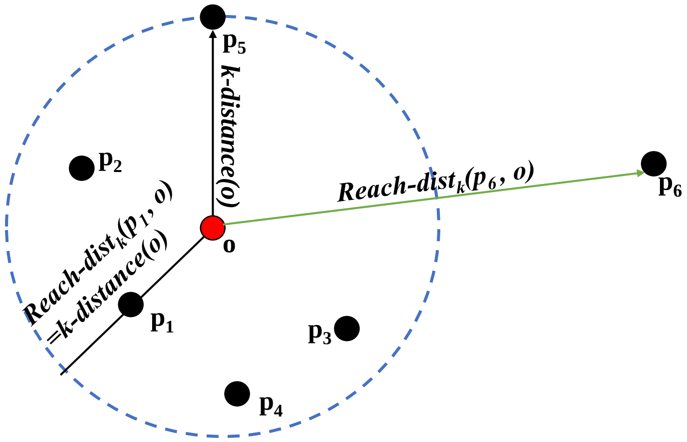


Figure 43: Illustration of the reachability distance for different data points p from point o assuming that k is equal to 5.

The concept of reachability distance given k is equal to five is illustrated in Fig. 43. If a data point (i.e., p_1, p_2, p_3, p_4 or p_5) is within the k-nearest neighbor of o then the reachability distance of each data point is the k-distance(o). Conversely, if a data point (i.e., p_6) is far away from o (i.e., not among the k nearest neighbor) the reachability distance is the actual distance, $d(p_6, o)$. The rationale for replacing the actual distance with k-distance(o) as the reachability distance for close data points to o is to reduce the statistical fluctuation of distance measurement (i.e., $d(p, o)$ where p is the ps close to o). This smoothing or normalizing effect is dependent on the value k. The higher the value of k the better the similarity of reachability distances for data points within the same neighborhood [101].

Definition 4: local reachability density of data point p

In defining outlier using density-based clustering, the two parameters that are functions of density are [101]:

1. $MinPts$ which specifies a minimum number of data point;
2. v which specifies a volume.

In order to calculate the density of a given neighborhood (i.e., p's neighborhood), the values of $reach\text{-}dist_{MinPts}(p, o)$, for $o \in N_{MinPts}(p)$ are used for volume measurement. Hence, the local reachability density of a data point p is defined as

$$lrd_{MinPts}(p) = 1/\left(\frac{\sum_{o \in N_{MinPts}(p)} reach\text{-}dist_{MinPts}(p, o)}{|N_{MinPts}(p)|}\right) \tag{23}$$

Definition 5: LOF of data point p: The LOF scores the degree of a data point being an outlier. LOF is the average ratio of the local reachability density of p and the data point in $MinPts$-nearest neighbors.

$$LOF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|} \tag{24}$$

When the LOF of an object or data point is approximately or less than 1 then the data point is not an outlier. However, if the LOF of a data point is greater than 1 then the data point is an outlier. In determining a data point p as an outlier using the LOF, the following conditions are used:

- $LOF(p) \approx 1$, it means p has a similar density as its neighbors (i.e., not an outlier),
- $LOF(p) < 1$, it means p has a higher density than its neighbors (i.e., not an outlier),
- $LOF(p) > 1$, it means p has a lower density than its neighbors (i.e., an outlier)

5.6.2 Training Process of Local Outlier Factor

As briefly described in Section 5.4 that 3000 RF signals from 10 devices are utilized, for the Bluetooth and WiFi devices, we selected 200 RF signals from each device as the training set and 100 RF signals as the evaluation set (i.e., test and validation). A total of 800 RF signals (non-UAV signals) are used to train the LOF algorithm. It is important to note that UAV control signals are not needed for training purposes (i.e., UAV control signals are not part of the training set) because the problem is formulated as a semi-supervised approach. The evaluation set consists of 400 non-UAV signals (i.e., Bluetooth and WiFi signals) and 1800 UAV control signals. The evaluation set is randomly split into the ratio of 70% to 30% for the test and validation set, respectively. To ensure the model is adequately trained, The author use the validation set to fine-tune the model (i.e., selecting the appropriate number of neighbors) and testing the model using the test set.

The analysis is performed on a Dell Inspiron 13 7000 Laptop running on Microsoft Windows 10 powered by an Intel(R) Core(TM) i7-8550U CPU at 1.80 GHz, 1992 MHz, 4 Core(s), 8 Logical Processors, and 16 GB RAM.

5.7 Experimental Results and Discussions

The evaluation is done during and after the training process of the LOF algorithm. The results are focused on three main points. These points are

- Selection of the number of neighbors to be used for LOF to classify signals;
- Performance metric of the detection algorithm;
- Model performance under varying SNR.

5.7.1 Selection of k (number of neighbors)

The Manhattan distance is used for computing the distance between two data points. A major hyperparameter of the LOF algorithm is the number of nearest neighbors (i.e., (k)) for estimating the local density of a data point. Determining the positive integer value of

TABLE 14: Average accuracy of LOF model based on the validation and test set when increasing the number of neighbors for local density estimation at 30 dB SNR.

Metric	number of neighbors																			
	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200
Validation Accuracy	43.9	95.1	95	94.7	94.1	94.7	95	96.2	96.5	**96.7**	96.5	96.5	96.4	96.5	96.5	96.4	96.2	95.8	95.8	95.8
Test Accuracy	45	95.4	95.8	96	95.4	95.3	95.6	96.8	96.6	**96.7**	96.7	96.2	96.5	96.5	96.7	96.4	97	96.8	96.8	96.8

k is done by using the validation set. The initial value of k is set to 10, and k is iteratively varied at an incremental step of 10. For every increase in k, the performance of the model is assessed using classification accuracy. Table 14 shows the accuracy of the LOF model as the value of k increases. At k equal to 100, the highest accuracy was obtained for the validation set, which stands at 96.7%. Based on this, the value of k for calculating the local density by the LOF model is selected as 100.

5.7.2 Model Performance based on Classification Metrics

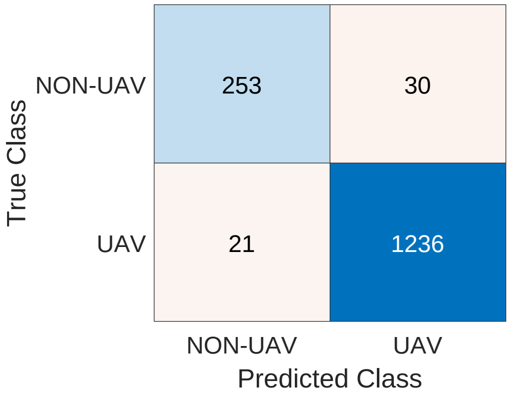


Figure 44: Confusion matrix of the LOF model ($k = 100$) when evaluating the model with the test set at 30 dB.

While the training of the algorithm is formulated as a one class-problem (i.e., training the model with only inliers), the evaluation can be done like a binary classification. This is because the detection algorithm can class an input as either an inlier (non-UAV) or outlier (UAV) signal. Accuracy, precision, recall, and F1-score are examples of metrics for evaluating the performance of a classifier.

By using the test set to evaluate the LOF model, accuracy of 96.7% is achieved at 30dB SNR. The confusion matrix of the model using the test set for evaluation is shown in

Fig. 44 at 30 dB SNR.

5.7.3 Performance at Difference SNR

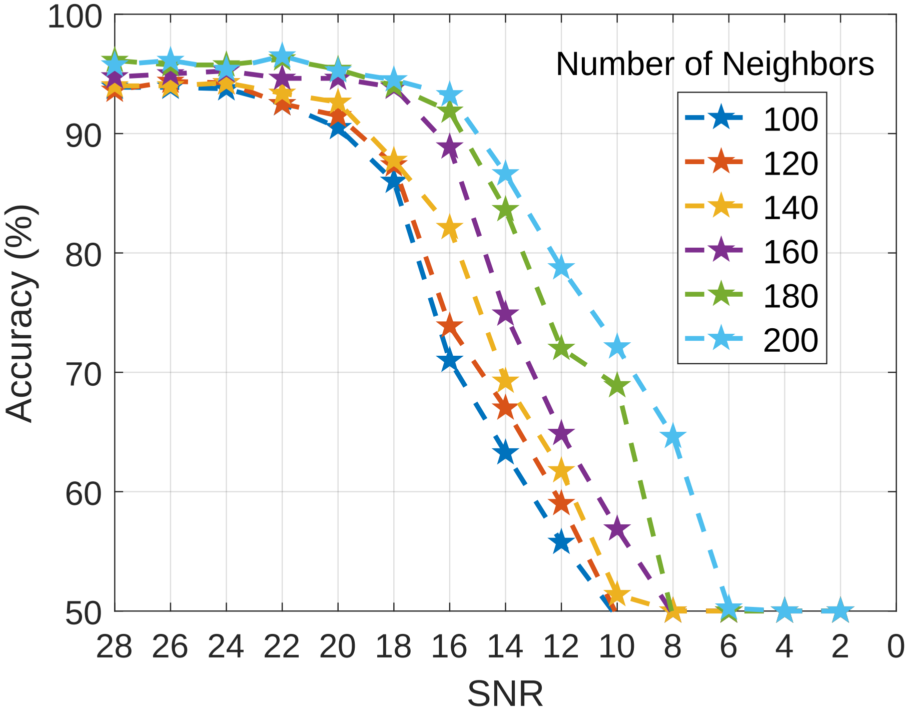


Figure 45: Classification accuracy under varying SNR and number of neighbors for the LOF model.

Because the model is trained using signals with high SNR, it is crucial to examine the model's performance when a signal (i.e., either a non-UAV or UAV signal) with low SNR is passed into the classifier. By degrading the SNR of the signals with Additive white Gaussian noise (AWGN), the underlying behavior of the classifier is examined. To avoid performance bias from using imbalanced data to evaluate a model, balanced test data (i.e., an equal number of UAV and non-UAV signals) is used to assess the underlying behavior of the classifier at varying SNR.

Fig. 45 shows the behavior of the classifier when reducing the SNR of signals and varying the value of k at step 20 starting from 100 to 200. The classification accuracy decreases as the quality of the signal decreases irrespective of the value of k. However, it was observed that the higher the value of k, the better the model performed at a lower SNR (i.e., from 10 dB to 28 dB). In Fig. 45, from 10 dB to 28 dB SNR, the accuracy of the classifier increased with an increase in k. For instance, at 12 dB SNR, when k is 100, 120, 140, 160, 180, and 120, the accuracy of the model is 55.75%, 59%, 61.8%, 64.9%, 72%, and 78.8%, respectively.

At 10 dB SNR, when k is equal to 100, the model's behavior can be viewed as a random process (i.e., accuracy is 50%) because any signal that goes through the model is classified as a UAV controller signal. On the other hand, at k equals 200, the model's accuracy is 72.1% at 10 dB SNR and did not make a random guess until the SNR is equal to or below 6 dB.

5.8 Conclusion

This chapter addresses the problem of UAV detection by introducing and adopting a semi-supervised anomaly detection approach using a LOF algorithm. It has been shown that by decomposing the transient state of RF signal using a two-level wavelet packet transform and obtaining the second-moment statistics (i.e., variance) of each packet as feature set (signal fingerprint) can be exploited for differentiating a UAV controller signal from other RF signals such as WiFi and Bluetooth. The feature set extracted from WiFi and Bluetooth signals is used to train a LOF algorithm to detect a UAV control signal as an anomalous signal. A classification accuracy of 96.7% is obtained at 30 dB SNR when detecting UAV control signals as anomalous signals. More so, the LOF performance increases with an increase in the number of neighbors used for determining the local density of a data point when classifying a signal with a lower SNR.

CHAPTER 6

HIERARCHICAL LEARNING FRAMEWORK FOR UAV DETECTION AND IDENTIFICATION

6.1 Introduction

This chapter introduces the use of an autoencoder (i.e., denoising autoencoder) for signal compression in UAV detection. The motivation for using a denoising autoencoder is because it is robust and resilient to corrupt or noisy data. Because of the robustness of the steady state of an RF signal to channel noise, the steady state of RF signals is exploited for RF signature extraction in this chapter. This is exploited for the UAV detection framework to achieve a high detection accuracy at lower SNRs (i.e., under high channel noise). Finally, the application of hierarchical learning for UAV classification is introduced which is a shift from the conventional flat classification approach.

6.2 Dataset Description

The CardRF is used for this work.

6.3 Stacked Denoising Autoencoder

One of the challenges of using RF sensing for UAV detection and identification is data explosion from sampling the signals. The effect is even more pronounced when performing spectral analysis on the captured signal [102]. According to [46], dimensionality reduction is a method that can be utilized to minimize the explosion of data. Hence, dimensionality reduction can be used to tackle the data explosion problem in a RF-based UAV detection and identification system.

Dimension reduction algorithms can be applied for signal compression to yield a latent representation of the signal. Principal component analysis (PCA) is one of the most frequently-used algorithms [46]. PCA is a linear transform that projects a set of variables (possibly correlated variables) into latent space representational variables which are mutually uncorrelated and ordered in variance with the expectation that the original information is retained [46, 103]. Another dimensionality reduction algorithm is an autoencoder which is an unsupervised deep neural network algorithm. Autoencoders are also called auto-associator neural networks in literature [104]. An autoencoder is used for data compression, decompression, and feature extraction. The main goal is to learn an efficient way to reconstruct its input data as the output under certain constraints, to ensure no direct copying or memorizing of the data in the network [105, 106].

An autoencoder has three parts which are the encoder, code layer, and decoder. The structure of a typical autoencoder is shown in Fig. 46 which has one hidden layer. The input to the hidden layer represents the encoder. Similarly, the hidden layer to the output layer represents the decoder. The hidden layer is the code layer. The encoder part compresses the input data into a latent representation that is embedded in the code layer. Conversely, the decoder decompresses the latent representation back to the original input as output. Let X be the input vector, let Y be the latent representation and let X' be the reconstructed X at the output. The key objective of the autoencoder is to learn a latent representation Y of X [105]. Given that the dimension of Y is d' and that X is d dimensional vector, if $d' > d$ then the Y representation is said to be an over-complete representation. On the contrary, if $d' < d$ then the Y representation is called an under-complete representation. Hence, the encoder can be expressed as [105]:

$$Y = f_\theta(X) = s(WX + b). \tag{25}$$

The parameters of the encoder θ is represent as $\{W, b\}$. where W is a $d' \times d$ weight matrix and b is the bias vector with d' dimension. Also, s is the activation. Similarly, the decoder is mathematically expressed as [105]:

$$X' = g_{\theta'}(Y) = s(W'Y + b'). \tag{26}$$

decoder parameters θ' is denoted as $\{W', b'\}$.

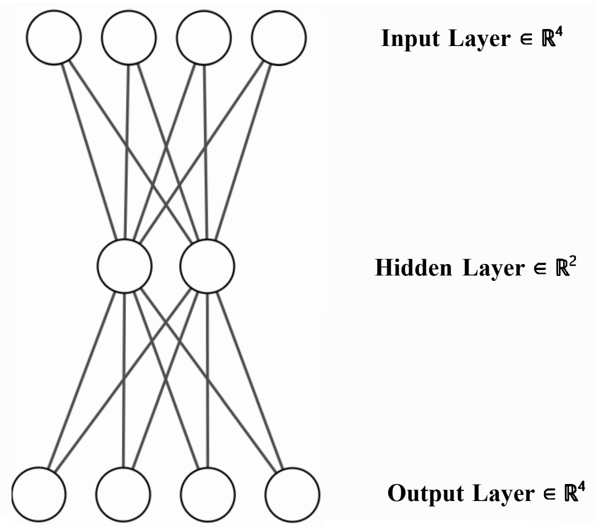


Figure 46: A neural network architecture of a typical autoencoder with one hidden layer and two nodes at the hidden layer. The hidden layer represents the code layer.

An autoencoder has the capability to project data in both linear and non-linear ways depending on the activation function used in each neuron of the network. Activation functions enable the nonlinearity attributes of autoencoder. Fig. 47 shows the graphical representation of commonly used activation functions. The definition of these activation functions are listed below:

1. ReLU: This is called rectified linear unit and it is mathematically defined as:

$$ReLu(x) = max(0, x) \tag{27}$$

   The key advantage of a ReLU is that it is simple and computationally efficient. However, dead ends can occur in ReLU. For instance, when the weights are greater than zero but the value of x is less than zero.

2. Leaky ReLU: It is a variant of ReLU with an improvement to reduce the dead ReLU problem. It is mathematically defined as:

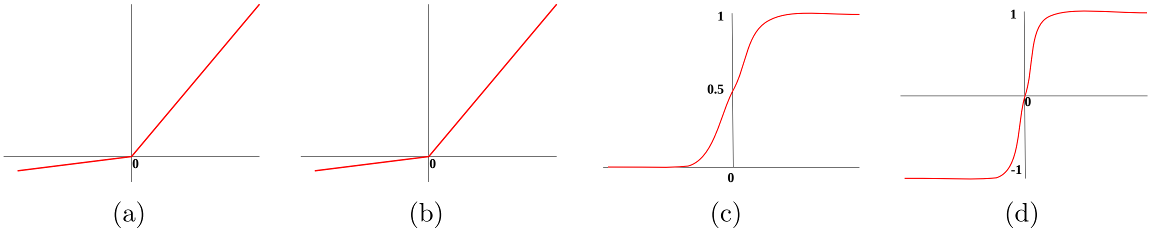
$$f(x) = max(\alpha x, x) \tag{28}$$

3. Sigmoid: It is applied as a logistic function. It is not centered on zero and it gives an output value between 0 and 1. It is mathematically defined as:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{29}$$

4. Tanh: It is called hyperbolic tangent function. It is centered on zero and it ranges from -1 to 1. It is computationally expensive.

$$Tanh(x) = \frac{2}{1+e^{-2x}} - 1 \tag{30}$$



(a) (b) (c) (d)

Figure 47: Common examples of activation functions: (a) ReLU, (b) Leaky ReLU (c) Sigmoid, (d) Tanh.

The training of an autoencoder involves minimizing the reconstruction error L, also known as the loss function. A good reconstruction of X by the autoencoder implies that the latent representation Y at the code layer retains enough information of the X. If the reconstruction of X by the autoencoder is X' then the loss function can be expressed as:

$$[L(X, X')], \tag{31}$$

Typical examples of a loss function include: mean absolute error, mean squared error and cross entropy. Cross entropy is preferred loss function when dealing with binary numbers. The choice of loss function is determine if X is a real or binary number. To ensure optimal parameters during the training process, optimization algorithms (i.e., optimizers) are usually adopted to minimize the loss function as:

$$argmin_{\theta,\theta'}[L(X, X')], \tag{32}$$

where θ represents the weights W and biases b of the encoder and θ' denotes the weights W' and biases b' of the decoder.

Examples of optimizers are stochastic gradient descent (SGD) [107], adaptive subgradient (AdaGrad) [108], RMSProp [109], and Adam [110].

The denoising autoencoder (DAE) is a variant of autoencoder that improves the capacity to learn non-linear latent representations of signals while being resilient to signal corruption or noise [105]. The denoising criterion is one of the constraints use by the autoencoder to avoid it memorizing its input (i.e., "simply copy the input") [105]. The two overarching goals of a DAE are [105]:

1. To learn a higher level representation that is stable and robust to corruption of the input.
2. To perform denoising task so as to capture a useful underlying structure in the input distribution.

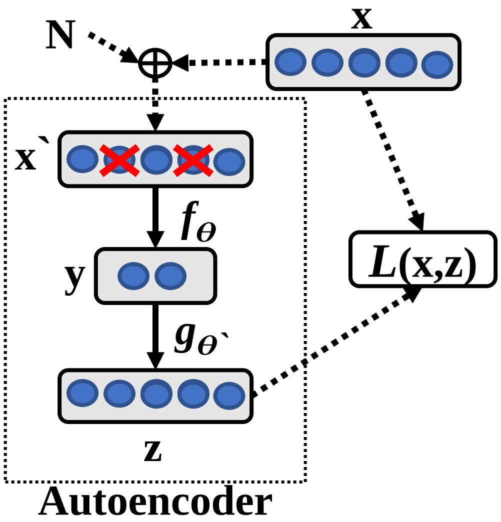


Figure 48: The schematic representation of a denoising autoencoder. An example x is corrupted by a stochastic process N to give an input x'. Input x' is passed into an autoencoder to map it to latent representation y through the encoder function f_θ. The decoder function $g_{\theta'}$ is used to reconstruct the input x' as output z. The reconstruction error is measured through the loss function $L(x, z)$.

The DAE is trained to reconstruct a clean input from a corrupted or defective input. Fig. 48 shows the schematic illustration of a DAE. A clean input x is first corrupted by a stochastic process N to form input x'. The corrupted input is passed into an autoencoder to learn a latent representation y using the encoder f_θ. The latent representation y is used by the decoder $g_{\theta'}$ to reconstruct the corrupted input x' as an uncorrupted input z with is comparable to clean input x. Both θ and θ' are trained to minimize the reconstruction error between x and z using a training set.

The stochastic process for corrupting input data can be additive white Gaussian noise (AWGN), masking noise, and salt-and-pepper noise [105]. AWGN is one of the most common types of noise and it is generally used for corrupting data that has real values. The salt-and-pepper noise is generally used for data that are binary numbers. On the other hand, masking noise is dropping certain values in the data to create a form of missing data or substituting certain values in data with a random value or default values.

6.3.1 Proposed SDAE Architecture

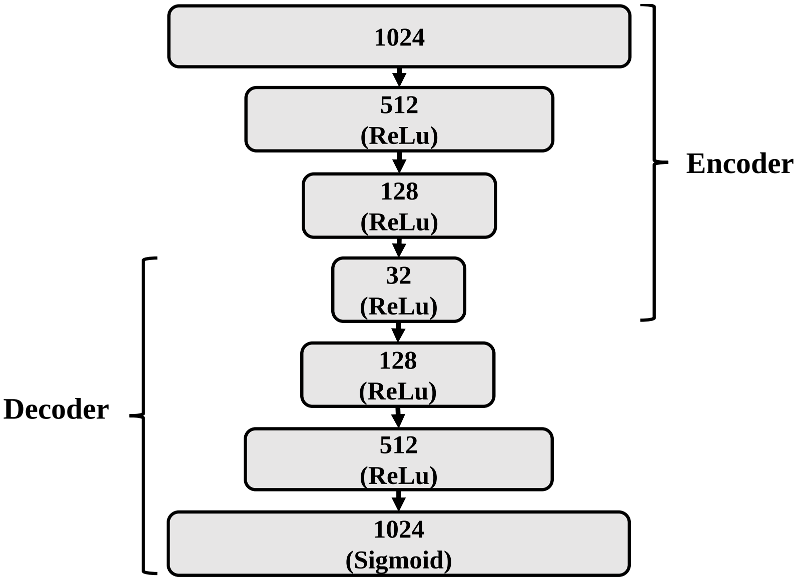


Figure 49: The architecture of the proposed stacked denoising autoencoder for signal compression and extraction of latent representation. The SDAE's input and output size are 1024. The input layer has no activation function but the output layer uses a sigmoid as the activation function. The SDAE has five hidden layers with size 512, 128, 32, 128 and 512, respectively. Each of the hidden layers uses a ReLu as the activation function. The dimension of the latent representation is 32.

The SDAE is used for signal compression and to learn an under-complete representation of a signal that is robust to channel noise. The architecture of the SDAE is shown in Fig. 49. It has an input layer, five hidden layers and an output layer. The input and output layer both have equal sizes (i.e. 1024 nodes). The size of the each hidden layer is 512, 128, 32, 128 and 512, respectively. Only the input layer has no activation function. The activation function of each hidden layer is a ReLu and for the output layer is a sigmoid. The SDAE compresses a signal with 1024 length to 32 dimensional latent representation.

6.4 Local Outlier Factor

The principle of LOF is discussed in chapter 5.

6.5 Training the SDAE-LOF Model

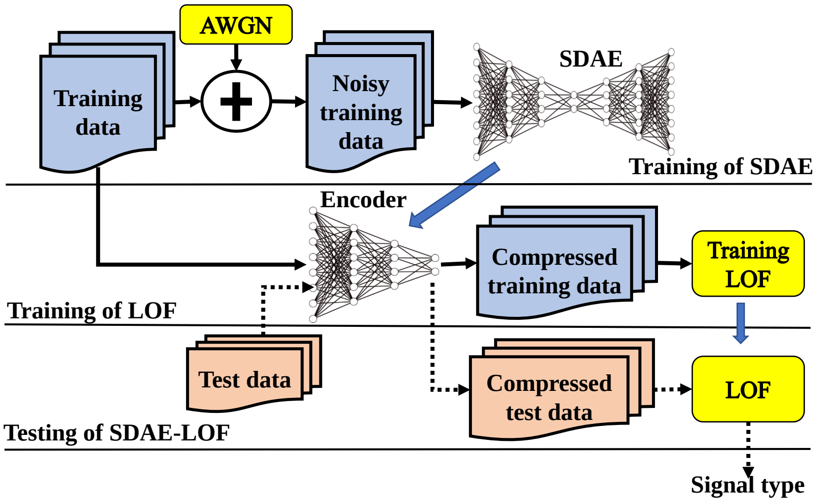


Figure 50: Stacked denoising autoencoder-local outlier factor (SDAE-LOF) training process for UAV Detection. The training process is a dual level process. The first level trains the SDAE and the second level trains the LOF. After the training of SDAE, the encoder is detached from the network and used in the second stage to compress the uncorrupted training data. The compressed training data is used to train the LOF model. Similarly, the testing data is first compressed using the encoder before passing to the trained LOF for UAV detection.

The training of the SDAE-LOF model is done in two stages as illustrated in Fig. 50. The first stage is training the SDAE model and the second stage is training the LOF model. In the first stage, the commonplace (i.e., Bluetooth and WiFi) signal x in the training set T is assumed to be a recognized signal propagating within the infrastructure under surveillance at a high SNR where there is little or no channel noise. It is important to note that the training set for the SDAE-LOF model only consists of signals from Bluetooth and WiFi devices (i.e., no UAV signals are included). Signal x is corrupted as $\acute{x}$ due to channel noise. $\acute{x}$ is given as:

$$\acute{x} = x + n, \tag{33}$$

where n is the AWGN. We assume that the channel noise is AWGN in this work.

Some signals x in T are randomly corrupted with AWGN. This is to increase the ability of the model to learn higher-level representation even in the presence of channel effects. The essence of the SDAE is to map $\acute{x}$ to x by forming a latent representation. To ensure a fast learning process by the SDAE, min-max normalization is applied on $\acute{x}$ and x which constrained the signal amplitude to fall between 0 and 1. The normalization of $\acute{x}$ and x is given as $\acute{m}$ and m, respectively.

$$m = \frac{x_i - min(x_{train})}{max(x_{train}) - min(x_{train})}, \tag{34}$$

$$\acute{m} = \frac{\acute{x}_i - min(\acute{x}_{train})}{max(\acute{x}_{train}) - min(\acute{x}_{train})}, \tag{35}$$

Normalized signals $\acute{m}$ and m are mapped into a latent representation h as

$$h = f_\theta(\acute{m}), \tag{36}$$

where f_θ is the encoder function of the SDAE.

By using h to reconstruct $\acute{m}$ through a decoder function as u;

$$u = g_{\theta'}(h), \tag{37}$$

where $g_{\theta'}$ is the encoder function of the SDAE.

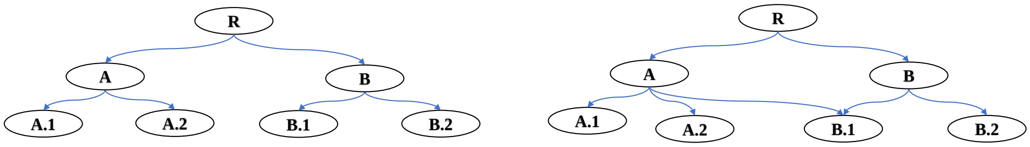
u must be nearly the same as m (i.e., $u \approx m$). Therefore, SDAE is trained to minimize the reconstruction error between u and m.

After the SDAE has been well trained, the decoder is detached from it. The encoder f_θ is used to encode the uncorrupted training data into a latent representation (i.e., a compressed training data with 32 dimensional length), which is used to train the LOF model.

The trained-LOF model is passed to the testing stage. The testing data consists of Bluetooth, WiFI, UAV and UAV controller signals. The encoder function f_θ is used to compress the testing set. The compressed signal can then be passed into the LOF model for detection of a UAV or UAV controller signal.

6.6 Hierarchical Classification

The majority of research studies have made use of flat classification (i.e., binary or multi-class classification) for UAV identification, especially where a UAV's flight modes are considered. Hierarchical classification is a learning strategy in which classes in a classification problem are methodically structured into a class hierarchy, forming a tree structure or directed acyclic graph. [111]. This implies that classes with high similarities are categorized or structured into meta-classes. For example, UAVs from the same manufacturer can create a meta-class that bears the manufacturer's name. Basically, any classification problem where the classes have an *'IS-A'* relationship can be considered or formulated as a hierarchical classification problem [111, 112].



(a) (b)

Figure 51: The two types of hierarchical structure: (a) tree-based structure, (b) DAG based structure.

According to [111,113,114], there are three criteria that differentiate one hierarchical classification method from another. These are:

1. Hierarchical structure: The two types of hierarchical structures are: tree-based and directed acyclic graph (DAG) [111]. In a tree-based structure, the child node cannot

have more than one parent node. On the other hand, in DAG, a child node can have more than one parent node. Fig. 51(a) and Fig. 51(b) show the tree-based and DAG structure, respectively. In Fig. 51(b), the child node $B.1$ has two parent nodes (i.e., node A and node B). This makes the structure a DAG structure.

2. The second criterion is the depth or level at which classification is performed in the hierarchy. Suppose that the hierarchical classifier is designed to only classify the leaf node of the hierarchy. In that case, it is referred to as mandatory leaf-node prediction (MLNP) [113] or virtual category tree [114]. Conversely, suppose the classifier is implemented to predict any node at any level in the hierarchy. In that case, such method is called non-mandatory leaf-node prediction (NMLNP), or category tree [114].

3. The third criterion is the approach in which the hierarchical structure is explored [111]. These can be: (i) a top-down approach where local classifiers are utilized at every node or parent node or level in the hierarchy; (ii) a global approach where a single classifier is designed to handle all the class hierarchy; (iii) a flat approach which disregards the hierarchy relationship among classes.

6.6.1 Local Hierarchical Classification

Local hierarchical classification (LHC) is a top-down approach where a local classifier is utilized at a node, parent node, or level in a hierarchy to improve classification based on the local information [111]. There are three ways of implementing LHC, and these include:

1. A local classifier per node (LCN): A binary classifier is utilized for each node in the hierarchy.

2. A local classifier per parent node (LCPN): A classifier is used for every parent node in the hierarchy to classify the children nodes. A multi-class classifier or binary classifier can be adopted depending on the number of children nodes or use case.

3. A local classifier per level (LCL): A single classifier for every level in the hierarchy to classify nodes in the given level.

6.6.2 Hierarchical Classification Algorithm Definition

A unified framework that defines the algorithm for hierarchical classification is proposed in [111]. A hierarchical classification can be described by four-tuple as shown:

$$HC = \langle \Delta, \Xi, \Omega, \Theta \rangle, \tag{38}$$

where

- Δ defines the ability of the algorithm to predict single or multiple paths in the hierarchy. This implies that this variable can only take two values which are: single path prediction (SPP) and multiple path prediction (MPP).

- Ξ defines the level or depth at which classification is performed. As previously discussed, this can be either MLNP or NMLNP.

- Ω is the hierarchy structure which can be a tree-based (T) or directed acyclic graph (DAG).

- Θ defines the mannerism of exploring the hierarchical structure. This can be local or global. If a local approach is adopted, then the variable can be LCN, LCL, or LCPN.

A new variable Λ is introduced to give the description of HC algorithm as five-tuple:

$$HC = \langle \Delta, \Xi, \Omega, \Theta, \Lambda \rangle, \tag{39}$$

where Λ is the learning algorithm (i.e., ML classification algorithm) that can be support vector machine (SVM), deep neural network (DNN), logistic regression, XGBoost, decision tree, ensemble, and so on. Introducing Λ gives complete information and definition of an HC algorithm.

6.6.3 Proposed Hierarchical Classification Algorithm

The proposed HC algorithm in this work is defined as:

$$HC = \langle MPP, NMLNP, T, LCPN, XGBoost \rangle, \tag{40}$$

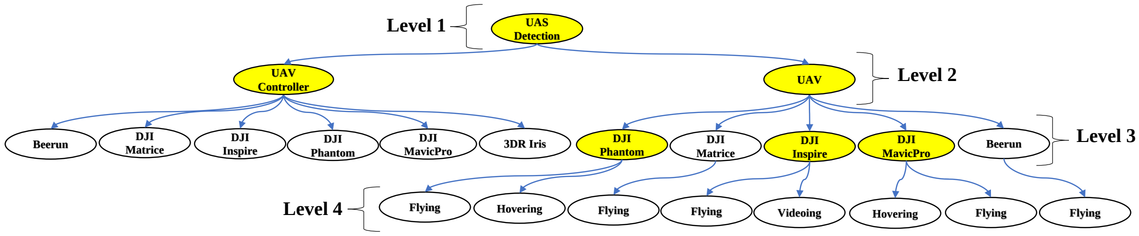


Figure 52: Tree based structure for UAS identification using hierarchical classification. The nodes in yellow are nodes with more than one child node. A classifier is employed at every node with more than one child node. For this reason, the hierarchical classifier is made up of six classifiers.

LCPN reduces inconsistent prediction and gives tolerance for class relationship [111]. The tree-based structure simplifies the class relationship. MPP helps in understanding the prediction flow, and NMLNP enables the adaptability of the algorithm for various use cases. XGBoost is the ML classifier which is a scalable tree boosting algorithm that sequentially combines several predictors (i.e., decision trees) [115]. Each predictor improves on the error of its predecessor.

6.7 Feature Extraction

The proposed feature extraction method exploits two techniques which are the Hilbert Huang transform and the wavelet packet transform. From analysis, the feature set from each transform is insufficient to give an optimal classification accuracy. Hence, the extracted feature set from WPT is used to augment the HHT feature set. Twenty-six statistical features are extracted from HHT, and another sixteen statistical features are extracted from the WPT of the signals.

6.7.1 Hilbert Huang Transform

Hilbert Huang Transform (HHT) is a time-frequency-energy domain analysis approach that can be utilized for extracting intrinsic and hidden characteristics of a signal. HHT exploits both empirical mode decomposition (EMD) and Hilbert transform (HT) for analyzing a signal [116].

EMD decomposes a signal into a collection of oscillating waves that reveals latent quasi-periodicity and characteristics of the signal. The set of oscillating waves is called intrinsic mode functions (IMFs) and residual. The decomposed signal $x(t)$ is expressed as:

$$x(t) = \sum_{i=1}^{n} x_i(t) + r(t), \tag{41}$$

where $x_i(t)$ denotes intrinsic mode function and $r(t)$ represent the residual component.

The decomposing of signal into IMFs is based on the sifting procedure [3, 117, 118]. The sifting procedure is described in Algorithm 1. The IMF must satisfies two properties [3]:

1. An IMF has only one extreme between zero crossings. This implies that the difference between the number of local minima and maxima is not more than one.
2. An IMF has a mean value of zero.

Algorithm 1 Sifting Procedure of Empirical Mode Decomposition [3]

```
1. Initialize: $i$ :=1, $r_0(t) = x(t)$
2. Extract the $i$-th IMF as follows:
    (a) Set $h_0(t) := r_{i-1}(t)$ and initialize $k := 1$
    (b) Locate all minima and maxima on $h_{k-1}(t)$
    (c) Construct an envelope on $h_{k-1}(t)$ using minima and maxima (i.e., $L_{k-1}(t)$ and
        $U_{k-1}(t)$ are the envelope from the minima and maxima, respectively).
    (d) Determine the mean signal $m_{k-1}(t)=\frac{1}{2}(U_{k-1}(t) - L_{k-1}(t))$ from the envelopes
    (e) Determine the $k$th component $h_k(t) := h_{k-1}(t) - m_{k-1}(t)$
    (f) Determine if $h_k(t)$ satisfies all the IMF criteria.
        i. If no, $k := k + 1$ and repeat step b
        ii. If yes, $x_i(t) := h_k(t)$ and $r_i(t) := r_{i-1}(t) - x_i(t)$
3. Determine if $r_i(t)$ is a residuum
        i. If no, $i := i + 1$ and start at step 2 again
        ii. If yes, stop the sifting process
```

Fig. 53 shows the EMD of signals from DJI Inspire and Beebeerun UAV with first two IMFs and the residual for each signals.

HT is used for analyzing the time-frequency-energy distribution of the signal. HT is used for spectral analysis of each IMF to extract the frequency information with respect to time and measures the variation of the oscillating wave at different time-spaces and locations [116,117]. For each IMF, the HT returns an analytical signal z as defined in (42).

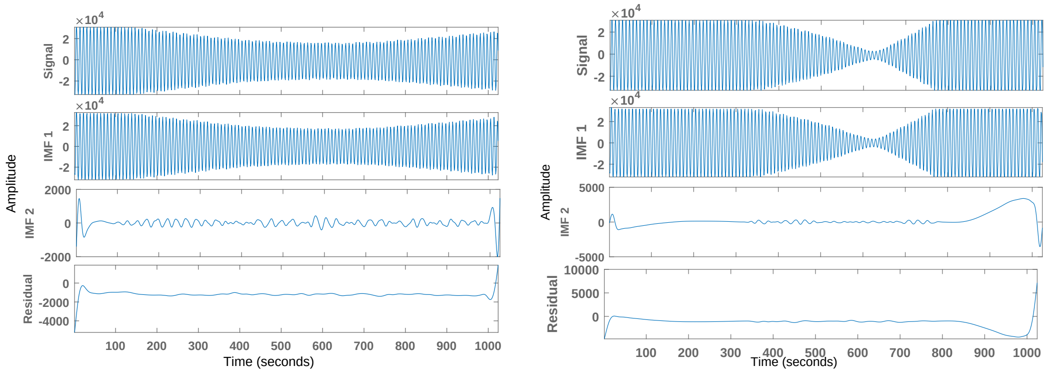
$$z_i(t) = imf_i(t) + jH\{imf_i(t)\}, \tag{42}$$

where $H\{imf_i(t)\}$ is the HT of $imf_i(t)$. The analytical signal, $z_i(t)$, can be represented in a polar form as (43):

$$z_i(t) = a_i(t)\ e^{j\theta_i(t)}, \tag{43}$$

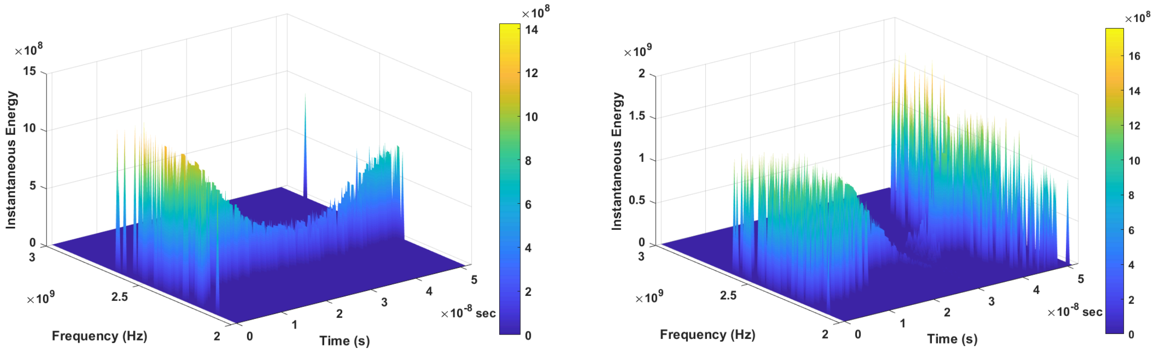
where $a_i(t)$ is the instantaneous amplitude and $\theta_i(t)$ is the instantaneous phase. The instantaneous energy is expressed as $|a_i(t)|^2$.

For feature extraction using HHT, only the first two IMFs are employed. For each IMF (i.e., $i = 1, 2$), thirteen statistical parameters statistically summarize the instantaneous energy as features giving a total of 26 features from HHT. The thirteen statistical parameters are described in Table 15. The HT for the first two IMFs of signals from DJI Inspire and Beebeerun UAV are shown in Fig. 54(a) and Fig. 54(b), respectively. This shows the magnitude of the energy distribution in the time-frequency domain.



(a) (b)

Figure 53: The empirical mode decomposition showing the first two IMFs and the residual of the captured RF signals (i.e., using the steady state) from: (a) DJI Inspire, (b) Beebeerun.



(a) (b)

Figure 54: The Hilbert spectrum showing the time-frequency-energy characteristics from the steady state of the captured RF signals: (a) DJI Inspire, (b) Beebeerun.

TABLE 15

Definition of statistical features used for feature extraction.

Features	Formula	Measures
Mean (μ)	$\frac{1}{N}\sum_{i=1}^{N} x_i$	Central tendency
Harmonic mean	$(\frac{1}{N}\sum_{i=1}^{N} x_i^{-}1)^{-1}$	Central tendency
Standard deviation (σ_T)	$\left[\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})^2\right]^{\frac{1}{2}}$	Dispersion
Variance	$\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2$	Dispersion
Kurtosis (k)	$\frac{\sum_{i=1}^{N}(x_i-\bar{x})^4}{(N-1)\sigma_T^4}$	Tail/shape descriptor
Root mean square (x_{rms})	$\left[\frac{1}{N}\sum_{i=1}^{N} x_i^2\right]^{\frac{1}{2}}$	Magnitude/Average power
Shape factor(x_{sf})	$\frac{x_{\text{rms}}}{\bar{x}}$	Shape descriptor
Peak value (x_{pv})	$\max(\mathrm{x_i})$	Amplitude
Peak to peak (x_{ppv})	$\max(\mathrm{x_i})-\min(\mathrm{x_i})$	Waveform amplitude
Interquantile range	Q_3-Q_1	Dispersion
Shannon entropy(H_s)	$-\sum_{i=1}^{N} {x_i}^2 \log {x_i}^2$	Uncertainty
Summation(S)	$\sum_{i=1}^{N} x_i$	Cumulative sum of amplitude

6.7.2 Wavelet Packet Transform

The theoretical concept of WPT has been discussed in Section 5. However, sixteen statistical features are extracted from the decomposition of a signal using the WPT. For each packet, four statistical parameters are used to describe the properties of the packet as a feature. These four statistical features are standard deviation, variance, peak to peak magnitude, and Shannon entropy. The definitions of these features are in Table 15. The primary goal of the feature set from the WPT is to improve the performance of the proposed model by augmenting the feature set from the HHT.

6.8 Experimental Results and Discussions

The proposed framework is evaluated, taking into consideration five primary directions. These directions are:

1. The performance of SDAE based on the reconstruction error
2. The performance of LOF based on: i) the selection of the number of nearest neighbors for determining local density; ii) the type of distance measure; iii) classification metrics; iv) amount of training data required; v) inference time; vi) impact of channel effect.
3. The performance of each composing model of the hierarchical classifier.
4. The performance of the LHC based on hierarchical classification metrics (i.e., hierarchical precision, hierarchical recall, and hierarchical F1-score).
5. The effect of feature set from the WPT on the LHC's performance.

6.8.1 Evaluation of SDAE

The training of the proposed SDAE model is done using some hyperparameters. These hyperparameters are: loss function, learning rate, optimizer, epoch, and batch size. The loss function utilized is mean absolute error, which measures the reconstruction error of the training process. The Adam optimization is used as the optimizer with a learning rate of 0.001. Similarly, the values for the number of epoch and batch size are 100 and 128, respectively.

During the training process of SDAE, 30% of the training set is used as the validation set. The learning curve of the training process is shown in Fig. 55 where the reconstruction error for both training and validation sets decreases steadily per epoch.

The test set for evaluating SDAE comprises signals from Bluetooth and WiFi devices just the same way as the training set consists of signals from Bluetooth and WiFi devices. However, the test set is not corrupted with AWGN (i.e., the test set's signals are at 30 dB SNR). When SDAE was evaluated using the test set, the total reconstruction error (MAE) is 0.00793. This implies that the SDAE was able to reconstruct the all the test set with approximate error equal to zero.

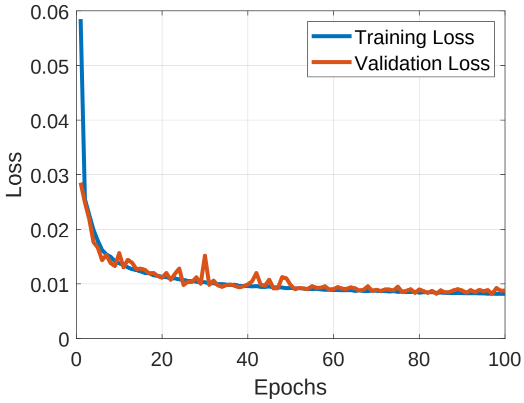


Figure 55: The learning curve of SDAE. This shows learning performance of the SDAE at every epoch. The loss function of the training set and validation set drop steadily per epoch.

6.8.2 Evaluation of LOF

The main objective of the LOF model is to classify a compressed signal as either an inlier or outlier. This model was evaluated using the same principles or metrics for evaluating a binary classifier. Therefore, classification metrics such as accuracy, precision, recall, and F1-score can be used to evaluate the classifier's performance.

The encoder of the trained SDAE is used to compress the training set into a 32-dimensional latent representation signal. The compressed training set is used to train a LOF model. The evaluation set for the LOF model consists of 100,000 recognized signals (i.e., Bluetooth and WiFi) and 100,000 UAS signals. The evaluation set is split into test and validation set in the ratio of 80% to 20%, respectively.

The key hyperparameters of LOF are the distance metric and the number of nearest neighbors for estimating the local density of a data point. The distance measure is used for measuring similarity in data. There are different distance metrics, and these include Manhattan (also known as $L1$ or city block), Euclidean (also known as $L2$), Cosine distance, Minkowski distance, and so on. For the essence of training the LOF model, Manhattan distance is used for the distance measure. However, we assess the impact of the distance measure on the model's performance using Euclidean and Cosine distance.

When using LOF, determining the optimal number of nearest neighbors to estimate the local density of a data point is not a simple operation because it influences the model's performance. To choose the number of nearest neighbors, the validation set is utilized. The number of neighbors is first set to 10 and then increased in steps of 10 to 200. At every step of 10, the performance of the model (i.e., classification accuracy) is assessed using the validation set. Fig. 56 shows the performance of the LOF model as the number of nearest neighbors is been fine-tuned by increasing at the step of 10. The validation set had the maximum classification accuracy of 89.49% when the number of neighbors n is 20. As a result, the LOF model estimates a datapoint's local density using a number of nearest neighbors of 20.

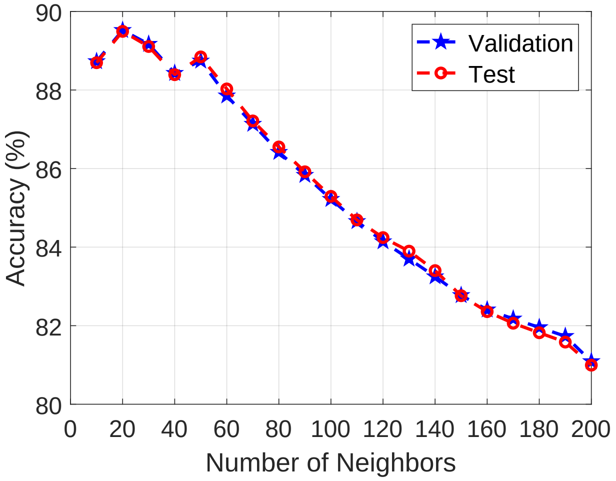
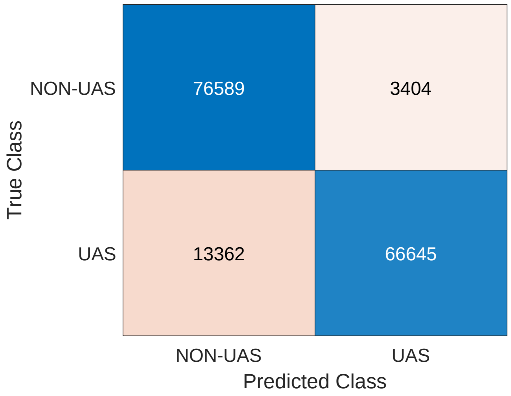


Figure 56: Average accuracy of the LOF model based on the validation and test set when increasing the number of neighbors for local density estimation at 30 dB SNR.

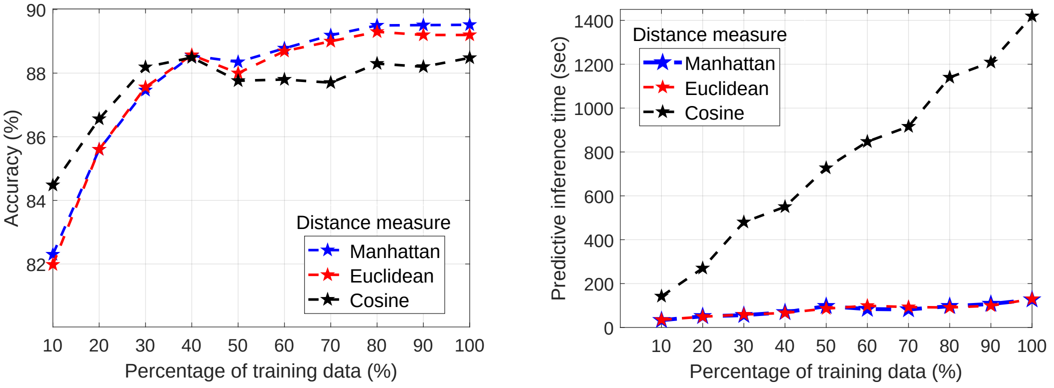
The test set is used to evaluate the LOF classifier's performance. The confusion matrix of the classifier using the test set is shown in Fig. 57. The confusion matrix's rows refer to the true class, and the columns show the predicted class. The LOF classifier is able to classify 95.7% of the Non-UAS signals and 83.3% UAS signals as the true class. The total



True Class \ Predicted Class	NON-UAS	UAS
NON-UAS	76589	3404
UAS	13362	66645

Figure 57: Confusion matrix of the LOF model ($k = 20$) when evaluating the model with the test set at 30 dB.

number of signals in the train set is 234,500 signals. So, by altering the amount of training data and the distance metric (i.e., using Euclidean and Cosine distance as a substitute for Manhattan distance), the effects of training set size on accuracy and predictive inference time are evaluated. The prediction inference time is the amount of time it takes the LOF classifier to infer the signal type in the test set.



(a) (b)

Figure 58: (a) The relationship between training set size and accuracy for the LOF model, (b) The relationship between training set size and inference time for the LOF model. Using either Manhattan or Euclidean distance has no significant differences on the LOF model. The Manhattan and Euclidean distance outperforms the Cosine distance.

Fig. 58(a) depicts the relationship between classification accuracy, distance metric, and training set size. When employing Manhattan and Euclidean distance, the accuracy of the LOF model increases as the training set grows. Both Manhattan and Euclidean distance have equivalent performance when using 10% to 40% of the training data. As we increase the training data from 40% to 100%, the Manhattan distance somewhat improves in accuracy. Similarly, when 30% or less training data is used, the Cosine distance outperforms both the Manhattan and Euclidean distances in terms of accuracy. When the training data is increased, both Manhattan and Euclidean distance outperform Cosine distance.

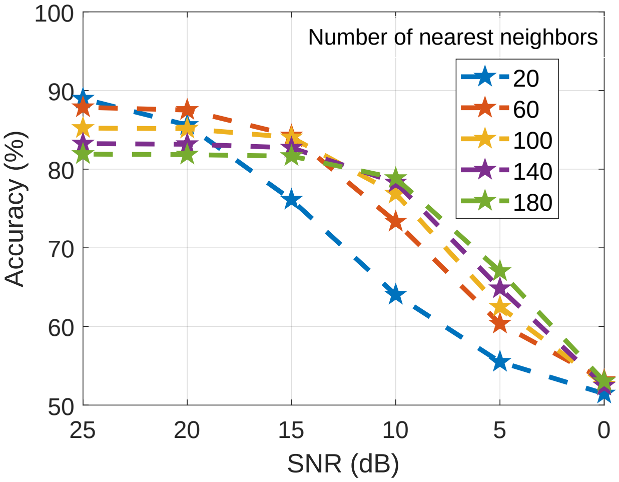


Figure 59: The classification accuracy of LOF model (distance measure is Manhattan) under varying SNR and the number of nearest neighbors. The number of nearest neighbors is increased at the step of 40 starting from 20 to 180.

The relationship between predictive inference time, distance metric, and training set size for the LOF classifier is depicted in Fig. 58(b). As the size of the training set grows, so does the time required for predictive inference. This is due to the fact that the LOF classifier is a non-parametric model. It computes the local density of each data point and assigns numbers to signify the degree to which the data point is an outlier. Predictive inference time is similarly comparable for Manhattan and Euclidean distance. The Cosine distance, on the other hand, takes longer than both the Manhattan and Euclidean distances. The average predicted inference time using all of the training data for Manhattan and Euclidean distance is around 150 seconds, whereas Cosine distance takes over 1400 seconds. When compared to the other two distance measures, the usage of Cosine distance takes longer.

In addition, by varying the SNR of the signals, the impact of channel noise on the LOF model is investigated. To change the SNR, AWGN is introduced to the signals. The effect of the number of nearest neighbors and the distance measure (i.e., utilizing Euclidean and Cosine distance instead of Manhattan distance) on varying SNR is analyzed.

The behavior of the LOF model under different SNR and number of nearest neighbors is shown in Fig. 59. With a reduction in the signal's SNR, the model's accuracy steadily decreases. The model's classification accuracy is around 89 percent at 25 dB, using 20 as the number of nearest neighbors. This is more effective than using a higher value (i.e., greater than 20) for the number of nearest neighbors. However, from 15 dB to 0 dB, the model's performance improves with an increase in the number of nearest neighbors. For example, at 10 dB, the model's accuracy is 64% , 73% , 77% , 78% , and 79% , respectively, when the number of nearest neighbors is 20, 60, 100, 140, and 160.

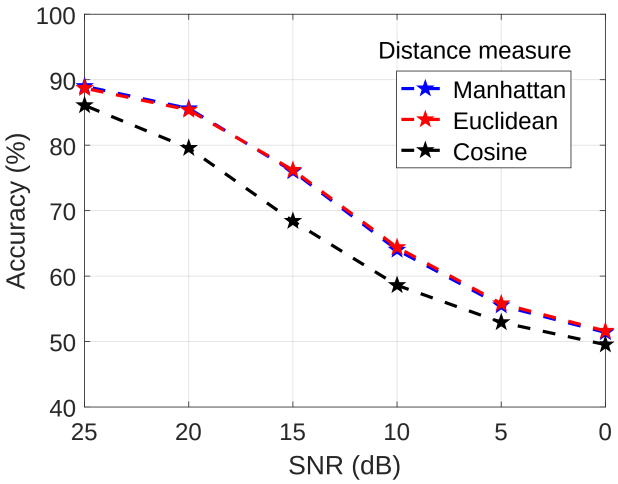


Figure 60: Performance of the LOF model (the number of nearest neighbors is 20) under varying SNR and distance measure. The Manhattan and Euclidean distance have similar characteristics when varying the SNR of the signal. Both Manhattan and Euclidean distance outperform the Cosine distance.

The model's performance while varying the signal's SNR and using the three distance measures is depicted in Fig. 60. At various levels of SNR, the Manhattan and Euclidean distances have comparable properties and outperform the Cosine distance.

6.8.3 Evaluation of each model in the Hierarchical Classifier

The hierarchical classifier is made up of six XGBoost models as shown in Table 16 which shows the performance of each model using the classification metrics defined in (17) to (20). The classifiers are first trained using the 26-feature set from the HHT. The UAS classifier, UAV controller classifier, UAV classifier, DJI Phantom classifier, DJI inspire classifier, and DJI MavicPro classifier, respectively, have accuracy of 87.10%, 71.19%, 81.45%, 87.37%, 94.46% and 81.40%. The 16-feature set from WPT is utilized to supplement the 26 features from the HHT in order to increase the classifiers' accuracy. The performance of the classifiers improves when the HHT-WPT feature set is used. The accuracy of the UAS classifier, UAV controller classifier, UAV classifier, DJI Phantom classifier DJI inspire classifier, and DJI MavicPro classifier increased to 91%, 73.19%, 82.49%, 88.58%, 95.28%, and 82.39%, respectively. The impact of the WPT feature set is particularly evident at the first level of the hierarchy (i.e., UAS classifier). The UAS classifier's accuracy improved by 3.9%. Similarly, the accuracy of the UAV controller classifier, UAV classifier, DJI Phantom classifier, DJI inspire classifier, and DJI MavicPro classifier increased by 2%, 1.04%, 1.21%, 0.72%, and 0.94%, respectively. This demonstrates that the WPT feature set increases the accuracy of the individual classifier in the hierarchy.

TABLE 16: Performance measure of each local classifier in the hierarchy.

Level	Model	HHT				HHT-WPT				Accuracy difference(%)
		Accuracy (%)	Precision (%)	Recall (%)	F_1-score (%)	Accuracy (%)	Precision (%)	Recall (%)	F_1-score (%)	
1	UAS classifier	**87.10**	86.20	86.37	86.28	**91.00**	90.23	90.60	90.42	**3.90**
2	UAV controller classifier	**71.19**	72.53	73.16	72.72	**73.19**	74.01	74.30	74.06	**2.00**
2	UAV classifier	**81.45**	81.19	81.46	81.27	**82.49**	82.18	82.50	82.25	**1.04**
3	DJI Phantom classifier	**87.37**	88.62	86.00	87.30	**88.58**	89.71	87.40	88.54	**1.21**
3	DJI Inspire classifier	**94.56**	95.70	93.28	94.50	**95.28**	95.98	94.45	95.21	**0.72**
3	DJI MavicPro classifier	**81.40**	86.77	74.52	80.18	**82.39**	88.12	75.17	81.13	**0.94**

6.8.4 Evaluation of the Hierarchical Classifier

There are no standard ways of evaluating hierarchical classifier [111]. However, for a local hierarchical classification, testing is systematically carried out such that the first level prediction is used to narrow down the second level prediction and so on. This is recursively performed until it reaches the leaf node or a specific terminal node.

Flat classification metrics are sometimes used for hierarchical classification [111]. However, errors at different levels should not have the same weight or penalty [111]. In [119], three hierarchical classification metrics that penalize the hierarchy classifier's errors at different levels are proposed. These are hierarchical precision (hP), hierarchical recall (hR), and hierarchical F_1-score (hF_1). The definition of these metrics are [119]:

$$hP = \frac{\sum_i |P_i^{\prime} \cap T_i^{\prime}|}{\sum_i |P_i^{\prime}|}, \tag{44}$$

$$hR = \frac{\sum_i |P_i^{\prime} \cap T_i^{\prime}|}{\sum_i |T_i^{\prime}|}, \tag{45}$$

$$hF_1 = \frac{(\beta)^2 + 1) \cdot hP \cdot hR}{\beta^2 \cdot hP + hR}, \beta \in [0, +\infty] \tag{46}$$

where $P_i^{\prime}$ is a set of predicted class(es) for test sample i and its ancestors, $T_i^{\prime}$ is a set of true class(es) for test sample i and its ancestors, β is equal to 1 provided precision and recall are equal weight.

TABLE 17: Performance measure of the local hierarchical classifier at different levels in the hierarchy.

Level	HHT			HHT-WPT			Precision
	Precision (%)	Recall (%)	F_1-score (%)	Precision (%)	Recall (%)	F_1-score (%)	difference(%)
1	**86.20**	86.37	86.28	**90.23**	90.60	90.42	**4.03**
2	**64.88**	64.88	64.88	**67.86**	67.86	67.86	**2.98**
3	**54.38**	54.36	54.37	**56.86**	56.90	56.88	**2.48**

The performance of the hierarchical classifier per level in the hierarchy is presented in Table 17 utilizing the hierarchical classification metrics. Because there is no erroneous propagation into the first level (i.e., root node) of the hierarchy, the flat classification metric is used. Because the error from level 1 is propagated to levels 2 and 3, this error must be penalized. As a result, for the two levels, hierarchical classification metrics are utilized. We assessed the LHC when only the 26 features from HHT are utilized to build each classifier in the hierarchy. A precision of 90.23%, 67.86%, and 56.86% is achieved for level 1, 2, and 3, respectively. When the feature set from HHT-WPT is utilized for modeling, the hierarchical metrics increase. The impact of WPT was seen more in the first-level classifier. The increase in hierarchical precision is 4.03%, 2.98%, and 2.48% for level 1, 2, and 3, respectively. The hierarchical precision of the model decreases as we traverse down the hierarchy.

6.9 Conclusion

This chapter introduces a framework for UAV detection and identification that incorporates SDAE-LOF and hierarchical learning. The framework exploits the steady-state of RF signals emanating from the communication between the UAV and its flight controller. Because the UAV communication link operates in the same frequency band (i.e., 2.4 GHz) as WiFi and Bluetooth devices, an SDAE-LOF model was used to detect the presence of a UAV or UAV controller signal with an accuracy of 89.52%. SDAE is used for the compression of signals into a latent representation that is robust to channel noise. The latent representation is utilized by a semi-supervised anomaly detection model (LOF) to detect a UAV or UAV controller signal among Bluetooth and WiFi signals. The LOF algorithm's most essential hyperparameters are the distance measure and the number of nearest neighbors. These two hyperparameters are utilized in estimating the local density of data points. The effect of these two factors significantly affects the performance of the proposed SDAE-LOF model. At low SNR, it was discovered that having a large number of nearest neighbors (more than 20) increases detection accuracy. In terms of predictive inference time and varying SNR, using either Manhattan or Euclidean distance as the distance metric produces equivalent results. The Manhattan and Euclidean distances, on the other hand, outperform the Cosine distance.

To classify the type of UAS signal, UAV model, and UAV flight mode, a hierarchical learning framework is employed. For feature extraction, the framework employs HHT and WPT. The WPT feature set supplements the HHT feature set to increase the framework's performance. The impact of employing HHT-WPT for feature extraction is significant at the root node of the hierarchy (i.e., determining the type of UAS signal) and gradually diminishes down the hierarchy.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

This dissertation focused on advancing the state of the art in RF-based UAV detection and identification systems. The proposed frameworks in this dissertation are not limited to UAV detection and identification. They can be extended to the detection of rogue RF devices in an environment. For instance, in a data center infrastructure where unknown or unwanted RF devices are restricted from operating to avoid compromise in network security.

It is demonstrated in this work that exploiting the steady state of a signal for UAV detection and identification is robust to channel noise when compare to the transient state. More so, using an image-based signature (i.e., scalogram or scattergram) with a pre-trained CNN-based model (i.e., SqueezeNet) performs better than coefficient-based signatures in term of classification accuracy even though it is computationally expensive to extract image-based signatures. Using scattergrams as a signature for UAVs identification outperforms scalograms in terms of classification accuracy and time cost for extracting its signature from a signal.

This work also demonstrated that having high detection or classification accuracy is not a sole factor to consider in designing a UAV detection and identification system; other design factors must be considered. Such factors include but are not limited to robustness to channel effect or noise, time and space complexity, and so on.

More so, similarities in UAVs from the same manufacturer (i.e., DJI) create classification errors that inhibit a classifier's performance. Effective feature engineering on the RF signals from UAVs or UAV controllers can help in reducing the classification errors.

TABLE 18: Comparison of pros/cons using supervised and semi-supervised learning approach for UAV detection and identification.

Properties	**Supervised Learning**	**Semi-supervised Learning**
Labeling	full labeling	partial labeling
Labeling cost	high	relatively low
Implementation of a classifier for the identification of UAV models and flight modes	easy	very difficult
Detection of UAVs not included in the training set	no	yes
Validation set	not necessary	very important

The semi-supervised anomaly detection approach is a potential solution to the problem of UAV detection and identification. Thus, a semi-supervised learning framework is introduced for UAV detection in this work. To the best of my knowledge, no research has exploited semi-supervised learning framework for UAV detection. This approach was proposed by using the LOF algorithm which is resilient in detecting outliers in a dataset.

Two semi-supervised learning frameworks are proposed. The first one is based on extracting features from the transient state of the RF signals by using time-frequency domain techniques (i.e., WPT). The second one is based on extracting features from the steady state of RF signal by using unsupervised learning approach (SDAE) to compress the signals into a latent representation. One important finding from the two proposed semi-supervised learning frameworks is that the performance of the LOF algorithm increases with an increase in the number of nearest numbers used in estimating the density of a data point at lower SNR. Distance measure is an imperative hyperparameter of LOF, as investigated in this work, utilizing either Manhattan or Euclidean distance as the distance metric which gives comparable results. Based on the results of this work, Table 18 lists the comparison of using supervising or semi-supervising for UAV detection.

Another integral part of this work is introducing a hierarchical classification approach to UAV identification. The UAV classification problem is constrained into a tree hierarchy to classify the type of UAS signal, UAV model, and flight mode of the UAV. One of the benefits of the hierarchical classifier is to exploit the granularity of classifiers in achieving an improved classification accuracy. Achieving a classification accuracy of 91% at the first level of the hierarchy (i.e., determining the type of UAS signal) depicts that the UAV signals are different from the UAV controller signals.

7.2 Future Work

The introduction of semi-supervised learning framework has set a pace for exploring out-of-order distribution or incremental learning for UAV detection and identification research. Hence, this work will be extended by investigating the application of incremental learning for UAV detection and identification. The future of UAV detection and identification systems is to allow certain UAVs to fly into a restricted area. For instance, Amazon Inc. has received approval from the FAA to use UAVs for delivery. If an individual has restricted their home from the operation of UAVs but wants to allow Amazon UAVs to provide a home delivery of goods, then the individual might require Amazon's UAV RF signatures to re-train their UAV detection and identification system. This will avoid the false alarm of detecting an intruding UAV by the system. Instead of retraining the system from scratch, incremental learning can be applied to fine-tune the model to learn the new signatures.

With the advancement in computer technology, the possibility of developing a multi-modular sensing technique for UAV detection and identification on an edge device is currently being explored in collaboration with the department of electrical and computer engineering, North Carolina State University. The outdoor experiment is currently being performed where audio, RADAR, video, and RF sensors are utilized. Some of the aspects in this work will be incorporated as part of the micro-architecture of the multi-modular sensing framework for UAV detection and identification.

REFERENCES

[1] Sinno Jialin Pan and Qiang Yang, “A survey on transfer learning,” *IEEE Trans. on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[2] Francois Chollet et al., *Deep learning with Python*, vol. 361, Manning New York, 2018.

[3] Angela Zeiler, Rupert Faltermeier, Ingo R Keck, Ana Maria Tomé, Carlos García Puntonet, and Elmar Wolfgang Lang, “Empirical mode decomposition-an introduction,” in *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2010, pp. 1–8.

[4] FAA, “UAS by the numbers,” 2020, ”Accessed: 2020-10-12”.

[5] Anna Banaszek, Aleksander Zarnowski, Anna Cellmer, and Sebastian Banaszek, “Application of new technology data acquisition using aerial UAV digital images for the needs of urban revitalization,” in *Proc. Environmental Engineering, Int. Conf.*, Vilnius, Lithuania, Apr. 2017, pp. 1–7.

[6] Bilal Hazim Younus Alsalam, Kye Morton, Duncan Campbell, and Felipe Gonzalez, “Autonomous UAV with vision based on-board decision making for remote sensing and precision agriculture,” in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2017, pp. 1–12.

[7] Seamus Coveney and Kenneth Roberts, “Lightweight UAV digital elevation models and orthoimagery for environmental applications: data accuracy evaluation and potential for river flood risk modelling,” *Int. Journal of Remote Sensing*, vol. 38, no. 8-10, pp. 3159–3180, 2017.

[8] Riham Altawy and Amr M Youssef, “Security, privacy, and safety aspects of civilian drones: A survey,” *ACM Transactions on Cyber-Physical Systems*, vol. 1, no. 2, pp. 1–25, 2016.

[9] NASA, “What is unmanned aircraft systems traffic management?,” 2020, Accessed: 2020-10-12.

[10] Kim Gittleson, “Data-stealing snoopy drone unveiled at black hat,” 2014, Accessed: 2020-10-12.

[11] Michael S. Schmidt and Michael D. Shear, “A drone, too small for radar to detect, rattles the white house,” 2016, Accessed: 2020-10-12.

[12] Doug Bolton, “Man arrested for landing radioactive drone on japanese prime minister’s roof,” Accessed: 2020-10-12.

[13] Gina Harkins, “Illicit drone flights surge along U.S.-Mexico border as smugglers hunt for soft spots,” Accessed: 2020-10-12.

[14] FAA, “UAS sightings report,” 2020, Accessed: 2020-10-12.

[15] "Fly safe – drone flying tips, policies regulations, and more – DJI," Accessed: 2020-11-12.

[16] "Applications and services," Accessed: 2020-11-12.

[17] Daniela Doroftei and Geert De Cubber, "Qualitative and quantitative validation of drone detection systems," in *Proc. Int. Symposium on Measurement and Control in Robotics*, Mons, Belgium, Sep. 2018, pp. 26–28.

[18] Gabriel C Birch, John C Griffin, and Matthew K Erdman, "UAS detection, classification, and neutralization: Market survey 2015," *Sandia National Laboratories*, 2015.

[19] Rick L Sturdivant and Edwin KP Chong, "Systems engineering baseline concept of a multispectral drone detection solution for airports," *IEEE Access*, vol. 5, pp. 7123–7138, Apr. 2017.

[20] Phuc Nguyen, Hoang Truong, Mahesh Ravindranathan, Anh Nguyen, Richard Han, and Tam Vu, "Cost-effective and passive RF-based drone presence detection and characterization," *GetMobile: Mobile Computing and Commun.*, vol. 21, no. 4, pp. 30–34, 2018.

[21] Ben Brown and Kevin Buckler, "Pondering personal privacy: a pragmatic approach to the fourth amendment protection of privacy in the information age," *Contemporary Justice Review*, vol. 20, no. 2, pp. 227–254, 2017.

[22] Roman V Yampolskiy and Marina L Gavrilova, "Artimetrics: biometrics for artificial entities," *IEEE Robotics & Automation Mag.*, vol. 19, no. 4, pp. 48–58, 2012.

[23] Wenyi Zhao, Rama Chellappa, P Jonathon Phillips, and Azriel Rosenfeld, "Face recognition: A literature survey," *ACM computing surveys (CSUR)*, vol. 35, no. 4, pp. 399–458, 2003.

[24] Claudia Stöcker, Rohan Bennett, Francesco Nex, Markus Gerke, and Jaap Zevenbergen, "Review of the current state of uav regulations," *Remote sensing*, vol. 9, no. 5, pp. 459, 2017.

[25] Mais Nijim and Nikhil Mantrawadi, "Drone classification and identification system by phenome analysis using data mining techniques," in *Proc. IEEE Symposium on Technologies for Homeland Security (HST)*, Waltham, MA, USA, May 2016, pp. 1–5.

[26] József Mezei, Viktor Fiaska, and András Molnár, "Drone sound detection," in *Proc. IEEE Int. Symposium on Computational Intelligence and Inf. (CINTI)*, Budapest, Hungary, Nov. 2015, pp. 333–338.

[27] József Mezei and András Molnár, "Drone sound detection by correlation," in *Proc. IEEE Int. Symposium on Applied Computational Intelligence and Inf. (SACI)*, Timisoara, Romania, May 2016, pp. 509–518.

[28] Xuejun Yue, Yongxin Liu, Jian Wang, Houbing Song, and Huiru Cao, "Software defined radio and wireless acoustic networking for amateur drone surveillance," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 90–97, 2018.

[29] Eren Unlu, Emmanuel Zenou, Nicolas Riviere, and Paul-Edouard Dupouy, "Deep learning-based strategies for the detection and tracking of drones using several cameras," *IPSJ Trans. on Comput. Vision and Applications*, vol. 11, no. 1, pp. 7, 2019.

[30] Arne Schumann, Lars Sommer, Johannes Klatte, Tobias Schuchert, and Jürgen Beyerer, “Deep cross-domain flying object classification for robust UAV detection,” in *Proc. IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS)*, Lecce, Italy, Aug. 2017, pp. 1–6.

[31] Muhammad Saqib, Sultan Daud Khan, Nabin Sharma, and Michael Blumenstein, “A study on detecting drones using deep convolutional neural networks,” in *Proc. IEEE Int. Conf. on Adv. Video and Signal Based Surveillance (AVSS)*, Lecce, Italy, Aug. 2017, pp. 1–6.

[32] Petar Andraši, Tomislav Radišić, Mario Muštra, and Jurica Ivošević, “Night-time detection of UAVs using thermal infrared camera,” *Transportation Research Procedia*, vol. 28, pp. 183–190, 2017.

[33] Jedrzej Drozdowicz, Maciej Wielgo, Piotr Samczynski, Krzysztof Kulpa, Jaroslaw Krzonkalla, Maj Mordzonek, Marcin Bryl, and Zbigniew Jakielaszek, “35 GHz FMCW drone detection system,” in *Proc. Int. Radar Symposium (IRS)*, Krakow, Poland, May 2016, pp. 1–4.

[34] Gihan J Mendis, Tharindu Randeny, Jin Wei, and Arjuna Madanayake, “Deep learning based doppler radar for micro UAS detection and classification,” in *Proc. IEEE Military Commun. Conf.*, Baltimore, MD, USA, Nov. 2016, pp. 924–929.

[35] Ruben Morales Ferre, Wenbo Wang, Alejandro Sanz Abia, and Elena Simona Lohan, “Identifying gnss signals based on their radio frequency (rf) features—a dataset with gnss raw signals based on roof antennas and spectracom generator,” *Data*, vol. 5, no. 1, pp. 18, 2020.

[36] Mohammad F Al-Sa’d, Abdulla Al-Ali, Amr Mohamed, Tamer Khattab, and Aiman Erbad, “RF-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database,” *Future Generation Comput. Syst.*, vol. 100, pp. 86–97, 2019.

[37] Phuc Nguyen, Hoang Truong, Mahesh Ravindranathan, Anh Nguyen, Richard Han, and Tam 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*, 2017, pp. 211–224.

[38] Martins Ezuma, Fatih Erden, Chethan Kumar Anjinappa, Ozgur Ozdemir, and Ismail Guvenc, “Detection and classification of UAVs using RF fingerprints in the presence of WiFi and bluetooth interference,” *IEEE Open Journal of the Commun. Society*, vol. 1, pp. 60–76, Nov. 2019.

[39] Amir Alipour-Fanid, Monireh Dabaghchian, Ning Wang, Pu Wang, Liang Zhao, and Kai Zeng, “Machine learning-based delay-aware UAV detection and operation mode identification over encrypted WiFi traffic,” *IEEE Trans. on Inf. Forensics and Security*, vol. 15, pp. 2346–2360, 2019.

[40] Igor Bisio, Chiara Garibotto, Fabio Lavagetto, Andrea Sciarrone, and Sandro Zappatore, “Blind detection: Advanced techniques for wifi-based drone surveillance,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 938–946, 2018.

[41] Caidan Zhao, Caiyun Chen, Zhibiao Cai, Mingxian Shi, Xiaojiang Du, and Mohsen Guizani, “Classification of small UAVs based on auxiliary classifier wasserstein GANs,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 206–212.

[42] Xiufang Shi, Chaoqun Yang, Weige Xie, Chao Liang, Zhiguo Shi, and Jiming Chen, "Anti-drone system with multiple surveillance technologies: Architecture, implementation, and challenges," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 68–74, 2018.

[43] Wei Zhou, Lei Wang, Bingxian Lu, Naigao Jin, Linlin Guo, Jialin Liu, Honglei Sun, and Hui Liu, "Unmanned aerial vehicle detection based on channel state information," in *IEEE Int. Conf. on Sensing, Commun. Networking (SECON Workshops)*, Hong Kong, China, Jun. 2018, pp. 1–5.

[44] Martins Ezuma, Fatih Erden, Chethan Kumar Anjinappa, Ozgur Ozdemir, and Ismail Guvenc, "Micro-UAV detection and classification from RF fingerprints using machine learning techniques," in *Proc. IEEE Aerosp. Conf., Big Sky, Montana*, Mar. 2019.

[45] Guoru Ding, Qihui Wu, Linyuan Zhang, Yun Lin, Theodoros A Tsiftsis, and Yu-Dong Yao, "An amateur drone surveillance system based on the cognitive internet of things," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 29–35, 2018.

[46] Mehmed Kantardzic, *Data mining: concepts, models, methods, and algorithms*, John Wiley & Sons, 2011.

[47] R Kurt Barnhart, Douglas M Marshall, and Eric Shappee, *Introduction to unmanned aircraft systems*, Crc Press, 2020.

[48] Federal Communications Commission, "FCC rules regulations for title 47," 2020, "Accessed: 2020-10-12".

[49] Gerton de Goeij, Eildert H van Dijken, and Frank Brouwer, "Research into the radio interference risks of drone," *Vianen, NL, May*, 2016.

[50] Dan Mototolea and Comelis Stolk, "Software defined radio for analyzing drone communication protocols," in *2018 International Conference on Communications (COMM)*. IEEE, 2018, pp. 485–490.

[51] Stephen A Rackley, *Wireless networking technology: From principles to successful implementation*, Elsevier, 2011.

[52] Howard C Choe, Clark E Poole, M Yu Andrea, and Harold H Szu, "Novel identification of intercepted signals from unknown radio transmitters," in *Wavelet Applications II*. International Society for Optics and Photonics, 1995, vol. 2491, pp. 504–517.

[53] Alan V Oppenheim, Alan S Willsky, and S Hamid Nawab, *Signals and Systems*, Pearson Educación, 1998.

[54] KJ Ellis and Nur Serinken, "Characteristics of radio transmitter fingerprints," *Radio Science*, vol. 36, no. 4, pp. 585–597, 2001.

[55] Irwin O Kennedy, Patricia Scanlon, Francis J Mullany, Milind M Buddhikot, Keith E Nolan, and Thomas W Rondeau, "Radio transmitter fingerprinting: A steady state frequency domain approach," in *Proc. IEEE Veh. Technol. Conf.*, Calgary, BC, Canada, Sep. 2008, pp. 1–5.

[56] Tadayoshi Kohno, Andre Broido, and Kimberly C Claffy, "Remote physical device fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 2, pp. 93–108, 2005.

[57] Oktay Ureten and Nur Serinken, "Wireless security through rf fingerprinting," *Canadian Journal of Electrical and Computer Engineering*, vol. 32, no. 1, pp. 27–33, 2007.

[58] Randall W Klein, Michael A Temple, and Michael J Mendenhall, "Application of wavelet-based RF fingerprinting to enhance wireless network security," *Journal of Commun. and Net.*, vol. 11, no. 6, pp. 544–555, 2009.

[59] Ben D Fulcher and Nick S Jones, "Highly comparative feature-based time-series classification," *IEEE Trans. on Knowledge and Data Engineering*, vol. 26, no. 12, pp. 3026–3037, 2014.

[60] Xiaoyue Wang, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 275–309, 2013.

[61] Jeyanthi Hall, Michel Barbeau, and Evangelos Kranakis, "Detection of transient in radio frequency fingerprinting using signal phase," *Wireless and Optical Commun.*, pp. 13–18, 2003.

[62] Jiawei Han, Micheline Kamber, and Jian Pei, "Data mining concepts and techniques third edition," *The Morgan Kaufmann Series in Data Management Systems*, vol. 5, no. 4, pp. 83–124, 2011.

[63] Christos Faloutsos, Mudumbai Ranganathan, and Yannis Manolopoulos, "Fast subsequence matching in time-series databases," *Acm Sigmod Record*, vol. 23, no. 2, pp. 419–429, 1994.

[64] Donald J Berndt and James Clifford, "Using dynamic time warping to find patterns in time series.," in *KDD workshop.* Seattle, WA, USA:, 1994, vol. 10, pp. 359–370.

[65] Chotirat Ann Ratanamahatana and Eamonn Keogh, "Three myths about dynamic time warping data mining," in *Proceedings of the 2005 SIAM international conference on data mining.* SIAM, 2005, pp. 506–510.

[66] Michail Vlachos, George Kollios, and Dimitrios Gunopulos, "Discovering similar multidimensional trajectories," in *Proceedings 18th international conference on data engineering.* IEEE, 2002, pp. 673–684.

[67] Chris Ding and Hanchuan Peng, "Minimum redundancy feature selection from microarray gene expression data," *Journal of bioinformatics and computational biology*, vol. 3, no. 02, pp. 185–205, 2005.

[68] Stéphane Mallat, *A wavelet tour of signal processing*, Elsevier, 1999.

[69] Paul S Adisson, "The illustrated wavelet transform handbook: Introductory theory and applications in science, engineering, medicine and finance," 2002.

[70] Olivier Rioul and Martin Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Mag.*, vol. 8, no. 4, pp. 14–38, 1991.

[71] Robert X Gao and Ruqiang Yan, *Wavelets: Theory and applications for manufacturing*, Springer Science & Business Media, 2010.

[72] Xiaojin Jerry Zhu, "Semi-supervised learning literature survey," 2005.

[73] Aurélien Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*, O'Reilly Media, 2019.

[74] Chapelle Olivier, Schölkopf Bernhard, and Zien Alexander, *Semi-Supervised Learning.*, Adaptive Computation and Machine Learning. The MIT Press, 2006.

[75] Zhiyuan Shi, Minmin Huang, Caidan Zhao, Lianfen Huang, Xiaojiang Du, and Yifeng Zhao, "Detection of lssuav using hash fingerprint based svdd," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–5.

[76] Yan-Yan Song and LU Ying, "Decision tree methods: applications for classification and prediction," *Shanghai archives of psychiatry*, vol. 27, no. 2, pp. 130, 2015.

[77] Jessa Bekker and Jesse Davis, "Learning from positive and unlabeled data: A survey," *Machine Learning*, vol. 109, no. 4, pp. 719–760, 2020.

[78] Shehroz S Khan and Michael G Madden, "One-class classification: taxonomy of study and review of techniques," *The Knowledge Engineering Review*, vol. 29, no. 3, pp. 345–374, 2014.

[79] Alex Gammerman, Volodya Vovk, and Vladimir Vapnik, "Learning by transduction," *arXiv preprint arXiv:1301.7375*, 1998.

[80] Carolyn J Swinney and John C Woods, "Unmanned aerial vehicle operating mode classification using deep residual learning feature extraction," *Aerospace*, vol. 8, no. 3, pp. 79, 2021.

[81] Mhd Saria Allahham, Tamer Khattab, and Amr Mohamed, "Deep learning for RF-based drone detection and identification: A multi-channel 1-d convolutional neural networks approach," in *Proc. IEEE Int. Conf. on Inf., IoT, and Enabling Technologies (ICIoT)*, Doha, Qatar, Feb. 2020, pp. 112–117.

[82] Ender Ozturk, Fatih Erden, and Ismail Guvenc, "RF-based low-SNR classification of UAVs using convolutional neural networks," *arXiv preprint arXiv:2009.05519*, 2020.

[83] Nasim Soltani, Guillem Reus-Muns, Batool Salehihikouei, Jennifer Dy, Stratis Ioannidis, and Kaushik Chowdhury, "Rf fingerprinting unmanned aerial vehicles with non-standard transmitter waveforms," *IEEE Transactions on Vehicular Technology*, 2020.

[84] Sanjoy Basak, Sreeraj Rajendran, Sofie Pollin, and Bart Scheers, "Drone classification from rf fingerprints using deep residual nets," in *2021 International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE, 2021, pp. 548–555.

[85] Chengtao Xu, Bowen Chen, Yongxin Liu, Fengyu He, and Houbing Song, "Rf fingerprint measurement for detecting multiple amateur drones based on STFT and feature reduction," in *Proc. Integrated Commun. Navigation and Surveillance Conf.e (ICNS)*, Herndon, VA, USA, Sep. 2020, pp. 4G1–7.

[86] Chengtao Xu, Fengyu He, Bowen Chen, Yushan Jiang, and Houbing Song, "Adaptive rf fingerprint decomposition in micro uav detection based on machine learning," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7968–7972.

[87] Martins Ezuma, Fatih Erden, Chethan Kumar Anjinappa, Ozgur Ozdemir, and Ismail Guvenc, "Drone remote controller rf signal dataset," 2020.

[88] AERPAW, "Aerial Experimentation Research Platform for Advanced Wireless," 2020, "Accessed: 2020-10-12".

[89] MATLAB, *9.9.0.1467703 (R2020b)*, The MathWorks Inc., Natick, Massachusetts, 2020.

[90] Joakim Andén and Stéphane Mallat, "Deep scattering spectrum," *IEEE Trans.on Signal Processing*, vol. 62, no. 16, pp. 4114–4128, 2014.

[91] Joan Bruna and Stéphane Mallat, "Invariant scattering convolution networks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.

[92] Stéphane Mallat, "Group invariant scattering," *Commun. on Pure and Applied Mathematics*, vol. 65, no. 10, pp. 1331–1398, 2012.

[93] Mallat Stéphane, "Understanding deep convolutional networks," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, pp. 20150203, 2016.

[94] Linlin Shen and Li Bai, "A review on gabor wavelets for face recognition," *Pattern analysis and applications*, vol. 9, no. 2, pp. 273–292, 2006.

[95] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size," *arXiv preprint arXiv:1602.07360*, 2016.

[96] Douglas M Hawkins, *Identification of outliers*, vol. 11, Springer, 1980.

[97] Rémi Domingues, Pietro Michiardi, Jérémie Barlet, and Maurizio Filippone, "A comparative evaluation of novelty detection algorithms for discrete sequences," *Artificial Intelligence Review*, pp. 1–26, 2019.

[98] Omar Alghushairy, Raed Alsini, Terence Soule, and Xiaogang Ma, "A review of local outlier factor algorithms for outlier detection in big data streams," *Big Data and Cognitive Computing*, vol. 5, no. 1, pp. 1, 2021.

[99] Varun Chandola, Arindam Banerjee, and Vipin Kumar, "Anomaly detection for discrete sequences: A survey," *IEEE transactions on knowledge and data engineering*, vol. 24, no. 5, pp. 823–839, 2010.

[100] John C Platt, John Shawe-Taylor, Alex J Smola, Robert C Williamson, et al., "Estimating the support of a high-dimensional distribution," *Technical Report MSR-T R-99–87, Microsoft Research (MSR)*, 1999.

[101] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander, "LOF: identifying density-based local outliers," in *Proc. ACM SIGMOD Management of Data*, 2000, pp. 93–104.

[102] Ian F Akyildiz, Brandon F Lo, and Ravikumar Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical communication*, vol. 4, no. 1, pp. 40–62, 2011.

[103] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The elements of statistical learning: data mining, inference, and prediction*, Springer Science & Business Media, 2009.

[104] Mark A Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.

[105] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.," *Journal of machine learning research*, vol. 11, no. 12, 2010.

[106] David Charte, Francisco Charte, Salvador García, María J del Jesus, and Francisco Herrera, "A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines," *Information Fusion*, vol. 44, pp. 78–96, 2018.

[107] Herbert Robbins and Sutton Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[108] John Duchi, Elad Hazan, and Yoram Singer, "Adaptive subgradient methods for online learning and stochastic optimization.," *Journal of machine learning research*, vol. 12, no. 7, 2011.

[109] Tijmen Tieleman and Geoffrey Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[110] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[111] Carlos N Silla and Alex A Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery*, vol. 22, no. 1, pp. 31–72, 2011.

[112] Feihong Wu, Jun Zhang, and Vasant Honavar, "Learning classifiers using hierarchically structured class taxonomies," in *International symposium on abstraction, reformulation, and approximation.* Springer, 2005, pp. 313–320.

[113] Alex Freitas and André Carvalho, "A tutorial on hierarchical classification with applications in bioinformatics," *Research and trends in data mining technologies and applications*, pp. 175–208, 2007.

[114] Aixin Sun and Ee-Peng Lim, "Hierarchical text classification and evaluation," in *Proceedings 2001 IEEE International Conference on Data Mining.* IEEE, 2001, pp. 521–528.

[115] Tianqi Chen and Carlos Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. Int. Conf. on Knowl. Discovery and Data Mining (ACM SIGKDD)*, 2016, pp. 785–794.

[116] Norden E Huang, Zheng Shen, Steven R Long, Manli C Wu, Hsing H Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *In Proc. of the Royal Society of London. Series A: Mathematical, Physical and Eng. Sci.*, vol. 454, no. 1971, pp. 903–995, 1998.

[117] Donghoh Kim and Hee-Seok Oh, ".EMD: a package for empirical mode decomposition and hilbert spectrum," *The R Journal*, vol. 1, no. 1, pp. 40–46, 2009.

[118] Angela Stallone, Antonio Cicone, and Massimo Materassi, "New insights and best practices for the successful use of empirical mode decomposition, iterative filtering and derived algorithms," *Scientific reports*, vol. 10, no. 1, pp. 1–15, 2020.

[119] Svetlana Kiritchenko, Stan Matwin, Richard Nock, and A Fazel Famili, “Learning and evaluation in the presence of class hierarchies: Application to text categorization,” in *Proc. Conf. of the Canadian Soc. for Computational Studies of Intelligence*. Springer, 2006, pp. 395–406.

CURRICULUM VITAE

NAME: Olusiji Oloruntobi Medaiyese

ADDRESS:
Computer Science and Engineering Department
J.B Speed School of Engineering
University of Louisville
Louisville, KY 40292
United States of America.

EDUCATION:
Ph.D., Computer Science & Engineering, August 2021
University of Louisville, *Louisville, KY, USA.*
M.Sc. in Computer and Systems Engineering, June 2018
Tallinn University of Technology, *Tallinn, Estonia*
B.Eng in Electrical and Electronics Engineering, July 2012
University of Ilorin, *Kwara, Nigeria*

WORK EXPERIENCE:
Graduate Assistant, *Aerial Robotics Lab, University of Louisville*, KY, USA, 2018-2021
Data Scientist Intern, *Honeywell Inc.*, Northford, CT, USA, 2020
Data Scientist (Hardware Analytics) Intern, *Facebook*, Menlo park, CA, USA, 2020
Data Scientist Intern, *Eltoro Inc.*, Louisville, KY, USA, 2020

PUBLICATIONS:

JOURNAL

1. **Olusiji O Medaiyese**, Martins Ezuma, Adrian .P Lauf, Ismail Guvenc. Wavelet Transform Analytics for RF-Based UAV Detection and Identification System using Machine Learning. **(under-review at Elsevier, Pervasive and Mobile Computing).**

2. **Olusiji O Medaiyese**, Martins Ezuma, Ayodeji .A. Adeniran, Adrian .P Lauf, Ismail Guvenc. Hierarchical Learning Framework for UAV Detection and Identification **(under-review at IEEE Trans. Veh. Tech.).**

CONFERENCE

1. **Olusiji O Medaiyese**, Martins Ezuma, Adrian .P Lauf, Ismail Guvenc. Semi-supervised Learning Framework for UAV Detection. **(accepted at IEEE PIMRC 2021).**

2. **Olusiji O Medaiyese**, Abbas Syed, Adrian .P Lauf. Machine Learning Framework for RF-Based Drone Detection and Identification System.**(under review).**

3. Christian Nwachioma, Martins Ezuma, **Olusiji O Medaiyese**. FPGA prototyping of synchronized chaotic map for UAV secure communication. IEEE Aerospace Conference, USA, March, 2021.

4. **Olusiji O Medaiyese**, Adrian .P Lauf. Machine Learning Based Adaptive Link Quality Prediction for Robot Network in Dynamic Environment. In Proc. Int. Symposium on Robotic and Sensors Environments (ROSE), Ottawa, Canada, June, 2019.

5. Raimund Ubar, Adeboye Stephen Oyeniran, **Olusiji O Medaiyese**. Minimization of the High-Level Fault Model for Microprocessor Control Parts. In Proc. Biennial Baltic Electronics Conference (BEC), Tallinn, Estonia, October, 2018.