

Bangor University

DOCTOR OF PHILOSOPHY

Tracking and Predicting Bee Behaviour and Movement Using Machine Learning

Morton Williams, Samuel

Award date:
2024

Awarding institution:
Bangor University

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



PRIFYSGOL
BANGOR
UNIVERSITY

School of Computer Science and Electronic Engineering

College of Environmental Sciences and Engineering

**Tracking and Predicting Bee Behaviour and
Movement Using Machine Learning**

Samuel Morton Williams

Submitted in partial satisfaction of the requirements for the

Degree of Doctor of Philosophy

in Computer Science

Declaration and Consent

I hereby declare that this thesis is the result of my own investigations, except where otherwise stated. All other sources are acknowledged by bibliographic references. This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree unless, as agreed by the University, for approved dual awards.

Yr wyf drwy hyn yn datgan mai canlyniad fy ymchwil fy hun yw'r thesis hwn, ac eithrio lle nodir yn wahanol. Caiff ffynonellau eraill eu cydnabod gan droednodiadau yn rhoi cyfeiriadau eglur. Nid yw sylwedd y gwaith hwn wedi cael ei dderbyn o'r blaen ar gyfer unrhyw radd, ac nid yw'n cael ei gyflwyno ar yr un pryd mewn ymgeisiaeth am unrhyw radd oni bai ei fod, fel y cytunwyd gan y Brifysgol, am gymwysterau deuol cymeradwy.

Statement of Originality

The work presented in this thesis is entirely from the studies of the individual student, except where otherwise stated. Where derivations are presented and the origin of the work is either wholly or in part from other sources, then the full reference is given to the original author. This work has not been presented previously for any degree, nor is it at present under consideration by any other degree awarding body.

Signed:

Statement of Availability

I hereby acknowledge the availability of any part of this thesis for viewing, photocopying or incorporation into future studies, providing that full reference is given to the origins of any information contained herein. I further give permission for a copy of this work to be deposited with the Bangor University Institutional Digital Repository, the British Library ETHOS system, and/or in any other repository authorised for use by Bangor University when the approved bar on access has been lifted. I acknowledge that Bangor University may make the title and a summary of this thesis/dissertation freely available.

Signed:

Abstract

This study sought to understand how machine learning could facilitate the tracking of bees and address the lack of automated bee-tracking tools that can count bee behaviour and movement from positional information. Harmonic radar tracking datasets of bumblebee movement were used to predict flight tasks. Random forest (RF), Support Vector Machine (SVM) and neural network (NN) algorithms were trained on the dataset and their performance was evaluated. Unsupervised clustering (lacking any human labels) was performed to investigate whether a simple binary classification of bee tasks (foraging or exploring) could be replaced with a multiclassification model with more complex behaviour modelling. Comparisons of optical and thermal camera systems were also undertaken and found that thermal cameras, which are less affected by sub-optimal lighting, are more suitable for automated bee flight characterisation, (inwards, outwards, and hovering) at the entrance to the hive. Gaussian mixture models and a Kalman filter were used to extract bee flights from recordings from bee hives. The recorded flights were pre-processed into a dataset to train SVM, RF, and NN algorithms to predict the three flight types. Finally, a Doppler radar was used to record bee entry and exit activity from a hive. The data was processed using Log Area Ratios, derived from Linear Prediction Coefficients, to create a dataset for training machine models. The goal was to create a system that could function in real-time using a Raspberry Pi to classify the activity at the entrance of a hive. This study demonstrates that machine learning could automate a data-intensive field of study and provide meaningful insights into the activities of bee species with uses in the apicultural sector, including research and conservation.

Acknowledgements

I would like to thank my supervisors Dr. Paul Cross and Dr. Cris Palego for their insightful feedback throughout the study.

The project was funded by Knowledge Economy Skills Scholarships (KESS 2) and the Welsh Government. I am grateful for their funding and support which allowed me to develop my research.

I extend special gratitude to my colleagues Dr. Nawaf Aldabashi, Sara Bariselli, and Dr. Jake Shearwood for the opportunity to work together and co-develop projects. Their humour, wit, and intelligence made my experience enjoyable even during long dark hours.

Finally, I would like to thank my whole family for their support. Their belief in me has kept my spirits high during this process. I thank especially my daughter Mia and fiancée Lucy for their unwavering patience and for being a constant source of encouragement and motivation.

Contents

Declaration and Consent.....	2
Statement of Originality	3
Abstract	4
Acknowledgements	5
1 Introduction, Motivation and Structure.....	11
1.1 Background and Motivation.....	11
1.2 Research Questions.....	16
1.3 Aim, Objectives, and Structure of Thesis	16
1.4 Contributions	17
1.5 Structure of the Thesis	19
2 Literature Review with Experimental and Computational Methods	21
2.1 Introduction	21
2.2 Literature Review	24
2.2.1 Manual Tracking	24
2.2.2 Harmonic Radar	26
2.2.3 Radio-Frequency Identification	28
2.2.4 Radio Telemetry	29
2.2.5 Battery-less Tag Tracking Technologies.....	33
2.2.6 The Costs of Tagged Technologies.....	35
2.2.7 Doppler Radar	37

2.2.8	Internal Hive Predictions.....	39
2.2.9	RF-Based Machine Learning.....	44
2.2.10	Visual, Thermal, and Light-Based Techniques	47
2.2.11	Time-series prediction	57
2.3	Machine Learning Algorithms.....	59
2.3.1	Neural Networks	59
2.3.2	Support Vector Machines	63
2.3.3	Random Forests	66
2.3.4	Cross Validation Bound Bayesian Hyperparameter Optimisation.....	70
2.3.5	Time Series Analysis	76
2.3.6	Performance Evaluation and Metrics.....	76
2.4	Video Processing Techniques.....	80
2.4.1	Gaussian Mixture Models for Background Subtraction.....	80
2.4.2	Kalman Filter for Movement Prediction.....	83
2.5	Signal Processing Techniques.....	86
2.5.1	Linear Predictive Coding.....	86
2.5.2	Mel-Frequency Cepstral Coefficients.....	90
2.6	Munkres Assignment Algorithm.....	91
2.7	Summary and Discussion	93
3	Early Prediction of Bumblebee Flight Task*	95
3.1	Introduction	95

3.2	Data	97
3.3	Filtering and Preparation Methods	100
3.4	Initial Experiment.....	103
3.5	Initial Results and Discussion.....	107
3.6	The Dangers of Subsampling, Manual Tuning, and the Question of Loss	116
3.7	Summary, Limits and Conclusion.....	119
4	A Comparison of Machine-Learning Assisted Optical and Thermal Camera Systems for Beehive Activity Counting*	122
4.1	Introduction	122
4.2	Materials and Methods.....	124
4.3	Results and Discussion.....	133
4.4	Feature Importance.....	138
4.5	Wasp Detection.....	140
4.6	Test Stage.....	141
4.7	Summary, Limits and Conclusion.....	143
5	Challenges in Developing a Real-time Bee Counting Radar using Machine Learning*	146
5.1	Introduction	146
5.2	Preliminary Work.....	147
5.3	Methods	151
5.4	Results	156
5.4.1	Preliminary Results.....	156

5.4.2	Exploring the Weaker Results	159
5.4.3	Testing Stage	170
5.5	Discussion.....	171
5.6	Summary and Conclusion	173
6	Summary, Future Work, and Conclusion	175
6.1	Discussion and Limitations.....	175
6.2	Future Work	177
6.3	Review of Research Questions	179
6.4	Review of Research Aims and Objectives	180
6.5	Conclusion	181
7	References.....	182
	Appendices	214
	A - Localization and Tracking Bees Using a Battery-less Transmitter and an Autonomous Unmanned Aerial Vehicle*	214
1.	Introduction.....	214
2.	Miniature Self-powered Radio Telemetry Tag	215
3.	Phased Array and Feeding Network.....	217
4.	Autonomous Tracking.....	219
5.	Discussion	223
6.	Conclusion.....	224
	B - Integration of 5.8GHz Doppler Radar and Machine Learning for Automated Honeybee Hive Surveillance and Logging*	225

1.	Introduction.....	225
2.	5.8 Ghz Doppler Radar.....	226
3.	Honeybee RCS Approximation.....	227
4.	Calibration Procedure.....	228
5.	Free Flying Honeybee RCS.....	229
6.	Doppler Radar Integration with Machine Learning.....	230
7.	Discussion	232
8.	Conclusion.....	232

C - A Machine Learning Integrated 5.8-GHz Continuous-Wave Radar for Honeybee Monitoring and Behavior Classification* 233

1.	Introduction.....	233
2.	5.8-GHz CW Radar Design	235
3.	Predicted Versus Measured Radar Signatures.....	237
4.	RCS and Range Increase Using Silver Coating	245
5.	Machine Learning	249
6.	Conclusion.....	258

1 Introduction, Motivation and Structure

1.1 Background and Motivation

Bees are major pollinators of crops and wild plants in our ecosystems. The current and projected decline in insect populations is the consensus among recent scientific studies [1–3]. A study in the UK found that two-thirds of bumblebee species had range losses between 1960 and 1980 [4]. In America, it has been shown that half of the species modelled with unlimited dispersal patterns are projected to lose ranges [5].

The importance of animal pollination to human crop production is apparent. By crop type, over 75% of crops rely on animal pollination whilst by volume 35% of global production also rely on animal pollinators [6]. In the wild, 80% non-agricultural plants rely on insect pollinators [7]. Wild- and honeybees contribute more than \$2900 per hectare to the production of insect-pollinated crops [8].

In addition to their unmistakable contribution to food production and natural biodiversity, bees are crucial for long-term sustainable goals. Bees potentially contribute to 15 of the 17 United Nations Sustainable Development Goals (SDGs) and a minimum of 30 SDG targets [9].

Large-scale studies have been undertaken to find broad overlapping trends in these instabilities to provide mitigation advice [10]. However, at a local level, it is difficult to pinpoint causes. Figure 1.1 shows a collated view of the global decline of varying insect species, capturing the threat faced by pollinators.

Focus on sudden insect biomass and species richness decline is now a trend, often citing causes such as climate change, pesticides, habitat loss, and invasive species [11, 12]. Studies focused on a particular group of insects are often inconclusive about the precise causes, due to the limited nature of current monitoring techniques [3]. Precise, detailed information captured by better sensor equipment can provide

more data to work with when forming conclusions, particularly to investigate overlapping issues for colonies.

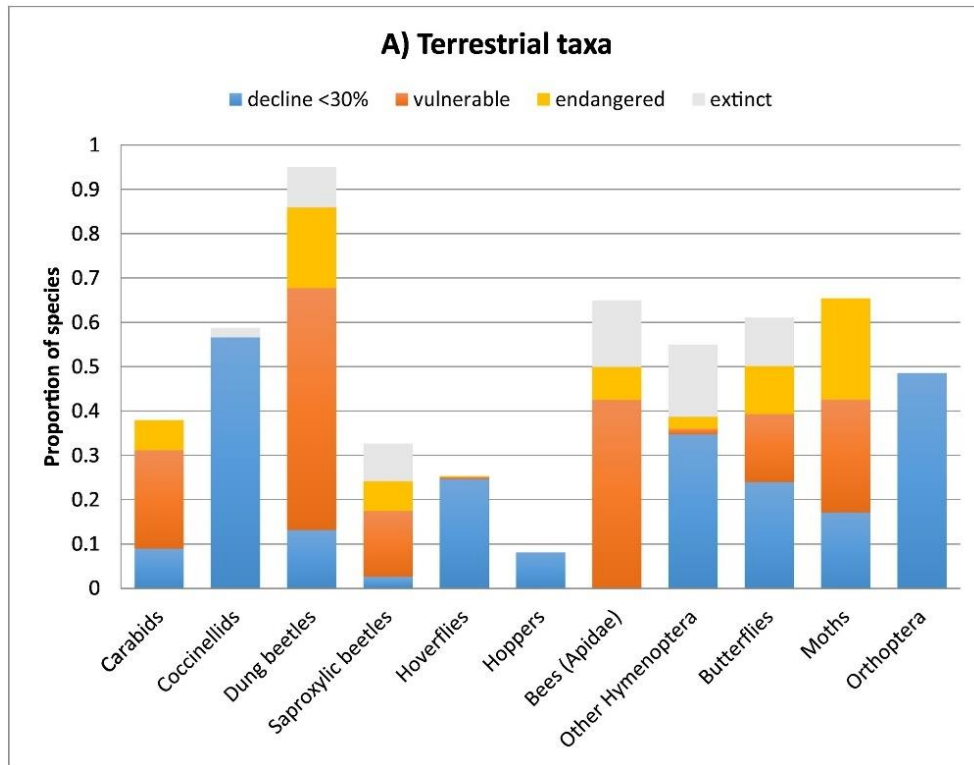


Figure 1.1 Proportion of threatened and extinct insect species covering a global 40-year period as collated from 73 reports [10].

Pesticides are a major factor in the decline of bees. In 2018, the European Union (EU) banned the use of three key neonicotinoids used in agriculture as pesticides [13]. These three chemicals were found to have drastic sub-lethal effects at typical exposure for bees [14]. Their replacements, acetamiprid and thiacloprid, have been found by the EU to bring minimal disadvantages to bees. However, studies have found evidence to the contrary for both [15]. Thiacloprid, in particular, is unique in that there is evidence that disease-bearing bees are affected differently from healthy bees [16].

Intensified farming modifies the landscape that both managed honeybees and wild bees inhabit. This is a key factor in the decline of populations over the last 50 years

[17]. The exact effect of monocultures, brought about by farming, on bees is difficult to measure. Studies have shown that wild bee species richness and abundance are positively affected by landscape complexity when insecticides are in use [18]. However, another study found that honeybees kept at diversified farms had increased colony weight and preoverwintering nutritional state, but suffered to the same degree as those kept on monoculture farms during late summer. This is a result of the stressors of living in low-diversity environments that poorly replicate natural ecosystems [19].

Increased urbanization creates challenges for bumblebees. When studying bumble bee species richness and abundance in allotment gardens, Ahrné et al. found overall decrease in abundance closer to the heart of Stockholm than in periurban areas, as well as a decrease in the number of species. This may indicate that changes to the environment are beyond the abilities of some species to adapt [20].

During the past 50 years, the invasive spread of the *Varroa destructor* mite has resulted in the death of millions of honeybee colonies [21]. *Varroa* mites cause minimal damage to their co-evolved ancestral host (*Apis cerana*) but they are lethal for their new host, the western honeybee. The debate as to whether western honeybees can adapt quickly enough to the new parasite, even with breeding programs, is not settled [22]. The dangers of an invasive species, which shows signs of rapid evolution itself, cannot be discounted.

Human activity modifies the environment and allows invasive species such as the Asian Hornet to invade and impair the population stability of native species [23]. The instability of insect species richness and biomass has secondary effects expected to undermine human industry, particularly food production [6, 24].

The Asian hornet invasion of Europe has been a centre point of research since the initial wave in 2004, located in Southern France [25]. Studies have been able to pinpoint their habits and preferred nesting sites. Hornet nests can be classified as primary and secondary nests. Primary nests are those that the foundress queen hibernates in over winter. From here, she nurtures a few critical workers until a stable colony is present to support her. Then, if the primary nest no longer suits the need, a secondary nest might be built in a place more suited to a larger colony.

76.9% of primary nests in Southern France were found in man-made structures. Conversely, 73.5% of secondary nests were found in trees, as discovered by Franklin et al. [26]. However, whether these data represent a fundamental pattern is undetermined. The authors note that primary nests are more likely to be found in urban areas due to alarm caused by layman's observations. In contrast, once a primary nest is found, investigations are undertaken to locate any associated secondary nests. However, it was a conclusion of this work that only 51.4% of nests were found and only 37.4% of nests were found while active.

The act of hornets hovering outside of bee colonies and predating is known as hawking. Asian hornets are three times more successful at hawking European bees than Asian bees, in part due to the lack of behavioural adjustments necessary to combat predation such as altered foraging and extra guards at the entrance to hives [27]. Also of note is the lack of evolved defensive tactics such as heat-balling, whereby bees cook an aggressive hornet by balling it up and vibrating until the temperature kills the hornet [28]. Similarly, European bees lack the behaviour of wing-shimmering, where the hive gathers to pulse their wings and disorient attackers [29].

The lack of defensive capabilities of endemic bees, as well as the lack of scientific ability to track down Asian Hornet nests, are exemplars of why more focused studies are needed. Mitigating technologies justifiably must be developed. This creates an opening for new technologies and algorithms to pinpoint exact causes with fully contextualised results, aimed at protecting species and monitoring risks.

There are many more drivers for bee population instability. Further context is provided by the work of Sanchez-Bayo, as demonstrated in Figure 1.2, showing the associated factors. This broad-view outlook is instrumental in spurring further research. However, given the number of potential causes, this data is less useful to a small-scale farmer or larger business interested in maximising pollination efficiency and preserving hives in place. In this case, precise threats need to be identified and addressed in real-time. Similarly, the most severe causes for decline referenced in these works are man-made. Agriculture, pesticides, and urbanisation can be limited

but may be an inevitable side effect of human population growth. Therefore, the best approach may be specialised mitigation tactics to preserve local species.

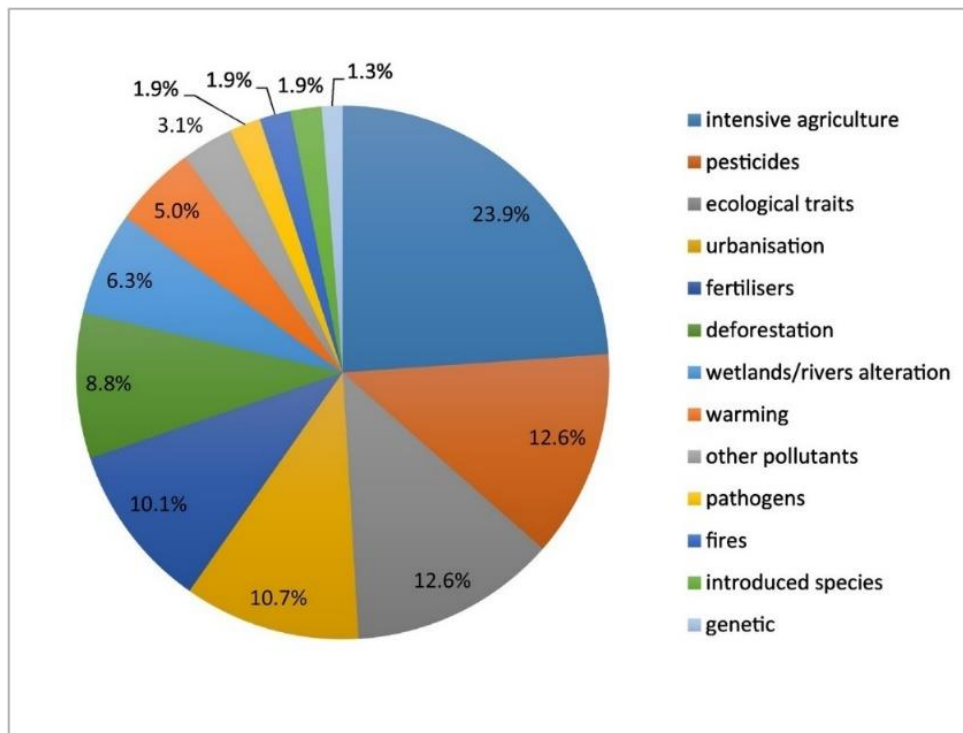


Figure 1.2 Associated factors of the decline of insect species, gathered in collated literature [10]. Clockwise segments match, in order, the list of causes.

If effective methods to monitor and predict insect behaviour are to be developed, the means to contextualise complex and technical information must be automated. This thesis will begin by providing an overview of some of the technologies that have been created to monitor bees and similar insects. The technical complexity of understanding the volumes of data that come from sensor equipment will be the justification that machine learning can provide ways of rapidly expanding the scope of technologies that currently face limits.

Three experiments have been undertaken that show how machine learning can enhance the ability to existing and in-development technologies to better track and predict insect movement. Using harmonic radar, the task a bee is undertaking upon leaving the hive has for the first time been accurately predicted, without the need for the full flight to be recorded. Significant progress has been made towards a real-

time, radar-based activity counter for the entrance of beehives to monitor detailed activities. Finally, the use of a thermal camera has been tested to count accurately the departures and arrivals of bees at the entrance of the hive.

1.2 Research Questions

This project investigates bee tracking technologies by combining them with machine learning approaches to determine how viable such combinations are for predicting and understanding bee behaviour.

The specific questions include:

- Which bee tracking technologies are most useful to use when paired with machine learning?
- Are existing model architectures capable of predicting bee behaviour as captured by these technologies?
- What are the limits of machine learning in this field?

The motivation is that there is a lack of machine learning usage to count and predict bee behaviour. Addressing this gap has the potential to create models and technologies that would aid in the research of bee population instability as well as bring advantages to industries that rely on bee pollination.

1.3 Aim, Objectives, and Structure of Thesis

The core aim of this thesis is to understand and report on the contribution that machine learning can make toward tracking and understanding flying insect behaviour.

This is encapsulated by a central set of objectives:

- Identify key tracking technologies with the greatest potential to integrate with machine learning. Two of these technologies were chosen as they were at the forefront of research at Bangor University and critically placed to help track bees.
- Design experimental setups to gather bee movement and/or behaviour data using these technologies.
- Generate suitable machine learning models to predict movement and behaviour using the acquired data.
- Evaluate the strengths and weaknesses of the models created (and their experimental setups) identifying the sources of any limitations. The focus is on the accuracy the models have in predicting bee movement and behaviour, rather than targeting specific technologies that may decrease cost, limit weight, or improve visualisation.
- Of the three technologies, provide informed discussion about which might benefit most or least from the machine learning integration.
- Discuss whether the experiments show support for using machine learning to aid in the design of future bee counting and modelling systems.

1.4 Contributions

This thesis is concerned with the development of systems to predict bee movement and behaviour, and as such the main contribution of this thesis is the design, measurement, and analyses of three prototypes to classify such behaviour.

The details of the contributions in this thesis are listed below:

- Data analysis, software development, and write-up for the development of a system to predict the function of a bumblebee's flight as it leaves the nest. This work is discussed in Chapter 3 and was also presented in the paper: S.M. Williams, N. Aldabashi, C. Palego, J.L. Woodgate, J.C. Makinson, P.

Cross, Early prediction of bumblebee flight task using machine learning, *Computers and Electronics in Agriculture*, Volume 184, 2021, 106065, ISSN 0168-1699, doi: 10.1016/j.compag.2021.106065

- Experimental design, data analysis, software development, and write-up for a comparison of two candidate camera systems to count honeybee activity at the entrance of their beehive. This work is presented in Chapter 4 and was also discussed in the paper: S.M. Williams, S. Bariselli, C. Palego, R. Holland, P. Cross, A comparison of machine-learning assisted optical and thermal camera systems for beehive activity counting, *Smart Agricultural Technology*, Volume 2, 2022, 100038, ISSN 2772-3755, doi: 10.1016/j.atech.2022.100038
- Experimental design, data analysis, software development, and write-up for a radar-based machine learning system to count activity near a beehive, along with detailed discussion on the current limits of creating such a system. This is present in Chapter 5 and in the paper: S. M. Williams, N. Aldabashi, P. Cross, C. Palego. Challenges in Developing a Real-Time Bee-Counting Radar. *Sensors* 2023, 23, Volume 11, Article 5250. doi: 10.3390/s2311525

In addition, some secondary contributions were achieved in other related projects as follows:

- Software development and data analysis for building self-piloting drone software to integrate with a novel localization system for battery-less bee transmitters, detailed in the paper: J. Shearwood, S Williams, N Aldabashi, P Cross, B M Freitas, C Zhang and C Palego., "Localization and Tracking Bees Using a Battery-less Transmitter and an Autonomous Unmanned Aerial Vehicle," *2020 IEEE/MTT-S International Microwave Symposium (IMS)*, Los Angeles, CA, USA, 2020, pp. 1263-1266, doi: 10.1109/IMS30576.2020.9223950
- Write-up, data analysis, and machine learning models to predict bee activity from radar signatures gathered from earlier prototypes of the radar used in Chapter 5, as discussed in two papers:

- N. Aldabashi, S. Williams, A. Eltokhy, E. Palmer, P. Cross and C. Palego, "Integration of 5.8GHz Doppler Radar and Machine Learning for Automated Honeybee Hive Surveillance and Logging," 2021 *IEEE MTT-S International Microwave Symposium (IMS)*, Atlanta, GA, USA, 2021, pp. 625-628, doi: 10.1109/IMS19712.2021.9574826
- Aldabashi, N., Williams, S.M., Eltokhy, A., Palmer, E., Cross, P. and Palego, C. (2023). A Machine Learning Integrated 5.8-GHz Continuous-Wave Radar for Honeybee Monitoring and Behavior Classification. *IEEE Transactions on Microwave Theory and Techniques*, [online] pp.1–11. doi: 10.1109/TMTT.2023.3248785
- Data collection and analysis for a paper on the biometric uses of the radar board featured in the thesis: N. Aldabashi, S. M. Williams, P. Cross and C. Palego, "A Printed Circuit Board Continuous Wave Doppler Radar for Machine Learning-Enhanced Biometrics," 2021 *IEEE MTT-S International Microwave and RF Conference (IMARC)*, KANPUR, India, 2021, pp. 1-4, doi:10.1109/IMaRC49196.2021.9714590

1.5 Structure of the Thesis

The subsequent content of the thesis will be ordered as below:

Chapter 2 (Literature and Methods): Literature Review of the current tracking systems for bees and similar insects. In addition, technologies related to tracking systems that instead monitor their behaviour. This is followed by an investigation into machine learning work undertaken using bee-tracking technologies most similar to the work undertaken in the thesis.

Following the literature review is a discussion of current methods, detailing algorithms employed across all experimental chapters. Experimental and

Computation methods are employed to record data, extract pertinent features, create and train models, and run assessments on the models and results generated.

Chapter 3 (Experimental): A chapter attempting to answer whether machine learning could predict the function of individual bumblebee flights, using positional coordinates gathered by harmonic radar, within a short time window of it leaving the nest.

Chapter 4 (Experimental): A comparison to determine whether a thermal camera might, using machine learning, perform comparably to a similar visible-spectrum system to count honeybee hive activity. Machine learning models were used to label flights, including partially recorded, broken, or double-counted flights as either outward or hovering flights.

Chapter 5 (Experimental): A chapter to determine the feasibility of developing a machine-learning integrated radar system to count the traffic at the entrance of a honeybee hive, using Support Vector Machines (SVMs) trained on Linear Predictive Coding (LPC) representations of radar signals.

Chapter 6: Discussion and limitations, future works, and conclusion.

Chapter 7: References.

Chapter 8: Appendix of published works with author contributions during the study period.

2 Literature Review with Experimental and Computational Methods

2.1 Introduction

This chapter contains both a literature review and a methodology section. The purpose of the literature review is to provide context for why the work in this thesis is both novel and useful.

The literature review contains two themes with no clear division. The first theme covers technologies that are used to track bees (or similar animals) regardless of whether they have been combined with machine learning. The second theme concerns systems that have used machine learning to predict, monitor, or count small insects and similar animals.

There is no division between the two themes for the following reasons:

- Some machine learning works have used general-purpose technologies not specific to insects such as video cameras. These technologies are commonplace and therefore are not discussed outside of the study in which they feature.
- Some machine learning studies use highly specific tracking systems that are only applicable (or have only been used) in the research that is being referenced. These are not discussed elsewhere as they are self-contained within the study.
- Some technologies have yet to be integrated with machine learning. It is important to discuss these as they represent gaps within the literature while also providing a background on which technologies have had the most impact.

A visual aid for understanding this chapter and its themes is given Figure 2.1.

When tracking insects there are many potential measures suitable for benchmarking success. Examples include weight, price, range, battery, accuracy, and risk to the insects. This thesis prioritises accuracy as it is a foundational concern for machine learning. However, it is important to be aware of the other concerns when tracking insects, particularly when it comes to risk. The literature review contains a section on the harm caused by tracking systems which use tags for this purpose.

There is no one tracking technology which can be integrated with machine learning which minimises the risk to insects while maximising the other metrics. The literature review aims to cover technologies that may compete with or complement those used in this thesis.

The second part of this chapter covers a methodology section which provides a discussion on the algorithms used to conduct the research within this thesis. This discussion covers how the algorithms work as well as where and for what purpose they have been used within the thesis. More specific implementation details are contained in the main chapters that use the methods; however, the methodology covers common implementation concerns and practices.

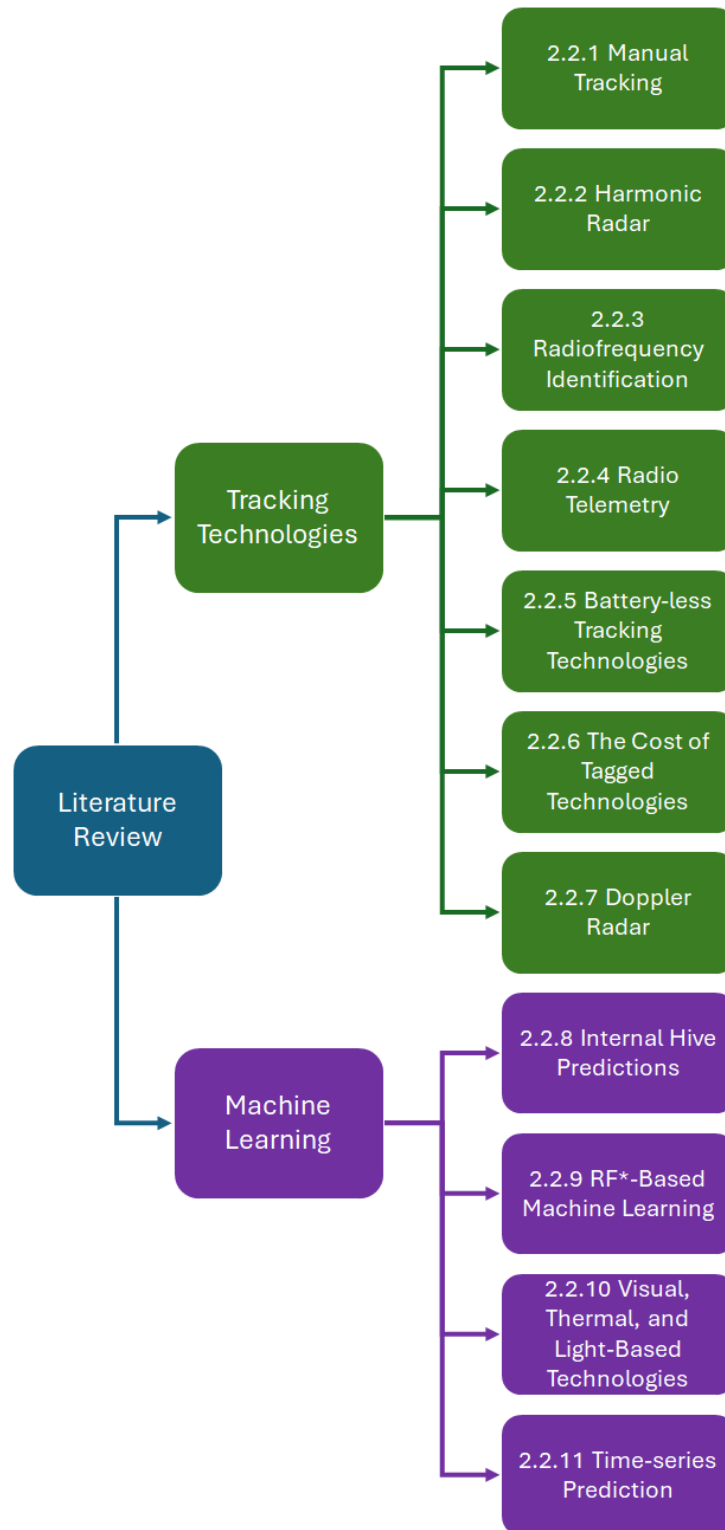


Figure 2.1 A thematic diagram of the literature review. RF in this instance refers to Radio Frequency technology.

2.2 Literature Review

2.2.1 Manual Tracking

Rudimentary tracking techniques have existed for a long time. Often, these are more efficient than more modern electronic approaches. One such technique, in China, involves baiting hornets into feeding on either meat or other insects in a managed area. Once distracted, a feather attached to a string is looped around the abdomen. The additional weight of the feather hampers the hornet as it returns home and the bright colours aid in keeping eyes on the hornet as it flies through the forest. One weakness of this approach, found during attempted replication by Western researchers, is that the hornet must not notice the feather attached or it will stop as soon as possible to remove it. In addition, the hornet is excluded from the hive on return based on the abnormality, making it not useful for repeated tracking [30].

Arguably the most well-known approach is the trap-and-release method, historically known as beelining [31, 32]. In this approach, multiple traps are placed where the tracking takes place. Bees, or hornets, are marked upon release and timed until return. This gives both direction and distance and, by combining multiple trap results, allows for a good estimate of the location to be determined. The process is visualised in Figure 2.2.

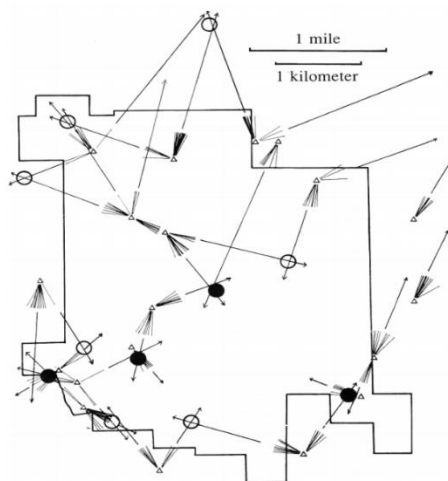


Figure 2.2 Visual representation of the beelining process: Map of the Arnot Forest showing beelining sites (triangles) and the observed bearings for bees as they departed these sites (arrows represent the mean of each cluster of bearings). The determined locations of nests are shown by circles, filled circles indicate nests which were located by following beelines back to them [32].

Traps vary significantly by the maker. An example is present in Figure 2.3, but all have some key requirements in common. The entries to the trap must be wide enough for the target to enter but not too large as to allow other insects to enter, with the exit being complicated enough that it becomes unfeasible. Additional exits can be present to allow smaller insects to leave. The addition of a gauze pad, or the use of a sponge, prevents these insects from drowning in the liquid lure. This lure can be a mixture of many fluids, such as beer, wine, honey, and blackcurrant syrup [33]. Beer and wine are used to repel bees when attempting to capture hornets [30].

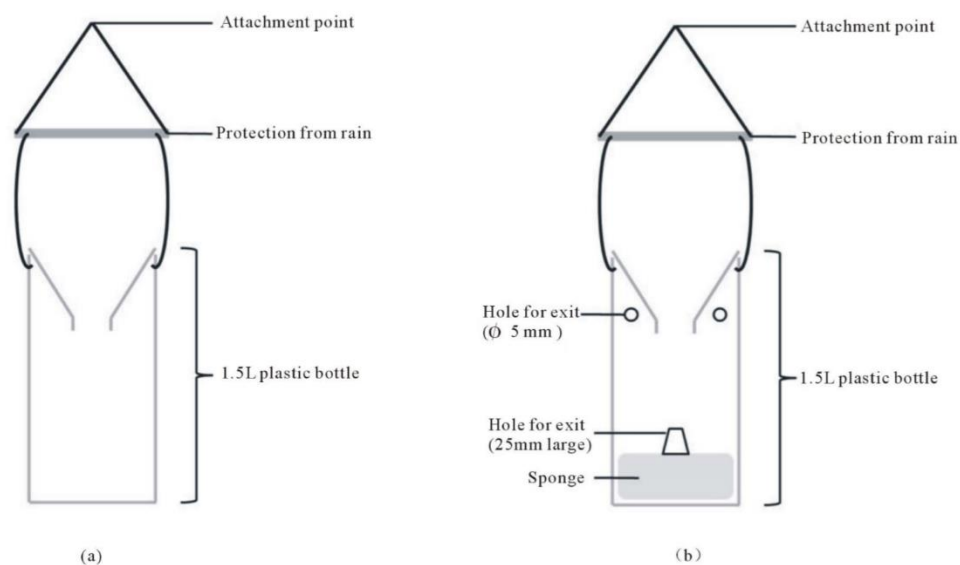


Figure 2.3 Structure of a trap designed to catch exclusively Asian hornets while releasing other insects [33].

Some negatives of these more manual techniques include limitations such as less than 1% of the catch being the Asian Hornet in Spain, severely limiting the capacity to track from traps alone [34]. Similarly, some approaches recommend killing early queens once they are found during the start of the season. This, in theory, prevents the development of primary nests and reduces the chance of secondary nests. Nevertheless, it has been found that Asian Hornets do so well in European climates that nests reach saturation quickly, leading to an overabundance of queens. These queens then fight over nests resulting in many queens dying [30]. As such, adding to the death of queens is redundant and removes the possibility of finding the nest itself.

2.2.2 Harmonic Radar

Radar has been used to study insects for over 50 years [35]. Both vehicle-mounted and handheld systems have been developed [36]. Larger, more powerful systems have a much greater range than the ~60m offered by handheld solutions, with the trade-off being much more difficult to deploy in remote areas [37].

Harmonic radar is successful due to the system broadcasting a signal that impacts a tag placed on the bee, producing a harmonic frequency. This frequency is reflected out and detected back at the receiver. Specific tag designs vary but ultimately reflect a unique signal back allowing for separation from the environment. Examples are included in Figure 2.4. Tags can weigh as little as 6-10mg [38]. There has been significant concern that the use of tags on arthropods can significantly change their behaviours, even when such tags are less than the typical loads of their pollen foraging [39].

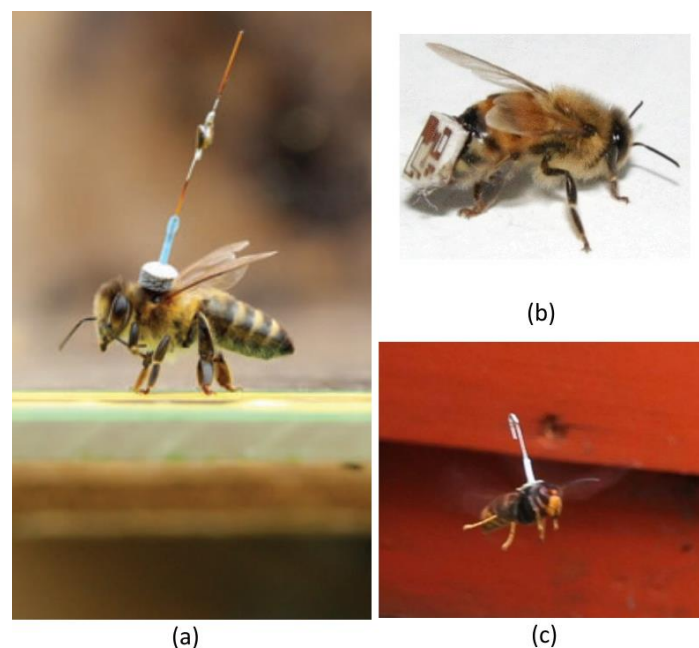


Figure 2.4 (a) A traditional antenna tag design in use for tracking bees [35], (b) A microstrip tag to reduce the burden on the bee [40], and (c) a harmonic radar tag on an Asian hornet [41].

Harmonic radar is particularly potent because it is long-range and has provided an understanding of how bees manage spatial memory. Menzel et al. investigated whether, like humans, bees can build effective pathing between learned and communicated landmarks [42]. Using harmonic radar tracking, they were able to

show that bees visiting a resource that they had learned themselves were then able to detour to a new resource that had been communicated to them via a waggle dance. This dance is used by bees to encode information about newfound resources. By detouring to that resource, they showed spatial awareness of the relative positions of two separate landmarks.

In 2014, Fischer et al. used harmonic radar and demonstrated the homing issues caused by neonicotinoids in bees [43]. 1.5 hours after exposure, bees were hampered during their return home and less likely to make the correct turn at landmarks. In a similar study during the same year, Wolf et al. showed similar homing issues in bees affected by the *Nosema ceranae* pathogen, again using harmonic radar [44].

Harmonic radar was used by Greggers et al. to show that successful swarms of bees have streakers that speed ahead of the main swarm to guide it to the location of a new nest [45]. A project spanning multiple years, spearheaded by Milanesio et al., developed a portable harmonic radar system. The system, shown in Figure 2.5, was able to detect Asian hornets up to 125m away, with the revised version capable of 150m [46, 47].



Figure 2.5 Harmonic radar for hornet tracking. The transmitting antenna is 180 cm long, while the receiving antenna, on top, is 50 cm long [46].

Harmonic radar is not without its downsides. It has been theorised that a small antenna with good efficiency and broad bandwidth is unfeasible to achieve [48]. Therefore, trade-offs must be made by sacrificing one metric to better the other two.

The terrain is very important with harmonic radar, causing instances of 'clutter' where the target is obscured [49]. These losses must be accounted for, and an attempt made to limit their effects. This can be as simple as limiting tracking to flat landscapes or increasing the height of the system so that it is above the foliage [46, 47, 50]. However, even low-to-the-ground foliage can interrupt the transmitted signal when the target stops foraging from a food source [38, 51].

Adjustments to the design of the harmonic radar itself have been made to increase the beam-width, reducing the ultimate range for the sake of wider detection capabilities [41]. These systems are effective out to 500m, which is significantly shorter than the projected 3km foraging range of these insects [36].

2.2.3 Radio-Frequency Identification

Radio-Frequency Identification (RFID) enables the unique identification of a tag wearer based on stored data [52]. The system sends out a signal which is received by the tag, and the tag then responds with a unique identification code. Tags can be powered or rely solely on the incoming signal for power needs.

The shortcoming of such systems is a limited range. While advancements have pushed detection ranges out as far as 64m, in practice the use of such systems for insect monitoring required checkpoints where the insects must cross the path of a narrow detection beam [53]. Figure 2.6 shows such a system, developed to identify bees feeding on a station. Practically, these systems are easily modified to function at the entrance of hives [54].



Figure 2.6 Example of a bee disturbed by the RFID tags. Aluminium housing (18.5 cm × 10.5 cm × 5.5 cm) and peripherals. (1) 4-pin power inlet; (2) micro-USB port; (3) Wi-Fi antenna; (4) GNSS antenna, and (5) RFID antennas and ports [55].

Due to the passive nature and small size of RFID tags (5.4mg [55]) they have led to breakthroughs with insects other than bees. Robinson et al. showed the mechanics by which ants select new nests by monitoring departures and arrivals at the candidate. Stronger candidate nests were less likely to be abandoned and something similar to a majority vote is used to determine the final site [56]. Harmonic radar has also been useful in the monitoring of the effects of neonicotinoids on beetles, as well as the study of other subterranean insects [57, 58]. Heavier tags have been used, weighing 81mg, however, these are impractical, especially for flying insects [59].

2.2.4 Radio Telemetry

Radio telemetry has a long history of use in tracking animals [60]. Early studies were limited by the technology at the time, such as using a 1cm spherical antenna ingested by Galapagos Tortoises [61]. As technology has moved forward, these tags have become smaller and more able.

Radio telemetry typically uses a battery-powered tag to increase the range of detection. This is then received by a handheld or vehicle-mounted receiver to isolate

location. Average detection distances are roughly 500m with peak detection at 1500m when the receiver is elevated above the tracking area [62].

Tags weigh as little as 200mg; however, this is still beyond the current flight capabilities of some bees and hornets. A demonstration of such a tag is found in Figure 2.7. A general rule is to limit tags to 12% of the bee's body weight [63].



Figure 2.7 (a) Transmitter attachment on a bee kept in a test tube with opened gauze where the transmitter is fixed. (b) Nectar collecting individual of bee having a transmitter attached. (c) Bee with an attached transmitter, foraging on red clover [63].

Nevertheless, recent research has shown that even such rules are misinformed or inaccurate [39]. Even a much stricter rule of 5% body mass for flying or swimming insects has been questioned. Radio telemetry tags constitute some of the heaviest tags in the field, a sacrifice for their gains, as shown in Figure 2.8. Tags can be expensive and have battery lives limited to days.

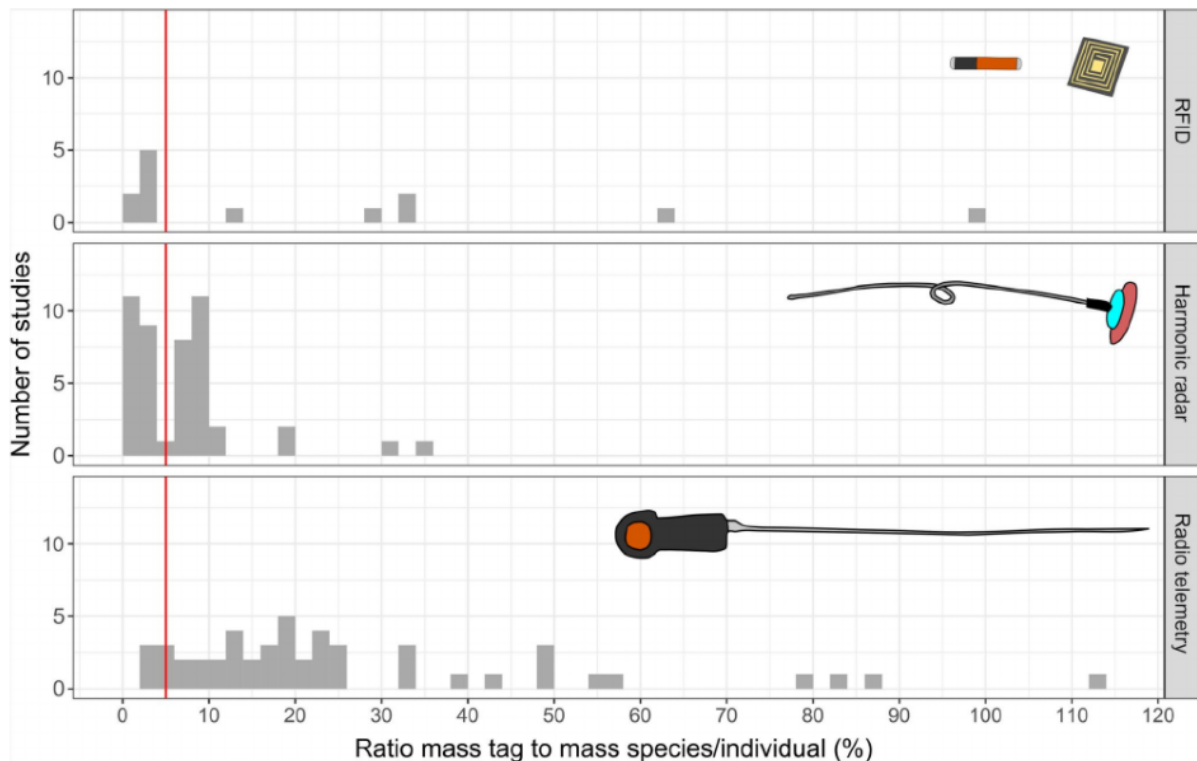


Figure 2.8 Histogram of the tag-to-body mass ratio for radio frequency identification (RFID), harmonic radar, or radio telemetry tags. The red line indicates the 5% threshold used for flying or swimming vertebrates [39].

Despite the drawbacks of radio telemetry, it has found its uses in insect tracking. Since it is simpler in principle than harmonic radar, it is cheaper to develop and typically doesn't require a large vehicle-mounted receiver [41].

It has been invaluable in tracking the ranges of various insect species in often complex terrain [64, 65]. In addition, it has contributed to the understanding of how pollinators make use of their environment space [63].

While manual tracking via radio telemetry is successful, it is limited by the manpower required. Typically, an approach very similar to that of beelining is used. The signal detected by a handheld receiver is matched to a compass bearing [66]. Several readings together can triangulate a colony or nest. A demonstration of this is present in Figure 2.9. Trained trackers must therefore carry the equipment often through difficult terrain and attempt to be consistent with one another in taking measurements.

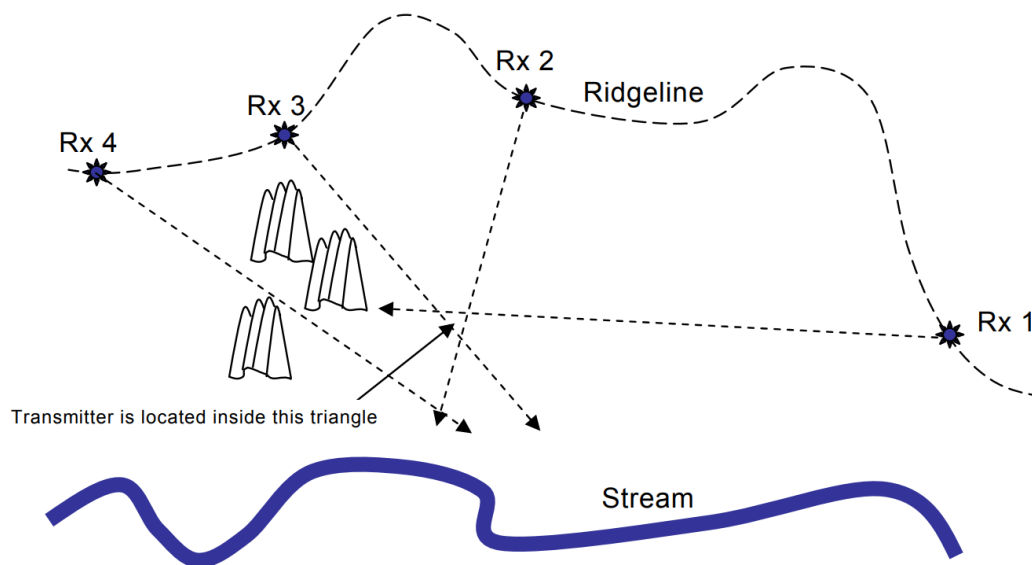


Figure 2.9 Manual radio telemetry localisation bearing a resemblance to the beelining process [66].

Automated approaches to tracking have been around for the last six decades [67]. Typically, these approaches use a primitive presence and absence system. No exact localisation is used in these techniques. Nevertheless, they can provide key knowledge. One such approach uses the Time Difference of Arrival (TDOA) to infer direction and distance when two receiving antennas are used alongside a pulsing broadcaster [68].

Based on the signal strength received by the overlapping fields of detection provided by each antenna, it is possible to provide an estimate for the target position. In 2011, Keys et al. were able to use such a system to monitor the positions of animals on Barro Colorado Island with ~50m accuracy if they were in the range of three towers [69].

In 2005, Stark et al. were able to use an acoustic variant of a telemetry system to track 12 squid over 300 km² for 37 days [70]. The study used "curtains" of receivers to segment bays and channels into areas and record when the squid approached said curtains.

One particularly potent advantage of radio telemetry is the use of unmanned aerial vehicles (UAVs) or drones. Given that the receiving antenna is relatively simple and small in structure versus harmonic radar, it is possible to mount such onto a drone.

Early experiments, using a facsimile of a drone, were successful circa 2009 [71]. Early drone prototypes took place in 2015, as drone technology matured enough to suit the need [72]. These experiments took place using birds as they are much more able to carry larger, more powerful tags.

As technology progressed, smaller and cheaper solutions became available. Gradual improvement has been seen across the years [73, 74].

A functional prototype is shown in Figure 2.10. This system was designed to track human targets with a large degree of success in 2019 [75]. Despite the potential of drones, minimal work has been undertaken and achieved to use the technology on insect targets, making this a key area for future research.



Figure 2.10 Prototype drone with antenna system [75].

2.2.5 Battery-less Tag Tracking Technologies

To compensate for the issues related to battery-based tags, augmented technologies have been in development since the 1970s [76]. These early systems typically used photo-voltaic cells to recharge batteries as they are depleted to allow for smaller

batteries with longer life spans. Since then, they have provided rechargeable systems to track birds over great distances and time with GPS [77].

Piezoelectric power generation converts impact energy from mechanical vibrations of life into electrical energy. Using such generation, it becomes possible to create tags that either supplement a battery or lack a battery altogether [78, 79]. Developed in the last thirty years, it has seen growing use in smaller applications [80]. In 2007, Takeuchi et al. were able to use a glass ball in a tube capped at both ends with a piezoelectric material to track school children entering and leaving the premises. Their walking triggered the glass ball to strike the material, generating energy that fuelled an RFID tag [81].

As the use of such techniques improved, they have been implanted in fish to enable tracking *in vivo* [82]. A strong example of piezoelectric for tracking larger than insect animals was done by Snowdon et al. in 2018 [83]. They used the technology to reinforce hawk tag batteries and enable periodic transmission in case of battery depletion, allowing them to function when the hawk had flown hundreds of miles out of retrieval range. This system, as shown in Figure 2.11, was especially useful as replacing the tags due to battery depletion over such a range is unfeasible.



Figure 2.11 Piezoelectric energy harvester used to track a hawk over large distances [83].

Further research developments have created tags suitable for insects. In 2009, Chang et al. developed a 1g tag suitable to harvest energy from a moth's flight to

generate 1 mW of electrical power at 1 VDC for general-purpose use [84]. Fast wingbeat insects offer an opportunity to generate significant power due to the thorax vibrations created. Aktakka et al. in 2011 were able to use the 85-100 Hz wingbeats of a Green June Beetle (*Cotinis nitida*) to generate 11.5 and 7.5 μW in device volumes of 11.0 and 5.6 mm^3 , respectively [85]. Placing two generators (one on each wing) resulted in more power per insect.

Most recently, Shearwood et al. were able to build a piezoelectric tag for honeybees [86]. This tag was entirely self-powered, with no onboard battery. This proof-of-concept design represents a shift towards piezoelectric systems designed around smaller insects.

2.2.6 The Costs of Tagged Technologies

The question of ethics surrounding the invasive tracking of wild animals has a long history [87]. As technology moves forward, the loss of public support has become a growing concern [88]. The lack of testing for the implications of tagging animals predates the use of such tags on insects [89]. Figure 2.12 details only some of the potential disturbances that could be associated with tagging insects.

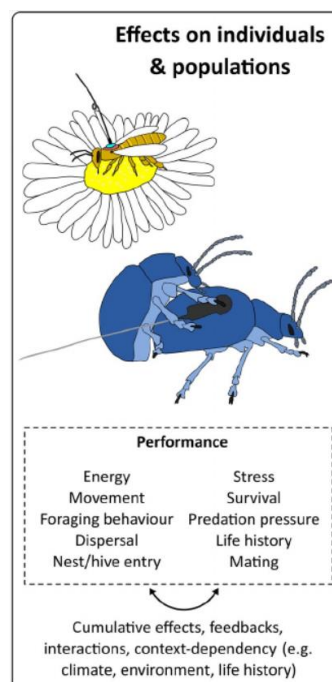


Figure 2.12 A conceptual diagram of the possible effects of tags on arthropods and the bias they can create in research results [39].

As early as the eighties, concerns for bats suffering from the weight of tags weighing only 5% of their body mass were made clear [90]. After the migration of such technologies to birds, researchers were quick to point out the lack of testing for after-effects [91]. Given how the tracking of birds has become commonplace and affordable, there have been growing analyses of the behavioural and ecological effects [92, 93]. A documented lower rate of return of tagged versus untagged birds has been found, putting a life cost on some of these technologies [94]. The often-quoted but non-existent 5% rule and lack of progress to encourage miniaturisation in birds have done little to further adapt equipment [95].

Where significant gains are necessary to aid in tracking, some have turned to implant tags to decrease the burden on the animal [96]. However, it is not hard to imagine that large-scale implanting of tags would not sit well with public opinion.

As the tracking of insects via tagging is much more recent, far fewer concerns have been voiced. It is also harder to test for detrimental effects. Concerns about weight in beetles (no more than 33%) have been raised, as well as concerns about the biological effects of glues used to fix tags to the body [97, 98].

A counterpoint to these concerns, a tag weight of 8% in the citrus fly was found to have no meaningful effect on activity over five days [99].

However, for pollen-bearing insects, there has been a documented significant energy cost increase when carrying pollen loads. The addition of a tag, regardless of how small, would compound such increases [100]. A 3mg tag was documented as causing a 27% reduction in take-off ability in honey bees [101]. There were also reduced pollination activities in tagged bees versus painted ones [102]. Diseased bees are less able to forage, therefore tagging such bees to study the effects of a disease might compromise the dataset [103]. Application of tags using low temperature or CO₂ (primary methods of disabling insects to apply tags) requires consideration of the long-term costs on the insects themselves [104].

2.2.7 Doppler Radar

One of the most appealing, though costly and difficult, approaches to insect tracking involves the use of a Doppler radar. Doppler radar functions by sending out high-frequency signals and analysing the minute differences in the signals reflected by objects moving in the environment. As such, no tag at all is required.

The study of high-frequency signals on bees and other insects is currently limited, however, in the available literature, there has only been small evidence of any effects of exposing bees to sub-1 GHz frequencies [105]. Most literature exploring higher frequencies finds no effect at all on bee behaviour [106].

Doppler radar typically works above 5.8 GHz, putting it well beyond the scope of what is known to harm bees [107]. The biggest shortcoming of such technology is the range and the scope of information available. Since no tag is present, the return power will be a fraction of that broadcast which significantly impacts the range. Laws in most countries place strict limits on broadcast power without a license. In the UK that is 25 mW Effective Isotropic Radiated Power (EIRP) at 5.8 GHz [108].

Another associated cost is that since there is no signal being broadcast from the target itself, there is limited information available. Speed and direction of travel, towards or away from the broadcaster, of the target, are available due to the nature of Doppler signals. Specific details such as the cardinal direction of travel are not easily determined. Figure 2.13 demonstrates the types of signals possible with a Doppler radar mounted on the entrance of a hive pointing outwards to monitor bees.

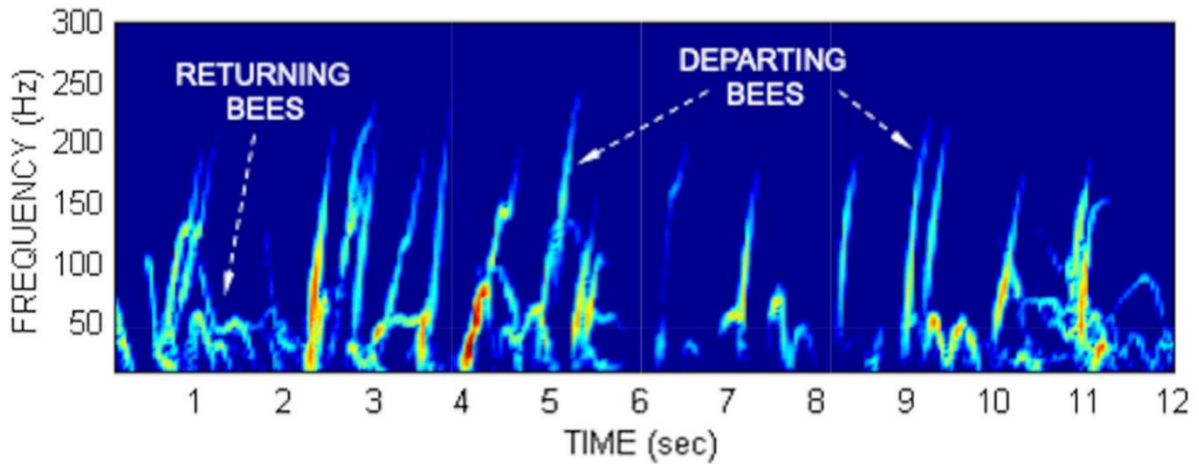


Figure 2.13 Signals detected of bees entering and exiting a hive via Doppler radar at 10.5 GHz [107].

Due to range limitations, almost all Doppler radars from 5.8 GHz up to 24 GHz are mounted at the entrance of a hive to catch bees leaving and returning [54, 109, 110]. Figure 2.14 shows a typical setup for this. The improvement of such technologies will be of great importance to future endeavours to track bees as this technology is far less invasive than others while still having the accuracy benefits of other RF technologies.



Figure 2.14 24 GHz Doppler radar mounted at the entrance to a beehive [109].

A hybrid technology exists in using a traditional microphone system alongside a Doppler radar, termed a Doppler microphone. Doppler radar typically creates data that is stored as audio files, therefore combining them with microphones creates two parallel audio samples that can be used to gather more detailed information than either alone [111].

2.2.8 Internal Hive Predictions

Precision beekeeping (PB), a branch of precision agriculture, is the concept of improved apiary management by monitoring individual bee colonies and predicting their needs [112–114]. PB systems are optimised to produce the best combination of sensors to gather relevant data that is fed into decision support systems. Their developments are driven to suit the needs of business interests, expected risks, and rewards. Their principal benefit is allowing the lessening of manual inspections through remote access to real-time colony health metrics.

Data collection in PB can be split into three groups [115, 116]:

- World parameters: weather information, seasonal variations, and video observations.
- Colony parameters: temperature, humidity, internal atmospheric composition, sound, video, and hive weight.
- Individual parameters: number of bees entering and leaving the hive, number of bees at the hive entrance.

Driven by the miniaturisation of sensors, coupled with increased capacities, PB systems have become more affordable and widespread. Naturally, this has created opportunities to include predictive algorithms to enhance their capabilities [117].

In 2016, Kridi et al. used thermal sensors to build thermal response patterns of bee colonies in Northeastern Brazil [118]. Their system was able to reduce the data transmission load by taking the crude readings and parsing them via k-means clustering to check whether conformance with a given pattern was present. This is shown in Figure 2.15. So long as the readings fell within an expected pattern, then no alert was necessary. When a deviant measurement was detected, a message

was sent to the base station. When three or more deviant measurements were taken, an alert was broadcast to initiate a response. The conceptual properties of this system are presented in Figure 2.16.



Figure 2.15 Conceptual creation of thermal patterns from raw data. These were the source patterns that new measurements were compared with to determine deviancy [118].

The goal of this system was to prevent the colony from overheating, which can lead to absconding. Absconding is an adaptive behaviour that Africanized honeybees have developed over their European counterparts. It is a response to the seasonal changes of semiarid areas, driving bees to move to more coastal areas where milder weather may be found [119]. For managed colonies this is not a desirable event, necessitating intervention.



Figure 2.16 The physical setup of Kridi et al., showing (a) the distance from the observation base, (b) the environment surrounding the apiary, and (c and d) the prototype in plastic and wooden protection, respectively [118].

In a mix of disciplines in 2016, Edwards-Murphy et al. used PB technologies to gather data that allowed them to use a decision tree algorithm to predict 10 classes of hive status, from normal and hibernating hives to damp or too-hot hives [120]. These tree algorithms were adaptively designed and managed with an average accuracy of 95.38%, with the final structure shown in Figure 2.17. Accuracy, in this case, is the proportion of correctly classified hives versus human expert classification. They also managed to use decision trees to correlate the coming of rain within the next six hours with the CO₂ level found inside the hive. The decision tree in Figure 2.17 was able to determine if a hive was too small (colony size/weight) to effectively determine the oncoming rain.

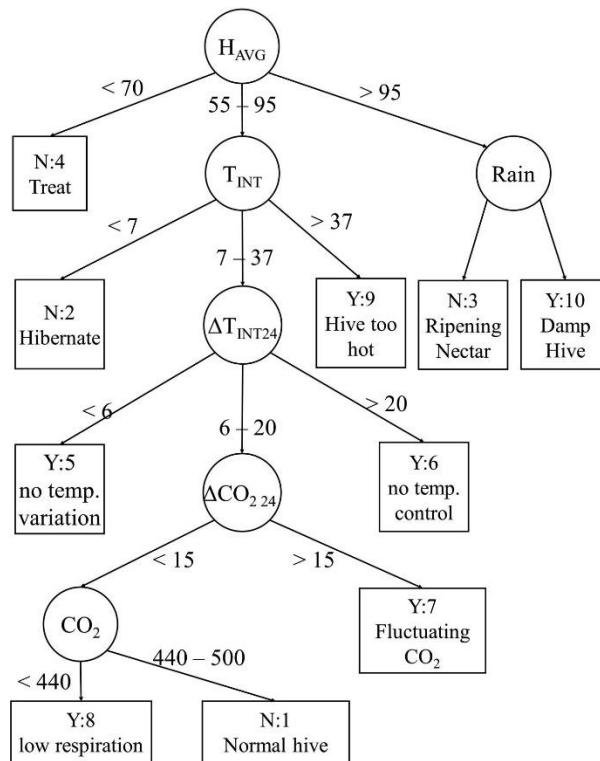


Figure 2.17 Decision tree structure for predicting hive status via Edwards-Murphy et al. [120].

In 2019, Robles-Guerrero et al. used an acoustic solution to correlate the sounds produced by a hive to determine if they were queenless [121]. They created a particularly accurate model (approx. 95% accuracy) from just five hives. They used the well-documented technique of Mel Frequency Cepstral Coefficients (MFCCs) to create model features. This technique turns a raw signal into a series of compact features representing the source signal. Feature selection and model generation were then performed using Lasso Logistic Regularisation. Further analysis looked at observing how the signature sounds changed depending on time and hive condition. A special note was made of how their data separability indicated the presence of many hive statuses detectable by acoustic signature alone.

Rafael Braga et al., in 2020, used PB techniques to predict colony health over time [122]. They utilised a combination of internal sensors (temperature, colony mass) and external sensors (temperature, wind direction, speed, precipitation, and daylight.) To create a class system to predict against, they used Healthy Colony Checklist (HCC), carried out manually once a week from six hives.

This checklist assesses the following:

- Presence of full brood spectrum.
- Sufficiency of adult bees.
- Presence of a young, productive queen.
- Sufficiency of water, forage, and food.
- Lack of stressors that affect colony prospects.
- Suitable space for current and near-term needs.

This checklist was binarised to form class labels. A k-nearest neighbour, random forest, and neural network algorithm were used as classification models. Hives were scored out of six according to the HCC and their final model, a random forest, was able to predict the hive score with over 90% accuracy.

This study targeted reducing the need for hive visits by an inspector. Such visits are drawn-out events and can be difficult in adverse weather or during the winter. One weakness of physical inspection is that each inspector may have minor biases or differences in approach. Having data captured by the sensor and assessed via a centralised algorithm could lead to the alleviation of these concerns.

Approaches like this study, which focus on the hive as a whole, make good use of machine learning and represent some of the cutting-edge work undertaken in the field. However, studies which do not account for the behaviour of individual bees are limited in scope. Whilst they can provide exceptional insight into colony-wide issues they cannot provide information about anything beyond that scope.

Studies which go further by monitoring individual bees have potential to provide even greater insight: in an ideal world, mapping an entire hives resource acquisition while modelling changes in hive health against the availability of plant species, presence of pesticides, proximity to human habitation, and weather will likely lead to further breakthroughs.

2.2.9 RF-Based Machine Learning

Early uses of machine learning coupled with RF (Radio Frequency) technologies took the form of identifying "clutter" from swarming insects, particularly for cleaning weather data to provide more accurate forecasts [123]. From this, the exploration of removing all clutter from a data set using artificial intelligence has been explored with radar. In the case of Islam et al. in 2012, removing ground, sea, and airborne clutter from radar readings targeting precipitation [124].

In 2008 Cabanes et al. used a Self-Organising Map (SOM) algorithm to infer ant behaviours based on them passing through gates in an artificial hive while tagged with RFID chips [125]. SOMs are a class of unsupervised machine learning, where the result is clusters of similar data points. These are generated independently from human observation and can provide new insight into previously undetermined behaviours. Figure 2.18 shows the experimental setup used which resulted in four behaviours being detected. These behaviours were attributed to ants specialised to care for the queen and brood, those that exclusively forage, generalist, and maintenance ants.

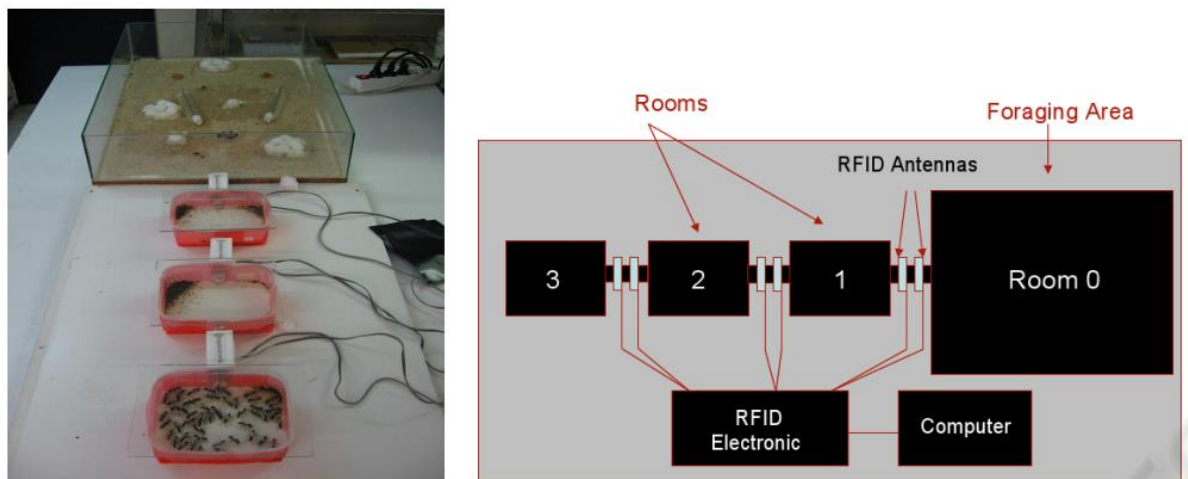


Figure 2.18 The RFID experimental device of Cabanes et al. [125].

A relatively simple machine learning technique, a Gaussian model, was used by Susanto et al. to perform curve fitting addressing shortcomings of RFID miss-

readings with clustered behaviours [126]. This allowed for the separation of bee behaviours into four clear categories from an RFID system at the entrance to the hive. They found the following:

- **Foraging (FG):** Bees were classified as foraging when the gap between successive detections is longer than six minutes.
- **Short Mission (SM):** Bees engaged in short missions were those with successive detection intervals between three and six minutes.
- **By The Entry (BTE):** Bees classified as being “by the entry” were those with successive detections of the same bee by an RFID antenna at a maximum time interval between successive readings of less than three minutes.
- **Departed bees (DB):** Bees that left the hive and never returned, either because they died or because they swarmed (including absconding).

In 2018, Hu et al. began the groundwork for a series of papers investigating the identification of insect species using radar [127]. Their initial paper looked at using a Support Vector Machine (SVM) that was trained to recognise species based on weight, wing beat, and body length-to-width ratio. They focused on 23 species that were commonly detected by radar when migrating, and their characteristics are demonstrated in Figure 2.19. Their final accuracy was over 80%, encouraging further work.

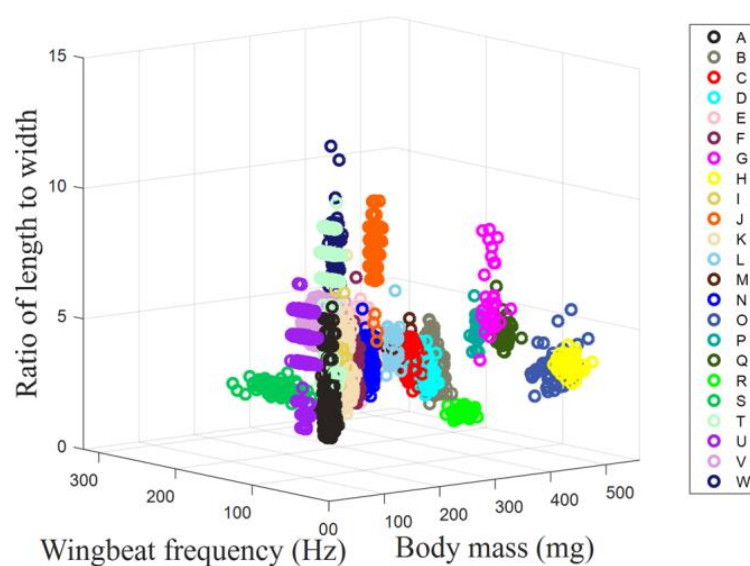


Figure 2.19 Parameters of 23 insect species that were fed into an SVM algorithm by Hu et al. [127].

In 2020, Hu et al. extended this work and attempted to use Radio Cross Section (RCS) to predict insect mass [128]. Again, using SVM, they recorded the cross-section of insects in a microwave anechoic chamber. The resulting data was fed into the SVM which was able to predict the mass of the target with 78% accuracy. As mass was one of the key variables used in their original work, this represents steps forward to predicting insect species from raw radar data.

In 2018, using RFID tags, Arruda et al. investigated separating individual bee species by looking at the mean, standard deviation, and sum of activity measured with each passing hour [129]. This data was fed into multiple machine learning algorithms; neural network (NN), classification and regression tree (CART), and random forest (RF). Two similar species were assessed in this manner, *Melipona fasciculata* and *Melipona seminigra*, across multiple days. Each hour provided three variables, leading to 72 captured data points per day. Variations in activity versus time of day were used as the foundation to predict species. The random forest proved strongest in this approach, resulting in a final accuracy of 87.41%.

Despite the clear power of machine learning in this field, implementations using other than RFID or meteorological radar are lacking. RFID has already shown how effective these techniques can be, with additional examples being the use of RFID to forecast bee activity or classify anomalous bee activity [130, 131]. The absence of further work using other forms of RF tracking can be attributed to the highly specialised and expensive nature of such systems for insects. There is not one broad-scope technology designed for many insects over large distances. This is not the case with RFID, which can use very small, un-powered chips, designed for a much broader range of applications than just insect tracking. Further work to demonstrate the efficacy of machine learning with other RF techniques should hopefully spur further development of more able, generalist systems.

2.2.10 Visual, Thermal, and Light-Based Techniques

2.2.10.1 Visual Tracking

Many algorithms have been specifically designed to predict insect behaviours as confirmation of ecological understanding. For example, using a model to predict whether bees share the foraging experience with recruits [42]. Some are machine-specific, direct algorithms to track insects via 2D video recordings [132, 133]. There have been recent efforts to develop intelligent algorithms capable of automating and improving many of these tasks [134, 135].

Visual identification of species using machine learning has been done with great effect; in the case of Urtaega et al. in 2016, identifying whether an image contained a poisonous or non-poisonous scorpion with an accuracy of 82.5% [136].

K-means clustering combined with Gaussian kernel filters has been used to assess crop damage from images taken by a UAV [137]. As shown in Figure 2.20, this approach was very effective, with the additional strength that k-means clustering is an unsupervised learning algorithm. This means that their implementation should work well with a wide range of similar images without needing further work. The kernel allows for manual adjustment where necessary to tune the images to the models.

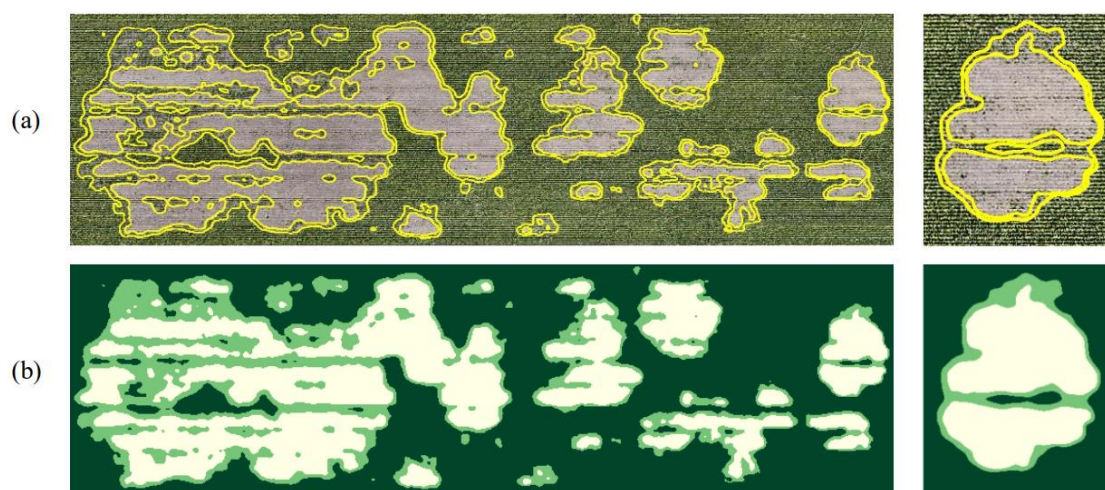


Figure 2.20 Farmland assessment of crop damage from images taken by UAV. (a) Raw image without health boundaries and (b) Membership map and decision boundaries after applying a soft K-means clustering with $K = 3$ clusters and Gaussian parameter $\delta = 8$ [137].

Some similar work was done by Alves et al. in 2020. They used semantic segmentation to split images of hive combs into component cells. They then classified those cells as being; eggs, larvae, pupa, honey, pollen, nectar, and others. Using 11 different learning models, they found the optimal solution to automate this process. They used well-known, pre-trained neural network algorithms and re-trained them to work on the new data set. The final accuracy for cell detection was 98.7% and the final classification of the contents was achieved with 94.3% accuracy. Finally, they developed this into free software known as DeepBee as demonstrated in Figure 2.21.



Figure 2.21 DeepBee© software developed for the interaction of the users with the predictions of comb cell contents [138].

Detecting dances and their meaning has been a subject of visual tracking. By taking videos of hive combs, individual bees can be tracked, and their movements interpreted into dances describing the location of resources that the hive may need. Veeraraghavan et al., in 2008, used Markov models to detect when bees were dancing [139]. Their results were within two video frames of expertly labelled data, paving the way to gather key video samples automatically much faster than humans, even if the decoding of such messages remains difficult.

Similar work was done by Blut et al. in 2017 [140]. They looked at tracking interactions other than dancing, namely offering, begging, trophallaxis, and antennation. Trophallaxis was indicated by the exchange of nectar between two bees. Antennation was present as the initiator in all interactions, where the bees maintain contact with their antennae. In all other cases, other behaviours then followed except in the case of antennation. In this case, individual bees were tagged with a visual barcode chip for unique identification. Their system was 93% accurate at detecting interactions. It is worth noting that they struggled to determine between individual interaction types as they focused on the time-length of interactions rather than other characteristics. Similarly, they used software called Janelia Automatic Animal Behavior Annotator (JAABA.) This tool is a machine learning-based animal behaviour decoder designed for use with video. It would be interesting to see if a more tailor-made algorithm for bees would be more effective than a generalised system designed for all kinds of animal detection.

Additional visual tracking was undertaken by Boenisch et al. in 2018. Their tags are shown in Figure 2.22. However, their work was very thorough. By combining multiple levels of machine learning, they were able to track 2,000 bees over 10 weeks to generate tens of millions of images. In the first stage, an SVM was used to predict that two incomplete or semi-obscured images of tags belong to the same bee. This allowed the formation of "tracklets." These were incomplete segments of a larger track. At the secondary level, a random forest was utilised to join individual tracklets into a complete whole. This allowed for the staged algorithm to track bees that became obscured, flew away on missions or otherwise made it difficult to maintain visual confirmation. Their final system reduced incorrect ID decodings from ~13% to 2%. In addition, their second stage allowed for 90.4% of tracklets to be correctly reformed into complete tracks.

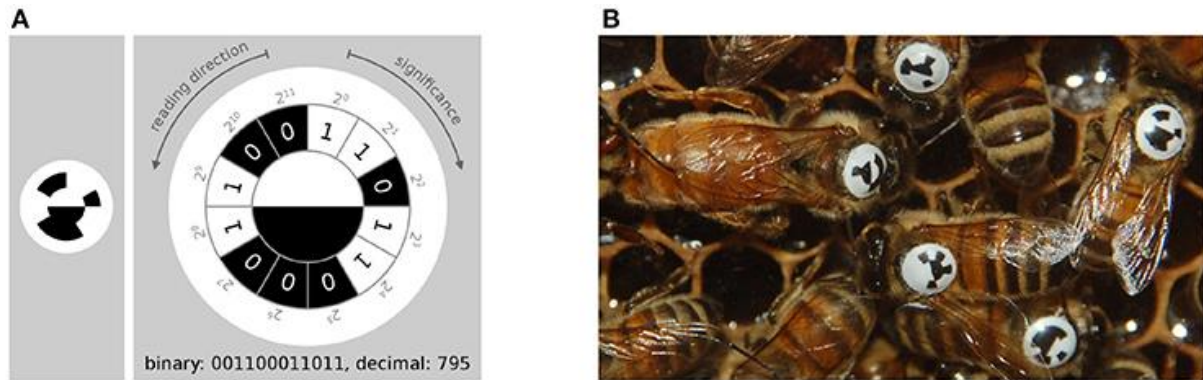


Figure 2.22 Visual tags designed by Boenisch et al. (a) 12 coding segments arranged in an arc around two semi-circles that encode the orientation of the bee. The tag is glued onto the thorax such that the white semi-circle is rotated toward the bee's head. (b) Several tagged honeybees on a comb. The round and curved tags were designed to endure heavy-duty activities such as cell inspections and foraging trips [141].

2.2.10.2 Thermal

It has been known for some time that colony insects both thermoregulate their bodies as well as their nests [142, 143]. As such there has been an interest in tracking insects back to their nests and locating such nests, using thermographic cameras. Such devices were historically expensive and very low resolution. Recent developments have increased their viability for such tasks. A reasonably priced thermographic camera can be expected to have a working resolution of 320x240 for a total of 76,800 pixels [144]. This is in contrast to budget digital cameras that have resolutions of 1920x1080 for a total pixel count of 2,073,600.

Nevertheless, despite their limitations, there is real potential for tracking insects. Two recent studies, by Roberts et al. and Lioy et al. both, explored the viability of such cameras to find hornet nests, as well as those of wild bees [144, 145]. They both had a large degree of success, with a sample of images collected by Lioy et al. present in Figure 2.23.

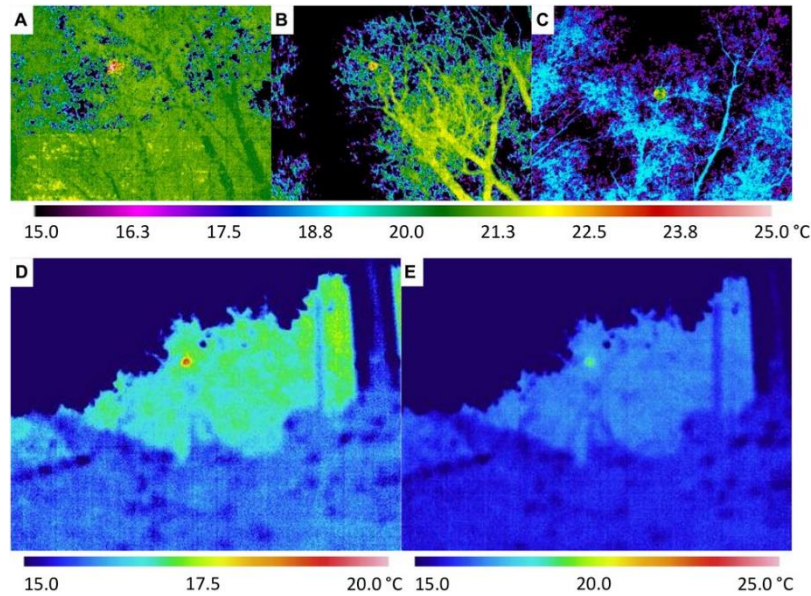


Figure 2.23 Application of thermal imaging for detecting *V. velutina* nests: (A) nest number one; (B) nest number two; (C) nest number three; (D) nest number one in the morning at 30 m from the operator; (E) nest number one in the evening at 30 m from the operator [145].

As both works were viability studies, no work was discovered that integrates any form of machine learning. This leaves a gap in research where further work can be done. Neural networks are particularly strong at dealing with low-resolution images and in cases where there is still ambiguity, super scaling techniques can aid via pre-processing techniques [146].

2.2.10.3 3D point cloud and spectral sensing techniques

Some specialised light-based techniques exist to identify, monitor, and track insects. The fluorescence of planthopper and moth species, a pest in China, were explored with light detection and ranging (LIDAR) techniques to accurately detect insects up to a distance of 50m, including then dusting other insects with fluorescent dyes to similar effect [147].

Hyperspectral imaging has been used to detect insect defoliation with machine learning [148, 149]. The red-edge band is useful when monitoring vegetation, as chlorophyll becomes almost transparent at wavelengths greater than 700nm. From then on, the internal structure causes this infrared light to be reflected. Different

states of cellular health respond with different intensities. Figure 2.24 shows such data gathered to be used in random forests and support vector machine prediction of the state of vegetation.

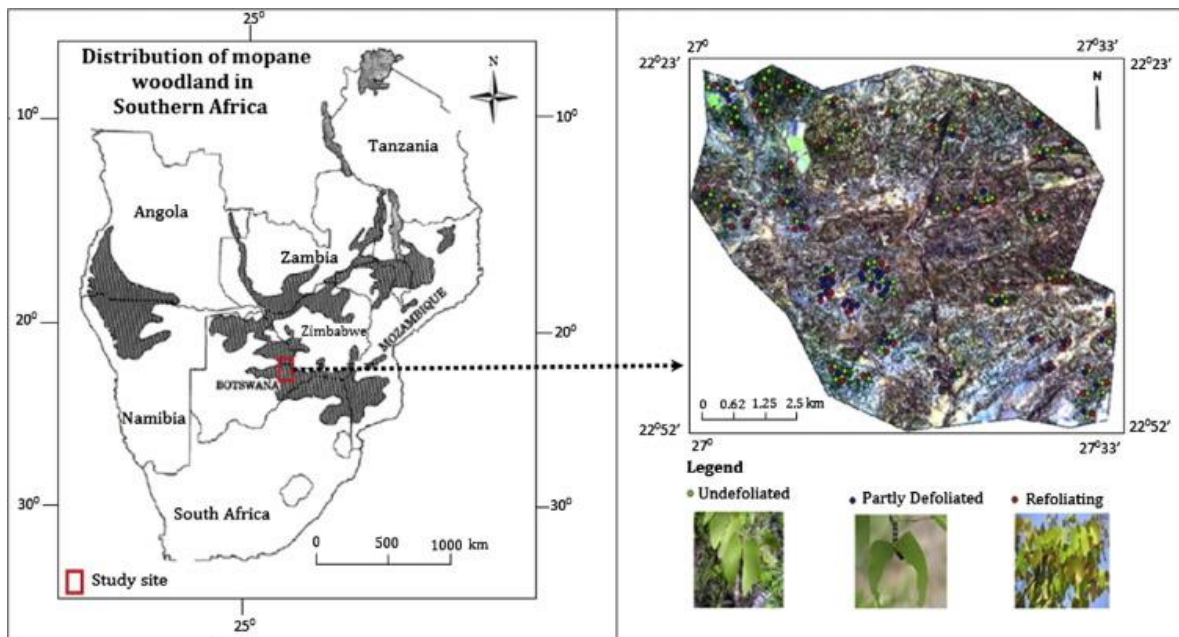


Figure 2.24 Distribution of mopane woodland in south eastern Botswana and the blue, green, and red band combination of a satellite image of the study area, together with Images for different defoliation levels [148].

In addition, there has been growing use of laser systems to detect insects. A trap can be manufactured such that when the insect enters or exits, it impedes a laser. Fluctuations in the laser cause patterns which can be attributed to different sources. Mullen et al. used the shadows caused by such insects to determine the species of insect [150]. Silva et al. took the approach of having a narrow laser impacting a transistor array as in Figure 2.25 [151, 152]. This allowed them to treat the incoming signal as an audio sample. Machine learning techniques involving Mel-Frequency Cepstrum Coefficients (MFCCs) and support vector machine allowed for an accuracy of >87% when determining from a sample of nine species including bees.

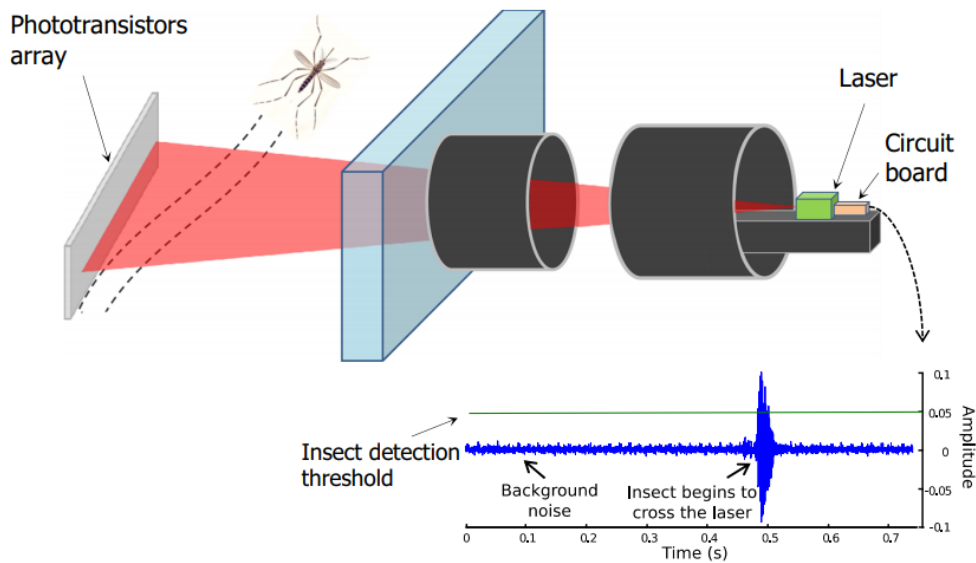


Figure 2.25 The logical design of the sensor used by Silva et al. A planar laser light was directed at an array of phototransistors. When an insect flew across the laser, a slight variation was registered by the phototransistors as a time series [152].

2.2.10.4 Audio Techniques

Tracking animals via acoustics is a challenging task. While many studies have been successful in water and enclosed spaces, the difficulty of tracking in open spaces with background noise has remained [153–155]. Systems have been developed capable of localising a sound signature out to between three and six meters [156, 157]. Machine learning has increased the accuracy of such systems up to around 90% [158]. Similarly, recent algorithms have allowed for the separation of different speakers in an office setting while maintaining the direction of arrival (DOA) accuracy [159]. The use of cooperative algorithms to identify near-field sound sources regardless of background noise has also been investigated, especially to aid in robotic applications [160, 161].

Another interesting, though equally challenging, prospect involves using UAVs to track insects. Several works have already made headway into a similar application, that of locating sound signatures during emergency search and rescue [162–165]. The biggest hurdle to overcome is the signal-to-noise ratio (SNR.) UAVs can easily be in the range of 60-80dB at one meter, compared with approximately 45dB for a

typical bee [166–168]. The very best algorithms can remain accurate at -10dB SNR, meaning the quietest drone at 60dB would be able to locate the loudest bees at 50dB, assuming the microphones were able to be carried at a 1-meter distance from the drone. Some exceptions allow for the SNR to be as low as -20dB in exceptional circumstances.

While promising, these small UAVs are typically unable to carry the microphone load required for the task of detecting insects. In most cases, circular or hexagonal arrays of microphones with as many as sixteen microphones are required, the frame to hold which is itself large and weighty necessitating a larger drone. A typical industrial drone, such as those provided by DJI, is often in the region of 80dB.

For example, a 3DR Iris+ drone is as quiet at 60dB with certain setups [168]. However, such a drone has a payload capacity of 400g. This is well outside the range of equipment necessary. Similarly, such a specific drone does not have a Software Development Kit (SDK) or the means to easily create one, to allow for automated piloting.

To continue, a DJI Phantom 4.0 rates as high as 76dB, with a payload capacity of 900g, potentially enough to carry the required load. Industrial standard drones such as the Matrice 210 series do not disclose their measured dB loudness as they are typically less concerned with maintaining low noise.

Another consideration in such a problem is the processing requirements of the localisation algorithms. Figure 2.26 shows two algorithms based on the Multiple Signal Classification (MUSIC) algorithms, comparing both accuracy vs SNR and processing on computer hardware. MUSIC is often used due to its ability to distinguish multiple audio signal sources in addition to being malleable to improvements that augment its ability to cope with noise.

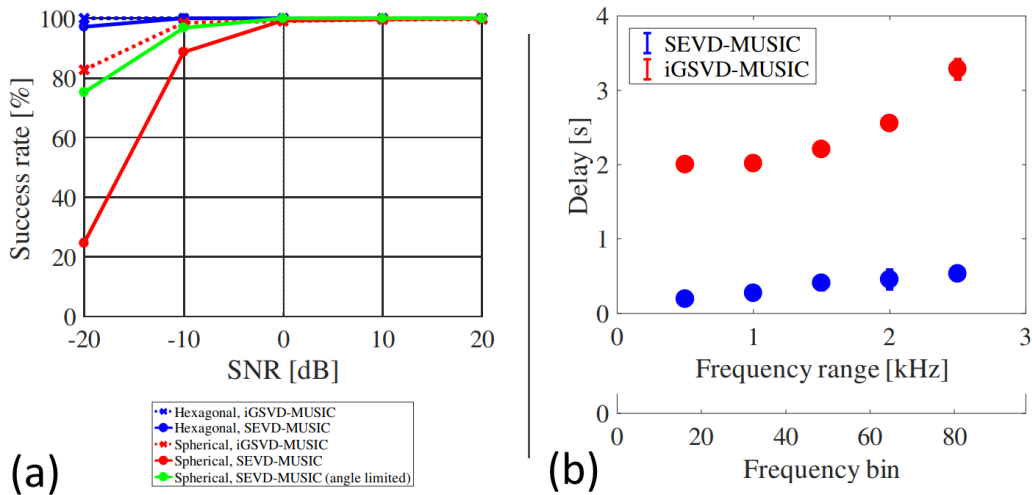


Figure 2.26 (a) Accuracy of two MUSIC derivative algorithms (SEVD and iGSVD) and (b) Processing cost of such algorithms [162].

Of special note to such systems, is that in most cases the data must be broadcast back to a ground station for processing. In the case of Hoshiba et al., this required transmission rates exceeding 5Mbps, limiting the distance between the drone and the ground station to 75 meters.

Some of the earliest use of machine learning with insect detection involved estimating the population of larvae in grain silos [169]. Coggins et al. used a time-delay neural network to track the vibrations of the larvae moving and eating the grain. This pattern has continued through recent decades. In 2014, Wang et al. used acoustics to determine the density of *Locusta migratoria migratoria*, serving the purpose of crop protection needs and environmental impact studies [170].

One particularly interesting example involves using the spectrogram of an image and feeding it into a convolutional neural network built to classify images. Dong et al. had some success with this in 2018. Their spectrograms present in Figure 2.27, can determine the species of insects by visualising the sounds that they produce [171]. In a similar experiment, Kulyuki et al. found that spectrograms performed worse than the other audio processing techniques [172], creating the need to evaluate on a case-by-case basis.

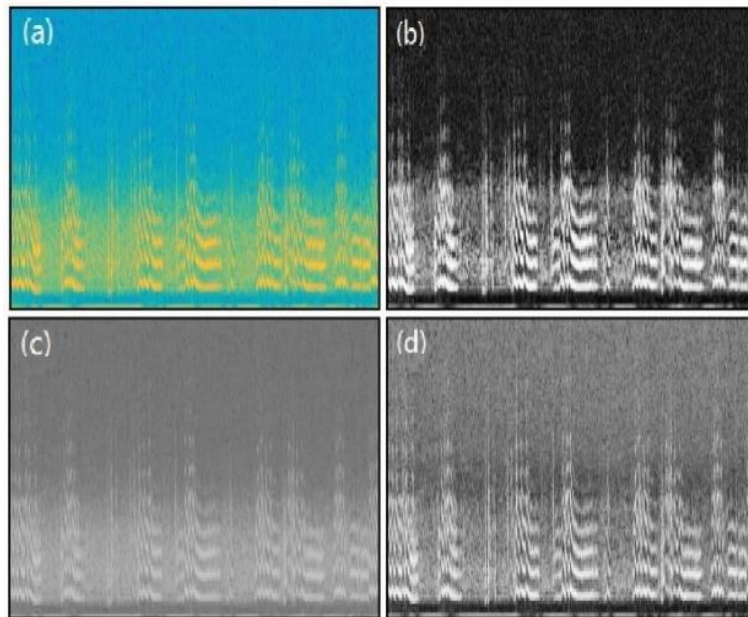


Figure 2.27 (a) Chromatic spectrogram (b) Enhanced R-space spectrogram (c) Gray spectrogram (d) R-space spectrogram used by Dong et al. for image classification based audio prediction of insect species [171].

For bees and hornets, more traditional techniques have been used to automatically classify based on acoustic signatures [173]. In such approaches, Mel-Frequency Cepstral Coefficients (MFCCs) are used to parameterise the acoustic signal to easily analyse it. A scale of pitches that maintains an equal distance below 1 kHz is referred to as a mel-scale, beyond 1 kHz the distance becomes logarithmic. This approximates how human hearing gives a bias to various frequencies and was originally designed to aid in speech pattern recognition. Such MFCCs can be fed handily into machine learning algorithms.

Heise et al. managed to tell between the arrival and departure acoustic signals of bees simply by analysing the shape of the acoustic envelope, with no machine learning to speak of [174]. This is promising for lightening the load on applications that identify the species by reducing the processing load associated with machine learning.

Devices have now been developed that work similarly to presence-and-absence RF-based systems. They are low-powered, solar rechargeable stations that listen to the

immediate acoustic environment [175]. Their primary use is to measure pollinator density in an area, but no exact location is determined.

While currently limited in scope for tracking insects, the audio approach remains appealing. With no tags and no requirements to broadcast any signals, the equipment can be simple and will not affect the insects themselves. The current major drawbacks are the low SNR of bees and other insects in their environment, limiting the range at which they can be heard.

2.2.11 Time-series prediction

Time series prediction is a potent method for classification or regression when previous input can provide context for current data. For example, time series is often utilised in text-to-speech where the previously detected words can provide context for the current spoken word, allowing a model to predict the word by using the context of the entire sentence.

Time series analysis has been used to great effect in determining the motion of larger animals, particularly humans. Here it is often used for fall detection and continuous human activity recognition [176].

In bees, time-series analysis has been used in the form of Hidden Markov Models (HMMs) to monitor and classify bee swarm activity using Mel-Frequency Cepstral Coefficients (MFCCs) of acoustic data gathered from a hive [177]. Additionally, video data has been used to track and predict bee activity within a hive using HMMs [178, 179]. Taking this concept further, time-series analysis has been used to detect and monitor bee dancing within the hive [139].

Dynamic time warping (DTW), another form of time-series analysis, has been used to identify bee species from acoustic signals and to detect the current state of a beehive (by monitoring the sounds of a queen to detect oncoming swarming behaviour [180, 181]).

Long Short-Term Memory (LSTM) neural networks are a recent development for time-series analysis and have already been used on bees by detecting that a beehive is currently queenless by analysing the audio from within [182]. Lastly, RFID alongside Recurrent Neural Networks (RNNs) have been used to forecast bee foraging activity by counting bees entering and leaving the hive [183].

These studies show that time series analysis and prediction is an active area of research within bee tracking and behaviour analysis. By informing current predictions and forecasts using previously recorded data, models can overcome one of the most common limits imposed by techniques that only consider current data. This limit is that the current data alone does not provide sufficient context for accurate analysis.

However, time-series analysis poses challenges. It often requires a continuous input of labelled data which can be taxing to produce. Similarly, it can require more data than other forms of machine learning to provide sufficient context over time. Non-time series models can often make better use of the data in cases where it has been unfeasible to capture or label previous data.

This is reflected in the presented studies which focus primarily on internal beehive data, especially acoustic and video, and are therefore more contained and easier to label. Only one study monitored bees outside of the hive or nest and this study was limited to RFID counting of entering and leaving bees [183].

2.3 Machine Learning Algorithms

2.3.1 Neural Networks

Neural networks take their name for their perceived similarity to neurons found in animal brains, originally coined perceptrons by Rosenblatt in 1958 [184]. They are a machine learning process that attempts to adapt their weights to the data they are tasked with learning. Foundationally, they are based on artificial neurons that take a form as described in Figure 2.29.

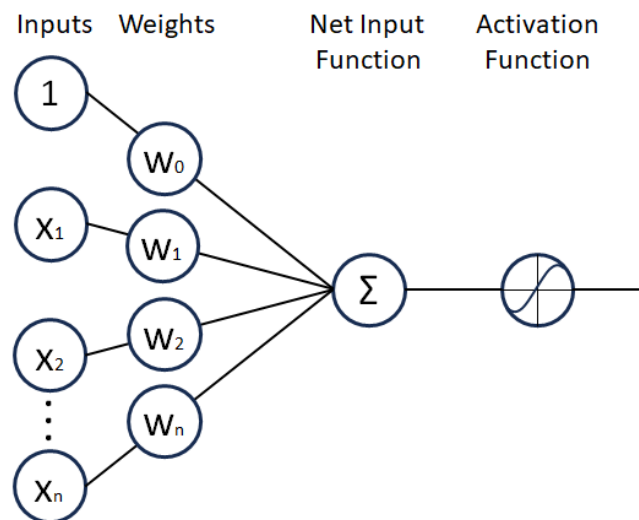


Figure 2.28

Figure 2.29 A diagram of the node within a neural network, showing the inputs, weights, input function, and activation function.

This neuron has n inputs which are coupled with an equal number of weights. An activation function determines where a neuron is fired (activated) or remains dormant (inactivated). The most common activation function is the Sigmoid function as in Equation 2.1, where x is the neuron output.

$$f(x) = \frac{1}{1 - e^{-x}} \quad 2.1$$

The neuron weights can be tuned during the learning phase where the network is adapted to produce the optimal output given a set of inputs. Neurons are placed

together in layers as demonstrated in Figure 2.30. Each layer takes its inputs as the output from the previous layer (or the original data in the case of the first layer). Layers between the first and last layer are referred to as hidden layers as their output is rarely retrieved directly.

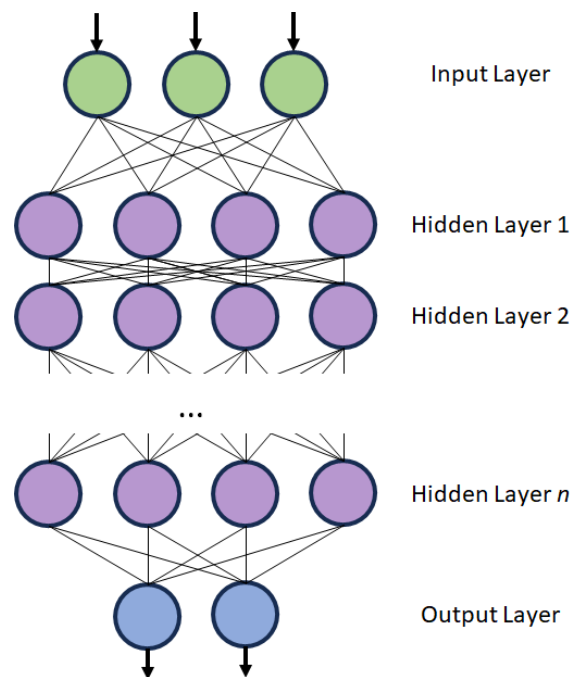


Figure 2.30 A diagram of a neural network showing the input layer, weighted interconnections, hidden layers, and output layer.

Training is the process of inputting data into the network for it to predict and be corrected. Training data has known values (such as class labels or regression values) for the network to attempt to predict. Learning is undertaken by a process known as backpropagation. Firstly, the network predicts an output based on its input data. This output is compared with the true, measured value for the original data. The deviation between the predicted value and the true value forms the *loss* metric for the network. The purpose of the learning is to minimise this loss value. To do this, the loss is fed back into the network and the weights within (for each neuron) are adjusted to decrease loss. The degree to which weights are adjusted is controlled by the *learning rate*. This rate is a value controlled during the learning process to fine-tune the network. If too aggressive, the network may skip over the minimum possible value for the loss. Various *optimiser* algorithms exist to manage the weights and loss

as the learning is undertaken. The work in this thesis used the Adaptive Moment Estimation (Adam) algorithm most frequently [185].

The output of each layer within the network can be summarised as in Equation 2.2, where $w_{k,j}^i$ is the weights applied to the connection between node k in the $i - 1$ layer and node j in the hidden layer at i . n_i denotes the number of neurons in the i^{th} layer.

$$h_i^j = f(\sum_{k=1}^{n_{i-1}} w_{k,j}^{i-1} h_{i-1}^k); i = 2, \dots, N \text{ and } j = 1, \dots, n_i \quad 2.2$$

The final output from the network can be written as in Equation 2.3. $w_{k,j}^N$ is the weight between the k^{th} node in the N^{th} hidden layer, and the j^{th} node in the next layer. Y is the vector of the output layer; F is the transform function and W is the weights for all hidden layers.

$$y_i = f(\sum_{k=1}^{n_N} w_{k,j}^N h_N^k); Y = (y_1, \dots, y_{N+1}) = F(W, X) \quad 2.3$$

After training is complete, the output from the network will consist of matrix Y which corresponds to the required predicted values. In the base case, the model is preserved with its current weights so that no more learning is undertaken. New data, lacking known output values, can be inputted into the network and predicted values generated for later use.

In this thesis, neural networks were used to predict categorical labels. In such cases, the size of Y corresponded to the number of classes within the problem. The position of the largest value in Y denotes the predict class label for the input data.

Neural networks are a popular machine learning tool and gave rise to the term deep learning. This term came about due to how scalable and specialised neural networks can become. Unlike other machine learning algorithms, neural networks can have more than a single set of hyperparameters. Hyperparameters are the values used to define how the model should behave and learn from the training data. Neural networks can have many layers of interconnected artificial neurons and there is no

limit to the number or shape of these layers, though overly large models can overfit on small datasets. These layers can have their own hyperparameters.

This leads to the possibility of specialised sub-regions within models consisting of unique architectures of neurons and layers that, when working together, transform or manipulate the data in a way that is beyond (therefore deeper) than previous machine learning algorithms.

A recent and famous example of this is the transformer model [186]. Here groups of layers and neurons function as “attention heads” which together contextualise the data within the model itself, allowing for key data to be amplified and redundant data to be diminished.

Some important hyperparameters when designing neural networks are:

- The number and shape of the hidden layers. This parameter gives rise to the possibility of specialised models, as each layer can have separate hyperparameters from the rest of the model.
- The connectivity and isolation of different layers. Layers can be grouped and shaped to perform sub-functions within the model by modifying their connections with other layers.
- The activation function, which controls how the neurons in a layer activate when presented with input. The activation of neurons is the way their input is signalled to be important to the process of the model prediction. It is also how, in a binary prediction case, the predicted class is decided.
- The optimiser function, as mentioned previously, is the algorithm that orchestrates model optimisation by coordinating the model's current prediction error and the weights within the model to create a set of updates that decrease error over time.
- Batch sizing, which is the number of data samples the network operates on before the optimiser updates the model. This parameter is often chosen as a limit of hardware resources rather than increasing model capability, however, a general understanding is that larger batch sizes smooth the learning curve and reduce the effect of outlying data samples on the training process.

- The number of epochs, which controls how many times the model iterates over the dataset while learning. Epochs, like batches, are often a hardware limitation. Given proper model metrics (discussed in Section 2.3.6) the model should converge, where the validation of the model shows no further improvement regardless of further epochs.

Given the complexity of designing a neural network, it is suggested that once moving beyond a few (two or three) hidden layers it is best to search the literature for a good neural network structure rather than attempt to design one as part of a study not focused on that task. This is because creating a new model can be an entire, self-contained, academic endeavour. This is reflected in this thesis, where the networks were small whilst in Chapter 5 where a predesigned set of architectures was chosen.

2.3.2 Support Vector Machines

Support Vector Machines (SVMs, [187]) are models for classification and regression problems. The core idea behind SVMs is simply that they create a hyperplane that separates the data into distinct classes. This is portrayed in Figure 2.31 where an optimal hyperplane has been determined which separates two classes of data with the maximum margin given at least two supporting vectors. A hyperplane in an n -dimensional Euclidean space is a flat, $n-1$ dimensional subset of that space which divides the space into two disconnected parts.

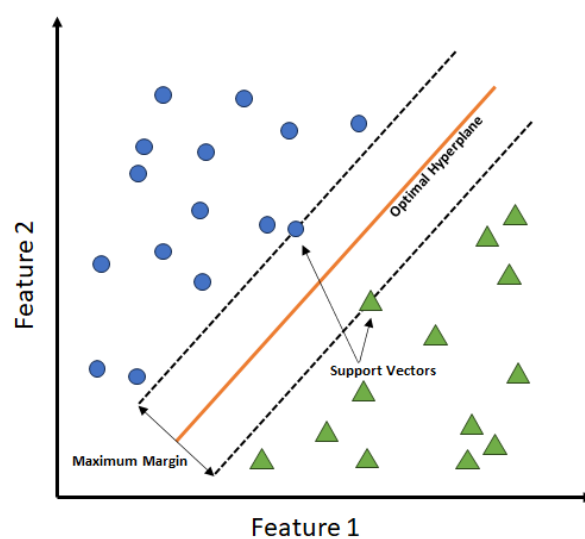


Figure 2.31 An optimal hyperplane between two datasets with a maximum margin to two support vectors.

The challenge arises when the data is not readily separable in n dimensions. To tackle this problem, SVMs use a kernel that projects the data into higher dimensional space where they can be readily separated, as demonstrated in Figure 2.32. SVMs use the ‘kernel trick’ which enables them to operate in this higher dimensional, implicit feature space without computing the coordinates of the data in that space by, instead, computing the inner products between the images of all pairs of data in the feature space. This is computationally cheaper than determining the coordinates within this higher dimensional space.

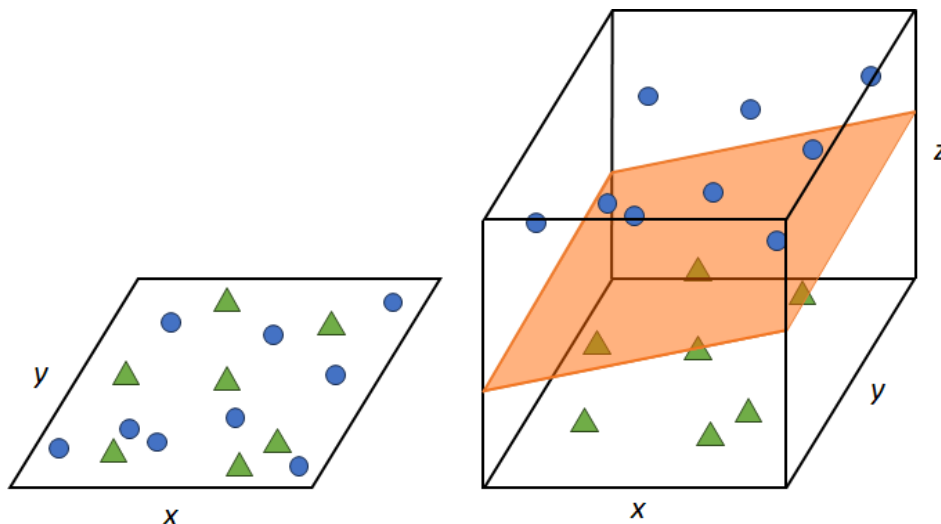


Figure 2.32 A Visualisation of the projection of 2D datapoints into a higher dimensionality by a kernel so that a hyperplane may bisect the two classes cleanly.

Many kernel functions exist, however, the most common are the linear (Equation 2.4), polynomial (Equation 2.5), and radial basis function (RBF, Equation 2.6) kernels. In these equations, x and y are vectors of features computed from input samples.

$$k(x, y) = x'y$$

2.4

$$k(x, y) = (x'y + b)^d \quad 2.5$$

$$k(x, y) = \exp(-\gamma \|x - y\|^2), \quad \gamma = \frac{1}{2\sigma^2} \quad 2.6$$

In the polynomial case, two parameters can be tuned to best effect (b and d). b is the regularization parameter which determines the influence of higher-order versus lower-order terms in the polynomial. d is the degree of the polynomial and controls the working output space dimensionality when calculating data separability. Higher values of d allow for more precise fitting of the data but can cause overfitting where the SVM can become too specialised on its training data and does not work well with new data.

In the case of the RBF, the tuneable parameter is the gamma γ . Gamma defines the reach of each point within the training data. If the gamma is low, each point in the training data will have greater reach, meaning that the decision boundary of the SVM will be highly flexed. Conversely, as gamma increases, the decision boundary will become increasingly linear. Gamma can decrease or increase the overall fit of the SVM model.

The C parameter is another important parameter (hyperparameter) when designing an SVM. This parameter is inversely proportional to the margin size, so a larger value of C decreases the width of the margin. Generally, it controls how strictly the SVM should seek to avoid misclassifying each training sample. A smaller value of C gives the classifier more flexibility but allows it to make more mistakes. C is a dominant hyperparameter when training an SVM as it can greatly improve model performance when properly tuned. However, if mishandled, it will create a model which overfits its training data and is less useful for validation and testing samples.

In most cases throughout this thesis the optimisation for the hyperparameters, including C , has been done via algorithms rather than manually. This increases the likelihood that the models will perform well and not overfit their training data. This optimisation is possible due to how SVM hyperparameters are implemented.

Unlike neural networks, the hyperparameters for an SVM are a single set of inputs. Therefore, it is easier to use algorithms to optimise these hyperparameters rather than relying on manual settings. This is discussed further in Section 2.3.4. SVMs are also less flexible and scalable when compared to neural networks. However, for smaller machine learning tasks they often compete well with neural networks. This is demonstrated throughout this thesis, where simple neural networks and SVMs are equally effective for predicting data classes.

Compared with Random Forests, which are discussed in the next section, SVMs offer similar performance but have slight biases towards different classification tasks. As SVMs use the concept of distance (in higher dimensional spaces) to perform classification tasks, they can work better with data where the features are interrelated. For example, in Chapter 5 the best model was an SVM. This is the case, in part, because the data was a compressed signal, where each feature column was an adjacent value in the compressed signal. SVMs may work less well when working with categorical or weakly related features, as the concept of distance between two samples may be less meaningful.

SVMs were used in this thesis across all three chapters, but primarily in Chapter 5. Radar signals taken from the entrance of a beehive were converted into Log Area Ratios (LARs) and used as both training and testing data for a support vector machine model.

2.3.3 Random Forests

Random Forests (RF) are ensemble learning methods for classification and regression that work by constructing multiple decision trees during training [188]. Decision trees are a simple machine learning structure that creates a cascade of questions to determine which class data belong to. Creating a decision tree involves choosing the most pertinent questions to ask in the form of weighing the variables within the dataset for their ability to split the data into two distinct groups.

For classification, the output of the random forest is the class selected by the largest number of trees. For regression, the average prediction of the individual trees is the returned value.

Random forests are generally preferred to decision trees alone because trees that grow very deep tend to learn irregular patterns and overfit their training sets. They become so specialised that they do not predict data that did not form part of their training set well. The random forest algorithm randomly samples features so that only a subset of variables is used to build each tree. Random forests then average the effects of individual these trees intending to reduce the variance of their predictions.

One subsampling procedure is referred to as bagging. Given a training set $X = x_1, \dots, x_n$ and responses $Y = y_1, \dots, y_n$ bagging selects a random sample (b) from the set, replaces the data, and fits a tree (f_b) to these samples, meaning a sample point can appear in multiple sub-samples. This process is visualised in Figure 2.33.

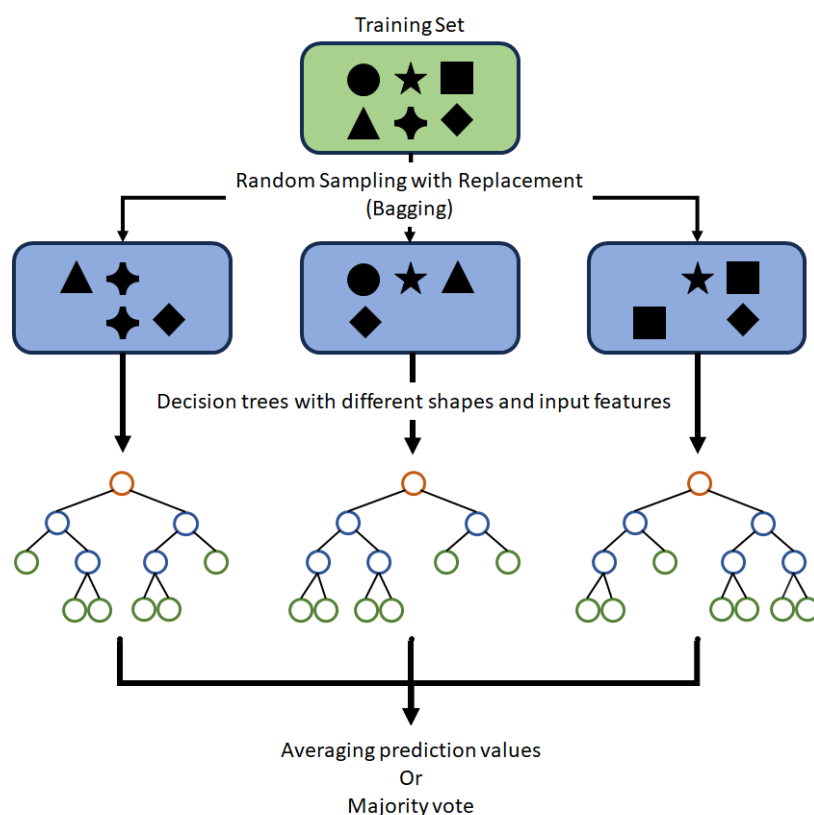


Figure 2.33 Visualisation of the process of generating and training a random forest.

After training, predictions (\hat{f}) for samples x' can be made either by averaging as in Equation 2.7 for regression or by taking the majority vote as in Equation 2.8 for a binary classification case. B is the total number of trees in the forest, therefore the sum of all tree outputs is divided by B in Equation 2.7 for the regression case, and if the sum of tree outputs is less than $\frac{B}{2}$ in the binary case Equation 2.8, then the vote for the overall output is 0.

$$\hat{f} = \frac{1}{B} \sum_{b=1}^B f_b(x') \quad 2.7$$

$$\hat{f} = \begin{cases} 0 & \sum_{b=1}^B f_b(x') < \frac{B}{2} \\ 1 & \sum_{b=1}^B f_b(x') > \frac{B}{2} \end{cases} \quad 2.8$$

Random forests were used in all three data chapters, but are most prominent towards the end of Chapter 3 and Chapter 4 where they deliver a strong performance both at predicting the flight of a bumblebee as it leaves the nest and predicting whether bees recorded by thermal camera are leaving the hive or hovering near the entrance.

Random Forests have several core hyperparameters, the most prominent of which is the number of trees within the forest. Unlike many hyperparameters, the number of trees does not by itself cause overfitting. As Random Forests take the average output of the total number of trees, overfitting is readily avoided because this process reduces variance while leaving any fundamental bias unchanged. Instead, increasing the number of trees will eventually cause a plateau in model accuracy where further trees do not produce better results. Optimising this parameter is simply finding the minimum number of trees to enter this plateau to reduce computational cost.

Another important parameter when creating Random Forests is the split criterion, which measures the quality when branches split within each tree to purify the data on each branch. A pure branch contains only one class of data.

There exist a few split criterion metrics, including loss metrics as explained in Section 2.3.6. Similarly, Gini impurity measures the frequency at which any element of the dataset will be mislabelled when it is randomly labelled. Split criterion is typically a trade-off between small increases in accuracy versus computational expense.

One parameter which can create a degree of overfitting is the maximum depth of the trees. This parameter limits the number of splits that can be performed in any single decision tree within the forest. A random forest of fully grown trees, where splitting continues until all branches are pure, might incur unnecessary variance and become overfitted. Tuning this parameter is a balance between increased training accuracy and overfitting.

It is possible to set the minimum samples to be present in each leaf (terminating node within the tree). A split at any depth will only be considered if it leaves at least this many samples in each of the left and right branches. A maximum number of features can be set to control which features are evaluated when looking for the best split.

In summary, Random Forests are one of the most flexible models which a single set of hyperparameters can control. However, they are still not as scalable or malleable as neural networks. In exchange, they are simpler to design. In this thesis, like SVMs, the hyperparameters for Random Forests have been optimised via an algorithm as they both use a single set of hyperparameters.

Random Forests can suffer when the number of features is large, but the fraction of relevant variables is small. This is because, when considering a branch split, an imbalanced ratio of relevant to irrelevant features decreases the chances that good features will be chosen to form the splitting decision.

Compared with Support Vector Machines, Random Forests can excel when working with features that are weakly related or are categorical. This is because random forests will analyse a subset of the features when making a split decision, making the relationship between features in a series less important. This effect is shown in Chapters 3 and 4 where Random Forest models performed best. The feature sets in these chapters were weakly connected variables describing bee flights captured by either camera or harmonic radar.

2.3.4 Cross Validation Bound Bayesian Hyperparameter Optimisation

Bayesian hyperparameter optimisation (or tuning) is the process of using Bayes Theorem to direct the search to find the maximum or minimum of an objective, in this case, the accuracy (or loss) of a machine learning model by changing the hyperparameters of the models [189]. It is a derivation of Bayes' theorem. Given data E , the posterior probability $P(M|E)$ of a model M is equal to the likelihood $P(E|M)$ of observing E given model M multiplied by the prior probability of $P(M)$ and divided by the probability of $P(E)$:

$$P(M|E) = \frac{P(E|M)P(M)}{P(E)} \quad 2.9$$

Since $P(E)$ is the probability of observing E which is the input, this equation can be rewritten to show that the posterior probability $P(M|E)$ of a model M is *proportional* to $P(E|M)P(M)$:

$$P(M|E) \propto P(E|M)P(M) \quad 2.10$$

Posterior probability is the revised (updated) probability of an event occurring after taking into consideration new information. Prior probability is the probability of an event occurring prior to receiving new information.

Hyperparameters are the parameters that define the structure of the machine learning model before learning is undertaken. Examples include the number of trees in a random forest, the kernel in a support vector machine, and the size and activation function of layers in a neural network. These are chosen before the model is exposed to any data but have a measurable impact on the capability of the final, trained model. Badly chosen hyperparameters can cause overfitting or poor performance once a model is trained.

The process works by iteratively training new models on the data and varying the hyperparameters using a search function to find the optimal values. This is different from manual tuning, where the programmer manually chooses the hyperparameters.

Manual tuning has advantages in that the programmer can gain a deep insight into the problem and its data by conducting their search for ideal parameters. This human learning element can then be used to further improve the data pool itself, for example by expanding the data if it is lacking. It also has advantages if the process requires domain-level knowledge to find the best values. The cons of manual tuning are that the found values are unlikely to be optimal and the process can be time-consuming.

Bayesian approaches keep track of the model capability as they evaluate new sets of parameters which they use to form a probabilistic model mapping hyperparameters to a score on the objective function (such as the accuracy of the model.)

In essence, they build a history based on past calls to the computationally expensive process which is training and evaluating the machine learning model. They then use this history to create a probabilistic model which can predict the next best sample point for hyperparameters that will perform well when calling the expensive training and evaluating process. The point is to reduce the number of calls to this process while maximising the increase in accuracy to save processing time [190].

This probabilistic model is referred to as the surrogate model for the objective function and the steps to a Bayesian search can be understood as:

1. Build a surrogate model of the objective function.
2. Find the hyperparameters that perform best on the surrogate.
3. Test these hyperparameters on the objective function.
4. Update the surrogate incorporating the new information.
5. Repeat steps 2-4 until a maximum number of iterations has been completed or a timer expires.

As an equation, this can be viewed as:

$$x_{opt} = \mathit{arg} \max_{x \in \chi} f(x) \tag{2.11}$$

Where $f(x)$ represents the objective function to maximise – the model accuracy, x_{opt} is the set of hyperparameters that yields the highest value, and x can take any value in the domain χ . χ is the domain the programmer sets, including the minimum and maximum, or set, for all hyperparameters.

A common way of conducting a Bayesian hyperparameter search involves using Gaussian Processes (GPs [191]) as a surrogate. In this case, the function f is assumed to be a realisation of a GP with mean function μ and covariance kernel K , giving:

$$f(x) \sim GP(\mu, K) \tag{2.12}$$

The covariance function describes assumptions about the data. The equations for the distribution of the prediction function, given the training observations, are highly sensitive to the covariation between the test locations and the training locations as expressed by the matrix K .

For the covariance function the exponential square function is a popular choice:

$$K(x_i, x_j) = \exp\left(-\frac{1}{2} \|x_i - x_j\|^2\right) \tag{2.13}$$

Where x_i and x_j represent the i^{th} and j^{th} samples, respectively. The process of determining the posterior distribution of $f(x)$ is:

1. Sample t observations as the training set:

$$D_{1:t} = \{x_n, f_n\}_{n=1}^t, f_n = f(x_n). \quad 2.14$$

2. Based on the function f , compute the function value $f_{t+1} = f(x_{t+1})$ at the new sample point x_{t+1} . According to the assumption of the Gaussian process, $f_{t:1}$ in the training set plus the function value f_{t+1} follows the $t + 1$ normal distribution:

$$\begin{bmatrix} f_{1:t} \\ f_{t+1} \end{bmatrix} \sim N\left(0, \begin{bmatrix} K & k \\ k^T & k(x_{t+1}, x_{t+1}) \end{bmatrix}\right) \quad 2.15$$

where:

$$f_{1:t} = [f_1, f_2, \dots, f_t]^T \quad 2.16$$

and:

$$k = [k(x_{t+1}, x_1), k(x_{t+1}, x_2), \dots, k(x_{t+1}, x_t)]. \quad 2.17$$

The Gaussian process returns the probability distribution over all possible values of f_{t+1} . If the training set is large enough, the Gaussian process can obtain an approximate estimate of the function $f(x)$ distribution.

After obtaining a surrogate of the objective function, Bayesian optimisation uses the acquisition function u to derive the maximum of function f . It is safe to assume that a high value of the acquisition function μ corresponds to a large value of the objective function f .

The simplest acquisition algorithm is the probability of improvement algorithm (PI) which tries to explore near the current optimal value point (x^+) to find the point (x) most likely to prevail over the current optimal value. The search process continues until the number of iterations of the algorithm reaches an upper limit. It can be expressed as:

$$PI(x) = P(f(x) \geq f(x^+)) = \Phi\left(\frac{\mu(x) - f(x^+)}{\sigma(x)}\right) \quad 2.18$$

where:

- Φ is the cumulative distribution function (CDF) of the standard normal distribution.
- μ is the mean of the mean of the standard normal distribution.
- σ^2 is the variance of the standard normal distribution.

However, this algorithm is greedy and only considers exploration and not the degree of improvement. Therefore, the sampling point is in a limited range and can easily fall into the local optimal solution and miss the global optimal solution. By adding term ε , ensuring that the improvement between $f(x)$ and $f(x^+)$ is not less than this amount, then a minimum threshold for improvement can be ensured in the form:

$$PI(x) = P(f(x) \geq f(x^+) + \varepsilon) = \Phi\left(\frac{\mu(x) - f(x^+) - \varepsilon}{\sigma(x)}\right). \quad 2.19$$

Thus, the two core components (the surrogate and acquisition algorithms) allow for the informed search for ideal hyperparameters. However, the process is not without limits. Referring to the descriptive algorithm Equation 2.11, the domain χ is set by the programmer before the Bayesian hyperparameter optimisation. Should χ be sufficiently broad to cover the ideal hyperparameters for any given problem, it may also be broad enough to allow for the model to overfit the training data. To refresh, overfitting can occur when the model complexity is so high that it can, and does, learn to optimise based on noise within the dataset rather than valuable information within the set.

As such, hyperparameter optimisation is often bound to a process called cross-validation. The core concept of cross-validation is:

- Partition the training data into several subsets.
- Holdout a different set each time the model is trained.
- Test the model on the remaining set.
- Repeat the process for each subset of the data.

The most common cross-validation algorithm is the k-fold cross-validation algorithm, where k subsets are created and used as a test while the rest of the data is used to train the model. Cross-validation allows for the detection of overfitting because when part of the data is withheld, the model accuracy will drop when testing this data if the model has overfitted on the remaining data. This process is visualised in Figure 2.34.

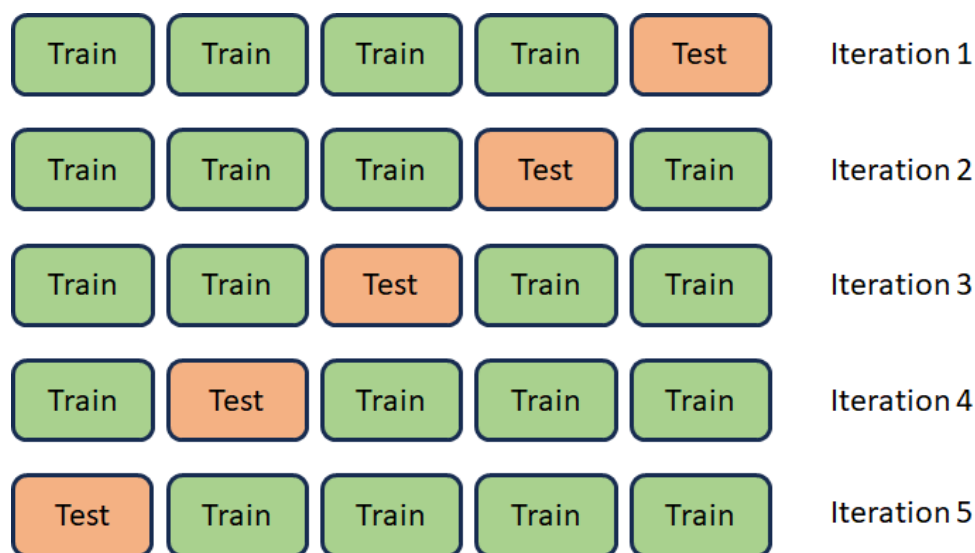


Figure 2.34 A simple visualisation of five-fold cross-validation which depicts how the data would be separated into five sets, with one set being held as testing data for five total iterations of evaluation.

With hyperparameter tuning, the validation accuracy taken as an average across all k folds is used to inform the acquisition algorithm. This reduces the likelihood that overfitting will take place as once it is prevalent the accuracy will begin to drop and the overfitting hyperparameters will be avoided by the optimisation process.

2.3.5 Time Series Analysis

Time series analysis and prediction is one of the primary avenues of future work, alongside those that will be discussed further in Section 6.2. However, time series work was not undertaken for several reasons:

- The unconstrained environment that this project utilised when recording bees. As mentioned in the literature review, most studies have focused on settings where the bees were contained in an enclosed environment making labelling activity a simpler task. Time series needs clear, sequential labelling which can be hard to produce in an open environment.
- Time series is costly as it requires an entire series of data to be labelled, leading to a dense but narrow dataset when labelling is challenging. This is problematic where a wider, more sparsely labelled dataset is preferred. This was the case in Chapter 5 (which provides more information) where different weather conditions were required to evaluate the overall functionality of the radar system across several days.
- It is good to evaluate whether simpler techniques work first before moving forward with more advanced models and systems. One of the first suggestions or criticisms, if time series analysis was unsuccessful, would be whether a simpler model was tested as a proof-of-concept first. This is true for Chapter 3 which was novel in predicting bee tasks upon leaving the nest. Validating that it was possible with a simpler model was crucial, especially since the source dataset was considered small which might be a limitation for time series analysis (more information is present in the chapter.)

Overall, times series analysis is an identified gap within the literature that this thesis does not address. There are clear barriers to utilising time series analysis in this work, but it is recognised that with more resources it would be a natural step forward for the work undertaken.

2.3.6 Performance Evaluation and Metrics

In addition to choosing appropriate machine learning algorithms, it is imperative to choose good metrics by which to measure the performance of such algorithms.

These metrics are used both by the engineer responsible for creating the final machine learning models, as well as the model generation algorithms themselves, to

determine how well a model is performing its task. Metrics used throughout this thesis include:

2.3.6.1 Accuracy

Accuracy is the number of correct responses generated by the model, expressed as a percentage (e.g. 91% or 0.91) in relation to the total number of predictions undertaken. It is often the headline metric used to convey failure or success on behalf of the model. However, accuracy can sometimes not fully express the capabilities of a model. For example, with an unbalanced dataset where one class of data is overabundant versus all others, a model may optimise itself to prioritize correctly predicting this overrepresented class. It may then poorly predict the lesser classes. The accuracy may still be high, due to these lesser classes being a small subset of the data. This may give an inaccurate presentation of how well the model is performing.

2.3.6.2 Loss

Loss in machine learning is a crucial component for many models, particularly neural networks. It quantifies the difference between the predicted outputs of an algorithm and the actual targets. It both considers the number of incorrect predictions and the confidence of the model when making those predictions. A greater number of incorrect predictions and a high certainty create a high loss value. Loss is important to neural networks because their architecture makes direct use of it, first calculating its value and then using it to update the architecture's internal state to reduce loss on subsequent predictions.

The loss value is usually considered ideal as it approaches zero but has no strict upper bound and depends on the choice of algorithm used to generate it. Loss is therefore best used to measure iterative improvement as a model is trained (with the expectation that it decreases over time) or to directly compare two separate models with similar accuracy to determine the best model. A common type of loss for binary predictions is Log Loss. Chapter 3 used this loss to determine that a random forest had room for improvement, which was later utilized, despite a neural network having moderately superior accuracy.

2.3.6.3 Confusion Matrices

A confusion matrix allows the visualisation of results produced by a model to assess its capabilities. A demonstrative example of a confusion matrix is presented in Table 2.1. The actual classes A, B, and C are shown by rows and predicted classes are presented by columns, for example, 4 of class A were predicted as C. An ideal confusion matrix would have all samples present along its diagonal (shaded cells in the table).

Table 2.1 An example of the structure of a confusion matrix.

		Predicted		
		Class A	B	C
Actual	A	24	1	4
	B	3	26	5
	C	7	6	31

Confusion matrices provide a quick and detailed evaluation of the results from a model. They become less useful as the number of classes increases as the overall performance becomes harder to discern from an abundance of information. They cannot present information on multi-label problems (where one sample may be both B and C), which do not feature in this thesis. Nor can they show results for hierarchical classification, where each class of A, B and C may be split by further classification (i.e. into A1 and A2). This type of classification is present in Chapter 5. While confusion matrices were not central to this thesis, the related metrics recall, and precision, and F1-score were used throughout.

2.3.6.4 Recall, Precision and F1-Score

Recall is a measure of a model correctly identifying true positives and is expressed as $\frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$. Like both accuracy and loss, it can be distorted by an

imbalanced dataset. In addition, in a multiclass problem recall must be calculated for each class and then combined into single metric, often via averaging. The averaging strategy plays a key role in evaluating model performance. If there is a class of interest (correctly predicting this class takes precedence over other classes) then it can be weighted to bias the recall appropriately. If classes are imbalanced, bias can be given to the sparser classes to provide a more meaningful value.

Precision follows a similar concept, as it is the ratio between the number of true positive cases and all positive predictions, expressed as $\frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$.

Like recall it can be distorted by some datasets and care must be taken when evaluating performance. Recall and precision are often discussed together as they will provide a deeper understanding than alone, or they are often combined into one metric, the F1-score. F1-score, also referred to as the harmonic mean, is expressed as $2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$.

To summarise, recall focuses as a means to ensure the model is correctly identifying all true positive cases, whereas precision ensures the model is not incorrectly assigning the positive class to a true negative. They are good metrics independently, but recall can be high with precision being low and vice versa. F1-score combines the two and gives a singular metric that provides a measure like accuracy for determining model capability. F1-score may not be appropriate where either recall or precision is valued more than the other, as it gives equal weight to both.

2.3.6.5 Micro- and Macro-Averaging

As discussed, the balance within a dataset can reduce the effectiveness of metrics at describing model performance. Macro- and micro-averaging offer a means to tailor the metrics depending on the dataset being used. For macro averaging the average metric for each class is calculated and then averaged only according to the number of classes. This gives equal weight to each class, regardless of class population. Conversely, micro-averaging also considers the population of each class when computing an average and weights each class by its population.

These weightings can expose classification challenges in problems where there are classes with low populations. This means that if the model performs well on high population classes and poorly on low population classes, metrics will fluctuate. This safeguards against models showing strong performance only because of class imbalance. Micro- and macro- F1-score were used to show that, in Chapter 5, a model was suffering from a class imbalance problem. This allowed for a greater exploration of the data to determine ways of improving the models.

2.4 Video Processing Techniques

2.4.1 Gaussian Mixture Models for Background Subtraction

Gaussian mixture models (GMMs) function on *K-independent* Gaussian distributions and are used to model *K* different clusters [192]. This is like k-means clustering. K-means clustering can be briefly envisaged by the following steps:

1. Choose the number of clusters *K*.
2. Initialise the centre of each cluster.
3. Assign each point in the problem to its nearest cluster.
4. Recalculate the centre.
5. Repeat until the central point ceases movement.

Unlike k-means clustering, each point gets associated with all the clusters with a probability value. To do this, GMMs define their mixtures (clusters) as the following parameters:

- A central mean μ .
- A covariance Σ that defines the width or shape of the mixture.
- A mixing probability π that defines the size of the Gaussian function.

An illustrative example of these parameters is present in Figure 2.35 where graph (a) shows the mean centre μ and the shape-determining factor Σ in a two-dimensional Gaussian map. Graph (b) shows two independent Gaussian distributions in one dimension, their centres μ and the resulting Gaussian mixture with shape determined by π .

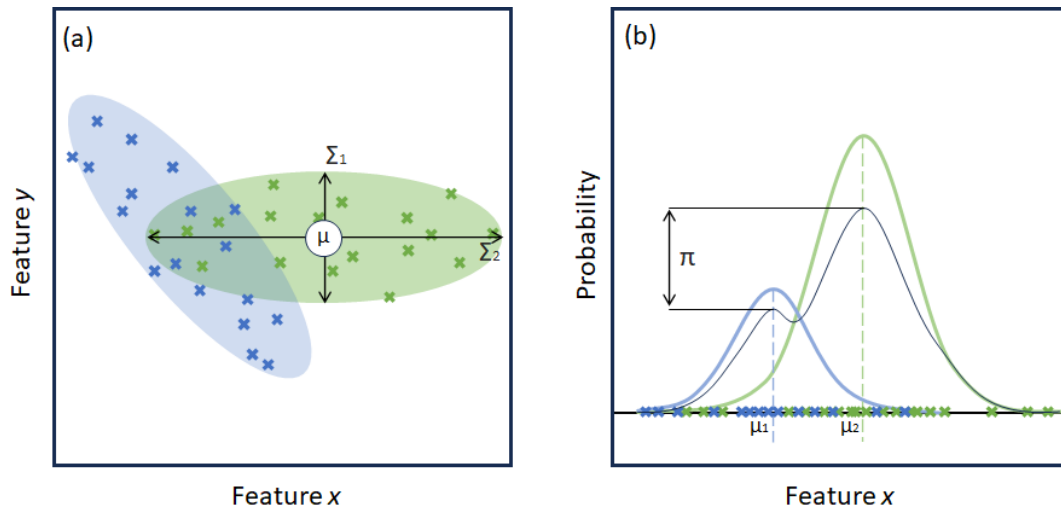


Figure 2.35: (a) Two-dimensional Gaussian distributions and (b) one-dimensional Gaussian distributions and an illustration of the resulting mixture.

These three parameters must be optimised to form an ideal solution, ensuring that each Gaussian fits the data points belonging to each cluster. First, the Gaussian density function is given by Equation 2.20.

$$\mathcal{N}(X|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)\right) \quad 2.20$$

where x represents the data points, D is the dimension of the points. μ and Σ are, as stated, the mean and covariance.

The log of the above equation can be represented as Equation 2.21.

$$\ln \mathcal{N}(x|\mu, \Sigma) = -\frac{D}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \quad 2.21$$

With this, it would be possible to take a derivative, set it to zero, and the problem could be solved. However, this only works for one component and the mixing

coefficients are unknown. In most cases, the more complex, interacting problems between multiple components are solved with the Expectation Maximisation (EM) algorithm.

The EM approach is an iterative one. Firstly, centres (μ) are created for each Gaussian distribution. Initially, this step can be handled by k-means clustering. All points within the dataset are then evaluated based on which centre most correctly describes their location. This is the estimation step.

The maximisation step tweaks values for the parameters μ , Σ , and π such that they better fit the points for which they have the most responsibility. The estimation-maximisation steps are continued until the convergence. This happens when changes to μ , Σ , and π are so small they fall below a predetermined threshold.

With a properly fitted GMM, new data points can be evaluated based on the probability that they belong to any of the K clusters in a way which is more flexible than k-means clustering. However, the process of building a GMM is computationally more expensive.

In this thesis, GMMs are used to segment the foreground in the video recordings in Chapter 4. This problem is common usage for GMMs, and in its simplest form can be thought of as a two-cluster problem. The first cluster is the background of the recording and the second is the objects moving in the foreground. A GMM can be fitted to model the two classes of data. When new frames are received, the difference between the new frame and the learned background image is first computed.

Then, decisions must be made. The differences detected may either be due to an object moving in the foreground, such as a car moving down a road, or they may be minor changes in the background, such as a gentle breeze in a tree at the side of the road.

The GMM is used to predict the probability that each difference belongs to either class. Those that are predicted to belong to the background class are used to update the learned background image. Those that are determined to belong to the foreground class are retrieved for further analysis.

In practice, several clusters are used rather than two. This is because multiple areas of the foreground may have drastically different effects on the background subtraction phase, for example, the difference between a car on a road and a pedestrian on the pavement nearby.

GMMs were used to segment foreground and background in this thesis for three reasons. The first is that foreground segmentation was not a novel aspect of the study, and the priority was to create the best model for predicting bee movement. Secondly, GMMs required less processing power and were easier to train when compared with more recent, deep learning approaches which increased the time that could be spent improving the bee prediction models. Lastly, a deeper learning model can often require more supervision, data preparation, and fine-tuning which would also increase overhead when the focus was better invested elsewhere. Moving to a deep learning segmentation model would be useful in future work to remove any inaccuracies introduced by the GMM system.

2.4.2 Kalman Filter for Movement Prediction

The Kalman filter is an algorithm for estimating the values in a measured system as they change over time, given some measurement uncertainty when those values are evaluated [193]. This filter was used in Chapter 4 to track bees across frames in a video recording.

The Kalman filter assumes the sensors in use are noisy and the output and noise can be modelled as a function of a Gaussian probability distribution. It also assumes that this is true for all measurements taken by the system.

A 2D Gaussian as used by the Kalman filter in this work can be described by a 1x2 matrix describing the centre (or mean) position x and velocity y , and a 2x2 covariance matrix as shown in Equation 2.22 and Equation 2.23. The 2x2 matrix contains the terms Σ_{xx} and Σ_{vv} which are the variances associated with the position (Σ_{xx}) and velocity (Σ_{vv}) along its diagonal. Adjacent to these is the term Σ_{xv} which serves as the correlation between the position and velocity errors.

$$\mu^t = \begin{pmatrix} \mu_x \\ \mu_v \end{pmatrix} \quad 2.22$$

$$\Sigma^t = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xv} \\ \Sigma_{vx} & \Sigma_{vv} \end{pmatrix} \quad 2.23$$

The diagonal of the covariance matrix is the variance of each dimension in the 2D system, and the off-diagonal elements are the covariance between position and velocity.

The filter will update the mean and covariance matrix when new observations are detected, hence this being referred to as the system state.

The Kalman filter, therefore, has two stages. The first is the prediction stage, which tries to predict the current state given the previous stage and the time that has passed. The update stage then combines the predicted state with a new, real measurement.

The prediction stage uses a state-transition matrix (F), or physical model, which relates the previous state to the updated state. The predicted mean for the predicted state is given by Equation 2.24, where u_t is a control signal that is often missing.

$$\mu^p = F\mu^t + Bu_t \quad 2.24$$

The covariance matrix must also be changed as the uncertainty of the system continues. Process noise denoted by Q is the uncertainty created by the time that has passed since the last real measurement and is not accounted for in the base physical model F . The covariance is calculated as in Equation 2.25.

$$\Sigma^p = F\Sigma^tF^T + Q \quad 2.25$$

The update stage works differently. Firstly, it is required to determine the Kalman gain K_g as in Equation 2.26. This gain is a factor using which new information is incorporated into the state and can be tuned to provide the best filter results.

$$K_g = H\Sigma^p(H\Sigma^pH^T + R)^{-1} \quad 2.26$$

In this equation, H is a transformation matrix which transforms the state into the measurement domain and R is the measured state covariant matrix. New measurements can now be taken as μ^m and Σ^m . New versions of both the mean and covariant matrix must be calculated as μ^u and Σ^u as in Equations 2.27 and 2.28 and will be the basis for all new predicted states.

$$\mu^u = \mu^p + K_g(\mu^m - H\mu^p) \quad 2.27$$

$$\Sigma^u = \Sigma^p - K_g(H\Sigma^pH^T + R)K_g^T \quad 2.28$$

In total, these equations give a means of iteratively predicting the state of a moving object, factoring in real measurements, and updating expectations as time progresses. This Kalman filter was used in the thesis to predict the movement of honeybees in a video recording between each frame. This allowed for the tracking of bees from one location to the next. For this thesis, the bees were assumed to be constantly changing speed and therefore the Kalman filter used a constant acceleration model, and process noise was tuned by hand for each video. When a Kalman filter estimates the motion of an object it must account for unknown deviations from the model, this is referred to as the process noise. Since bees are capable of intricate, highly complex flight even in small spaces, the process noise was generally assumed to be large.

2.5 Signal Processing Techniques

2.5.1 Linear Predictive Coding

Linear Prediction (LP) analysis is one of the most capable tools in Digital Speech Processing. Feature extraction of speech is one of the most important issues in speech recognition. Linear Predictive Coding (LPC) is used for source coding of speech signals particularly for compression on cellular networks and represents the spectral envelope of a digital signal in compressed form.

LPC is a source-filter model in that a source goes through a filter as in Equation 2.29. The source, $e(n)$, originally modelled the vocal cords, while the resonant filter $h(n)$ modelled the vocal tract [194].

$$x(n) = h(n) * e(n) \quad 2.29$$

The resonances give rise to formants, which are enhanced frequency bands in the sound produced. These closely match significant local peaks within the spectral envelope. LPC analyses the signal by estimating these formants, removing their effects from the signal, and estimating the characteristics of the remaining signal. This process is called inverse filtering and leaves behind an artifact known as the residue. The formants and residue can then be transmitted elsewhere to reconstruct the signal.

That LPC extracts features from the audio in the form of a compressed spectral envelope is useful for machine learning. Feature selection is a core research area for machine learning, aiming to improve models by removing irrelevant or redundant data which can reduce computation time, improve accuracy, and facilitate a better understanding of the learning model [195]. Selecting good features often requires domain knowledge and often utilises existing algorithms that can describe data succinctly while preserving nuance that is useful to machine learning models.

LPC has gained traction for analysing signals that fit the source-filter model but are not speech or audio signals. Radar signals match this profile in that the source is the reflected waves from the object being measured and the resonant filter is the environmental- and hardware-sourced noise that impacts that signal [196].

LPC determines the coefficients of a finite impulse response filter that predicts the next value in a sequence from current and previous inputs. This type of filter is also known as one-step forward linear predictor. LP analysis uses an all-pole filter described as:

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 - \sum_{k=1}^p a_k \cdot z^{-k}} \quad 2.30$$

where $\{a_k \mid (1 \leq k \leq p)\}$ are the predictor coefficients and p is the order of the filter.

In the time domain, Equation 2.30 matches Equation 2.31, and predicts a signal sample as a weighted sum of past samples.

$$s'(n) = \sum_{k=1}^p a_k \cdot s(n - k) \quad 2.31$$

$s'(n)$ is the predicted value based on the previous values in the source signal. LP analysis estimates the parameters for a segment of signal, the idea being to find $a_k \cdot s$ so that Equation (above) provides the closest approximation. The error, e , between predicted and correct value is:

$$e(n) = s(n) - s'(n) \quad 2.32$$

The summed square error E is used to find a set of candidate equations, as the minimum value of E occurs when the derivative is zero with respect to each of the parameters a_k . The matrix form of these equations is

$$\begin{bmatrix} r(0) & r(1) & r(p-1) \\ r(1) & r(0) & r(p-2) \\ \dots & \dots & \dots \\ r(p-1) & r(p-2) & r(0) \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_p \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \\ \dots \\ r(p) \end{bmatrix} \quad 2.33$$

where $r(i)$ is the autocorrelation of lag i computed as

$$r(i) = \sum_{m=0}^{N-1-i} s(m) \cdot s(m+i) \quad 2.34$$

and N is the length of the signal segment $s(n)$.

The Levinson-Durbin algorithm solves the n^{th} order system of linear equations

$$R \cdot a = b \quad 2.35$$

for the case where R is a Toeplitz matrix and b is identical to the first column of R shifted by one element.

The autocorrelation coefficients $r(k)$ are used to compute the LP coefficients $a_i, i = 1 \dots p$ by solving the set of equations

$$\sum_{i=1}^p a_i \cdot r(|i-k|) = r(k) \quad 2.36$$

where $k = 1 \dots p$.

The following set of equations are solved using Levinson-Durbin recursion:

$$E(0) = r(0) \quad 2.37$$

$$k_i = \frac{r(i) - \sum_{j=1}^{i-1} a_j^{i-1} \cdot r(i-j)}{E(i-1)} \quad 2.38$$

$$a_i^{(i)} = k_i \quad 2.39$$

$$a_j^{(i)} = a_j^{i-1} - k_i \cdot a_{i-j}^{(i-1)} \quad 2.40$$

$$E(i) = (1 - k_i^2) \cdot E(i-1) \quad 2.41$$

where $1 \leq j \leq i-1$ and $1 \leq i \leq p$. The parameters k_i are also referred to as the reflection coefficients. If $|k_i| \leq 1$ the roots of the polynomial predictor lie within the unit circle and the all-pole filter is stable.

LPCs themselves are not without drawbacks. Line spectral frequencies (LSFs) are often used to represent LPCs for transmission over a channel as they are more efficient while also being less prone to quantisation noise [197]. This is the error generated when the number of bits assigned to each LP coefficient does not allow sufficient differentiation.

Line spectral frequencies are generated from the roots of two polynomials which are constructed from the prediction filter $A(z)$ (see Equation 2.30) as:

$$P(z) = A(z) + z^{-(p+1)}A(z^{-1}) \quad 2.42$$

$$Q(z) = A(z) - z^{-(p+1)}A(z^{-1}) \quad 2.43$$

The LSF representation of the LP polynomial consists of the location of the roots of P and Q . An algorithm to determine these roots is to evaluate the polynomial at a sequence of closely spaced points around the unit circle, observing when the result changes sign which indicates the root lies between the points tested.

LSFs are the de facto representation of LPCs in modern speech coding. However, they are computationally expensive to calculate for many coefficients and therefore limit the capacity for their use for machine learning purposes when many features can be required.

Log Area Ratios (LARs) are another method of representing LPCs that, while not as noise-free as LSFs, are simpler to compute [198]. The relationship between LAR and LPC is:

$$LAR = \log \left(\frac{A_i}{A_{i+1}} \right) = \log \left(\frac{1+a_i}{1-a_i} \right), A_{p+1} = 1. \quad 2.44$$

where a_i is the i^{th} parcor coefficients which can be found by:

$$a = a_i^{(i)}, 1 \leq i \leq p \quad 2.45$$

where $a_i^{(i)}$ is the i^{th} LPC calculated by the LPC model.

LPCs, LSFs, and LARs were used in Chapter 5 as feature extraction techniques for complex radar signatures of bees at the entrance of beehives captured by a 5.8 GHz doppler radar. They proved particularly potent, even over other, more standard approaches to signal classification.

2.5.2 Mel-Frequency Cepstral Coefficients

The mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a signal, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency.

The MFC has frequency bands that are equally placed on the mel scale, which approximates the human auditory system's response to sound more closely than linear frequency bands. Mel frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC [199]. They are computed as follows:

- Take the discrete Fourier transform of a discrete-time signal divided into windows.
- Map the powers of the spectrum into the mel scale using the transform present in Equation 2.46, using triangular overlapping windows.
- Take the logarithmic scale of each of the mel scale frequency windows.
- Invert the mel scale algorithm to return the signal to the standard domain.
- Compute the discrete Fourier transform of the new signal.

$$\phi = 2595 \log_{10}\left(\frac{f}{700} + 1\right) \quad 2.46$$

MFCCs have a long history of information retrieval in music, from MP3 encoding to feature extraction for machine learning. Though unusual in the realm of radar, transforming the scale of the underlying data can be useful. While not directly used in this thesis, an earlier related contribution explored adapting the mel-scale to match radar data to determine if could be useful for classification. This is covered in Appendix C and summarised in the introduction to Chapter 5.

2.6 Munkres Assignment Algorithm

The Munkres Assignment Algorithm is a variation of the Hungarian Assignment Algorithm [200]. An assignment algorithm seeks to distribute n tasks between n workers in such a way to find the minimum assignment cost (assuming each task creates a cost when assigned to a given worker.) This is useful in computer science in general to optimise how work is performed and find the best candidates in a given optimisation problem.

A simple interpretation of the process is as follows, where potential assignments are marked (using prime and star in this case):

1. Create an n by n matrix representation of workers and jobs.

2. For each row of the matrix, find the smallest element and subtract it from every element in that row, creating a zero somewhere in that row (the minimum element, representing the lowest cost assignment for one row.)
3. Find a zero (Z) in the matrix. If there is no starred zero in its row or column, star Z . Repeat for each element in the matrix. Go to step 4.
4. Cover each column containing a starred zero. If all columns are covered the starred zeros now represent a complete set of ideal unique assignments. The process can terminate. If this is not the case, continue to step 5.
5. Find a noncovered zero and mark it with prime. If there are no starred zeros in the row containing this primed zero, continue to step 6. Otherwise, cover this row and uncover the column containing the starred zero. Continue in this manner until there are no uncovered zeros left. Save the smallest uncovered value and go to step 7.
6. Construct a series of alternating primed and starred zeros as follows: Let Z_0 represent the uncovered primed zero found in step 4. Let Z_1 be the starred zero in the column of Z_0 if there is one. Z_2 will denote the primed zero in the row of Z_1 (which is certain to exist.) Continue in this manner until there is a primed zero with no starred zero in its column. Remove the star marker from each zero in the series, place a star marker on each primed zero, remove every prime marker and uncover the matrix completely. Return to step 4.
7. Add the value found in step 5 to every element of each covered row and subtract it from every element of each uncovered column. Return to step 4 without altering any stars, primes, or covered lines.

When the process is finished, each row and column will have one value highlighted with a star marker. This is the optimal assignment for that row and column in the context of minimising the total cost within the matrix.

This Munkres assignment algorithm is commonly used when assigning potential movement-tracking candidates to previously recorded entities. This is how it was used in Chapter 4, where new bee detections were required to be assigned to existing tracks found. It was possible to detect new bees by setting a maximum cost value. This maximum cost would remove a candidate from the assignment problem when a new bee detection was too costly to assign to any existing bee path. Bees

that were not assigned any new detections (because they were not a minimum cost assignment) were flagged and removed from consideration after three frames. Therefore, it was possible to detect bees leaving the frame using the cost assignment.

2.7 Summary and Discussion

This chapter contains a literature review and methodology section. The literature review discusses current tracking solutions for bees and similar animals. It also provides an overview of the current machine learning work that has been done to predict insect movement and behaviour. A theme within the literature review is the importance of understanding the different metrics that can be used to measure the success of a bee tracking system, such as the range, accuracy, and risk to the bees. This thesis prioritises accuracy more than any other metric as the integration of machine learning with tracking systems is a new topic within the field.

The methodology section discusses the algorithms used throughout the thesis, their background, implementation, and some limitations. More information is included within the main chapters as the methodology covers common implementation details.

Several gaps within the literature have been identified. One, time series prediction, has already been discussed, and additional gaps are present. There is a gap in the literature given that no machine learning has been integrated with harmonic radar tracking of bees. Similarly, the literature contains few examples of using radar or cameras alongside machine learning to count the activity of bees near the beehive entrance. This thesis includes three chapters which each address one of these gaps.

In general, the literature review shows that despite concern over the long-term population stability of bees there are few efficient machine-learning integrated tracking systems. Machine learning can automate the data- and labour-heavy process of counting bee activity.

Concerning the research questions for the thesis, this chapter supports the investigation into identifying key technologies to pair with machine learning to track bees. The chapter further supports evaluating the feasibility and success of these systems. Lastly, discovering the limitations of such technologies is important to providing useful output for later researchers.

3 Early Prediction of Bumblebee Flight Task*

3.1 Introduction

This chapter details the development of machine learning models capable of predicting bumblebee flight task shortly after leaving the nest, and was used to publish a paper [201]. In addition to the content of the paper, this chapter adds discussion which contextualises the results for the thesis and adds Section 3.6 which further explores Random Forest models.

Before the onset of this thesis project, the author was contracted to help develop a self-piloting drone to track bee flights using a battery-less piezoelectric tag and antenna system at the university. This created a string of successful experiments in iteratively improving the design of the tag system and the self-piloting software [86, 202]. The original project goal for this thesis was to extend this system to 3000 meters tracking distance.

A brief overview of this system is illustrated in Figure 3.1, showing both a conceptual design and photographs of the bee-tracking drone system.

An immediate concern became apparent due to the nature of the drone and target. The drone in question, the DJI Matrice 200, was a workhorse drone. Able to carry larger payloads at great speed, it lacked agility compared to the bees.

In addition, this drone would not fit between gaps that would accommodate the bee such as small hedgerows, or between farm buildings. Interference could affect data capture when the bee flew under cover such as shrubbery. It was apparent the drone would need a system to either predict the bee's next position or make use of partially recorded flights. Classification of flight tasks could serve as an auxiliary aid to drones that can match the characteristics of the flight at hand.

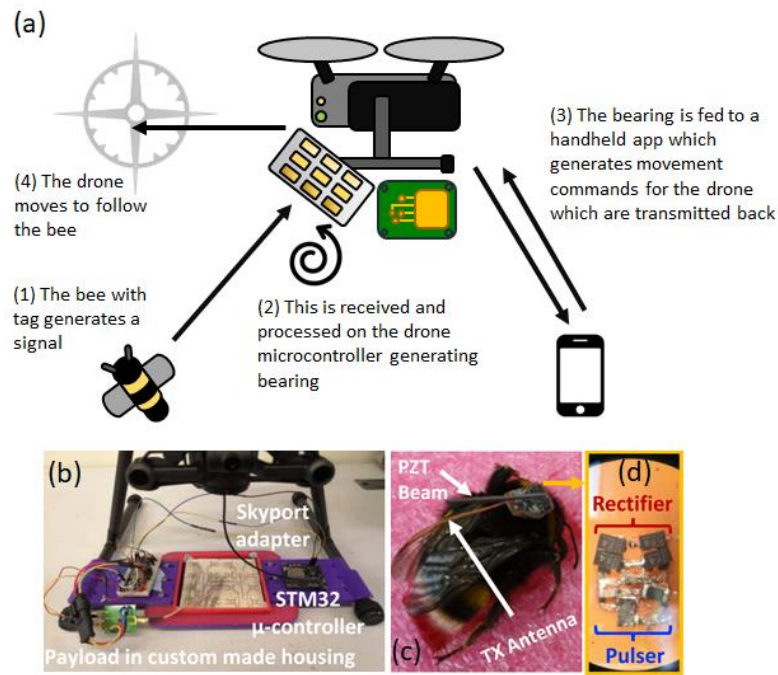


Figure 3.1 (a) The design of bee tracking drone, showing how the signal is converted into command signals for the drone to follow. (b) The antenna system mounted between the drone's feet. (c) The piezoelectric tag on a bumblebee. (d) A closeup of the tag circuitry. An immediate concern became apparent due to the nature of the drone and target. The drone in question, the DJI Matrice 200, was a workhorse drone. Able to carry larger payloads at great speed, it lacked agility compared to the bees.

By making use of incomplete data, a robust algorithm for the early classification of flight purposes could make it possible to monitor how colonies of bees divide their labour resources between exploration for new floral resources and exploitation of those already known. This would allow researchers or commercial users of bumblebee pollinators to monitor the efficiency of pollination, and colony health and predict future needs in time to respond flexibly to them: an increase in exploration flights might suggest that currently known resources are insufficient to support the colony; while a drop in exploitation flights could predict upcoming starvation. Furthermore, these methods would scale well, opening the potential to monitor pollination services over large areas or allow researchers to investigate interactions between colonies in resource exploitation.

In this chapter, data, shared by the Rothamsted Research group, was used to generate models to predict bumblebee flight tasks. This data was taken by their harmonic radar, a system different from the drone with a lower time-resolution.

However, the data provided a useful example of the type of data that might be gathered by the drone.

3.2 Data

Woodgate et al. used harmonic radar for lifetime tracking of bumblebee workers, exploring how their flight patterns developed with experience [203]. The authors described an algorithm for classifying flights into those which explore new resources versus those focused on foraging from known floral sources. This algorithm was inherently simple and effective. An *exploitation* flight was a flight that consisted of a single loop where the bee stopped for a length of time at a location it had previously investigated. All other flights, including those with multiple loops to and from the nest, were considered *exploration* flights. In this work, the concern lies with improving the data acquisition of similar studies and investigating the automated early classification of these flights.

Individual bees seemed to follow a loose pattern of some initial exploration flights followed by periods of exclusive exploitation flights. This, in turn, could be included in the dataset as information to assist with prediction. This is true only if each bee can be uniquely always identified. Radio-frequency techniques do not easily afford this. Therefore, it was decided to exclude this so that the algorithm generated can function with a wide range of technologies, some of which may not uniquely identify the bee.

Some fixed conditions could influence the outcome such as experiments being undertaken during daylight hours in good weather. The primary concern with the dataset was that there were too many potential variables to fully describe all drivers that might affect bee behaviour. This range includes things as simple as local temperature up to small undetectable air currents.

The data consisted of 37591 unique data points taken from 244 bee flights. It represented coordinates as the distance in meters from the source nest. Using this data in this study allowed for the extraction of meta-data such as current speed, average speed, distance from the nest, and perpendicular distance from the average bearing. A summary of these is given in Table 3.1.

An additional metric was used as described in the original work, named digressiveness. This measure was a numerical representation of flight efficiency, with a value of 1 representing a perfectly efficient flight. Given that the work here focuses on machine learning, a small adjustment was made to this metric to enable better normalisation. This original equation is shown in Equation 3.1 with the changes made as in Equation 3.2.

$$D = \frac{\textit{Flight Distance}}{2 * \textit{Optimal Distance}} \quad 3.1$$

$$D = \frac{\textit{Flight Distance}}{\textit{Optimal Distance}} - 1 \quad 3.2$$

These changes were made as the emphasis is on early detection of a task, where the insect may not yet have returned to the nest, therefore only the outward journey is important necessitating the removal of the factor of two in the original equation.

With this new equation, a perfect flight would be $D = 0$ rather than $D = 1$ as per the original specification, allowing better normalisation of digressiveness over the series. A simple demonstration is present in Figure 3.2 showing the digressiveness value for a selection of lines.

Table 3.1 Available variables within the source dataset and derivative variables used to train the machine learning models.

Source	Variable	Description/Count
Source dataset variables	Exploration flight readings	24285
	Exploitation flight readings	13306
	Flight number	Unique flight ID
	Bee number	Unique bee ID
	Digressiveness	See Equation 3.1
	Date and time	Date of flight and time of day
	X, Y position	X and Y coordinate (meters) from nest
	Radar X and Y	X and Y coordinate (meters) from nest
	Nest latitude, longitude	Global position of nest
	Radar latitude, longitude	Global position of radar
Bee latitude, longitude	Global position of bee	
Additional, calculated variables used as model input	Current speed	Distance between coordinates over time between measurements
	Average speed	Average distance covered over time
	Distance from nest	Distance (meters) from nest
	Perpendicular distance from average bearing	Distance between average bearing coordinate and current location (both equidistant from nest.)
	Digressiveness	See Equation 3.2

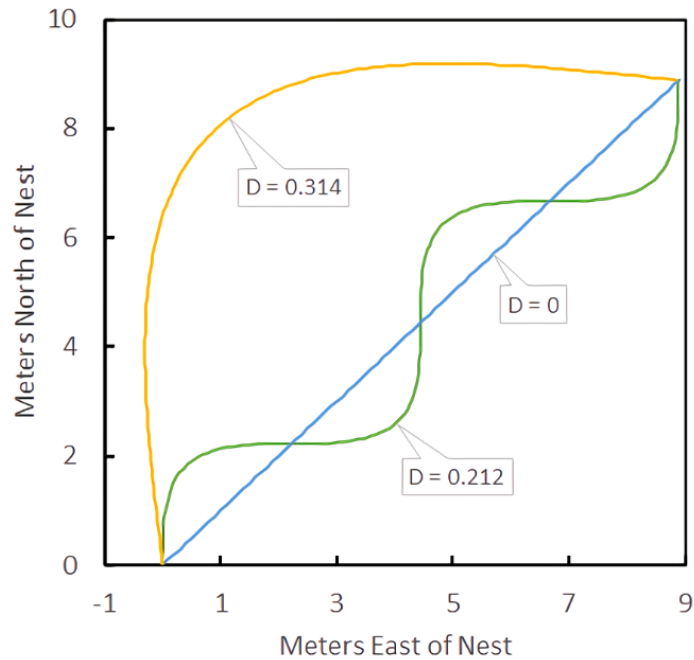


Figure 3.2 Example digressiveness metrics for hypothetical bee flight patterns.

It is noted that additional metrics could be used that describes the bumblebees' trajectory from the nest, which would likely aid in classification such as the exact heading from North which would highlight the bearing of flowers the bee is targeting. However, this work aimed to strip positional information from the data to allow any generated models to work with other colonies with different geographical features. By not including bearing as a metric, the characteristics of the landscape for this nest are not dominant in determining predictions. Because the data is expressed in more general terms such as distance from the nest, current speed, and average speed, the data from other colonies might be substituted effectively.

3.3 Filtering and Preparation Methods

There is a caveat when using the data in that localisation error can occur [203]. For instance, occlusions such as those caused by the bee flying behind a tree would mask the detection leading to dropouts in the dataset. An example of this is shown in Figure 3.3, detailing the missing elements of otherwise strong tracks. In the case of flight 83, there is a large distance gap which could affect the data. Similarly, in flight

130, the bee was likely busy collecting pollen from shrubbery for some time which led to it being occluded. In the current work, the concern is with when the bee is both moving and able to be seen therefore both tracks must be filtered down to the relevant parts.

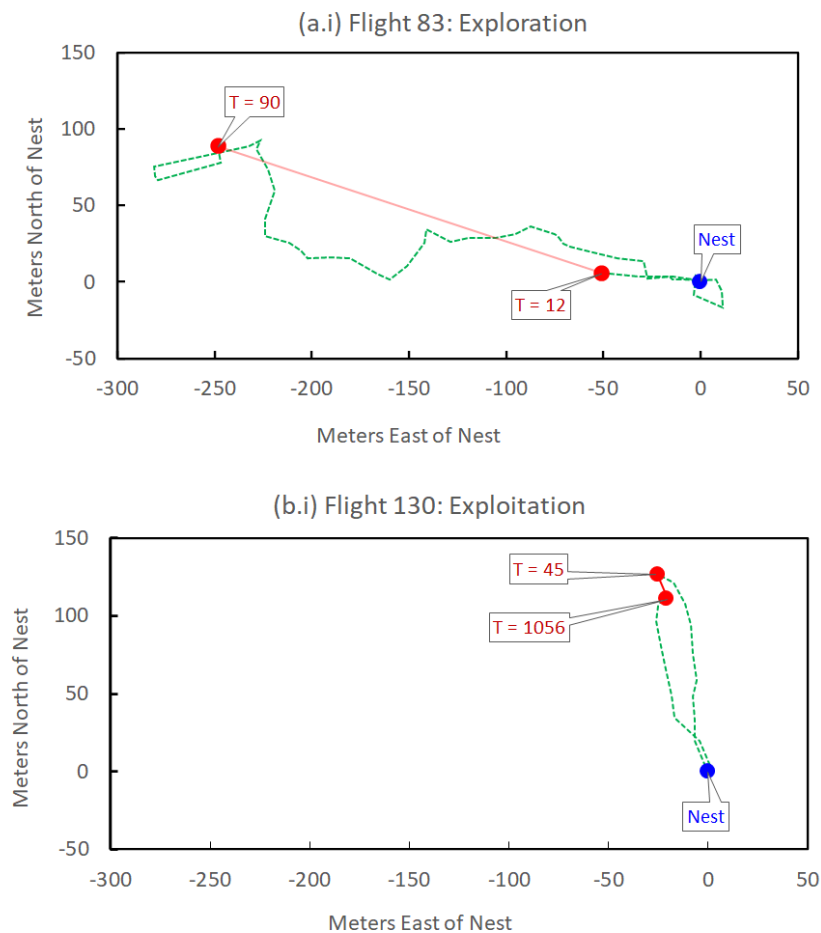


Figure 3.3 Details of (a) flight 83 and (b) flight 130 showing lost segments of flight (as solid red lines), with T being time since track beginning.

As the work was focused on the early prediction of the bumblebee flight task to replicate the conditions faced by the drone (which was to benefit from and continue from this work), a sampling window of 50 positional readings was used. This is the first 50 positions read by the radar if they satisfy the data filtering process and was equivalent to two and half minutes from first to final detection. This was chosen as a reasonable time for the drone to track the bee while still to make a prediction as to

the bee's flight purpose if the bee signal was subsequently lost. An additional reason for limiting the number of points for consideration was that some bee flights were extremely long with one case of a bee not returning until the following day.

To aid the machine learning, an additional set of filters was originally designed to strip out noisy values (potential radar errors or where bees were behaving erratically) from the dataset.

After filtering for lost time, the secondary chosen approach for filtering the data was to focus on three variables: current speed, perpendicular deviation from current bearing, and digressiveness. Perpendicular deviation takes the bearing of the flight, including the current point, and evaluates the absolute distance between the average bearing and the bee's true position, serving as a method of understanding how quickly the bee changes direction without using specific positional information.

The distribution of unfiltered data was explored by normalising the data between 0 and 1 for each feature to view the distribution of values. This was done by subtracting the minimum value and dividing the result by the remaining maximum. This process was reversible so that realistic limits could be evaluated to determine if they improved data distribution by inspecting the data distribution pre- and post-normalisation. These limits were determined to be a max speed of the bumblebee of 8 meters per second, a perpendicular deviation of no more than 8 meters, and a digressiveness of 2 or less. The normalised distribution of the data is presented in Figure 3.4.

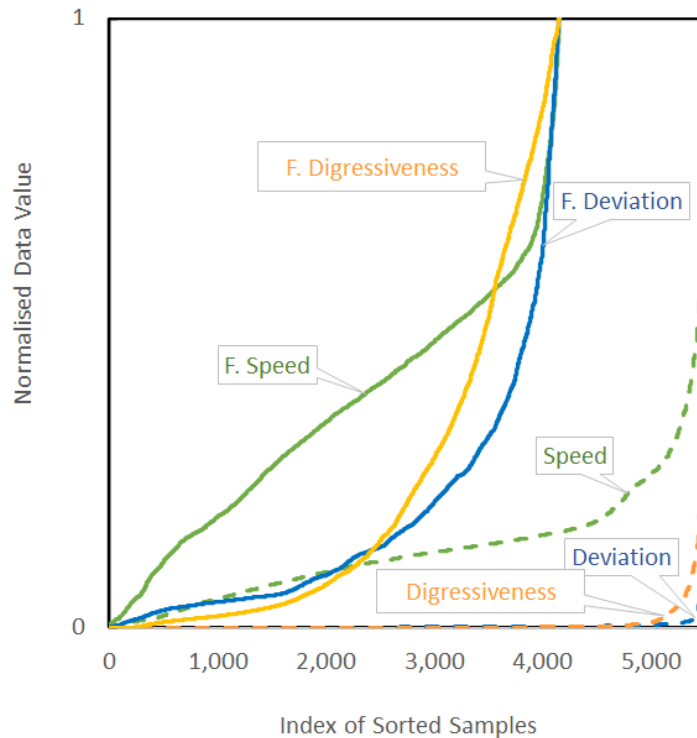


Figure 3.4 Filtered (Solid line) and unfiltered (Dashed line) normalised data distributions of the dataset.

The filtration of digressiveness is key as some rare flights had multiple, short-distance occlusion losses. Coupled with the radar's accuracy (two meters) this could give the appearance of the bee looping multiple times in a short distance, skewing results.

It is also noted that the limitations on speed are less than with similar research where the maximum speed of a bumblebee is around 15 meters a second [50], however, this could be due to previous research noting a range of 3 to 15 meters a second depending on environmental conditions such as wind resistance.

3.4 Initial Experiment

The filtered data was split into two similar-sized sets for both classes (n~1130 each) before being fed into the following models;

- A random forest classifier (RF [188]) of 1000 nodes initialised to a random state.
- A support vector machine classifier (SVM [187]) with a radial basis function kernel, regularization of 1, and a kernel coefficient of 0.5.
- A neural network (NN [204]) with two hidden layers of scaled exponential linear units (SELU) with a final sigmoid activation layer and a dropout rate of 0.5.

The size of each dataset is the number of points present in the first 50 readings in each flight, filtered based on the discussed parameters. There was an approximate 3:1 ratio of available points for exploitation and exploration respectively pre- and post-filtering.

Under-sampling was first utilised as imbalanced datasets are challenging to train models upon. To strengthen the conceptual results of the work, the learning outcomes for both filtered and unfiltered data are included. This allows the demonstration of patterns discovered from filtered data matching patterns in unfiltered data.

To confirm prediction validity, both accuracy and loss were evaluated. Binary cross-entropy loss, also known as log loss, was used as a loss function for all the models generated. This loss can loosely be interpreted as the proportion of incorrect predictions produced by the model in a set, in addition to its confidence in those predictions. A perfect loss would have a value of zero, as demonstrated in Equation 3.3. In this case, i is the index of a given prediction, y_i represents the target value output and \hat{y}_i is the predicted value.

$$loss_i = y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log(1 - \hat{y}_i) \quad 3.3$$

To calculate the loss for both the RF and SVM, these models were created in scikit-learn with SVM probabilities determined by Platt's method [205].

For the original learning outcomes, an 8:2 ratio was used to split data into training and testing sets for the models, with the neural networking taking a small sample

(20%) of its training set to act as a validation set. In this method, the flights were disassembled into constituent context points and reformed into randomised composited sets.

In the case of the neural network, the check-pointed model with the highest accuracy against the validation set was used as the outcome to limit any residual overfitting.

For predictions over time, 180 flights were used as training with 64 flights held for testing. As the filtering process could shorten a flight to less than 50 measurements, three separate instances of each model were trained. The 50 measurement limit was, as discussed, to prioritise predicting the bee's task soon after leaving the nest, with 50 readings being approximately two and a half minutes.

One triplet of models would contain one NN, one SVM, and one RF. Each triplet of models shared the same set of flights. The average accuracy across sets of triplets was used as the final measurement.

In essence, each of the three datasets was a random permutation of the original set, split further as discussed into a training and testing set. This served the purpose of reducing possible bias introduced by the training data having a disproportionate number of flights with the full 50 possible values versus the testing set.

An important observation of this data is that measurements were taken every 3 seconds by the radar. The initial measurement was not guaranteed to be the first possible point of the bumblebee leaving the nest as the bee could be obscured by the nest itself or masked by the rotation of the radar. To counteract this, predictions are made from the second point onwards so as not to make faulty assumptions as to a target's current speed.

It is also prudent to mention that Woodgate et al. noted that their classification of flight tasks used an algorithm created to match human observation [203]. However, they note that such classification was not necessarily suited to the nuances of bee behaviour. Notably, they discuss the trade-off between capturing what was happening precisely while also creating a method that was as simple and universal as possible. This creates the possibility that where the machine learning structures

disagree with the original method, some of these instances may be due to the more complex nature of machine learning and represent the ground truth. This also formed part of the study.

To investigate further, the unfiltered data were explored using Ward's method of clustering to build a set of data labels by calculating the incremental sum of squares. Following this, the method works by creating a simple nearest centroid classifier to estimate a label based on proximity to the nearest Ward cluster centroid [206]. As shown in a dendrogram present in Figure 3.5, there are indications that there exist multiple sub-flight types. Clustering the data into six, eight, ten and twelve clusters allowed investigation into whether successive points shared clusters, rather than neighbouring points belonging to unrelated clusters. This would indicate sub-flight characteristics. Further evidence that sub-flight types were present came as neighbouring points continued to share clusters each time the number of clusters varied. The goal was to determine if more than the described two categories of flight exist by predicting them over a small set of flights. This also allows exploration of whether multiple flight categories can be present in a single flight, indicating the possibility of either adaptive tasks or strict tasks.

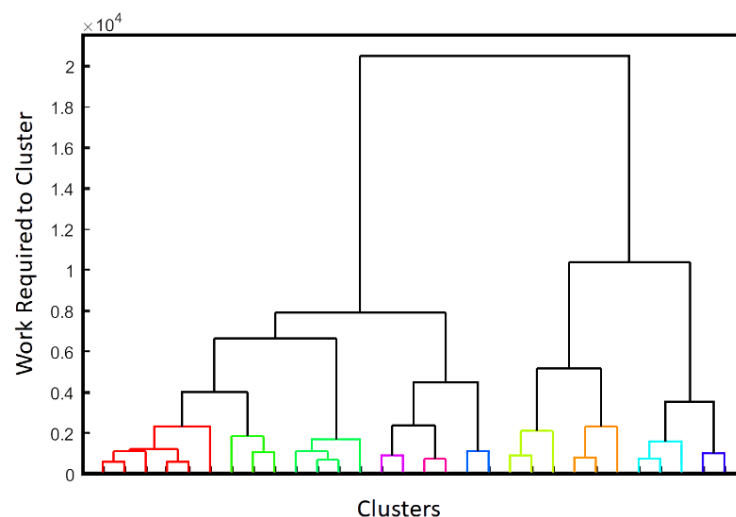


Figure 3.5 Dendrogram of unfiltered data, with the cluster threshold set to ten to match cluster analysis.

3.5 Initial Results and Discussion

A final accuracy, on the initial 50 readings of each flight, was achieved as 91%, 81%, and 85% for the neural network, support vector classifier, and random forest classifier respectively. This is the result when flights were disassembled and individual points re-composited to form sets, stripping them of intra-flight relationships. These are detailed in Table 3.2, showing that the random forest performs very similarly regardless of filtering the data, even doing somewhat better with unfiltered data. Both the neural network and the support vector machine benefit from filtering the data, however, the SVM does have lower loss with unfiltered data. These results do indicate that the patterns within the data exist both with and without filtering.

As the dataset was limited to 244 flights, it was not possible to disaggregate the dataset by either ambient temperature or time of day. For example, the random forest improved to 90% accuracy and 0.27 loss when the time of day was included. However, this may be an overstatement of the algorithm's capabilities. After approximately 6pm, all recorded flights (6 total flights) were exploitation flights. This could mean that bees leave the nest that late in the day only for food or that there was not enough data to capture the truth of the matter, so the algorithm would likely (incorrectly) assign exploitation to all flights in this slot. More flights would provide the correct ratio of classes for the algorithm to attribute a label. Given this limitation, time was not used though it may be reincorporated as the dataset is further expanded. This also supports previous arguments in favour of keeping the variables as general as possible.

Table 3.2 Initial Learning results: Strongest results in bold.

<i>Method</i>	<i>Filtration</i>	<i>Accuracy</i>	<i>Loss</i>
Neural Network	Filtered	91.0%	0.42
	Unfiltered	75.2%	0.60
Random Forest	Filtered	85.1%	0.35
	Unfiltered	86.9%	0.32
Support Vector Machine	Filtered	81.3%	0.59
	Unfiltered	71.2%	0.55

More interesting for this study was the accuracy versus flight time. As each subsequent point contains more context about the flight, such as a more refined value for average speed, it is expected to see an increase in accuracy over time.

It proved prudent to evaluate the results as an average of models across three sets. To take the example of the neural network, results were 50%, 69%, and 47% for each triplet respectively on the initial prediction (2 measurements taken.) While some sway is to be expected, there is a 22% accuracy shift between the weakest and strongest results. For comparison, the RF managed 74%, 71%, and 73%. This is a much more typical sway with a gap of 3% accuracy.

However, as time progresses this shift tapers out such that at the peak accuracy of the neural network, the results are 94%, 93%, and 90%. Comparatively, at this point, the RF returned an accuracy of 79%, 81%, and 72%, now producing a 9% shift in accuracy.

The likely reason behind these shifts is due to imperfect filtering of the data, a smaller number of test flights than ideal, a residual error left from the radar itself, and the discussed random flight order in the dataset. As previously mentioned, the first reading of the bee was not always adjacent to the nest which created the possibility of feed-forward error as points taken for learning purposes contain continuously more context about the flight. Flights were validated on whether their initial distance from the nest (x) was less than or equal to the distance between the first and second reading (y). Assuming acceleration upon leaving the nest, the bee must have been airborne for longer than the three-second rotation of the radar if x is greater than y . 53% of recorded flights began with the bee already beyond y distance from the nest, making speed at these early points unknowable.

With a larger number of flights recorded, an adaptive filter could be developed such that these issues may be addressed. For example, having a lower speed limit at the start of a flight. This might be expected as the bee accelerates away from the nest and would serve to curtail the use of flights where the bee had already covered some distance. With 244 flights, such an adaptive filter would lack the proper context to correctly determine a mean value for this speed.

The binary cross-entropy loss across model triplets is much more stable, though the trend for SVM results to spike remains. This supports the idea that accuracy issues stem from data points that are either incorrectly filtered or form part of a set that was labelled incorrectly by the original algorithm.

As shown in Figure 3.6, the loss values are strongest with the random forest. The SVM results lack stability, and while both the neural network and random forest start competitive, the random forest in time outpaces the competition.

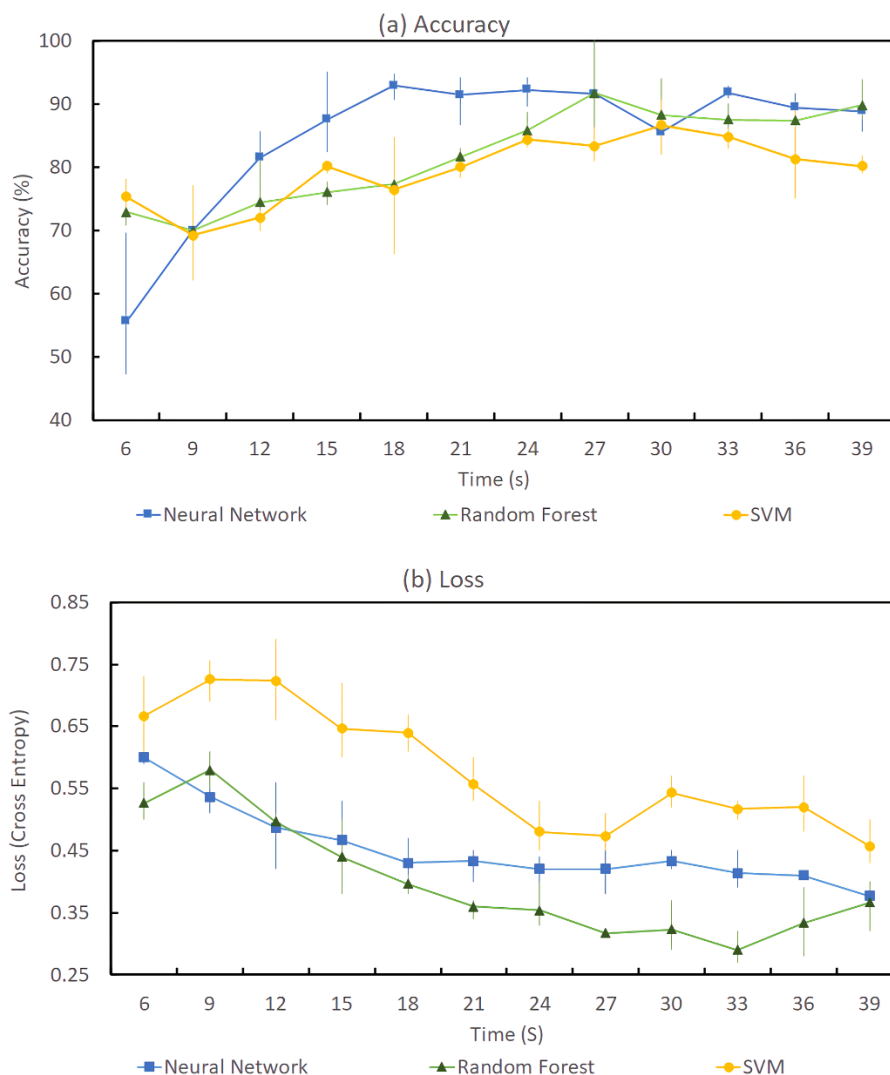


Figure 3.6 (a) Average accuracy and (b) loss over time of the models trained on the dataset. Error bars are for the min and max values across model triplets. The horizontal axis is the time from the first detection by the radar, regardless of the time of day.

Conversely, with accuracy, the results favour the neural network. It is worth noting that both the random forest and support vector classifiers manage the initial

classification with higher accuracy than the neural network. However, the neural network reaches its highest accuracy faster, with 81% at the 4th measurement (12 seconds) and a peak of 92% at the 6th measurement (18 seconds.) The neural network is also the strongest model over the entirety of the dataset.

Observing both loss and accuracy together, the neural network is strongest. Before 18 seconds, the neural network and random forest are almost equal in loss value, but the neural network quickly climbs to its peak accuracy whereas the random forest takes much longer to reach similar results.

Another way of interpreting these results lies in looking at model predictions on specific flights. Figure 3.7 shows two flights, one exploration (flight 233) and one exploitation (flight 89.) Flight 233 had a total flight prediction accuracy of 90%, though even at its third prediction it made a mistake. Flight 83 had a final accuracy of 75% and made a substantial error on its ninth prediction by producing incorrect results with almost certainty.

Both flights produce a majority vote in favour of the correct result within the first four measurements, supporting the initial findings presented here.

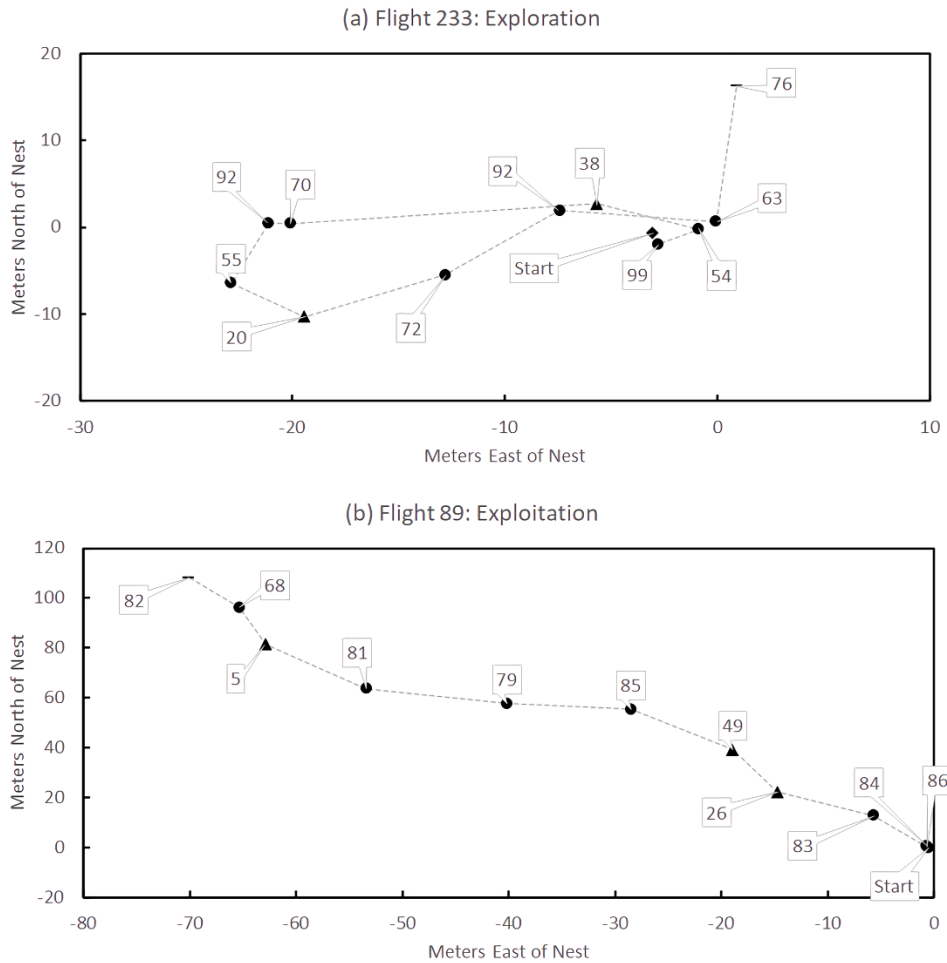


Figure 3.7 First 12 positions of two flights, (a) flight 233 and (b) flight 89, with confidence scores. Scores below 50 indicate an incorrect prediction represented by ▲. A score of 0 would indicate total confidence in an incorrect prediction, and a score of 100 indicates a perfect prediction. Correct predictions are marked by ●.

Interestingly, these early results provide context surrounding some of the incorrect predictions created by this approach. Flight 64 is an exploration flight, yet the neural network predicted only 10% of the points correctly. Context is provided in Figure 3.8 and shows that for the first 12 measurements, this flight bears a striking resemblance to flight 89. However, the full flight plot shows the characteristic looping and backtracking associated with exploration flight.

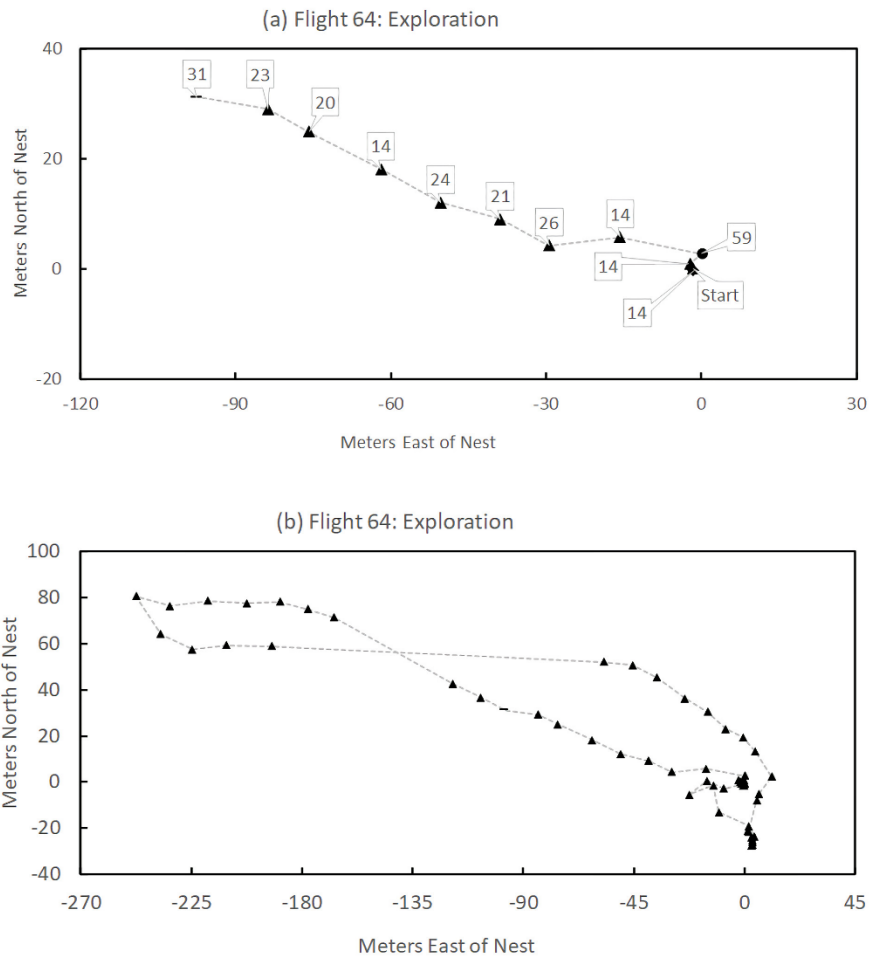


Figure 3.8 (a) First 12 samples of flight 64 with predictions plus (b) total plot of flight 64 showing characteristics of exploration.

As mentioned, cluster analysis was also performed. No filter was used, however, as with the models themselves, only the first 50 measurements were evaluated. The goal was to determine if there is justification for there being more than two types of flight present in the dataset and whether flight 64 could be classed as a hybrid flight.

Ten clusters were formed from the entire dataset minus flights 64, 89, and 233. The cluster number was chosen to achieve good granularity when separating these flights into sub-flight types, allowing a comparison to be drawn between the structure of the flights. With these cluster centroids, predictions were undertaken using the one-nearest neighbour classifier approach. Results of predictions for flight 64 in Figure 3.9 show that segments of the flight remain intact, rather than there being

randomly clustered data. This indicates that even a relatively simple algorithm can segment a flight into similar sections. Furthermore, there appear to be repetitive clusters based on flight parameters. Using the example of cluster 5 (C5), it is present before the major arc of the flight and immediately following, indicating the same behaviour both before and after this arc.

It is important to note the functions of the clusters would require further analysis. This would also likely need a larger sample of flights to provide better definitions.

However, it is important to evaluate the possibility of hybrid flights, or perhaps flights that do not fall under the umbrella of either exploitation or exploration.

Looking at the composition of the three flights in question, again shown in Figure 3.9, shows that it is likely that 64 does fall under a hybrid category. Like the exploitation flight 89 that it was mischaracterised as, it contains four component clusters rather than the two present in exploration flight 233. On the two overlapping clusters C5 and C6, flight 64 falls directly between flight 89 and 233 in terms of proportion. The exploration flight is strictly composed of these two clusters, with an overwhelming majority of C6 (88%). Flight 64 is much more like 89 with 52% and 38% respectively.

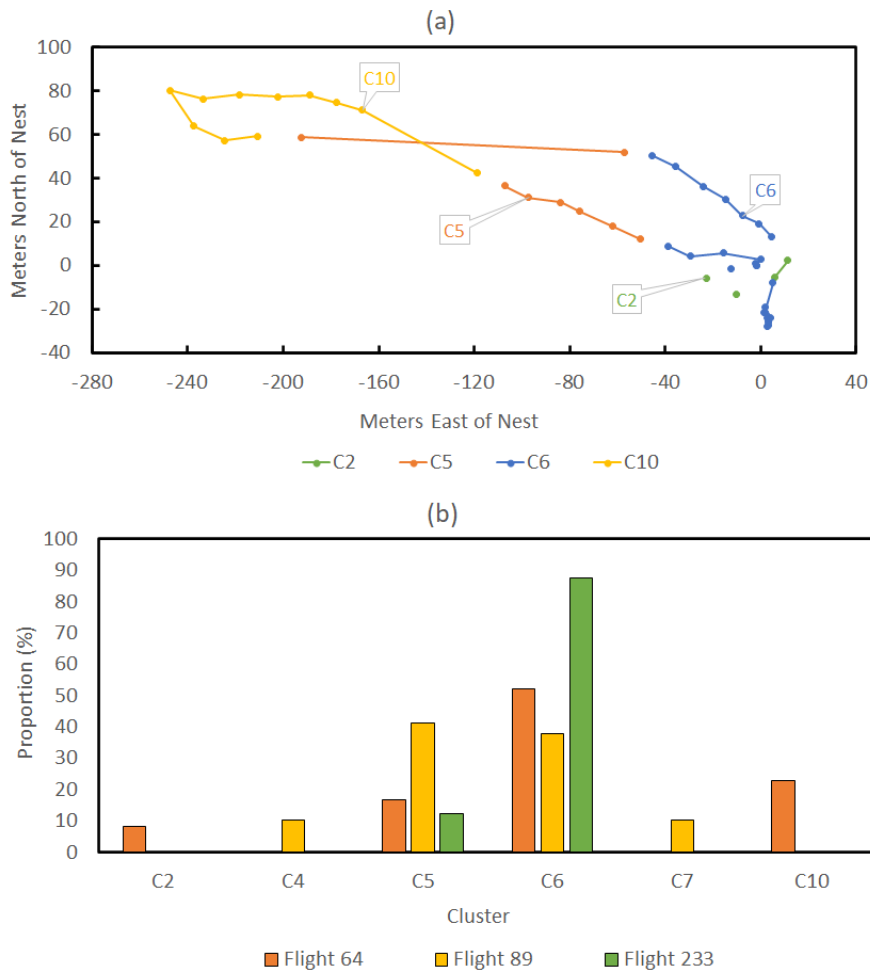


Figure 3.9 (a) Flight 64 labelled with nearest centroid classifier and (b) proportional constitutions of flights 64, 89, and 233 of their component clusters.

An expanded look at this experiment is presented in Figure 3.10. The distributions of clusters between flights remain reasonably consistent, with flight 64 appearing to be somewhere between the exploitation and exploration flights. This is true until there are only six clusters, where the flight patterns become similar.

This is evidence that hybrid flights exist. In this case, it could be that the bee went in search of additional food after a first visit. Further work could investigate correlating clusters with other behaviours. Verification of these clusters would need to be completed, seeking to find meaningful and resilient functions for each. Some clusters may trivial behaviour, such as a cluster being: Bee flying cruising towards or away from the hive. Such clusters would not be relevant to overall flight function.

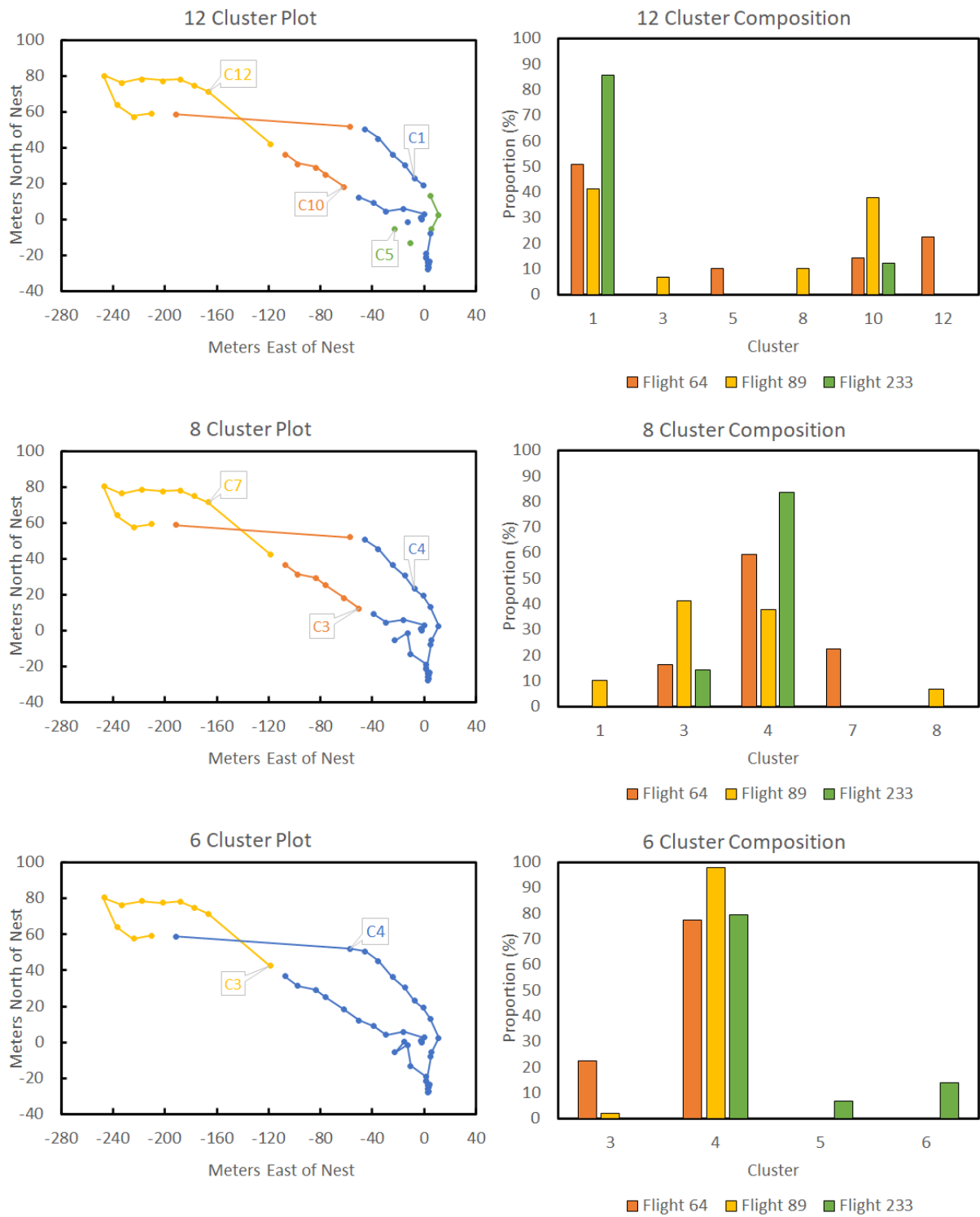


Figure 3.10 The results from clustering when varying the total number of clusters.

3.6 The Dangers of Subsampling, Manual Tuning, and the Question of Loss

The low loss value for the Random Forest encouraged further investigation. In addition, subsampling leaves large portions of the data unexplored, meaning that results could be different when using a different subset.

Sample weighting is a means to allow a model to use the entire dataset by giving the underrepresented class a greater weight when fitting the model. It is not a guarantee of a better model but is a useful way of measuring how the model might perform when exposed to the entire dataset.

Hyperparameters are the parameters that define the Random Forest such as the number of trees, the depth of each tree, and the criterion that causes a branch on each tree to split. By setting these manually, the model is limited by the operator. Hyperparameter tuning is a method of using a search function (such as a Bayesian Search) to determine the best fit for these parameters automatically. This would be expected to improve results moderately.

The hyperparameter limits for the random forest were as follows:

- Between 10 and 1000 trees in increments of 10.
- A split quality criterion of either gini impurities or entropy.
- A number of samples to split of either \log_2 total features or the square root of total features.
- A max tree depth of 10 and 1000 in increments of 10.
- Minimum samples to split in set [2, 5, 10, 15].
- Minimum samples per leaf in set [1, 2, 4, 6, 8].

Lastly, rigour can be improved by using cross-validation. K-fold cross-validation splits the dataset into k number of subsets (folds) and uses one as a testing set while training on all others. It iteratively ensures that each fold is used as a testing set once, meaning for 10-fold cross-validation, ten versions of the model are generated.

Putting all these techniques together, a random forest model was trained using weights with all the data. The results of Figure 3.11 show how the model performed in each fold of a 10-fold cross-validated training and testing session. Both filtered and unfiltered data have been evaluated. The average accuracy for filtered data was 87.96% and for unfiltered was 90.72%.

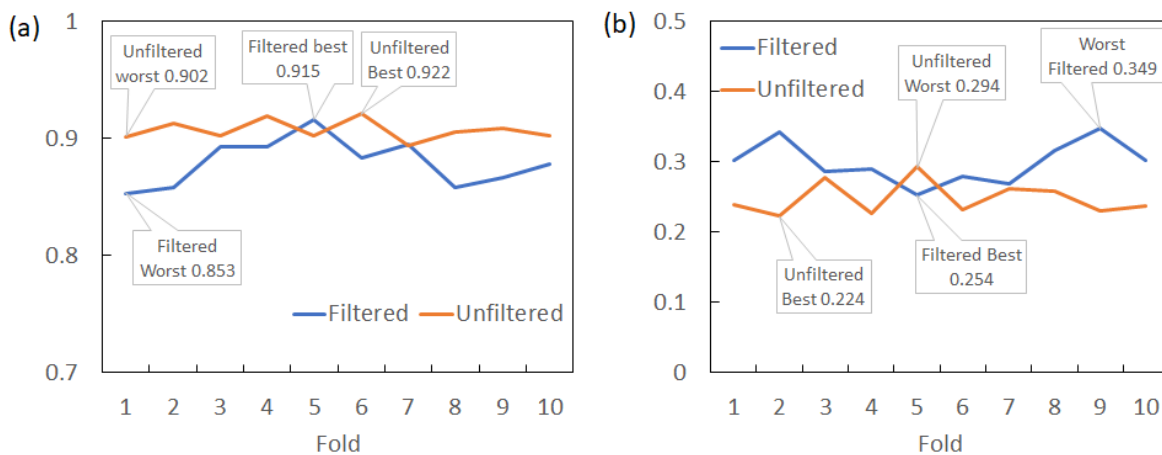


Figure 3.11 (a) Accuracy (b) Loss - Per fold results from the 10-fold cross-validated experiment.

This follows the under-sampling results, where the unfiltered accuracy surpassed the filtered accuracy. The random forest is resilient to the effects of the unfiltered data and does better with the whole dataset present. The best-case accuracy for both filtered and unfiltered data also surpassed the neural network on the under-sampled data, however, the averages are a better measure of overall performance.

The experiment was also conducted by varying the number of folds. The results of this experiment are presented in Figure 3.12. Again, the model does better with unfiltered data, showing less variance as the number of folds increases. This variance for filtered data is likely to be caused by the smaller folds containing harder to classify data that do not have a representation in the remaining training set. The filtered data surpasses the unfiltered data in the best-case scenario, approaching 95% but this is not a stable result for the set because it is a one fold in many and does not represent the entire set.

The high accuracy and low loss for this dataset are evidence that the random forest did not benefit from the filtering algorithm. When coupled with the hyperparameter

tuning and sample weighting the random forest remained fully capable of predicting flight labels. In the future, with a larger dataset, it would be beneficial to investigate whether this result remains. It could be that filtering the dataset removes easy-to-classify outliers, rather than adding noise into the set which makes learning difficult.

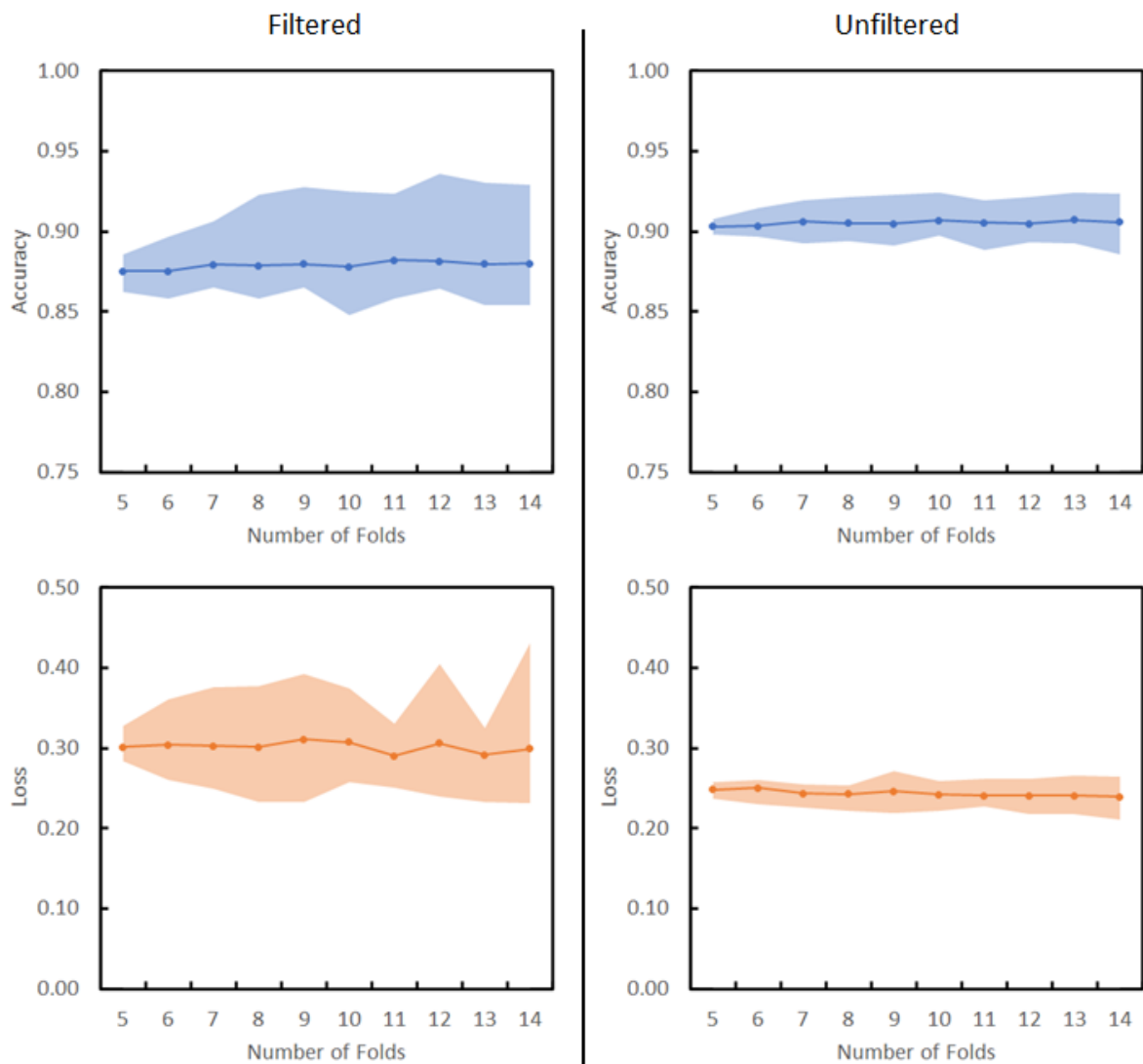


Figure 3.12 Results from varying the number of cross-validation folds in the experiment. The line shows the average result. The shaded area illustrates the range encapsulated by the minimum and maximum results.

3.7 Summary, Limits and Conclusion

This chapter detailed the development of a bee flight task classification system with a focus on early prediction. There were models created that were more than 90% accurate at predicting this task. A breakdown of different filtering techniques is presented as well as their effect on the model accuracy. In addition, k-fold cross-validation and automated hyperparameter tuning have been investigated to improve the models.

Limits exist in the methods used in this chapter. Whilst there are many thousands of individual points, there were only 244 flights. Results from the accuracy-vs-time assessment are only proof of concept. It was always the goal of this thesis to expand upon this work and provide a larger dataset for similar evaluations. Future work may face challenges as more hard-to-classify flights are recorded by the equipment at the University.

This is a weakness that cannot be addressed with the current data. Data of this kind are still rare and require expensive, sparse equipment to gather. The drone project that was to be done as part of this thesis was designed to alleviate this very problem. However, the project was discontinued when the hardware engineer who created the sensor system left the university and there were insufficient funds to hire a replacement to develop and maintain the equipment.

Conversely, studies such as this, though limited, are crucial in laying the groundwork and securing interest for future more thorough work using larger datasets. It is hoped that the drone project will be revived, and a more complete version of this work can be undertaken.

The final validation for the algorithm must be tested in the field. Real-time execution of the algorithm with third-party assessment would refine measurement precision. This could be in the form of observing bee pollen load on return to differentiate between foraging and other flight types. Additional harmonic radar data gathering will enable further classification of the detected clusters (for example predation and disease). Early classification of behaviour types can improve unmanned drone airtime both in prioritising bees to save battery life and in enabling adaptive flight to

match bee flight patterns. This in turn will serve as more proof that the algorithm is correctly predicting to match field observations.

These proof-of-concept results show that a rapid automated prediction of bumblebee flight tasks could be possible. This is a prediction of whether a bumblebee is exploring new resources or exploiting ones already known.

These results open pathways to expand on the radar tracking of insects by allowing fast determination of flight errands. This could allow for automatic prioritisation of exploring bees over foraging bees for longer-range tracking to build maps of nest resource acquisition. Similarly, it would allow for a shorter-range system to sit near a nest to monitor the number of bees leaving for each task over time, without needing knowledge of a bee's destination.

With more development, this would lead to being able to monitor resource use and pollination efficiency in near-real-time so that interventions can be made to improve them, such as moving nests to help them make better use of crops or providing supplementary food when colonies are in need.

Further work could augment the process to predict the future needs of the colony and nest health. Too many exploration flights might suggest they are not getting enough food; too few exploration flights might suggest the colony is over-reliant on a small number of food sources and will be in trouble if those plants stop flowering.

This could scale to monitoring multiple colonies over a large area, such as an entire farm, and allow for moving colonies to areas that are not getting enough pollination. In addition, it would be possible to monitor wild bees to work out where conservation resources should be concentrated and researchers can start to look at how the foraging decision of one colony affects another. This could provide insights into how the pattern of nest foraging vs exploring affects the nest's overall health.

Time series analysis was not undertaken in this chapter but is recognised as future work. The limiting factor was the small number of total flights (244) which would present challenges for time series analysis. More data would be required to fully explore the technique, a limitation that this work overcomes by working with points

within flights as disconnected prediction tasks. In addition, the work focused on providing a proof-of-concept that demonstrated that a bee's task could be predicted upon leaving the nest. This work is novel within the field and the results show that a comparatively simple model is successful, without exploring time series analysis. Gaps in the detections of bees when they were foraging within bushes, as well as the variable lengths of flights, might be barriers to time-series analysis. However, time series prediction may provide the means to detect multi-purpose flights, detecting the transition from one flight behaviour to another. This would be particularly useful in the drone project by, for example, detecting that a bee intends to return home just before the drone loses sight of it. The drone could fly back to the nest and validate the return of the bee, without needing to track it the entire way.

Given the 3-second time delay of the tracking system, a faster system, such as the one in development, might be able to produce better results. The additional resolution offered by more measurements in a shorter time frame offers a potential boost in performance. This may result in classification being possible in a shorter period which would further reduce the range requirement of a classification device, thereby potentially making them easier and cheaper to manufacture.

These models represent proof of concept that real-time evaluation of bumblebee tasks can be carried out successfully and could aid in automated tracking solutions for bumblebees and other colony insects.

4 A Comparison of Machine-Learning Assisted Optical and Thermal Camera Systems for Beehive Activity Counting*

4.1 Introduction

This chapter details the development of twin camera systems to classify outward bee flights from the entrance of beehives, and was used to publish a paper [207]. In addition to the content of the paper, this chapter adds further context for how the results relate to the concerns of the thesis. No additional sections were added to this chapter.

This study was inspired by the work of fellow student Sara Bariselli. Her work is focused on determining whether the foraging and exploration flights of bees are affected by the local magnetic field.

As part of her requirements, she needed to reliably gather the details of all outwards flights from a hive across the trial period. This totalled hundreds of flights per hour of investigation, a labelling task that would take weeks to conduct manually but could be achieved trivially by sufficiently capable machine learning.

Some flights were difficult to classify. As with the work of Susanto, some flights exiting the hive were not true foraging or exploration flights [126]. These were instead local short flights which were not relevant to the work. There exists an abundance of literature about using cameras to track bees [208]. However, few of these use machine learning to classify flight behaviour.

In addition, studies so far only used visible spectrum camera systems. A noted weakness of this approach is the lack of adaptability of such systems to difficult lighting conditions. For example, optical cameras require good lighting and contrast to accurately count activity and are susceptible to shadows. During periods of insufficient natural light bulbs can be used but the heat and light generated by these may impact bee behaviour [208]. Similarly, infrared sensors require the modification of a hive entrance by adding an emitter [209].

Thermal cameras have the potential to overcome all of these drawbacks. Capturing the body heat of the bees as a contrast image versus a background, they will operate in any environment where the temperature is lower or higher than that of the bee. As shown in Figure 4.1, the bee's body is not uniform in temperature and they will appear contrasted against most backgrounds. The thorax of the bee is warmer than the abdomen and legs, therefore should the environmental temperature match the thorax the abdomen and legs will still appear against nearby objects.

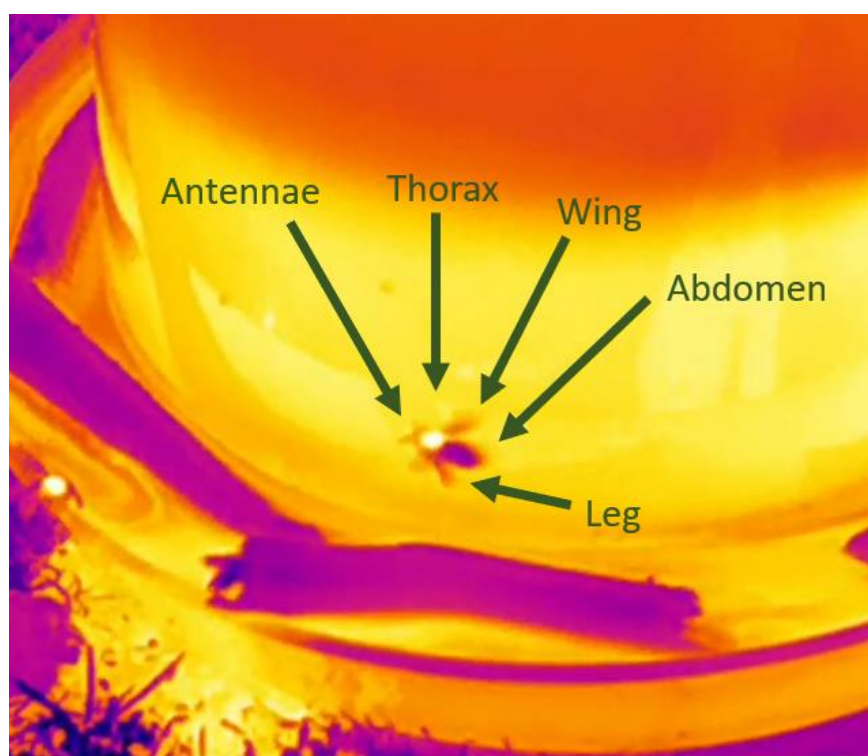


Figure 4.1 Anatomy of a honeybee under thermal camera, flying in front of a warm sucrose solution feeder. The thorax was $21^{\circ}\text{C} (\pm 1^{\circ}\text{C})$

This work implemented a hierarchical camera system where a higher resolution, wider field of view optical camera can provide information as to whether a bee is leaving, entering, or hovering near the hive entrance. To summarise, this work is an investigation into whether machine learning would allow a thermal camera, with lower base specifications than a competing optical camera, to operate with the same efficiency counting beehive entrance activity. Showing that this is the case by using

machine learning to boost the thermal camera, the work demonstrates the versatility of thermal cameras and is an argument for their use considering that they do not rely on lighting conditions, suffer from the effects of shadows, or need modification to hive entrances. The use of movement characteristics for machine learning creates an opportunity for future work to predict the presence of predators such as the Asian Hornet (*Vespa Velutina*) by using an expanded version of the system developed.

4.2 Materials and Methods

The thermal camera used was a HT-301 designed by HTI-Instruments. It is a microbolometer camera with a resolution of 384 by 288 pixels, operating at 25 Hz with a field of view of 28.2° by 21.3°. The primary optical camera used was a GoPro © Hero 7 Black model with a resolution of 1920 by 1440 pixels, operating at 30Hz with a field of view of 94.4° by 72.2°.

The 25 Hz framerate represents how frequently the resistance is measured from the pixel sensors, which are affected by an increase in temperature caused by incoming infrared radiation at the target frequencies. Larger, more sudden shifts in resistance take longer to dissipate resulting in variable data change rates within the frame, causing fast-moving objects to appear as streaks.

A Panasonic © compact system camera (DC-GX800) using a resolution of 1920 by 1080 pixels, 50 Hz framerate, and variable field of view was used as the reference camera. This camera was used when a flight could not be confidently labelled solely on the information of the two test cameras. These cameras are shown in Figure 4.2.



Figure 4.2 (a) The thermal camera (b) The GoPro © camera and (c) the Panasonic © camera

The two primary cameras were suspended above the entrance of the hive facing downwards at a distance of one meter, whereas the ambiguous decision camera was fixed to a tripod approximately 2.5 m from the hive directly facing, and parallel, to the hive entrance. Cropped views from each camera are shown by Figure 4.3, indicating their relative positions. Raw pixel values from the thermal camera are interpreted through a colour map embedded in the software provided by the manufacturer. The particular colour map used made no difference to the tracking software and was chosen to aid with the label correction process. The thermal camera had the option to embed minimum, maximum, and centre-point temperatures as labels into the video frames. The labels would act as moving objects within the frame and would interfere with object tracking tools. Therefore, this overlay was disabled resulting in only the contrast images being recovered.

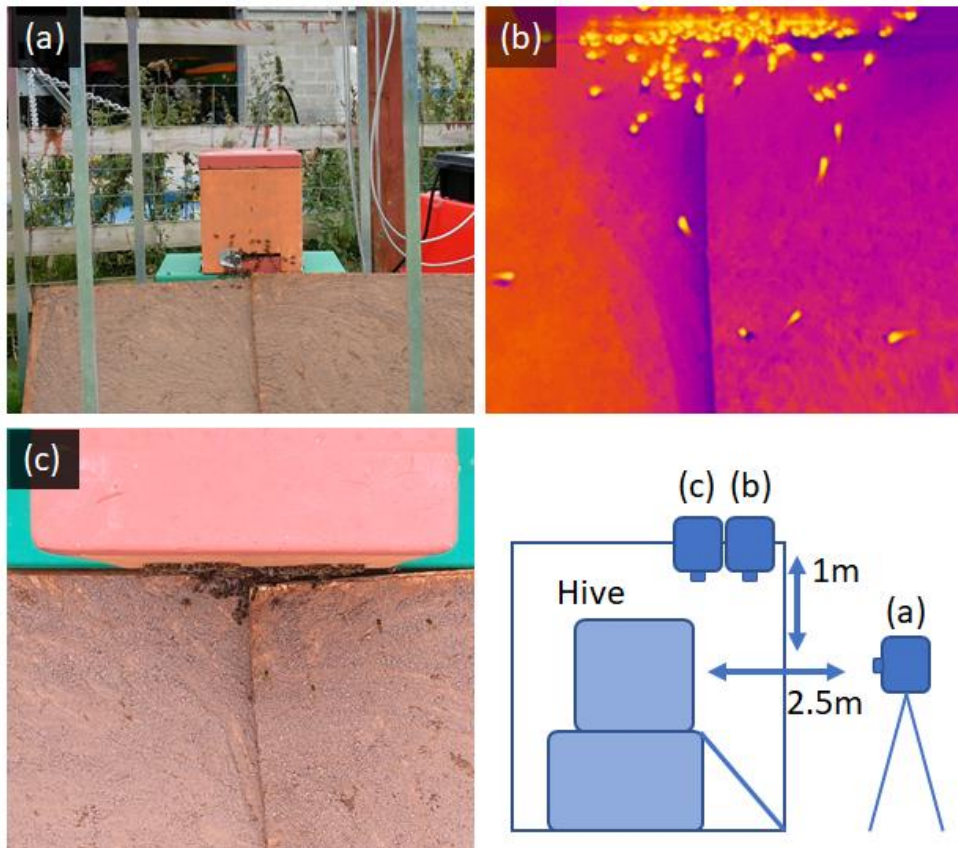


Figure 4.3 Cropped views from (a) Ambiguous decision camera (b) Thermal camera (c) Optical camera, and a diagram of the camera arrangement.

Videos were recorded across six days, between 10 am and 4 pm, with between 30 min and an hour of footage per day. Ambient temperature varied between 12 °C and 16 °C. Bees were given warm sucrose feed at least two hours before recording to encourage activity. Artificially increasing bee activity allowed an improved understanding of the model performances under high load. The feeder was placed more than fifty meters away from the hive to reduce unnecessary hovering activity. Overcast days were favoured to produce a neutral environment for the optical camera. In bright conditions, this camera would be susceptible to bee shadows creating false detections that would impact the optical track extraction and labelling algorithms, but also required sufficient light to detect the bees. Techniques exist in literature to compensate for other conditions but thermal cameras do not need these, necessitating keeping a fair environment for both cameras.

Flights were extracted from the raw video files using software written in MATLAB. Gaussian mixture models generated a foreground detector by comparing each frame in the raw videos with the learned background model [192]. The parameters for this foreground detector were selected by hand for each video from each camera. The detected shapes were analysed based on their size to determine eligibility for being considered bees, aiming to remove small shapes resulting from minor movements such as blades of grass.

Each eligible shape was assigned a Kalman filter unless otherwise attributed to an existing flight [210]. Between frames, assigning objects to existing flights was handled by computing the costs as the pixel-wise distance between the predicted locations generated by the Kalman filter from the previous frame's objects and the new detections from the current frame. The Munkres variant of the Hungarian algorithm was used to minimise the total cost of assigning each object to an existing flight [200]. A base cost was used, and tuned much like the foreground detector, to determine non-assignment. Unassigned objects could go on to become new flights and flights that had no new object assignments in three frames were flagged as completed and no longer considered.

At the end of processing, the video resulted in several flights consisting of continued x and y pixel coordinates, the longest and shortest axis of the extracted shapes per frame, the frame time of each detection, and a unique ID. Predicted Kalman coordinates were not included.

Initially, a simple naive filter was attempted. Similar in principle to that found in [133] it used a set of exit points around the edge of the camera frame. Flights that started nearest the hive point and ended nearest the exit point could be labelled as "out" whereas those that started near an exit and entered could be labelled "in." Finally, null points were used near the centre of the frame to allow for labelling of invalid hover flights, those that neither started nor ended near the entrance to the hive.

This filtering system is visualised in Figure 4.4, however, emergent issues were impacting the quality of extracted flight information when considering the field of view of the cameras.

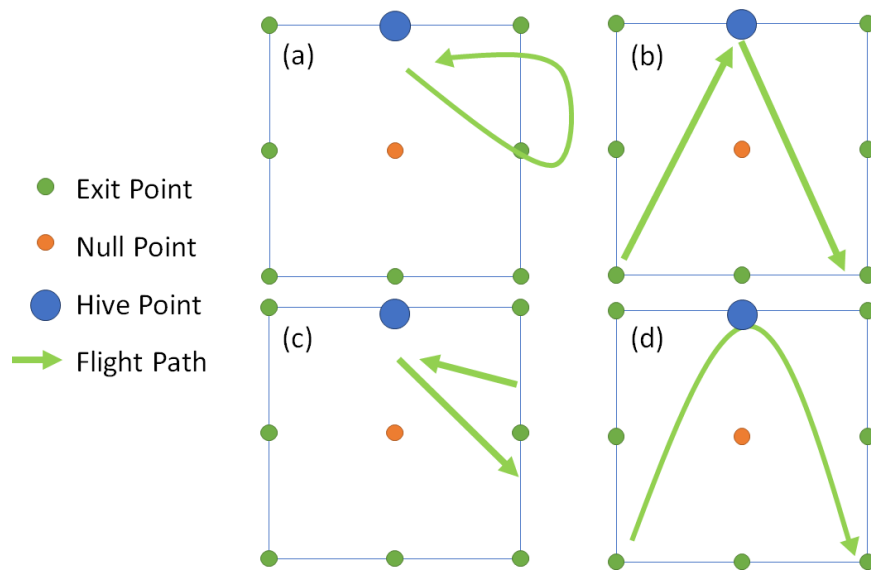


Figure 4.4 Naive filter and visualised problematic flights. Squares represent camera frames and their borders. Flights (a) and (c) show flights that exited the frame border limit, and outgoing and incoming flights may not be the same individual. Trajectories (b) and (d) show how two flights may be fused if the end of the incoming flight occurs within the same frame as the outgoing second flight.

For example, flight (a) in Figure 4.4 starts and ends near the entrance of the hive. It should be discarded as an invalid flight as the bee did not leave the vicinity.

However, it crosses the edge of the frame and appears as in (c). This means one exit and one entrance flight would be recorded. Bees were noted to fly out as far as 1.5 m only to return to the hive, putting them out of view of both primary cameras.

The filter produced issues not unique to the field of view. Flights in (b) are one valid ‘in’ flight and one valid ‘out’ flight. However, if the two flights overlapped it would be possible for the extraction algorithm to fuse the flights, resulting in the flight present in (d) meaning one hover flight was recorded rather than one incoming and one outgoing flight.

One way to discern the difference between these cases was by analysing the dimensions of the extracted shapes. Figure 4.5 shows four flights captured by the camera systems. In (a) and (c) the bee is flying at speed in a straight line, whereas in (b) and (d) it is flying to the side at a slower speed. At the edge of the frame, this may be enough to ascertain the difference between a bee leaving the area and one

soon to return as part of an invalid flight. However, this is much less clear on the optical camera given the lower contrast and higher frame rate.

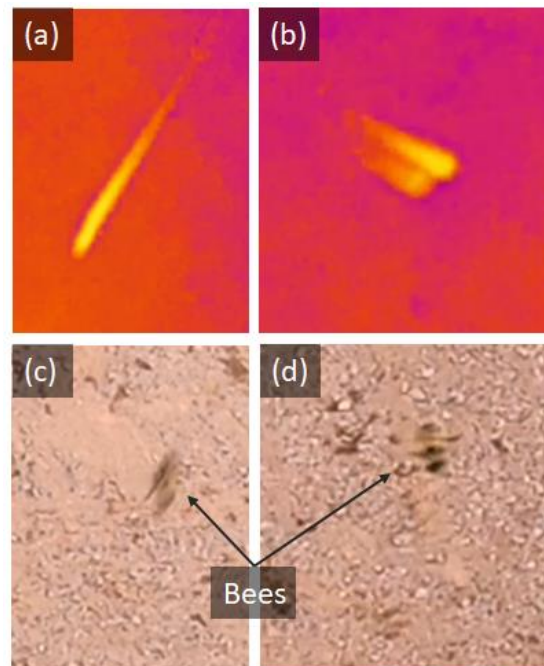


Figure 4.5 Flights showing different exit strategies captured by the cameras. Images (a) and (c) show two different bees flying straight ahead, whereas (b) and (d) show two different bees flying more horizontally. Note the smudged nature of these high-speed flights on the thermal camera, caused by the resistance change delay associated with microbolometer technology.

Also of concern were flights involving a loop where the bee turned around shortly after leaving the hive to face the entrance before continuing its flight as shown in Figure 4.6. As such, classifying these correctly would mean that some bees moving to the side were classifiable as outward flights. This would be true where the flight was truncated by the edge of the frame and a full loop was not captured.

To address these issues machine learning techniques were explored. These aimed to correctly label flights when the edge of the frame caused truncated flights and where other issues with the flight extraction software emerged. Cases such as overlapping shapes breaking the tracking of a flight between frames and fused flights were both examples of issues with the tracking system.

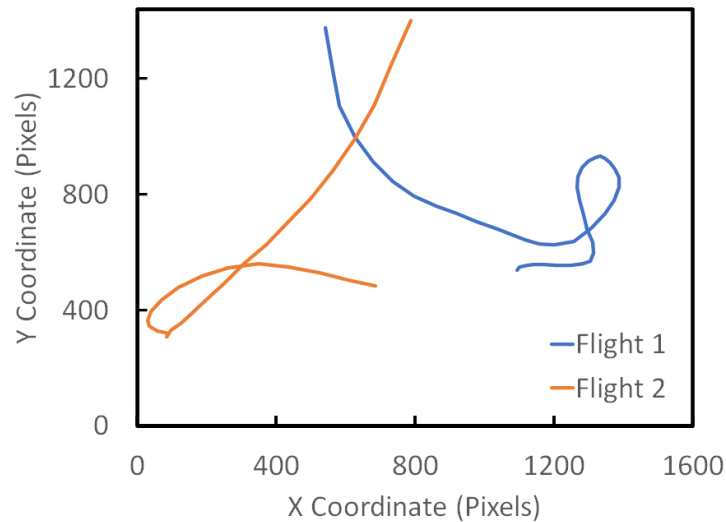


Figure 4.6 Two flights where the bee turned back to face the hive before flying away, behaviour associated with new foragers orientating themselves with respect to the hive. Flight 1 went on to complete a second loop but this was not captured fully by the tracking algorithm.

Incoming flights were not subject to machine learning. This was because the naive filter was adequate for counting inward flights and valid incoming flights would have the same profile by being at the edge of the frame and returning to the hive entrance. A pipeline showing the process of classifying flights is presented in Figure 4.7.

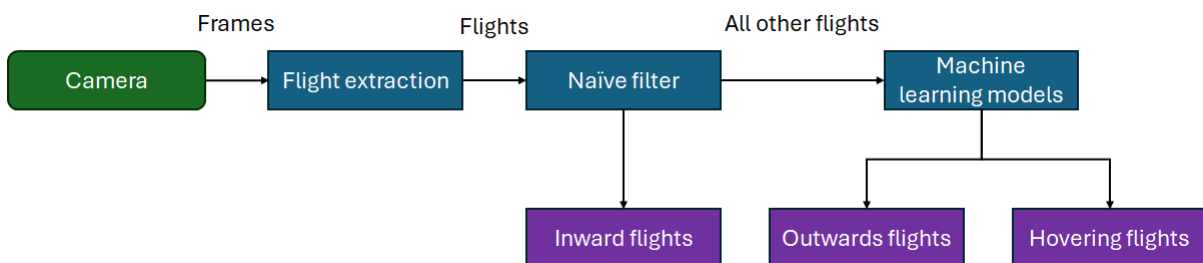


Figure 4.7 Flight classification pipeline, showing the naive filter which removes flights on start and end position. Machine learning models used flight behaviour data extracted from all coordinates for a flight.

Figure 4.8 shows two overlapping flights taken from an annotated video. These two bees (8287 and 8286) have distinct profiles when flying into the hive. Flight 8287 is more direct and linear. Flight 8286 is much slower with one big loop and a final partial loop. Their behaviours could be a result of space at the entrance (8287 entered first), the size of their bodies, and many other unknown factors. As such, predicting specific behaviour from these types of flights (such as return from hover/return from foraging) would prove difficult.

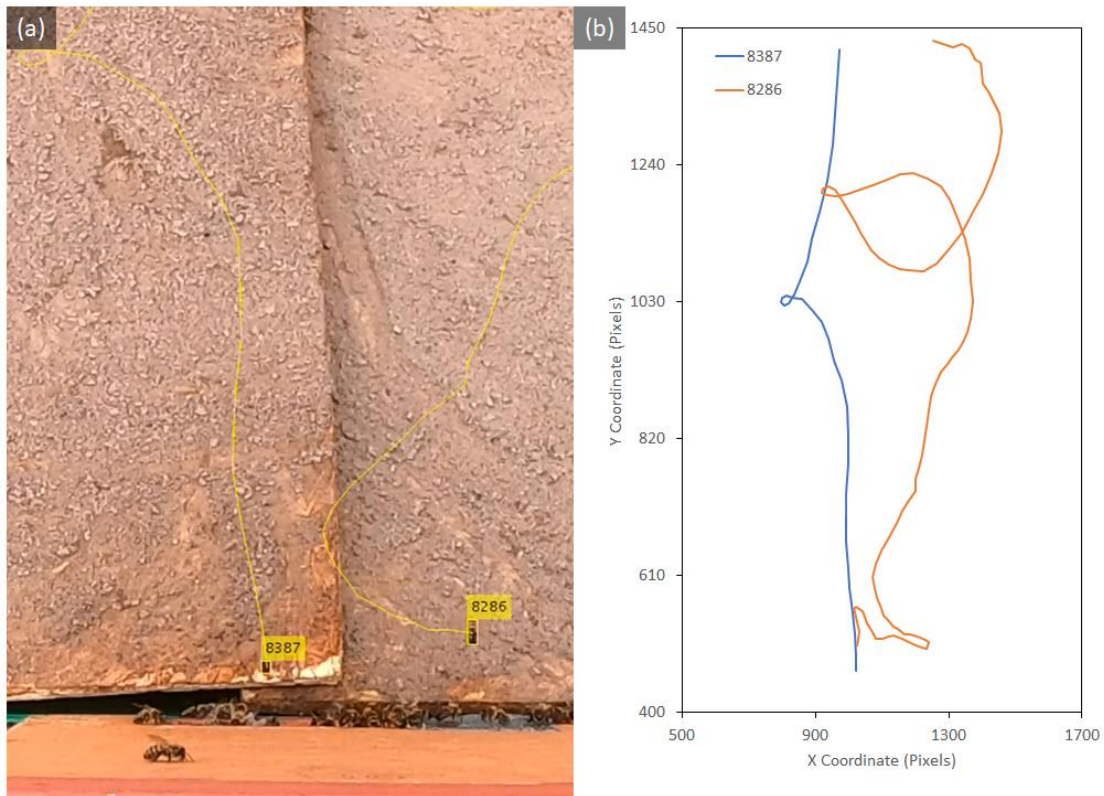


Figure 4.8 Two overlapping inward flights (8387 and 8286) showing different behaviours, (a) shows a screenshot from an annotated video and (b) shows a plot of the two flights.

The machine learning was based on 43 features, typically focused on the minimum, maximum, average, and last three measures of the following variables;

- Shape longest and shortest axis.
- Shape ratio and growth.
- Track life in seconds.
- Distances of the first detection to the hive and the nearest exit.
- Distances of the last detection to the hive and the nearest exit,
- Closest distance to the hive and the nearest exit.
- Current angle versus average bearing angle.
- Current angle versus perpendicular to the hive entrance.
- Change in angle over time.
- Number of loops and loop lifespans.
- Duration and size of dropouts (frames where tracking failed).

- Acceleration and speed

Most of the more valuable information of a flight was encoded during the last three measurements taken of the location of the bees, thus the inclusion of minimum, maximum, and average of these 'end' variables in addition to the general copies.

Learning and testing took place in three stages. First, the naive filter results were hand-corrected by a human observer. Second, machine learning algorithms were trained on the labelled dataset. Finally, the algorithms were tested on a new video. This process was repeated multiple times, with the learning algorithms taking place of the naive filter in subsequent iterations. This continuous process allowed for the monitoring of test accuracy and the identification of weaknesses in both the features used and the underlying flight extraction algorithm.

Once the test accuracy for the algorithms began to plateau after three training and testing phases, a final test was arranged with a new recording and the procedure was assessed for veracity against a human counter.

The algorithms used were K-nearest neighbour (KNN), neural network (NN), support vector machine (SVM), and random forest classifiers (RF) [184, 187, 188]. A Bayesian search was enacted to tune the hyperparameters for the KNN, SVM, and RF approaches choosing from the possibilities as follows [211];

- For KNN: Number of neighbours and nearest neighbour algorithm.
- For SVM: Regularisation parameter, kernel, kernel degree, and kernel coefficient.
- For RF: Number of estimators, split criterion, and the number of features to consider for a split.

For the neural network, the model used composed of three hidden layers of 64 neurons each using the rectified linear unit activation (ReLU) function [212]. The shape of this network was chosen by manual testing to find the best hyperparameters, investigating the width and depth of the network compared to the final accuracy. Minor changes to this structure did not yield significant change across either dataset.

4.3 Results and Discussion

Extraction times for both camera systems is shown in Figure 4.9, showing that the thermal camera system demonstrated faster extraction and did not grow exponentially, explained by fewer pixels being processed and clearer shape extraction. The exponential growth in the visual camera system is attributed to visual clutter such as debris and raw computational complexity due to resolution and field of view. The greater field of view included more objects that require foreground extraction, filtering, and labelling. This includes undesirable extractions such as blades of grass increasing the number of artefacts requiring attention.

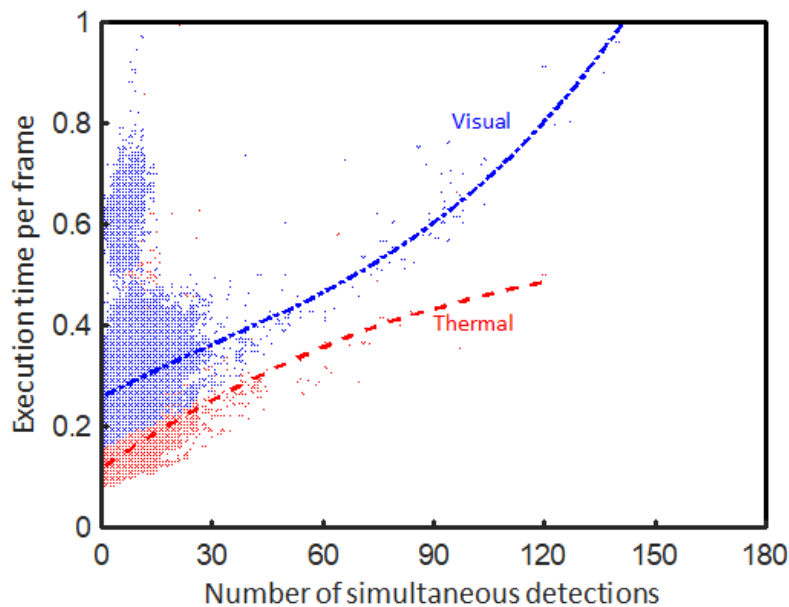


Figure 4.9 Time cost per frame to extract flights from both thermal and visual camera systems, with a simple 3rd degree polynomial fit.

The labelling of individual flights encountered issues when tracks split. Figure 4.10 (a) shows two flights where labelling could become ambiguous. In the Figure, a loop is formed as the bee flies away from the hive but it is broken by the edge of the frame. Deciding which section to label as an outward flight and which to disregard is challenging when the second section was missing for other flights. Labelling the first section as an outward flight would add a flight into the learning pool that, by itself, is

not statistically clear as an outward flight because it is curving back towards the hive. This would increase the residual error in the models generated.

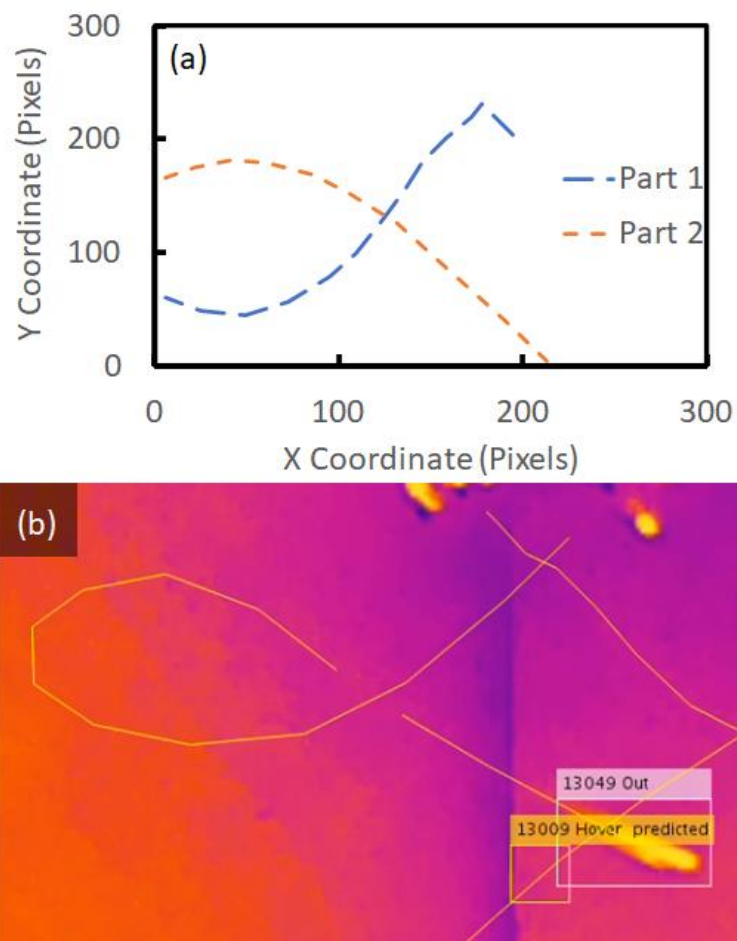


Figure 4.10 (a) Trajectory of an unclassifiable flight caused by the edge of the frame and (b) a screenshot illustrating an interrupted flight.

Similarly, in the screenshot flight Figure 4.10 (b), the track has been interrupted when two bees passed close together. The yellow bounding box, showing the predicted location of the first section as obtained by the Kalman filter, has been correctly labelled as invalid. The white bounding box has been correctly labelled as an out flight. In other cases, the second section of the flight may not be recovered and the remaining first section is unclassified. It could either be removed from the data set, meaning that a flight is labelled as missing, or labelled as an out flight and add error to the generated models.

The first stage of the training process involved using the naive filter to count outgoing flights. In the video recorded as a sample, using the naive filter resulted in 3220 outgoing flights counted. Of these, 1687 visual out flights were extracted correctly with 1533 flights later rejected by human expertise. This means that the naive filter achieved an accuracy of 52.39% and almost doubled the true number of out flights counted.

For the thermal video, 3735 outgoing flights were counted by the naive filter. This was then corrected to 1685 outgoing flights and 2050 incorrect flights, resulting in an accuracy of 45.11%.

The accuracy of the inward flight count by naive filter was wholly dependent on the underlying motion tracking system. Flights detected at the edge of the frame and ending near the entrance of the hive were required to be seen as inward flights. The inverse was not true for outward flights. Inwards flights which were truncated due to poor tracking were unavoidable, but distinguishing these using machine learning would be difficult. Inwards bees would hover if the entrance was congested, and this would be hard for any machine learning algorithm to separate from other hovering bees. A better underlying tracking algorithm, especially a deep learning approach, might prove useful in the future. Once inward flights were removed, the remaining hovering and outwards flights were inputted into the machine learning models. To do this, the features discussed were computed from the raw positional coordinates to form one row of data per flight, which was then given a label and used to train the models.

As these two videos were aligned by hand, a comparison of flight links between systems was possible. The mode difference in time between thermal and visual extracted flights was 0 s, with an average difference of 0.0468 s. 75.1% of flights occurred across systems within 0.5 s of each other. The larger differences can be explained by the fused, split, and frame-edge flights as discussed above. Fused flights may cause significant time variance across systems as an outgoing flight may be fused with a longer hover flight due to extraction errors, and this hover flight may have been present for many frames before the outgoing flight. This increases the discrepancy between the two systems.

The machine learning models were trained from these initial results Table 4.1. With thermal data, the KNN and NN are equally efficient algorithms, followed by RF and SVM. Thermal data outperformed optical data by <2%.

The most accurate models from this training dataset were then used to label a new video. For the thermal data, this is the neural network. For the visual data, the RF was used with the hyperparameters chosen by Bayesian search as entropy for the criterion, \log^2 features considered for a split and 1611 estimators used.

Table 4.1 Results of the first training. Precision is measured as true positive count over true positive count plus false positive count. Recall is measured as true positive over true positive plus false negative. F1 score is a combination of recall and precision, known as the harmonic mean.

Video Type	Algorithm	Accuracy (%)	Precision	Recall	F1-score
Thermal	NN	92.40	0.912	0.904	0.908
	RF	89.93	0.884	0.874	0.879
	SVM	90.55	0.887	0.881	0.884
	KNN	92.40	0.913	0.903	0.908
Optical	NN	90.50	0.892	0.891	0.891
	RF	91.07	0.894	0.893	0.893
	SVM	90.69	0.900	0.901	0.900
	KNN	89.25	0.871	0.897	0.884

Under test conditions, the optical results were that 4053 flights were predicted as out, with only 1149 being correctly labelled. In addition, 5471 flights were predicted as invalid flights with 221 of these being incorrectly labelled. The overall accuracy of these predictions was 67.19%, however, precision was 0.84 and recall 0.28. These results were because the test dataset was several times larger than the training set, increasing the possibility of flights that bore no resemblance to any flight seen prior, making classification more difficult. The wide field of view of the optical camera meant many more variations of flights were possible. Lastly, the removal of the naive filter meant many flights that were previously removed for consideration by this filter now needed classification by the trained model. This was because some outgoing flights were filtered out by the naive system as the underlying tracking algorithm did not successfully track them to the edge of the frame.

Results for the thermal camera showed 1727 flights were labelled as outgoing of which 1328 were correctly labelled. 6647 flights were predicted as invalid flights and 111 of these were incorrectly labelled. This gave an accuracy of 93.91%, a precision of 0.77, and a recall of 0.92.

The data from this test was then added to the training set and further train/test iterations as in Table 4.2. Iterations identified features likely to aid in classification. The number and duration of loops were subsequently added, as was the nearest and furthest distance measure from both hive and closest exit.

Table 4.2 Results of the two following training iterations and the second test.

Type	Algorithm	Accuracy (%)	Precision	Recall	F1-score
Thermal Train 2	NN	93.06	0.907	0.905	0.906
	RF	93.02	0.921	0.898	0.909
	SVM	93.02	0.920	0.889	0.904
	KNN	91.72	0.886	0.894	0.889
Thermal Test 2	RF	95.56	0.902	0.919	0.911
Thermal Train 3	NN	94.35	0.931	0.887	0.909
	RF	94.70	0.927	0.904	0.916
	SVM	94.41	0.916	0.908	0.912
	KNN	93.89	0.908	0.900	0.904
Optical Train 2	NN	92.29	0.889	0.912	0.900
	RF	92.34	0.896	0.899	0.898
	SVM	92.19	0.894	0.899	0.897
	KNN	89.75	0.865	0.875	0.870
Optical Test 2	RF	95.93	0.923	0.955	0.939
Optical Train 3	NN	94.23	0.919	0.915	0.917
	RF	94.37	0.926	0.917	0.922
	SVM	93.81	0.917	0.912	0.915
	KNN	92.56	0.893	0.909	0.901

In addition, refinements were made to the parameters controlling the track extraction in MATLAB, particularly as the test results from the first optical test showed that flights were missing from the collected data. These refinements improved the

selection of control parameters for the Gaussian models used in flight extraction based on the observations of the human labeller.

Accuracy started to plateau by the third training iteration. The more complex learning algorithms became noticeably more accurate and precise than the simpler KNN. In both cases, the algorithm selected for the second test was random forest, which became the strongest model in the final training iteration. For thermal data, the RF used entropy as the criterion, the square root of feature count to consider for splitting and 316 estimators. For optical data, the RF used the same parameters except for 2500 estimators, significantly higher than the thermal data, owing to the need to rely on more features to make decisions. The issues from the first optical test were no longer present by the second test, explained by the growth in the dataset covering more complex flights.

4.4 Feature Importance

To confirm that the need for more complex models to classify optical data was due to the shape metrics, feature importance was extracted from the random forest and the support vector machine using a linear kernel. For random forest, the feature importance was calculated as the mean impurity decrease within each tree when splitting using a set feature, scaled so that the set of importances summed to one. For the linear kernel of the SVM, the weights represent the vector coordinates which are orthogonal to the hyperplane and their direction indicates class. Again, these were scaled to sum to one.

The top ten features from both RF and SVM for both thermal and visual data are presented in Table 4.3. Textual columns are the feature type and the adjacent numerical column is the feature importance. Importance for all records within one table sum to 1.0. Only the first 10 records are shown. A key is present in the table footer, as each column is coloured based on the type of feature represented.

For thermal data, max shape ratio, max shape length, and final growth were the most important features across both models. Max shape ratio for SVM and max shape length for RF were the most valuable variables for classification and twice as important as any other unrelated variable.

Table 4.3 Feature importances for all model and video types. (a) is the table for thermal random forest data, (b) for thermal support vector machine data, (c) for visible spectrum random forest data, and (d) for visible spectrum support vector machine data.

(a)

Thermal RF	
Max Shape Size	0.1574
Max Shape Ratio	0.0819
Max Angle Vs Final Bearing Angle	0.0750
Last Distance to Hive	0.0694
Average Angle Vs Final Bearing Angle	0.0620
Exit Angle versus Final Bearing Angle	0.0560
Average Jump	0.0371
Max Jump	0.0354
Max End Change in Angle	0.0334
Final Bearing Angle Vs Bearing Angle	0.0326

(b)

Thermal SVM	
Max Shape Ratio	0.2233
Final Growth Ratio	0.1797
Acceleration Summary	0.1120
Final Acceleration Summary	0.0923
Life In Seconds	0.0859
Number of Loops	0.0336
Growth Ratio Total	0.0317
Max Time Difference	0.0311
Bearing Vs Exit Angle	0.0262
Average Angle Vs Final Bearing Angle	0.0203





(c)

Visual RF	
Bearing Vs Exit Angle	0.1183
Exit Angle versus Final Bearing Angle	0.0970
Average Jump	0.0846
Max End Change in Angle	0.0698
Max Jump	0.0675
Average End Change in Angle	0.0584
Max Angle Vs Final Bearing Angle	0.0520
Average Acceleration	0.0393
Last Distance to Exit	0.0357
Average Angle Vs Final Bearing Angle	0.0305

(d)

Visual SVM	
Min Change in Angle	0.1480
Max Shape Ratio	0.0933
Life In Seconds	0.0733
Final Acceleration Summary	0.0707
Growth Ratio Total	0.0602
Min Shape Size	0.0592
Average Change in Angle	0.0514
Average Jump	0.0471
Exit Angle versus Final Bearing Angle	0.0424
Average Angle Vs Final Bearing Angle	0.0412

Key

Shape	
Angle	
Speed	
Misc	

For optical data, the smallest change in bearing and overall bearing versus exit angle was most important for the SVM and RF respectively. Only two shape metrics appeared in the five most important features for each algorithm and no type of metric was twice as predominant as the next. This showed that, for optical data, learning was more nuanced, involving greater use of the features available.

4.5 Wasp Detection

In addition to bee tracks, wasps (*Vespula Vulgaris*) were also detected. There were too few of these detections for machine learning, at most four detections in a 20 min period. The presence of wasps supports that future development of this technique could include models able to count and monitor other insects that interact with a hive. For example, in Figure 4.11 a wasp was observed to fly close to a hive and hover near the entrance, before finding an isolated bee on the hive platform. There, it flew in proximity to the bee until that bee then fled off the frame. The wasp then returned to the entrance of the hive before flying away when it was clear that there was no access to the hive.

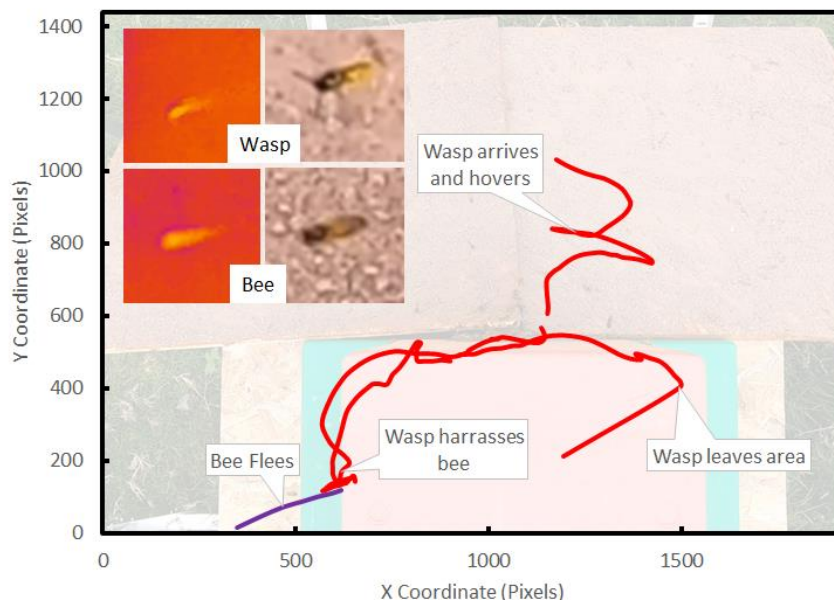


Figure 4.11 The flight of a wasp captured investigating the hive and harassing bees, including image of the wasp taken by thermal and optical cameras compared with similar images of bees.

Other interactions were observed, such as wasps flying directly to the entrance of the hive to be met by guard bees that then discouraged further investigation and tailed the wasp for a short distance as it left the area. Wasps will probe honeybee hives to test whether ingress to steal honey is feasible [213]. Thermal images of a wasp did not differ significantly from those of bees, especially when moving and the signature was blurred. Wasps appeared characteristically brighter than honeybees with optical cameras which will be useful for their identification. Their movement patterns are discernible from incoming, outgoing, and hovering bees. Interactions between these two species create more possible data permutations than just bee flights alone, which will necessitate more comprehensive data gathering.

4.6 Test Stage

Once the final training iteration was completed, a test was conducted using a final recording. For this recording, human expertise was used to attain the true number of in, out, and invalid flights working with a recording as demonstrated in Figure 4.12. Video times were cropped across camera systems to be an exact match, beginning and ending in synchronicity. Fused flights were identified and counted as lost flights, as were any double-counted flights as per split flights. Flights lost to failings in the underlying track extraction algorithm were also counted (Table 4.4).

The underlying tracking algorithm was 95.08% effective at recovering flights from optical recordings and 93.88% with thermal recordings. These are the useful flights (those that are not invalid.) The overall accuracy of the algorithms was 96.16% for optical data and 97.92% for thermal data, though the accuracy metrics were inflated by the greater size of the true negative (invalid/hovering flight) class. More useful measures would be a precision of 0.91 and recall of 0.90 for optical data. For thermal, precision was 0.92 and recall 0.90. For completeness, the original naive filter correctly labelled 549 flights with optical data and 286 with thermal data, far below the models created here.

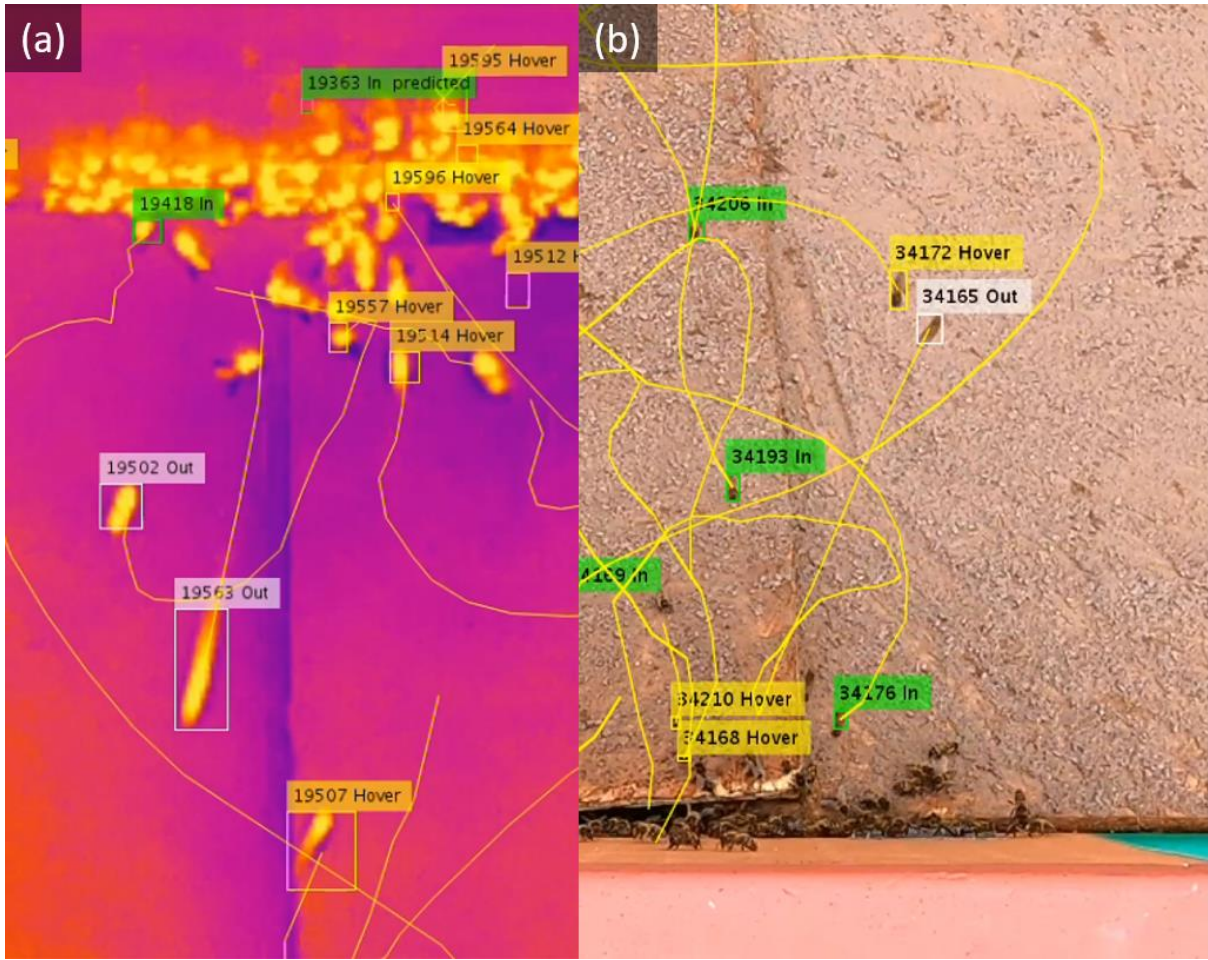


Figure 4.12 Cropped screenshot of each camera system during the test phase, showing (a) thermal camera results and (b) optical camera results.

Table 4.4 Results of the test recording.

Type	Missed Flights	Counted In	Counted Out	False Positives	False Negatives	True Negatives
Visual	82	902	681	65	62	2562
Thermal	102	907	668	64	55	4992

Both cameras performed well in classification and flight extraction, however, the thermal camera missed an additional 20 flights because they occurred outside of the frame. This results in an unavoidable 24.39% increase in missed flights over the optical camera system. The optical camera system achieved an accuracy of 95.53%, with precision of 0.92 and recall of 0.96, in its second to last test stage. This is a noticeably better result than the thermal camera at any of the test or train stages. The efficacy of each system will vary according to the environmental conditions on the days that they are used. The thermal camera can function under a wider range of lighting conditions compared to the optical system, making them more responsive under variable environmental conditions.

Further data is required to fully develop the dataset to account for all possibilities associated with seasonal changes affecting ambient temperature, wind conditions, and particularly rain. Rain would appear as a darker object moving within a frame before colliding with the hive entrance or ground and could be mistaken for bees. Wind will make the flights of bees more erratic. These issues could be addressed in future work. During this study it was noted that bees were quick to return to the hive and reduce activity when conditions became too windy or wet.

4.7 Summary, Limits and Conclusion

The work in this chapter has demonstrated that thermal cameras are a contender for bee counting applications. To do this, it directly compared the results when classifying bee flights from both a thermal and optical camera. Neural Network, Random Forest, Support Vector Machine and Nearest Neighbour models were developed and evaluated on both optical and thermal data.

Thermal cameras offer significant benefits. Their flexibility in deployment is greater than that of optical cameras, working in poorly lit conditions and without visual aids. They require less modification of beehive entrances and are passive devices that will not disturb bee behaviour.

The results suggest that despite the smaller field of view, fewer pixels, and lower frame rate, the thermal camera is comparable to the optical camera with the aid of machine learning. Both systems are at least 93% successful at extracting flights and 96% successful at classifying these flights. Recall and precision metrics demonstrate that the thermal system is equal and often better for classification due to the improved shape metrics gathered and the models are not as complex. Flights are more cost-efficient to recover from thermal footage due to fewer pixels to consider. As thermal cameras continue to decrease in price and increase in capabilities this advantage will grow.

The limits of this work come from two sources. Firstly, it was difficult to gather sufficient data and provide labels. The study is limited by the smaller data set. Secondly, bees were labelled by human observers. This introduces the possibility of observer bias into the labelling process, specifically the chance that there could be disagreement over a particular assigned label.

The end goal of this work would include building a system that removed the human element from the labelling process which would reduce or remove the possibility of ambiguous labelling. However, to build this system, data must be supplied and currently this can only be done with human labelling for now.

Expanding the dataset for this work will allow the inclusion of variable weather conditions and environmental temperatures. This would create a robust model for the system described to function in all weather conditions.

Further work in this developmental pathway could include a more intelligent flight extraction algorithm, able to reconstitute broken flights from frame-edge and collision losses. A reinforcement model, able to tune the extraction parameters, would increase the fidelity of the data and permit improved analyses.

Consideration could also be made to spot other insects near the entrance of the hive and classify them based on shape and movement metrics similar to those here. An example would be the Asian Hornet which is known for hawking behaviour at the entrance to beehives.

Lastly, investigation to correlate flight behaviour with younger bees, particularly investigating orientation loops, would allow for the monitoring and counting of fresh foragers from a hive as a health metric for managed hives.

5 Challenges in Developing a Real-time Bee Counting Radar using Machine Learning*

5.1 Introduction

This chapter details the investigation to build a real-time radar-based counter for the entrance of a beehive, and was used to publish a paper [214]. This chapter closely matches the content of the paper and only contains small additions to provide further discussion around the key motivations of the thesis.

This work discusses the challenges of counting bee activity using machine learning and radar. It provides a breakdown of the data processing and models used. The radar was constructed during the Ph.D. work of fellow student Nawaf Aldabashi, and the need for such a radar was borne by S & A Produce, a company specialising in using bee pollinators for soft fruit.

This company wanted to keep good, accurate records of the productivity of their bees highlighting those that were not thriving. In addition to the requisite hardware and software needed to be developed that could interpret radar signals reflected from bees and produce labels to describe the activity taking place.

The final radar module supporting the effort was a 5.8 GHz continuous-wave (CW) radar Printed Circuit Board (PCB). The PCB module integrated an in-phase/quadrature (IQ) mixer for the discrimination of positive and negative Doppler shifts. The IQ mixer fed 2 channels with identical 60-dB custom-designed Variable Gain Amplifiers (VGAs) and 100-dB common mode rejection ratio (CMRR) for amplification of the Intermediate Frequency (IF) signal. The VGAs additionally included low-pass filters limiting the IF output to the DC-408 Hz range. The VGA's output was fed to a laptop using an external USB sound card with a 44.1 kHz sampling rate.

The work that follows provides the details of creating machine learning models to run on portable hardware that would function to generate labels for the activity at the entrance of a beehive being measured.

5.2 Preliminary Work

Prior to this chapter, two studies were performed by the author using machine learning (ML) with the radar for proof of concept [215, 216]. At first, a binary (inwards versus outwards) classification problem was undertaken to validate that the radar was suitable for extracting signals of sufficient quality for machine learning. In addition, these experiments were taking place outdoors and ensured the capability of ML to classify noise-prone signals.

The radar used in these experiments was an earlier prototype utilising singular breadboards for each component, placed two meters from the hive with input captured by a laptop. A camera allowed for labelling each segment of captured data. This setup is shown in Figure 5.1.

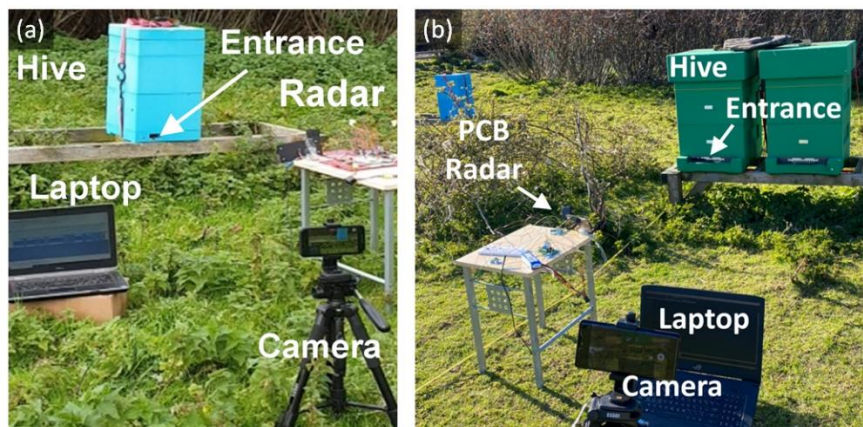


Figure 5.1 (a) Early prototype of the radar, placed in front of this hive to capture activity. (b) Later implementation of the radar in a repeat of the experiment.

This preliminary study utilised an image-recognition neural network (MobileNetV2 [217]) on spectrograms generated from raw signals with axes that were limited to 200Hz and then removed, with samples displayed in Figure 5.2. Results were 87.8% accuracy at distinguishing inwards versus outwards bees using this method.

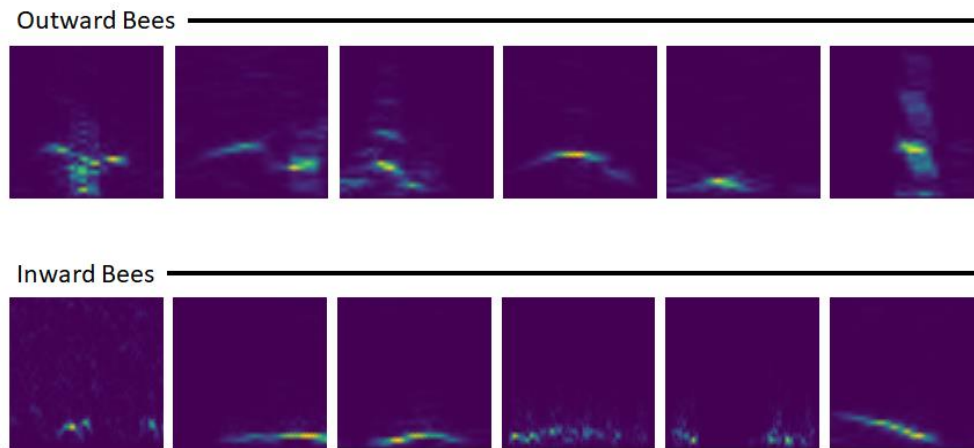


Figure 5.2 Spectrograms of Outwards and Inwards bees as captured by the radar, in 0.4 second samples limited between 0 - 200 Hz and with axes removed.

A follow-up study was conducted to find alternative feature extraction systems for the signals [216]. Both spectrogram generation and image recognition networks are computationally expensive and would hinder progress towards a real-time, low-power deployment solution.

In summary, the feature extraction techniques utilised were:

- **Linear Predictive Codes (LPCs [218]) and Line Spectral Frequencies (LSPs [219]):** These techniques are formerly signal compression techniques that are for reduced transmission load when broadcasting signals. However, they encode complex signals in a spectral envelope form describing the most prominent peaks. Therefore, they are suitable for feature extraction using those prominent peaks.
- **Mel-Frequency Cepstral Coefficients (MFCCs [220]) and alternative form BFCCs:** These techniques generated descriptors of the signal's short-term power spectrum, after a linear cosine transform of a log-power spectrum with an alternate, non-linear scale. With MFCC this scale is designed to approximate the weight applied by human hearing. The alternative form, BFCC, sought to find a better scale for bee radar signatures.
- **Temporal Features:** More traditional descriptors for signals included mean amplitude, root mean square, zero-crossing rate, short-time energy, spectral centroid, kurtosis, skew, standard deviation, mean, variance, and energy.

In addition, substitute machine learning models were explored that would potentially allow better utilisation of these new extracted features. These models were:

- **Neural Network (NN [204]):** A small sequential model using two densely connected layers of 32 neurons each, activated by the scaled exponential linear unit function (SELU.) This model was optimised by the Adam algorithm.
- **Support Vector Machine (SVM [187]):** An SVM with runtime hyperparameter tuning using Bayesian Searching. Limits were a regularization parameter between 1e-6 and 100, a kernel coefficient of 1e-6 to 100, and a polynomial kernel function degree between 1 and 5. Kernel choice was between linear, polynomial, radial basis function, or sigmoid.
- **Random Forest (RF [188]):** An RF with parameter tuning. The estimator count varied between 100 and 1000, in increments of 100. Branch split criterion was either decided using Gini impurities or entropy for information gain. The number of features to consider for a split was either the square root or \log_2 of the total number of features.

Finally, the problem was expanded to include hovering bees. Hovering was defined as all behaviour where the bee might fly close to the entrance of the hive but make no attempt to enter or leave the area. The bee might move closer, or further away, from the radar. It also included extended inward flights that were longer than the sampling window of 0.4 seconds. This was to train the models not to double count long inward flights where the bee faced congestion and hovered prior to entry.

Classifying this behaviour was valuable in both a commercial and research setting, first by removing the potential for these hovering flights to be falsely classified as entry and exit. In addition, it may prove that standalone hovering flights, or hovering before leaving, can be attributed to bee orientation flights, which can be a good indicator of growth, measured by the rate of young bees first leaving the hive [221].

Results for predicting between inwards, outwards, and hovering bees were strong. These results are presented in Figure 5.3. To summarize, in a binary problem, all feature-extraction methods bar one failed to compete with the more capable image-recognition deep learning model. The BFCC algorithm, a modified MFCC, achieved slightly better results at 91.1%.

When the problem was expanded to three classes (inwards, outward, hovering bees) all feature extraction and modelling algorithms lost potency except for the models learning from the Line Spectral Frequencies (LSPs.) These models did well, especially the support vector machine which gained more than 20% accuracy and achieved 93.4%.

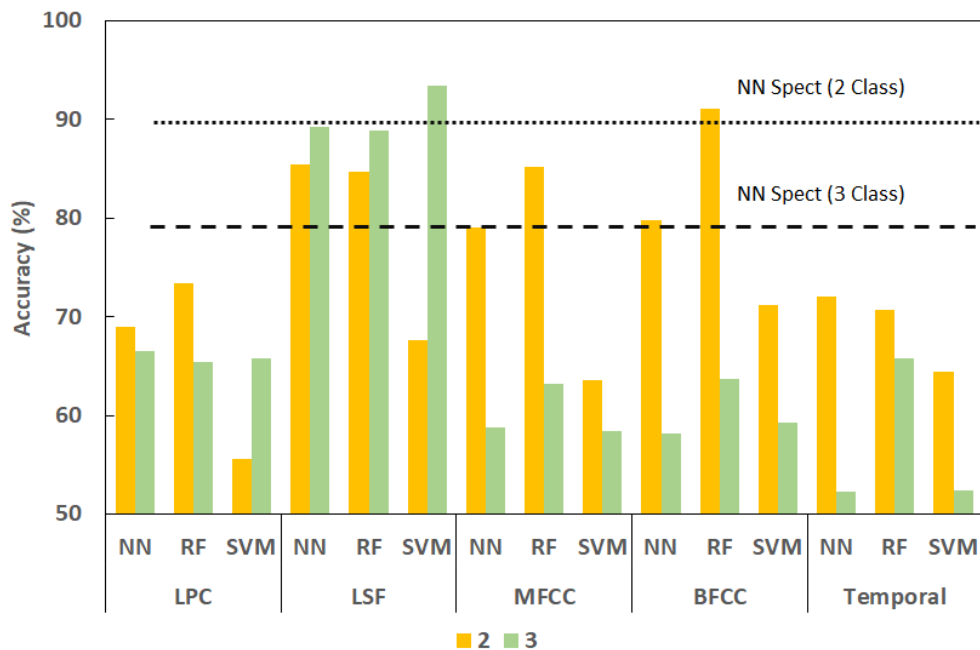


Figure 5.3 Visual plot of the results from different feature extraction methods for both binary and ternary classifications. Included is a comparative (dotted lines) for the deep-learning image recognition model using spectrogram representation of the bee signatures.

The results from this stage informed the experiments in this chapter, where LSPs were used alongside spectrogram deep-learning as a benchmark. These two approaches were the strongest in their respective previous studies and thus were targeted to handle the more complex next stage.

LSPs were later deemed to also be too computationally expensive and were replaced by Log Area Ratios (LARs.) LARs are not as efficient as LSPs for their original intended purpose (requiring more values to encode a signal, meaning more transmission bandwidth). However, they are significantly simpler to compute, meaning they make a good feature extraction technique where a minor increase in the number of features is not troubling.

5.3 Methods

The computing system was designed to minimize both cost and power consumption and was centred on a Raspberry Pi 4B. Without an AI Accelerator or equivalent, the Pi was not suitable for a deep learning approach. Instead, this system would leverage Support Vector Machines (SVMs [187]) to match previous work [215, 216].

The sampling time was limited to 0.4 s. This window represents the smallest observed complete event in the original dataset. Even within 0.4 s, most recorded samples included one or more hovering bees as well as the other classes. In 4.4% of samples, both an inward and outward bee event took place within 0.4 s. This is true of overlapping inward and outward bees as well. A smaller percentage (0.18%) contained multiple overlaps such as two inward and one outward.

Other research studies, without machine learning or automatic counting, have placed the radar onto the hive surface, facing outward [107, 109, 110]. The approach was chosen to overcome the following challenges of such placements:

- It removed the need to modify the hive. This is advisable given that the system may be used on wild bees.
- Bees crawl at the entrance and may cover either antenna, as in Figure 5.4.
- Antennae have a radiation pattern that may cause flights to be lost from the detection cone if, for example, they walk to the edge of the hive before takeoff.
- While offering some protection against hovering bees, surface-mounted radar may still be obscured more infrequently.
- Limited research suggests that bees may be sensitive to the frequencies used and the equipment will function as a source of heat, which may affect behavior [222, 223].

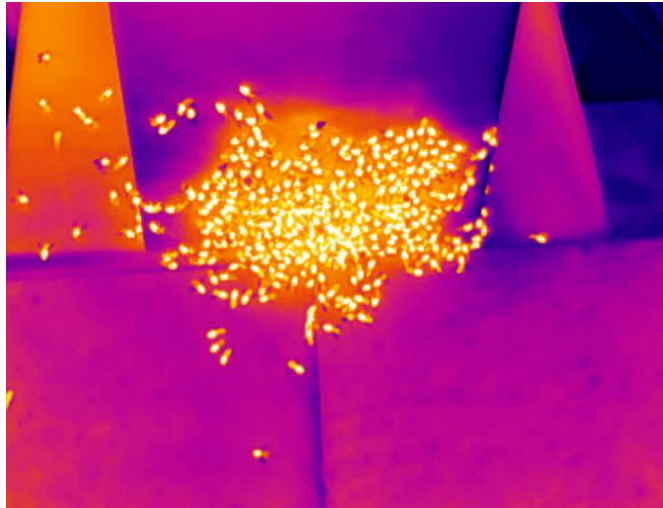


Figure 5.4 A thermal imaging camera capture of bees crawling over the entrance of a busy hive.

The position in this study ensured that the entire front surface of the hive was in view of the radar, and it was less invasive and the setup quicker. Hovering bees and weaker power reflection at the entrance of the hive remained an issue because of the free space between the radar and hive entrance.

Challenges were expected from the outset because there was no effort to standardize bee flights or control flight direction. Bees were free to leave in any direction, even crawling along the edge of the hive until takeoff on a side face. Similarly, on approach, bees could arrive from any angle and could be as quick or slow to enter as needed. When the entrance was congested, bees would often hover on arrival until there was space to enter, mimicking other hovering bees and obfuscating other activity when flying close to the antennae. The free-flying bees created complex radar samples that could not be intuitively labeled solely on signature structure alone.

Initial data were gathered across three days, consisting of twelve recordings with a maximum duration of 20 min each. Different hives were used during each day. Replacing the radar between sample gathering periods was not precise, because the system needed to be flexible, so long as it was placed within the expected range (1–2 m) of the hive as in Figure 5.1. When working with wild colonies it would not be

possible to guarantee the same distance or angle, nor would it be advisable to force such placement if minimal disturbance was desired.

Each radar stream was accompanied by a video from a digital camera. The video recording was initiated first, and the radar data were aligned by the operator counting down to the commencement of the radar recording. This was suitable to align within half a second. Two or three clean bee events would, by matching video frames to timestamps, allow complete alignment.

This source dataset was gathered to train the algorithms. Once trained, these would then be used to label entire videos. The operator would provide corrections of the sample labels where needed and the resultant datasets fed into the training data pool.

The machine learning models would be expected to label entire videos. Therefore, an additional dataset was later included that featured one full-length recording that was disaggregated into 0.4 s samples, and this was labeled and included in its entirety.

An overlapping window of 0.1 s was used to extract samples from consecutive or extended events, such as long hovering flights and background samples. A flexible approach was used when samples were not an ideal length for sub-division, modifying the final overlap to ensure all source data were used. For example, a signal of 0.6 s would be split into two 0.4 s samples with an overlap of 0.2 s.

Feature extraction for the primary system was achieved by using Log Area Ratios (LARs) derived from Linear Predictive Codes (LPCs) [198]. LPCs and their derivatives are a means of expressing the spectral envelope of a signal in compressed form. Their use in machine learning for radar data is relatively new and has successfully classified other, non-acoustic, signals [152, 196, 224].

The LARs were used to train a support vector machine with Bayesian hyperparameter optimization. Five different models were trained to evaluate whether there were benefits to breaking the classification problem into sub-problems with models for each. These problems were:

- Four-way classification.
- Background samples versus all others.
- Hover samples versus in and out.
- Three-way classification (hover, in, and out).
- Binary classification (in and out).

These models were chosen to allow multiple potential classification pathways. Either four-way brute classification, or splitting the problem into multiple, potentially easier, problems as demonstrated in Figure 5.5. These separate pathways were developed to maximize the opportunity for binary classifications that can favor SVM models [225, 226].

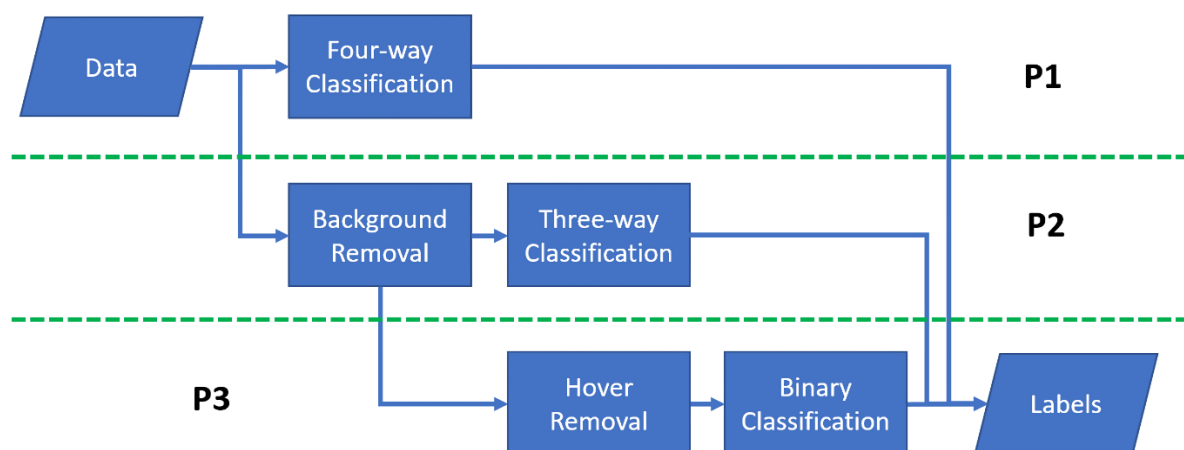


Figure 5.5 Three prediction pathways (P1, P2, P3) toward labelling samples. Continued binary classifications may favour Support Vector Machine (SVM) architecture over multi-class problems.

To provide context, similar models to those in the author’s previous work were used [215]. This was firstly a DenseNet deep learning architecture with a custom head network [227]. This network would operate on spectrograms generated from the 0.4 s samples. While unlikely to be lightweight enough to run on portable hardware, this model would provide a crucial understanding regarding the suitability of the data for machine learning.

To compare with other networks, MobileNetV2, VGG19, and a custom image processing network were also evaluated [217, 228]. The structure of both the custom network and DenseNet201 are shown in Figure 5.6.

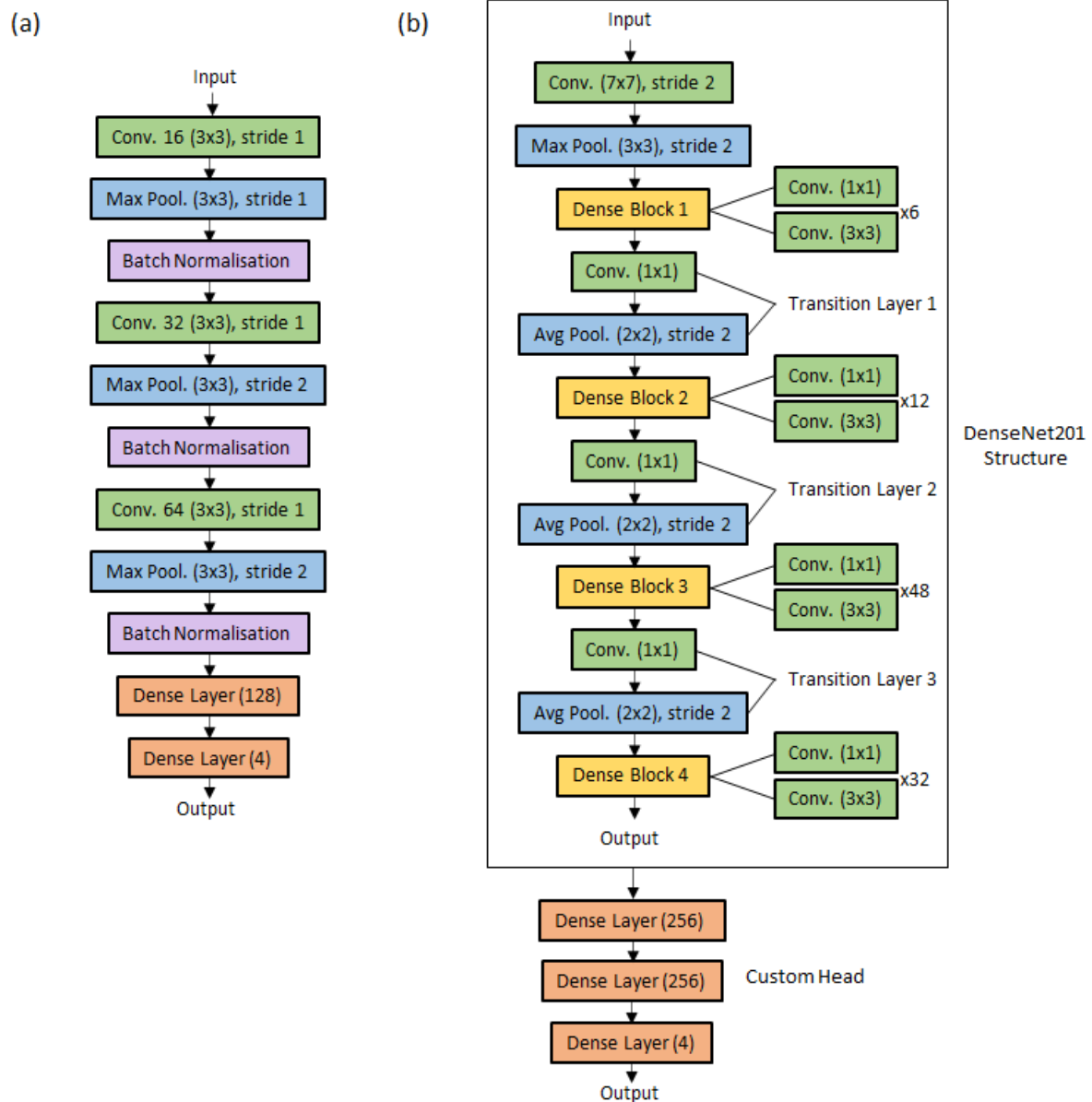


Figure 5.6 (a) Structure of the custom image processing network and (b) Structure of DenseNet201 and custom header network.

All of these networks were trained on the spectrogram data to measure performance and determine if image processing networks were still suitable for classifying the bee signals.

5.4 Results

5.4.1 Preliminary Results

Generated spectrograms of the signal samples provided evidence that signatures would have less information above 300 Hz (see Figure 5.7). This would exceed the typical flight speed of a bee at 8 m/s. As image processing networks require small inputs of no more than a few hundred pixels square, this work limited the upper range of spectrograms to 300 Hz and then 150 Hz to maximize image quality. The change to 150 Hz was initiated as accelerating and decelerating bees were always much slower than their cruising speed and spectrograms contained little information above 150 Hz. Any information here was lost in the contrast limits of the generated images and would only penalize the models. Empty space in already small images would reduce the resolution of the lower-frequency, more powerful signatures.

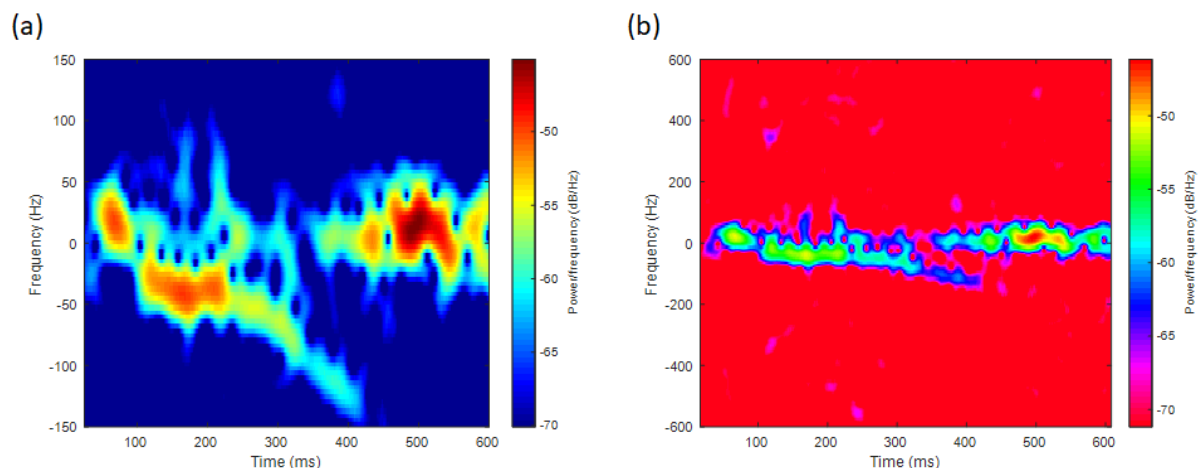


Figure 5.7 (a) A complete signal sample (outward bee) spectrogram limited to 150 Hz matching the images that were inputted into the deep learning models. (b) A larger range, high contrast (wider colour gamut) spectrogram of the same signal shows a paucity of information beyond 150 Hz.

However, results from the deep learning approach were weak, with the most precise being DenseNet with an accuracy of 46.73%. Given the four-way classification nature of the problem, this is better than a random choice, but the results warranted further investigation.

For comparison, the custom network achieved 45.39%, VGG19 delivered 42.93% and MobileNetV2 44.87%. The similarity in results shows that the inability to predict the correct class reliably is likely a limit created by the data representation rather than the capability of the models themselves. This is supported by the fact that the custom, untrained network was second in accuracy to DenseNet201 and performed more accurately than the alternatives.

By using LARs, it was possible to achieve a preliminary accuracy of 75.12% in a four-way scenario (Figure 5.9). In all cases, a 9:1 split of training and testing data was used and the results were gathered as an average of tenfold cross-validation. The Figure shows the outputs of running the experiment with three sets of data:

Set A: The single channel, manually gathered Doppler data from the radar.

Set B: The dual channel, manually gathered IQ data from the radar.

Set C: The dual channel, complete IQ dataset including both the manual set and the full recording breakdown dataset.

A visualisation of the sets and their sources is given by Figure 5.8.

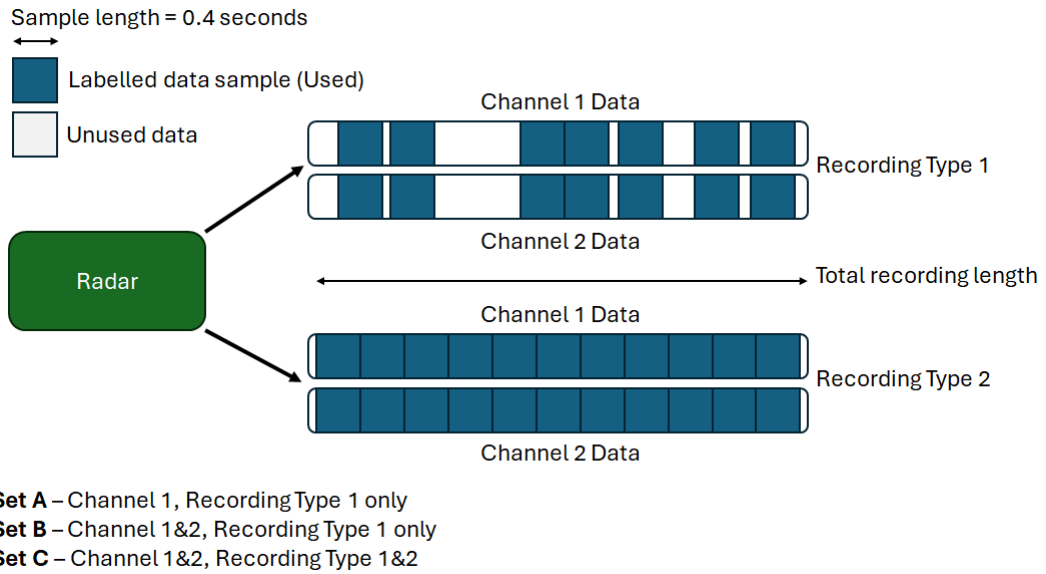


Figure 5.8 Visualisation of the classification set types and their sources.

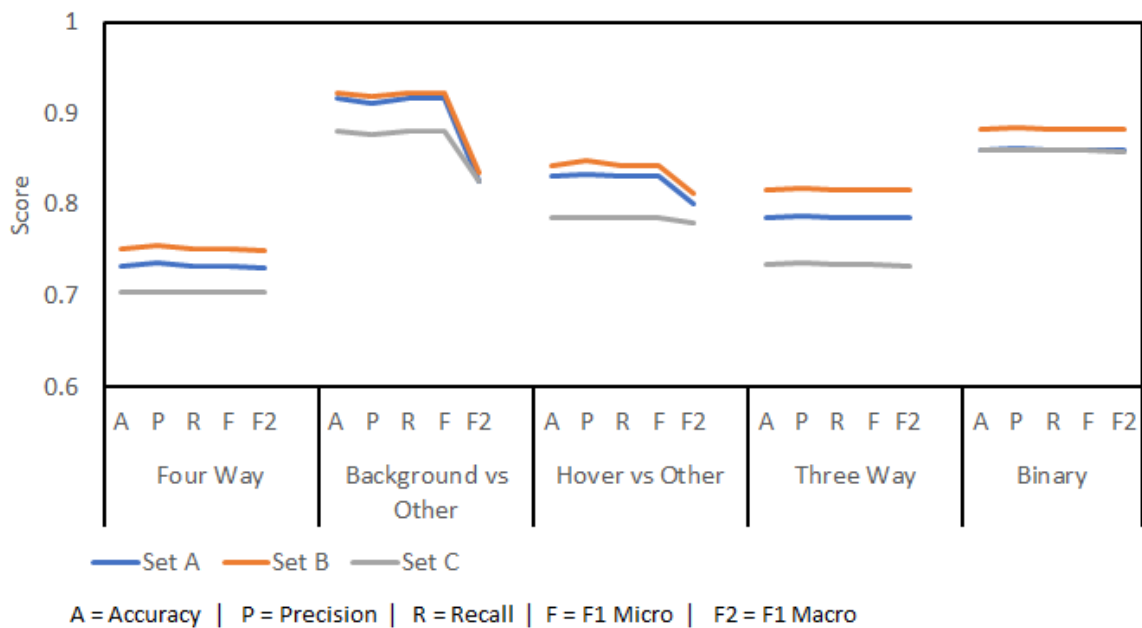


Figure 5.9 Results from preliminary machine learning models (Support Vector Machines) using Log Area Ratio (LAR) implementation.

Set C would be the dataset used in the testing phase of the work. This shows a performance penalty associated with fully captured datasets rather than hand-chosen samples. This is not unexpected, as more difficult samples (such as those with overlapping events) were required to be included. The results show that

complete IQ datasets are more suited for machine learning than single-channel results.

Separating the problem into smaller challenges did not create better results. While background prediction is good (91.59%), this would then be followed by either hover prediction (83.19%) or three-way prediction (78.65%); together these would fall below base prediction accuracy (75.12%). The targets are the labels generated by the final classification, the inward and outward bees. Knowledge of background and hovering signals is useful but is not the goal of this work.

A potential route for better classification would be to use speed-distance profile images of the bees in flight. Much like spectrograms, these could be used to train deep image-recognition models. A concern would be the effect of noise and multiple bees, which may have more of an effect in a speed-distance profile than the already significant effect in a spectrogram (see section 5.4.2). Given that the spectrogram approach performed poorly, and that the LAR approach was promising, the decision was made to prioritise the LAR approach. An additional benefit was the reduction in processing requirements as LARs are much simpler to compute and build models for.

5.4.2 Exploring the Weaker Results

The weaker-than-expected results spurred a further investigation into the spectrograms generated. Complex signals, difficult to classify, became apparent due to the free-flying nature of the targets. Figure 5.10 shows an ideal sample of four consecutive outward flights of bees, which quickly accelerate toward the radar before passing by in proximity as confirmed by video recording. The first two flights overlap on the spectrogram, hindering the ability of the machine learning to count them separately as they exist in one 0.4 s window.

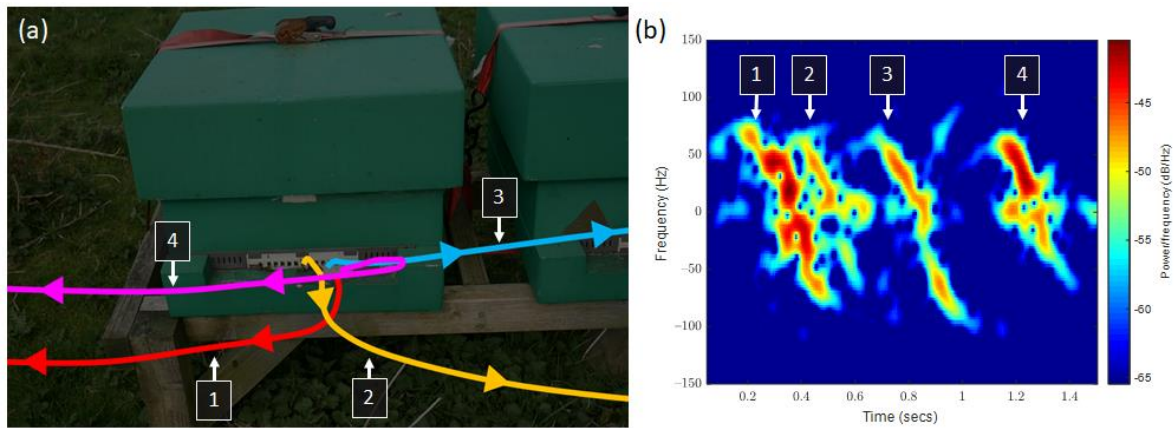


Figure 5.10 (a) Image showing the trajectories of four bees and (b) spectrogram recording of this event. The first two overlapped, limiting attempts to separate them. The numbered lines are the four flights recorded both in the video and spectrogram.

However, not all flights were clean. Figure 5.11 shows both a visual record and a spectrogram of complex overlapping events. These events are as follows:

1. Takeoff for a single bee.
2. Flight of the first bee to the right and behind the radar.
3. A hovering bee emerges from under the radar and flies off-screen to the left.
4. Vertical takeoff of two bees, one does not approach the radar.
5. The second of the two bees loops, increasing speed, and exits the frame.
6. The inward bee from the screenshot appears.
7. The first of the three bees in the screenshot takes off.
8. Two more bees take off after the first.
9. Closest approach of the exiting bees.
10. Inward bee enters the hive.
11. The last view of the exiting bees, flying away from the radar both left and right.

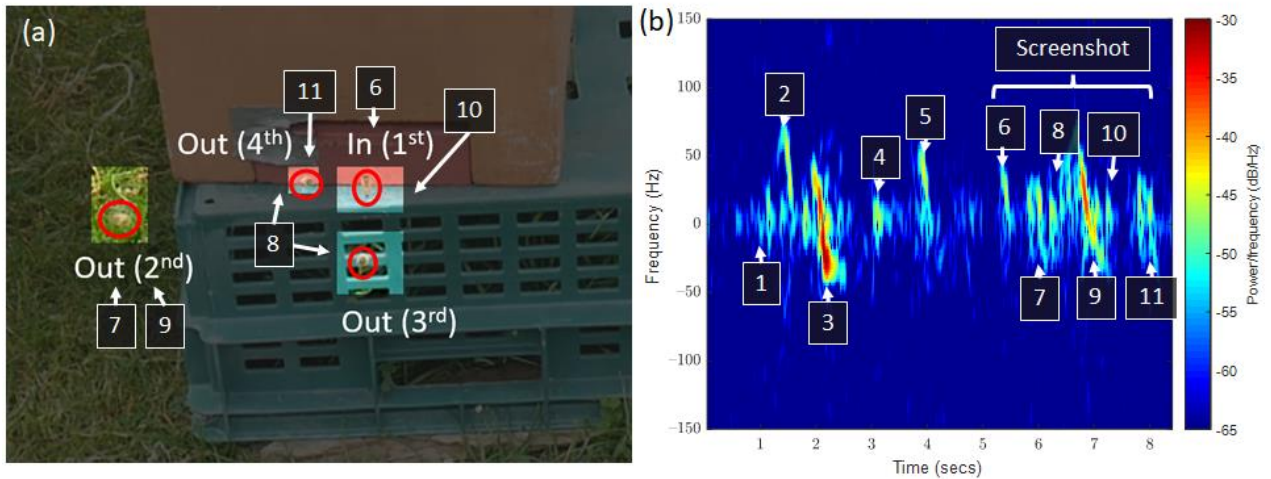


Figure 5.11 (a) A screenshot of the video recording of an event and (b) the corresponding spectrogram representation of the signal, showing complex overlapping elements. The red circles are the bees recorded across the events and the numbered items show correlation between spectrogram and the video recording.

While the signal happened across eight seconds and would be broken down into smaller, easier-to-classify samples, there is a paucity of information when multiple overlapping events took place. Specifically, between events 7 and 10, there is a compounding of the signals, justifying that the spectrogram approach would be met with failure.

Some events were too similar in the target frequencies to separate visually. An example of these is provided in Figure 5.12. The first event (a) is of a hovering bee that moves both towards and away from the radar with variable speed. The second event (b) is two inward bees flying towards the entrance of the hive; however, there is a sudden uplift of wind, which makes their flight difficult, and they struggle to fly along a fixed path.

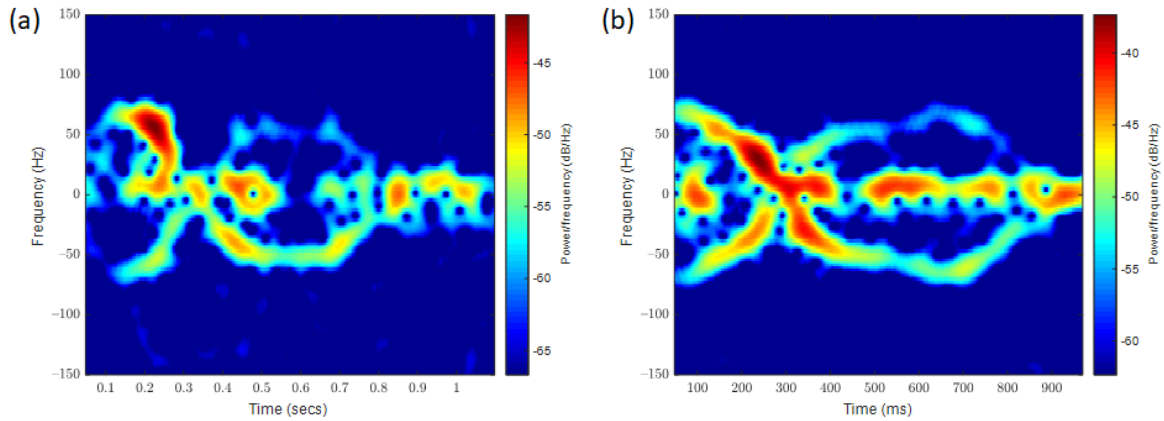


Figure 5.12 Two signals (a) showing a hovering bee signal and (b) showing an inward bee signal.

Figure 5.13 shows three hovering bees, none of which enter the hive or leaves the area during the segment. At 0.4, 0.75, and 1.5 s some examples are like the outward signals present in the ideal sample. Multiple hovering bees in a signal recording were common.

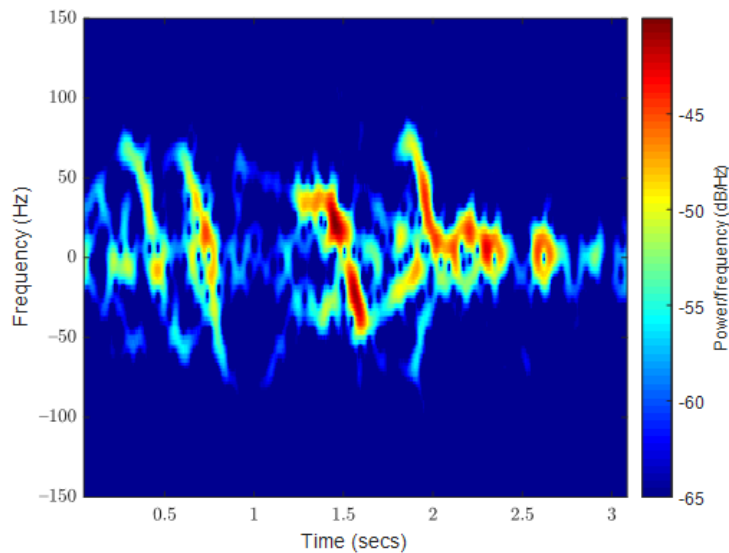


Figure 5.13 A hovering signal of three bees shows similarities to outward bee signals.

These signals are a close visual match to other, less ideal outward signals. In the samples collected, there were matches between all four classes. A spectrogram deep learning approach would encounter a point of no improvement due to the restraints of the visualization format. In the future, as this dataset is expanded, the visual overlap will continue to grow.

Given this limitation, questions emerged regarding the signal compression techniques and mild success. To understand how the data allowed the models to perform well, several exploratory investigations were undertaken.

The major disparity between these results and others found in literature was the number of LARs used in this work. It is common to expect 10 or fewer LP coefficients (equivalent in number to LARs) for each small window, itself less than 100 milliseconds [196].

In contrast, the models required that the 400-millisecond signal not be subdivided, and as such, the number of coefficients climbed at first to 240 per window for a 44.1 KHz sample rate and 100 per window for a down-sampled 3.5 KHz rate. This high number of coefficients is problematic. As the number of coefficients increases the algorithm quickly includes noise from the source.

Using the full number of coefficients, accuracy response as a function of the sample rate was assessed. The results are presented in Figure 5.14. This shows that accuracy required a sampling rate of greater than 3 KHz to achieve a plateau of growth. The exception to this was predicting background and binary signals, which had a strong response from any sampling rate, expected as these are simpler predictions.

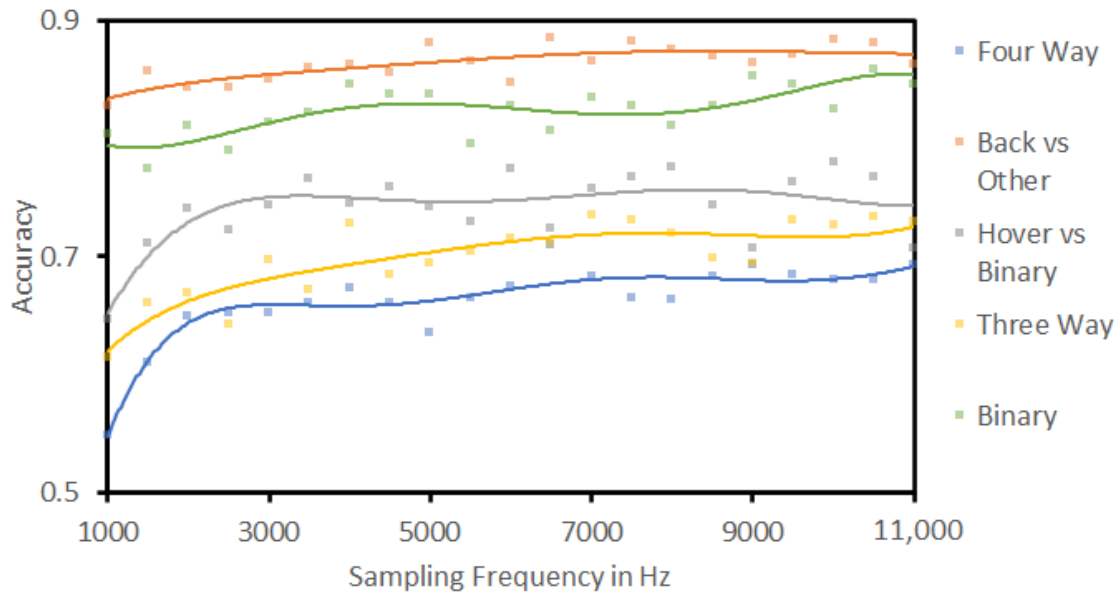


Figure 5.14 Accuracy versus sampling rate across the different prediction pathways, showing that accuracy changes in response to varying the sampling rate of the signal.

To investigate signal sub-division to match other works in the literature, the Raspberry Pi was first benchmarked to confirm limits to the number of coefficients that could be used. The results are presented in Table 5.1 and ‘times required’ have been measured to include running a prediction. This is to ensure the process happens faster than the 0.4 s window.

Table 5.1 Possible sub-window sizes on the Raspberry Pi © and the maximum number of coefficients per window possible.

Sub-Window Size	Encoding Limit	Total Number of Features per Channel	Time Required
40 ms	76	760	350 ms
50 ms	84	672	348 ms
80 ms	96	480	349 ms
200 ms	110	220	352 ms
400 ms (full window)	240	240	351 ms

Generating many coefficients for a 0.4 s window is computationally taxing. By using multi-core processing to handle each channel separately, the Raspberry Pi could encode 240 LARs in a 0.35 s window.

With these limits, a benchmarking routine was created to determine accuracy as a measure of the sub-window size and number of coefficients. The experiment was

also conducted when downsampling the signal to 3 KHz and 1 KHz to measure whether lower frequency components become more important when sub-dividing the window.

The findings are presented in Figure 5.15, demonstrating that the sub-division of the sample window decreases accuracy. For completeness, all sub-window lengths with the full 240 LARs are included, which would not be possible to run in real time on the Raspberry Pi. Even with all coefficients available, LPC derivative machine learning accuracy decreases as the signal is segmented. As LPCs are compression techniques, it can be understood that segmenting the signal further decreased the information in each resulting window. A comparison would be the segmentation of four similar spoken words into small time windows, which would decrease the overall context included as opposed to encoding the entire words with one compression window.

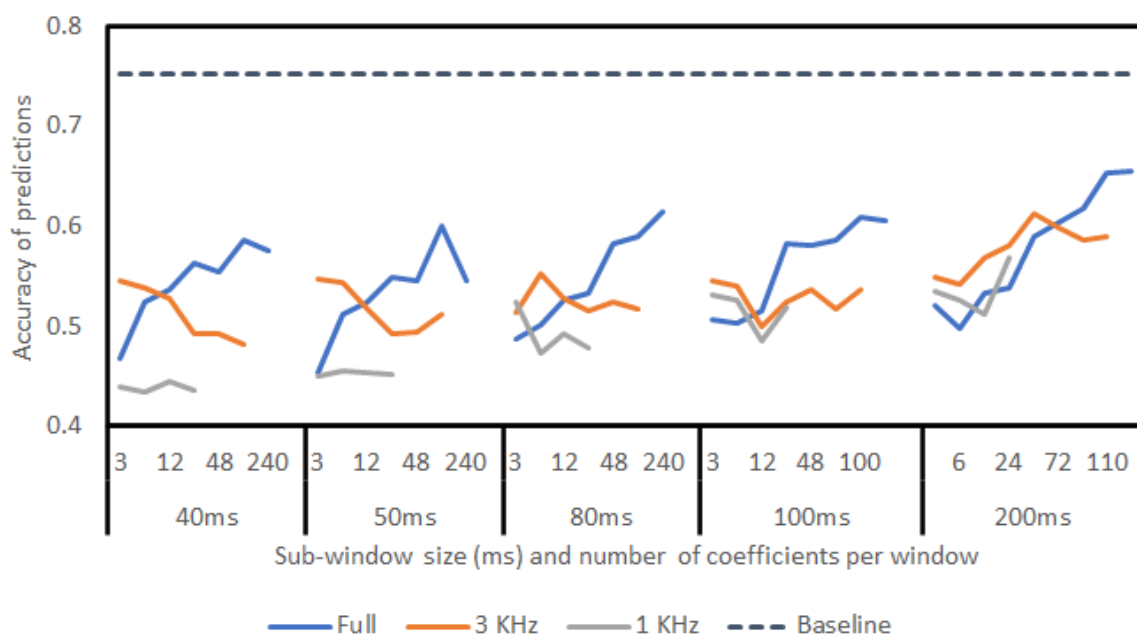


Figure 5.15 Results from sub-windowing the signal with differing coefficient numbers (Support Vector Machines). This includes accuracy at 44.1 KHz sampling rate and change in accuracy at both 3 KHz and 1 KHz. At 1000 Hz, some window/coefficient combinations could not be run due to insufficient data.

It became clear that there were either high-frequency and/or low-power components to the signals that were not easily shown on a spectrogram. These elements were crucial for machine learning success. The signal could not be further segmented without decreasing accuracy. Together, these findings supported that these components are being obscured by background noise.

It had been an expected evolution of the work to begin creating filtering algorithms to strip out the clutter associated with outdoor recordings in variable weather. However, the complexity of the filters will now become more challenging. Preserving complex patterns while removing the effects of wind and other clutter will be challenging.

However, without filtration, the machine learning models would be unlikely to adapt to new recordings. The existing data were recorded as subsets each from a single or group of videos, each with its own setups and environmental conditions. This could be introducing noise into the dataset, which meant that models were unprepared for new sets of data from previously unseen conditions.

The following question was whether leaving the sampling rate at the maximum 44.1 KHz was introducing needless noise that was affecting the feature encoding stage. Another routine was designed to measure how accuracy reflected the number of coefficients at differing sample frequencies. Lowering the sampling rate decreases accuracy, as shown in Figure 5.16. However, at lower sampling frequencies, accuracy requires fewer encoding coefficients. A notable plateau is present at 100 coefficients or more with a sampling frequency of 3.5 KHz, followed similarly by other sampling frequencies with the same number of coefficients.

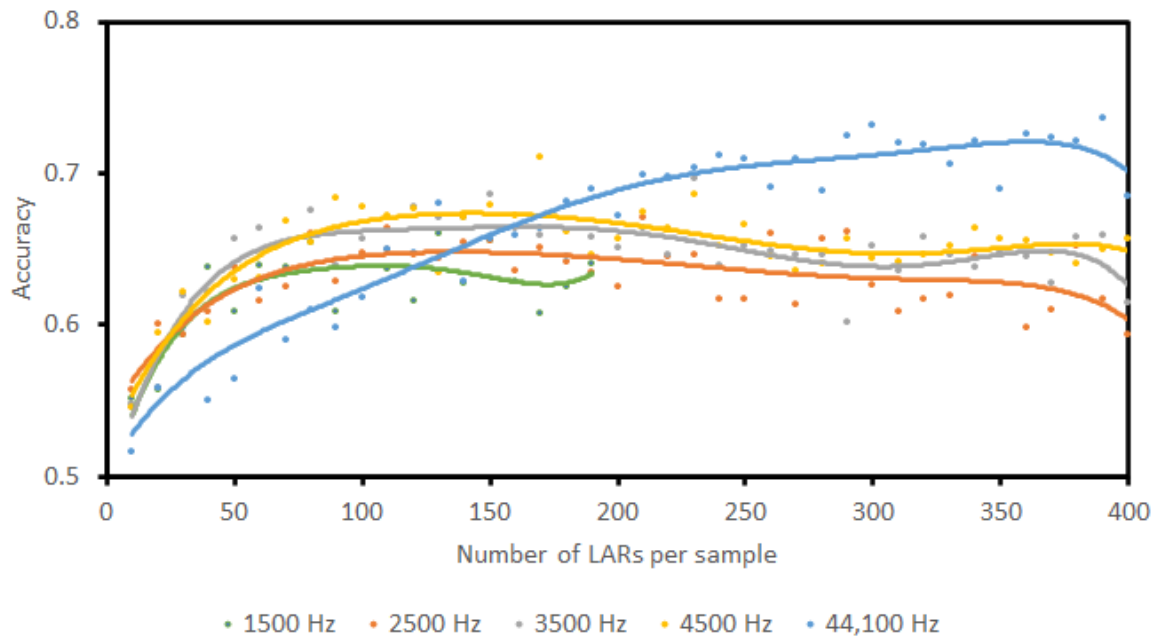


Figure 5.16 Accuracy versus the number of encoding coefficients for a range of sampling frequencies. Legend indicates sampling frequency in Hz. When using a 1.5 KHz sampling rate, it was not feasible to include large numbers of coefficients as the data became sparse.

While these results compare poorly to allowing an unrestricted sampling frequency, they show that the models require fewer LARs at lower frequencies to achieve maximum accuracy. This could indicate that the models may have been learning more general patterns in the data when given lower sampling frequencies to work with. When running the final tests, the results of lower-frequency, fewer-coefficient encoding would be included to measure whether models could become more generalized.

Now that it had been determined that the models were not influenced by noise included with an unrestricted sampling rate, it became prudent to analyze the signals in greater depth. LPCs are a compressed form of the spectral envelope of a signal. As such, it was useful to generate the spectral envelope for each signal and produce a standard deviation per class. Spectral envelopes were generated by determining spectral peaks, using short time spectrum estimate, within the signal and then using shape-preserving piecewise interpolation with a moving average to generate the envelope [229]. These were generated over sampling windows matching those of the LPC data. In Figure 5.17(a), the standard deviation of all spectral envelopes in each class is shown up to 1.5 KHz. Standard deviation is preferred as averaging spectral

envelopes would remove most peaks, illustrated in Figure 5.17(b). Here, the classes have very similar profiles except that each is separated by marginal strength difference.

The standard deviation in the background class is the flattest, except for several peaks centered at 1 KHz, which is faint noise in the signals, often masked by the bees themselves, caused by the recording equipment. The mean of the classes shows that this peak is very prevalent across samples.

The bees themselves are visible as a strong peak of deviation at sub 150 Hz frequencies, matching the signatures seen on spectrograms. Outward signals have a peak slightly higher in frequency, which can be explained by bees rapidly accelerating away from the hive. Inward bees decelerate and hovering bees are unlikely to reach a maximum speed near the hive. Notable peaks can be seen at 400 Hz and 800 Hz. Smaller peaks can be seen throughout, some more pronounced in one class over others but these are minor.

The range of the classes is shown in Figure 5.17(c), calculated as the difference between the strongest and weakest sample at each frequency. This shows that all classes have a similar magnitude of range. Even the background class has a notable range matching inward, outward, and hovering bees at the frequencies of interest (0-200Hz). This is most likely caused by bees in a few background samples that were near the radar but out of sight of the camera, perhaps directly behind the radar or off to the left and right. Unfortunately, without a 360° view, this is not avoidable. These could also be caused by low frequency environmental clutter, such as particularly strong gusts of wind that caused the equipment to vibrate.

The hovering class has the largest range across most frequencies. This is not unexpected at the class would be the most complex. Hovering was typically triggered when the entrance became congested and as such, many bees were seen hovering at the same time. Hence, the frequencies content of these signals was diverse and amplified by the number of individuals. Inward and outward bees were more commonly alone or sampled alongside one or two hovering bees and therefore have a smaller range.

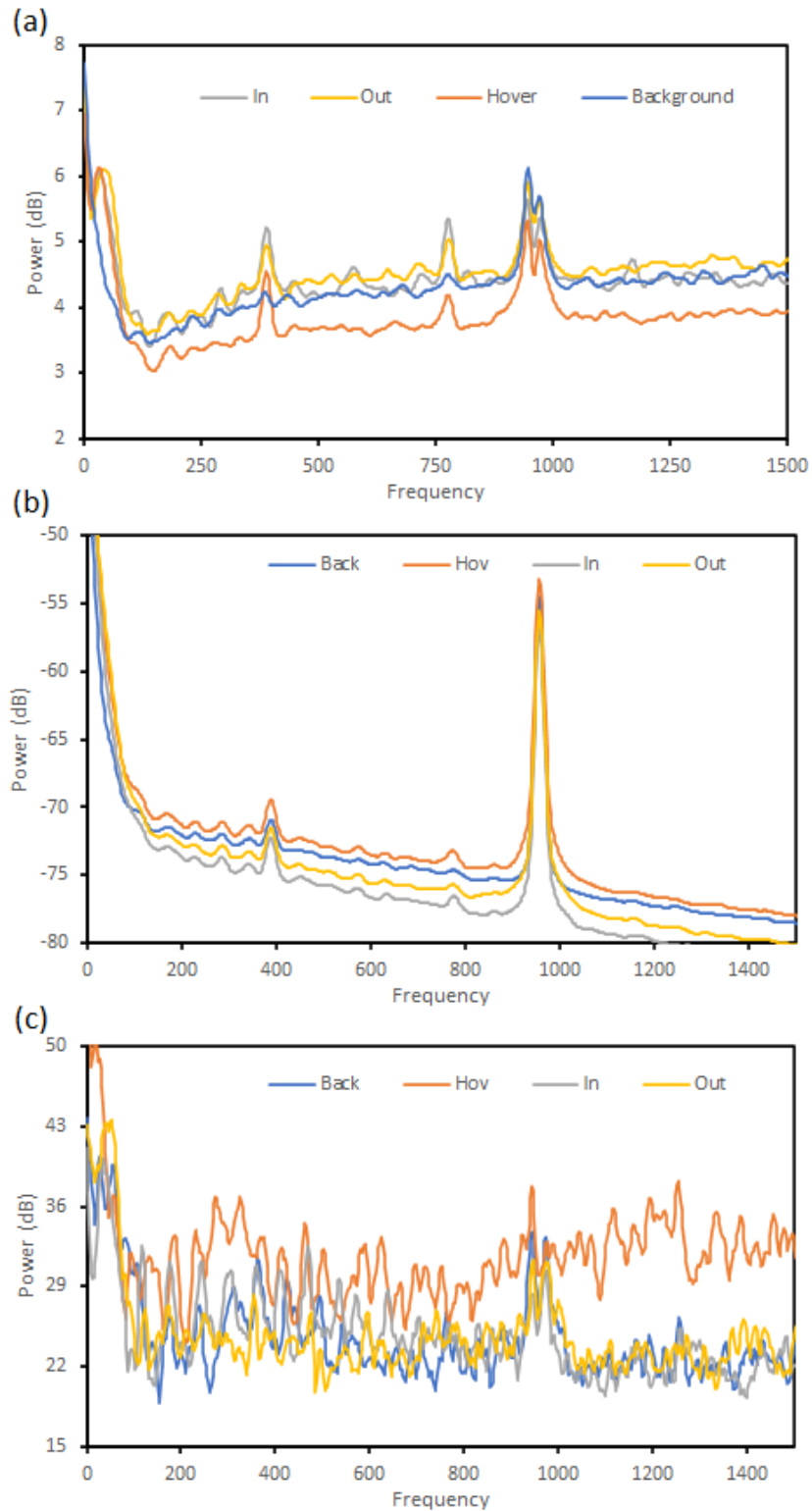


Figure 5.17 (a) The standard deviation of each class' spectral envelope (b) the mean of each class' spectral envelope and (c) the range of each class' spectral envelope.

5.4.3 Testing Stage

The machine learning was assessed on its accuracy in predicting the entire test set with all other data included as learning data (Figure 5.18). Significant penalties when using a separate setup are apparent. When exposed to new data, from a new radar position in differing conditions, the models lose their capabilities. Four-way classification accuracy drops to 70%, with a precision of 0.63 and recall of 0.70 due to imbalanced class sizes.

Sets in this figure are as follows:

- **Set A:** the complete training dataset was used, sampled at 44.1 KHz with 240 LARs.
- **Set B:** the complete training dataset was used, sampled at 3.5 KHz with 100 LARs.
- **Set C:** the smaller, manually extracted dataset with higher training accuracy was used, sampled at 44.1 KHz with 240 LARs.
- **Set D:** the smaller, manually extracted dataset with higher training accuracy was used, sampled at 3.5 KHz with 100 LARs.

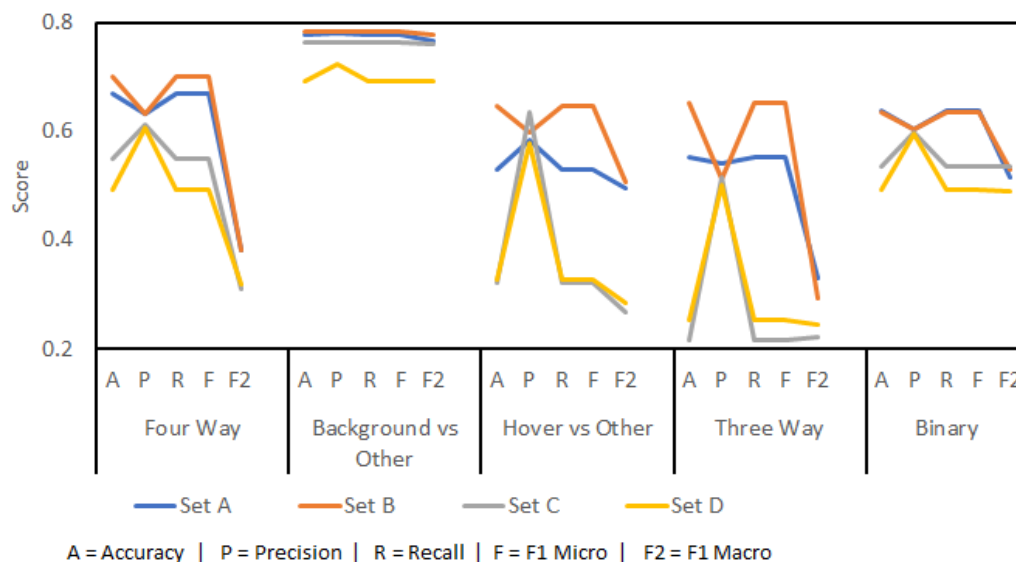


Figure 5.18 Testing results (from Support Vector Machines) from the final stage that show a decrease in performance versus the preliminary results. This is an effect of recording in outdoor spaces with variable conditions.

For completeness, the results for a down-sampled dataset at 3.5 KHz with 100 coefficients are included. Overall accuracy improved by 1–12% despite the lower training accuracy. A critical note for the four-way classification is that no inward bees were predicted correctly (121 samples or 4.8% of the data to label). The figures for this four-way classification are skewed by the much larger hover and background classes. This is evident when looking at the F1 macro scores, which expose accuracy bias caused by imbalanced classes.

Set B outperformed Set A despite lower training-stage results. This supports that different frequency bands and coefficient numbers benefit some classifications despite lower training accuracy. While adding more recordings, from differing weather and hive conditions, will improve the results further, the results above suggest that future gains will be ever-diminishing.

To achieve complete capability in this system, filters are a requirement. These filters will be challenging because of the complex signatures that form part of the machine-learning process.

5.5 Discussion

Compared to previous work, the results from this work are poorer [216]. For three-way classification, 93.37% accuracy was achieved, and 91.13% binary accuracy was achieved in the last effort. Similar results for this work were 81.67% and 88.33% accuracy for three-way and binary classification respectively (see Figure 5).

However, some key changes in the experimental setup explain the differences. This study used no data augmentation as the volume of data was considered sufficient. Data augmentation improves smaller datasets by creating a larger pool for training but can also make a set more homogenous and therefore easier to classify. The data recorded here were gathered across multiple days from more than one hive, which differs from previous studies where one hive was used on one day. The changes in radar distance and angle, coupled with varying weather, introduce more difficulty.

These additional challenges were inevitable in the development of a real-time implementation radar classification system.

Nevertheless, the expected outcome of this study was to meet or exceed previous results. Without this being achieved, there is further work remaining to overcome the shortcomings highlighted in this study.

The closest study in the literature to this work comes from Souza Cunha et al. in 2020 [107]. This study used the root mean square (RMS) of a Doppler radar as a measure of activity at the hive entrance, validating this by manually counting bees during recordings using a handheld clicker. RMS has key benefits as it is a simple, non-ML approach that gives a good measure of activity, which they were able to show correlates to hive health. As such, this approach is closer to field deployment readiness than the work here. However, they admit that 'non-foraging' bees (equivalent to hovering bees in this work) are counted in the RMS signal and there is no discernment between inward and outward bees using the radar. This work is an attempt to overcome these limitations and once fully developed will provide more precise information for future study.

The results show a pattern in that so long as sufficient data are available for each hive, distance, and weather condition, then the models are reasonably accurate. As soon as new conditions are introduced, the models lose accuracy. This is not unexpected, but the degree to which minor signal elements are necessary for good classification was not anticipated. These minor elements would too easily be removed by simple filters for environmental conditions.

Hovering bees introduce unique challenges in that, given the resolution of the radar, they appear to mimic the flights of other bees. They do this by passing close to the entrance of the hive while accelerating or decelerating, but not stopping. Minor differences in the signals will be useful to detect the difference between a slowing bee and one that stops. Again, these differences will be subject to interference from the environment.

Despite lower performance during initial training, models trained on subsampled signals with fewer LARs performed better than those with the complete data. This

supports the interpretation that the bulk of useful information is contained at lower frequencies. This is also shown when investigating the spectral envelope of each class, which shows more deviation at lower frequencies. However, the identification of which exact frequency bands are most important is challenging. Further work could look at performing statistical analysis of the signals in depth. This could provide guidance when developing filters as to which frequency bands are most important.

Hand-picked samples provided better training accuracy than the dataset containing all available data. The dataset containing all the data was more useful at the test stage. This is evidence that a hybrid approach may be useful in the future, with a dataset containing a core set of hand-chosen, clearer samples to provide a strong foundation. This is in addition to containing entire recording breakdowns, which will provide many hard-to-classify ambiguous samples.

This work is useful as no similar attempt has been made to classify honeybee activity at the entrance of a beehive using Doppler radar. Early experiments such as the one presented are necessary to identify the limits of existing technologies and algorithms as well as provide guidance for overcoming such restrictions.

This research implies that further work is needed to create a deployable real-time radar. A greater understanding of radar bee signatures is required so that good filtration can be enacted that does not remove the weaker signal elements.

5.6 Summary and Conclusion

An investigation into generating machine learning models to classify real-time radar data on honeybees has been detailed. These models aimed to monitor and count activity at the entrance to the beehives. Data gathered in this fashion, which are automatically labelled by machine learning models, would provide valuable data for ecological research and for businesses looking to improve their use of honeybees. The models generated in this work achieved an accuracy of 70%, though, by other metrics, the class imbalance created biased results.

Data were gathered from multiple hives across a few days from beehives kept at a farm. The data were split into 0.4 s samples, labelled by using video camera recordings of each event, and transformed into Log Area Ratios. These were then used to train Support Vector Machines to predict labels for new samples.

Challenges in progressing further have been identified. It is argued that a filter is needed, as high-frequency, weak signal elements appear to be needed for successful classification. These high frequencies are subject to interference and contain weak signal components that will be difficult to preserve. A greater understanding of these weak signal components is needed.

The limits of this work are clear. Four days of data were used from a small selection of beehives. To develop the solution further, many more hives would be required. Data would need to be captured that reflected all feasible weather conditions. Some, such as rain, may render the system incapable of predictions at all. In addition, an intelligent filter must be investigated to provide a means of removing much of the radar clutter that is unavoidable when recording outdoors while preserving weak but vital signal elements.

No further machine learning work is advised until filters are developed. Though additional data will result in increased accuracy, the system will not be resilient until environmental changes can be addressed. This work has functioned to provide specifications that future filters will need. With suitable further study, the work supports that the capability will exist to classify honeybee activity in real time.

6 Summary, Future Work, and Conclusion

6.1 Discussion and Limitations

This thesis explored whether machine learning algorithms can be used with current tracking technology to classify bee behaviours and movement. Each chapter tackles this question using a different technological approach. Each of the studied systems present functional advantages and disadvantages. The research was motivated by urgent need to monitor land use by pollinators such as bumblebees and that availability of large data volumes generated when tracking thousands of individuals. This volume of data can be utilised in real-time when driven by AI.

Machine learning was useful for classifying the activity of bees. However, there are several key limitations which affected data analysis in all the chapters. These were:

- The studies were limited by the availability and expense of gathering data, and the human error introduced when labelling data.
- The outdoor, wild, and unconstrained nature of bee activity allowed for complex interacting behaviours that became difficult to classify with current machine-learning capabilities.

All three analytical chapters suffered from insufficient data to cover all in-field use cases for the machine learning models being generated. Data of this type is challenging to collect, quantify and label, often taking months.

Determining which technology has the most potential for combined machine-learning success is complex. The auto-piloted drone/harmonic radar approach (better referred to as the long-distance tracking approach) presented more exciting possibilities. An early idea was to consider developing a system that would automatically build a map of which resources were visited at a given stage of development/time of year. Similar studies have been achieved using long range tracking, but not with an automated approach. The increased scale and automation would provide a level of detail of the development of bee colony that would exceed

current literature and provide visual maps that would be useful for public engagement to bolster support for bee-friendly legislation.

The thermal/optical camera approach was the most appropriate system for practical use. While requiring more computational power than the radar, it was less affected by environmental conditions and presented a two-dimensional view of the target rather than the equivalent one dimension of the radar.

Compared to the two previously mentioned systems (auto-piloted drone/harmonic radar or thermal/optical camera), radar was cheaper, more power efficient, and robust. It has potential to be mass-deployed to monitor multiple hives.

The heterogeneity of the machine learning models used is a result of the current best practices in literature, the hardware available, and the type of data being studied. While it would have been interesting to investigate a regression task (rather than the classification problems presented in this thesis), the data in question required classification. Similarly, neural networks, support vector machines, and random forests are standard approaches across the literature and are used in all three chapters. It would have been novel to explore other, more experimental machine learning models. However, given that the problem at hand was already taxing and in a niche area, compounding the complexity by using models other than tried and tested approaches would have potentially created unnecessary obstacles.

The author briefly investigated other classification models for each chapter, but their accuracy scores were so poor compared with the existing systems that they were quickly abandoned. Similarly, the author did consider attempting to build larger, more in-depth neural networks with specialist functions (such as a long-short-term memory networks or transformer networks.) However these typically require significant hardware expenditure that was out of range for the project. There was an interest in keeping hardware requirements low throughout the project to create solutions that would be available in real-time, or close to such.

6.2 Future Work

Recent developments have seen the release of affordable time-of-flight (ToF) camera sensors. ToF sensors allow for three-dimensional capture of an environment by timing the return of photons emitted by the sensor at several sample points (the effective pixels of the camera). These cameras have comparable image quality to the thermal camera but would additionally provide depth information. A fully three-dimensional rendition of the activity near a beehive would be more useful for machine learning classification if it could be shown that the camera was suitable for the task.

Further work could also look to bypass the 0.4-second window used in the radar study. This window was chosen as it matched the smallest sample in an early dataset. It also benefitted the study as it was not unreasonable for an observer to provide labels for this data on a 0.4-second scale. Smaller windows would prove increasingly challenging to label. Smaller windows might make it difficult to compute output faster than the data is gathered, making a real-time radar system unlikely.

However, it was possible for a 0.4-second sample to contain multiple bee signatures with no clear means of separation. By using smaller, partially overlapping windows it might be possible to detect multiple signatures, but this would also be an unreasonably challenging approach for an observer to provide labelling. It would also become likely that the models double-count a single bee if it appears in overlapping samples.

Another approach might be to further develop the thermal camera machine learning model to assist the radar system by providing accurate and automated labelling. A sufficiently capable model would be able to provide 34-millisecond precision when labelling bee flights if the camera and radar feed were synced appropriately. Thus, a camera-based model would generate large, precise datasets that could be used to train the radar model, eventually removing the need for the camera. However, any small imperfections in the camera-based machine learning would be inherited by the radar model.

It could potentially be further developed by converting the model to a regression system, whereby a regression value is predicted for inward, outward, and hovering bees. Peaks in the inward and outward predicted values (over time) would correspond to bees landing or first taking off, respectively for inward and outward bees, allowing for very fine counting of overlapping signals. The hovering value could be used to infer the total number of hovering individuals. This would be conditional on the radar hardware itself having sufficient resolution.

Combining different technological approaches and using machine learning to bridge gaps or provide additional functionality will be an area of increasing interest moving forward. It could be possible to increase the resolution at which bees can be monitored through a “combined arms” approach where different technologies interact to provide greater fidelity. This could include using penetrating radar (or terahertz imaging), if safety is not a concern, to map the positions of bees within a hive with a particular focus on the queen.

The thermal camera could be used to monitor movement inside an intact hive box, as it does not rely on any lighting or infrared emitters (which may generate heat and affect bee behaviour). This could be a novel research pathway to investigate interactions that happen within the hive. However, the lens for the camera would likely become dirty quickly and obscure the view. This would be a challenging project coupled with the low field of view of thermal cameras. Thermal camera lenses are made of expensive germanium as it does not absorb the observed infrared frequencies. Recently works have looked at developing cheaper, plastic alternatives that could replace these lenses. It might become possible to create a window into the side of a hive, allowing the thermal camera to see inside while blocking natural light frequencies, protecting the camera from dirt.

The long-range tracking project (utilising drones and harmonic radar) could be readily enhanced by using drone gathered datasets to ensure that machine-learning outputs are the same as harmonic radar data sets. Assuming that this is successful, expanding the work on unsupervised clustering of bee flights might yield insights into how bees make use of their environment given that an unsupervised algorithm is not biased by human-provided labels.

An expanded dataset would also allow for more thorough development of the flight task prediction models, particularly by identifying outlier flights that may not be present in the small harmonic dataset. By successfully classifying the bee flight task, it might become possible to tailor the code used to automatically pilot a drone based on which task the bee is undertaking, ensuring that tracking is more successful. A foraging bee is likely to be more direct, meaning the drone would need to prioritise acceleration and ignore minor deviations in favour of an overall projection of the bee's path. An exploring bee is likely to be more changeable meaning the drone would need to be more flexible and prioritise minor changes in the bee's path. It may follow that a machine learning model could pilot the drone by directly tuning flight parameters based on the target's movements in real time.

6.3 Review of Research Questions

In the following manner, the research questions for this thesis have been addressed:

- **Which bee tracking technologies are most useful when paired with machine learning?** This question has been answered during the summary discussion in this chapter. In short, there is no superior system. Each of the chosen technologies has strengths and weaknesses that make them strong candidates for being the best choice. All the systems would benefit from future work to improve them and remove some of the identified limitations.
- **Are existing model architectures capable of predicting bee behaviour as captured by these technologies?** This question has been answered in each chapter. The work in this thesis shows that current machine learning models can predict bee behaviour as captured by the systems used in this thesis. Each chapter contains a thorough breakdown of the success of each model and technology pair. As before, more work is required to better understand the data that is captured and engineer new models with greater performance.
- **What are the limits of machine learning in this field?** A common thread throughout this thesis has been to identify the limitations of each system.

This question has been answered in all chapters including this final chapter. The limitations of the systems included are clear and ideas have been presented for future improvement and work.

6.4 Review of Research Aims and Objectives

The core aim of this thesis was to understand and report on the contribution that machine learning could make toward tracking and understanding bee behaviour.

This was encapsulated by a central set of objectives. These objectives have been met as follows:

- **Identify key tracking technologies with the greatest potential to integrate with machine learning:** Several candidate technologies have been investigated to determine how well they might integrate with machine learning to track and predict bees. Each main chapter (3, 4, and 5) contains different hardware and software systems.
- **Design experimental setups to gather bee movement and/or behaviour data using these technologies:** Across chapters, experiments have been designed and conducted to evaluate the suitability of a technological approach. Data from these experiments have been gathered for two of the main chapters (4 and 5) and used to train models. In chapter 3, data was provided by an external group that was similar enough to the proposed system when the proposed system became unusable.
- **Generate suitable machine learning models to predict movement and behaviour using the acquired data:** Models have been generated in chapters 3, 4, and 5 for each of the candidate systems.
- **Evaluate the strengths and weaknesses of the models created (and their experimental setups) identifying the sources of any limitations:** Models have been evaluated from the perspective of best prediction accuracy for either predicting bee tasks or bee activity near the hive/nest.
- **Of the three technologies, provide informed discussion about which might benefit most or least from the machine learning integration:** This objective has been met in this final chapter, where discussion about the

strengths and weaknesses of the technologies (when compared with each other) has been provided.

- **Discuss whether the experiments show support for using machine learning to aid in the design of future bee counting and modelling systems:** The overall finding from this thesis is that there exists strong support from the data and evidence for pursuing further development of machine learning integrated bee counting and modelling systems. However, as has been addressed in this conclusion chapter, several technical and labour limitations exist which must be addressed.

6.5 Conclusion

This project contributes to the current literature as there has been a documented need for more intelligent, data-intensive methods for monitoring bee movement and behaviour. It has identified strong candidate technologies to facilitate better tracking of bees and has investigated integrating machine learning to automate the monitoring process. The output is a foundation of data, machine learning models, discussion, and future ideas that are ready for further development.

7 References

1. Potts, S. G., Biesmeijer, J. C., Kremen, C., Neumann, P., Schweiger, O., & Kunin, W. E. (2010). Global pollinator declines: Trends, impacts and drivers. *Trends in Ecology and Evolution*. <https://doi.org/10.1016/j.tree.2010.01.007>
2. Dirzo, R., Young, H. S., Galetti, M., Ceballos, G., Isaac, N. J. B., & Collen, B. (2014). Defaunation in the Anthropocene. *Science*. <https://doi.org/10.1126/science.1251817>
3. Hallmann, C. A., Sorg, M., Jongejans, E., Siepel, H., Hofland, N., Schwan, H., ... De Kroon, H. (2017). More than 75 percent decline over 27 years in total flying insect biomass in protected areas. *PLoS ONE*, *12*(10). <https://doi.org/10.1371/journal.pone.0185809>
4. Williams, P. H. (1982). The Distribution and Decline of British Bumble Bees (*Bombus Latr.*). *Journal of Apicultural Research*, *21*(4). <https://doi.org/10.1080/00218839.1982.11100549>
5. Sirois-Delisle, C., & Kerr, J. T. (2018). Climate change-driven range losses among bumblebee species are poised to accelerate. *Scientific Reports*, *8*(1). <https://doi.org/10.1038/s41598-018-32665-y>
6. Klein, A. M., Vaissière, B. E., Cane, J. H., Steffan-Dewenter, I., Cunningham, S. A., Kremen, C., & Tscharntke, T. (2007). Importance of pollinators in changing landscapes for world crops. *Proceedings of the Royal Society B: Biological Sciences*. <https://doi.org/10.1098/rspb.2006.3721>
7. Ollerton, J., Winfree, R., & Tarrant, S. (2011). How many flowering plants are pollinated by animals? *Oikos*, *120*(3). <https://doi.org/10.1111/j.1600-0706.2010.18644.x>
8. Kleijn, D., Winfree, R., Bartomeus, I., Carvalheiro, L. G., Henry, M., Isaacs, R., ... Potts, S. G. (2015). Delivery of crop pollination services is an insufficient argument for wild pollinator conservation. *Nature Communications*, *6*. <https://doi.org/10.1038/ncomms8414>

9. Patel, V., Pauli, N., Biggs, E., Barbour, L., & Boruff, B. (2021). Why bees are critical for achieving sustainable development. *Ambio*, 50(1).
<https://doi.org/10.1007/s13280-020-01333-9>
10. Sánchez-Bayo, F., & Wyckhuys, K. A. G. (2019). Worldwide decline of the entomofauna: A review of its drivers. *Biological Conservation*.
<https://doi.org/10.1016/j.biocon.2019.01.020>
11. Leather, S. R. (2018). “Ecological Armageddon” – more evidence for the drastic decline in insect numbers. *Annals of Applied Biology*.
<https://doi.org/10.1111/aab.12410>
12. Vogel, G. (2017). Where have all the insects gone? *Science*.
<https://doi.org/10.1126/science.356.6338.576>
13. Stokstad, E. (2018). European Union expands ban of three neonicotinoid pesticides. *Science*. <https://doi.org/10.1126/science.aau0152>
14. Van der Sluijs, J. P., Simon-Delso, N., Goulson, D., Maxim, L., Bonmatin, J. M., & Belzunces, L. P. (2013). Neonicotinoids, bee disorders and the sustainability of pollinator services. *Current Opinion in Environmental Sustainability*. <https://doi.org/10.1016/j.cosust.2013.05.007>
15. El Hassani, A. K., Dacher, M., Gary, V., Lambin, M., Gauthier, M., & Armengaud, C. (2008). Effects of sublethal doses of acetamiprid and thiamethoxam on the behavior of the honeybee (*Apis mellifera*). *Archives of Environmental Contamination and Toxicology*, 54(4).
<https://doi.org/10.1007/s00244-007-9071-8>
16. Brandt, A., Gorenflo, A., Siede, R., Meixner, M., & Büchler, R. (2016). The neonicotinoids thiacloprid, imidacloprid, and clothianidin affect the immunocompetence of honey bees (*Apis mellifera* L.). *Journal of Insect Physiology*, 86. <https://doi.org/10.1016/j.jinsphys.2016.01.001>
17. Goulson, D. (2003). Conserving wild bees for crop pollination. *Food, Agriculture & Environment*, 1(1).

18. Carrié, R., Andrieu, E., Ouin, A., & Steffan-Dewenter, I. (2017). Interactive effects of landscape-wide intensity of farming practices and landscape complexity on wild bee diversity. *Landscape Ecology*, 32(8).
<https://doi.org/10.1007/s10980-017-0530-y>
19. St Clair, A. L., St Clair, A. L., Zhang, G., Dolezal, A. G., O'Neal, M. E., Toth, A. L., & Toth, A. L. (2020). Diversified Farming in a Monoculture Landscape: Effects on Honey Bee Health and Wild Bee Communities. *Environmental Entomology*, 49(3). <https://doi.org/10.1093/ee/nvaa031>
20. Ahrné, K., Bengtsson, J., & Elmquist, T. (2009). Bumble bees (*Bombus* spp) along a gradient of increasing urbanization. *PLoS ONE*, 4(5).
<https://doi.org/10.1371/journal.pone.0005574>
21. Martin, S. J. (2001). The role of varroa and viral pathogens in the collapse of honeybee colonies: A modelling approach. *Journal of Applied Ecology*, 38(5).
<https://doi.org/10.1046/j.1365-2664.2001.00662.x>
22. Eliash, N., & Mikheyev, A. (2020). Varroa mite evolution: a neglected aspect of worldwide bee collapses? *Current Opinion in Insect Science*.
<https://doi.org/10.1016/j.cois.2019.11.004>
23. Keeling, M. J., Franklin, D. N., Datta, S., Brown, M. A., & Budge, G. E. (2017). Predicting the spread of the Asian hornet (*Vespa velutina*) following its incursion into Great Britain. *Scientific Reports*, 7(1).
<https://doi.org/10.1038/s41598-017-06212-0>
24. Ricketts, T. H., Regetz, J., Steffan-Dewenter, I., Cunningham, S. A., Kremen, C., Bogdanski, A., ... Viana, B. F. (2008). Landscape effects on crop pollination services: Are there general patterns? *Ecology Letters*, 11(5).
<https://doi.org/10.1111/j.1461-0248.2008.01157.x>
25. Rortais, A., Villemant, C., Gargominy, O., Rome, Q., Haxaire, J., Papachristoforou, A., & Arnold, G. (2010). A New Enemy of Honeybees in Europe: the Asian Hornet, *Vespa velutina*. *Atlas of Biodiversity Risks*.

26. Franklin, D. N., Brown, M. A., Datta, S., Cuthbertson, A. G. S., Budge, G. E., & Keeling, M. J. (2017). Invasion dynamics of Asian hornet, *Vespa velutina* (Hymenoptera: Vespidae): a case study of a commune in south-west France. *Applied Entomology and Zoology*, 52(2). <https://doi.org/10.1007/s13355-016-0470-z>
27. Tan, K., Radloff, S. E., Li, J. J., Hepburn, H. R., Yang, M. X., Zhang, L. J., & Neumann, P. (2007). Bee-hawking by the wasp, *Vespa velutina*, on the honeybees *Apis cerana* and *A. mellifera*. *Naturwissenschaften*, 94(6). <https://doi.org/10.1007/s00114-006-0210-2>
28. Ken, T., Hepburn, H. R., Radloff, S. E., Yusheng, Y., Yiqiu, L., Danyin, Z., & Neumann, P. (2005). Heat-balling wasps by honeybees. *Naturwissenschaften*, 92(10). <https://doi.org/10.1007/s00114-005-0026-5>
29. Koeniger, N., Koeniger, G., Gries, M., Tingek, S., & Kelitu, A. (1996). Observations on colony defense of *Apis nuluensis* Tingek, Koeniger and Koeniger, 1996 and predatory behavior of the hornet, *Vespa multimaculata* Pérez, 1910. *Apidologie*, 27(5). <https://doi.org/10.1051/apido:19960502>
30. Turchi, L., & Derijard, B. (2018). Options for the biological and physical control of *Vespa velutina nigrithorax* (Hym.: Vespidae) in Europe: A review. *Journal of Applied Entomology*. <https://doi.org/10.1111/jen.12515>
31. Kohl, P. L., & Rutschmann, B. (2018). The neglected bee trees: European beech forests as a home for feral honey bee colonies. *PeerJ*, 2018(4). <https://doi.org/10.7717/peerj.4602>
32. Visscher, P., & Seely, D. (2010). Bee-lining as a research technique in ecological studies of honey bees. *American Bee Journal*, 129(8), 536–539.
33. Monceau, K., Bonnard, O., & Thiéry, D. (2012). Chasing the queens of the alien predator of honeybees: A water drop in the invasiveness ocean. *Open Journal of Ecology*, 02(04). <https://doi.org/10.4236/oje.2012.24022>
34. Goldarazena, A., de Heredia, I. P., Romon, P., Iturrondobeitia, J. C., Gonzalez,

- M., & Lopez, S. (2015). Spread of the yellow-legged hornet *vespa velutina nigrithorax* du Buysson (Hymenoptera: Vespidae) across Northern Spain. *EPPO Bulletin*, 45(1). <https://doi.org/10.1111/epp.12185>
35. Chapman, J. W., Drake, V. A., & Reynolds, D. R. (2011). Recent insights from radar studies of insect flight. *Annual Review of Entomology*, 56. <https://doi.org/10.1146/annurev-ento-120709-144820>
 36. Beekman, M., & Ratnieks, F. L. W. (2000). Long-range foraging by the honey-bee, *Apis mellifera* L. *Functional Ecology*, 14(4). <https://doi.org/10.1046/j.1365-2435.2000.00443.x>
 37. Psychoudakis, D., Moulder, W., Chen, C. C., Zhu, H., & Volakis, J. L. (2008). A portable low-power harmonic radar system and conformal tag for insect tracking. *IEEE Antennas and Wireless Propagation Letters*, 7. <https://doi.org/10.1109/LAWP.2008.2004512>
 38. O'Neal, M. E., Landis, D. A., Rothwell, E., Kempel, L., & Reinhard, D. (2004). Tracking insects with harmonic radar: A case study. *American Entomologist*, 50(4). <https://doi.org/10.1093/ae/50.4.212>
 39. Batsleer, F., Bonte, D., Dekeukeleire, D., Goossens, S., Poelmans, W., Van der Cruyssen, E., ... Vandeghechuchte, M. L. (2020). The neglected impact of tracking devices on terrestrial arthropods. *Methods in Ecology and Evolution*. <https://doi.org/10.1111/2041-210X.13356>
 40. Tsai, Z. M., Jau, P. H., Kuo, N. C., Kao, J. C., Lin, K. Y., Chang, F. R., ... Wang, H. (2013). A high-range-accuracy and high-sensitivity harmonic radar using pulse pseudorandom code for bee searching. *IEEE Transactions on Microwave Theory and Techniques*, 61(1). <https://doi.org/10.1109/TMTT.2012.2230020>
 41. Maggiora, R., Sacconi, M., Milanesio, D., & Porporato, M. (2019). An Innovative Harmonic Radar to Track Flying Insects: the Case of *Vespa velutina*. *Scientific Reports*, 9(1). <https://doi.org/10.1038/s41598-019-48511-8>

42. Menzel, R., Kirbach, A., Haass, W. D., Fischer, B., Fuchs, J., Koblöfsky, M., ... Greggers, U. (2011). A common frame of reference for learned and communicated vectors in honeybee navigation. *Current Biology*, 21(8). <https://doi.org/10.1016/j.cub.2011.02.039>
43. Fischer, J., Müller, T., Spatz, A. K., Greggers, U., Grünewald, B., & Menzel, R. (2014). Neonicotinoids interfere with specific components of navigation in honeybees. *PLoS ONE*, 9(3). <https://doi.org/10.1371/journal.pone.0091364>
44. Wolf, S., McMahon, D. P., Lim, K. S., Pull, C. D., Clark, S. J., Paxton, R. J., & Osborne, J. L. (2014). So near and yet so far: Harmonic radar reveals reduced homing ability of nosema infected honeybees. *PLoS ONE*, 9(8). <https://doi.org/10.1371/journal.pone.0103989>
45. Greggers, U., Schöning, C., Degen, J., & Menzel, R. (2013). Scouts behave as streakers in honeybee swarms. *Naturwissenschaften*, 100(8). <https://doi.org/10.1007/s00114-013-1077-7>
46. Milanesio, D., Sacconi, M., Maggiora, R., Laurino, D., & Porporato, M. (2016). Design of an harmonic radar for the tracking of the Asian yellow-legged hornet. *Ecology and Evolution*, 6(7). <https://doi.org/10.1002/ece3.2011>
47. Milanesio, D., Sacconi, M., Maggiora, R., Laurino, D., & Porporato, M. (2017). Recent upgrades of the harmonic radar for the tracking of the Asian yellow-legged hornet. *Ecology and Evolution*, 7(13). <https://doi.org/10.1002/ece3.3053>
48. Hansen, R. C. (1981). Fundamental Limitations in Antennas. *Proceedings of the IEEE*, 69(2). <https://doi.org/10.1109/PROC.1981.11950>
49. Tahir, N., & Brooker, G. (2011). Recent developments and recommendations for improving harmonic radar tracking systems. In *Proceedings of the 5th European Conference on Antennas and Propagation, EUCAP 2011*.
50. Osborne, J. L., Clark, S. J., Morris, R. J., Williams, I. H., Riley, J. R., Smith, A. D., ... Edwards, A. S. (1999). A landscape-scale study of bumble bee foraging range and constancy, using harmonic radar. *Journal of Applied Ecology*, 36(4).

<https://doi.org/10.1046/j.1365-2664.1999.00428.x>

51. Riley, J. R., & Smith, A. D. (2002). Design considerations for an harmonic radar to investigate the flight of insects at low altitude. *Computers and Electronics in Agriculture*, 35(2–3). [https://doi.org/10.1016/S0168-1699\(02\)00016-9](https://doi.org/10.1016/S0168-1699(02)00016-9)
52. Lee, J. W., Kwon, H., & Lee, B. (2006). Design consideration of UHF RFID tag for increased reading range. In *IEEE MTT-S International Microwave Symposium Digest*. <https://doi.org/10.1109/MWSYM.2006.249638>
53. Wang, J., Zhang, J., Saha, R., Jin, H., & Kumar, S. (2019). Pushing the range limits of commercial passive RFIDs. In *Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2019*.
54. Nunes-Silva, P., Hrcir, M., Guimarães, J. T. F., Arruda, H., Costa, L., Pessin, G., ... Imperatriz-Fonseca, V. L. (2019). Applications of RFID technology on the study of bees. *Insectes Sociaux*. <https://doi.org/10.1007/s00040-018-0660-5>
55. de Souza, P., Marendy, P., Barbosa, K., Budi, S., Hirsch, P., Nikolic, N., ... Davie, A. (2018). Low-cost electronic tagging system for bee monitoring. *Sensors (Switzerland)*, 18(7). <https://doi.org/10.3390/s18072124>
56. Robinson, E. J. H., Smith, F. D., Sullivan, K. M. E., & Franks, N. R. (2009). Do ants make direct comparisons? *Proceedings of the Royal Society B: Biological Sciences*, 276(1667). <https://doi.org/10.1098/rspb.2009.0350>
57. Easton, A. H., & Goulson, D. (2013). The Neonicotinoid Insecticide Imidacloprid Repels Pollinating Flies and Beetles at Field-Realistic Concentrations. *PLoS ONE*, 8(1). <https://doi.org/10.1371/journal.pone.0054819>
58. Silcox, D. E., Doskocil, J. P., Sorenson, C. E., & Brandenburg, R. L. (2011). Radio frequency identification tagging: A novel approach to monitoring surface and subterranean insects. *American Entomologist*, 57(2).

<https://doi.org/10.1093/ae/57.2.86>

59. Barlow, S. E., O'Neill, M. A., & Pavlik, B. M. (2019). A prototype RFID tag for detecting bumblebee visitations within fragmented landscapes. *Journal of Biological Engineering*, 13(1). <https://doi.org/10.1186/s13036-019-0143-x>
60. Cochran, W. W., & Lord, R. D. (1963). A Radio-Tracking System for Wild Animals. *The Journal of Wildlife Management*, 27(1).
<https://doi.org/10.2307/3797775>
61. Mackay, R. S. (1964). Galapagos tortoise and marine iguana deep body temperatures measured by radio telemetry. *Nature*, 204(4956).
<https://doi.org/10.1038/204355a0>
62. Levett, S., & Walls, S. (2011). Tracking the elusive life of the Emperor Dagonfly Anax imperator Leach. *Journal of the British Dragonfly Society*, 27(1).
63. Hagen, M., Wikelski, M., & Kissling, W. D. (2011). Space use of bumblebees (*Bombus* spp.) revealed by radio-tracking. *PLoS ONE*, 6(5).
<https://doi.org/10.1371/journal.pone.0019997>
64. Wikelski, M., Moxley, J., Eaton-Mordas, A., López-Uribe, M. M., Holland, R., Moskowitz, D., ... Kays, R. (2010). Large-range movements of neotropical orchid bees observed via radio telemetry. *PLoS ONE*, 5(5).
<https://doi.org/10.1371/journal.pone.0010738>
65. Pasquet, R. S., Peltier, A., Hufford, M. B., Oudin, E., Saulnier, J., Paul, L., ... Gepts, P. (2008). Long-distance pollen flow assessment through evaluation of pollinator foraging range suggests transgene escape distances. *Proceedings of the National Academy of Sciences of the United States of America*, 105(36).
<https://doi.org/10.1073/pnas.0806040105>
66. Neill, E., & Jensen, P. (2014). Ground-based radio tracking: a best practice protocol. New Zealand Department of Conservation.

67. Cochran, W. W., Warner, D. W., Tester, J. R., & Kuechle, V. B. (1965). Automatic Radio-Tracking System for Monitoring Animal Movements. *BioScience*, 15(2). <https://doi.org/10.2307/1293346>
68. Searcy, S. W., Schueller, J. K., Bae, Y. H., & Stout, B. A. (1990). Measurement of agricultural field location using microwave frequency triangulation. *Computers and Electronics in Agriculture*, 4(3). [https://doi.org/10.1016/0168-1699\(90\)90020-P](https://doi.org/10.1016/0168-1699(90)90020-P)
69. Kays, R., Tilak, S., Crofoot, M., Fountain, T., Obando, D., Ortega, A., ... Wikelski, M. (2011). Tracking animal location and activity with an automated radio telemetry system in a tropical rainforest. *Computer Journal*, 54(12). <https://doi.org/10.1093/comjnl/bxr072>
70. Stark, K. E., Jackson, G. D., & Lyle, J. M. (2005). Tracking arrow squid movements with an automated acoustic telemetry system. *Marine Ecology Progress Series*, 299. <https://doi.org/10.3354/meps299167>
71. Körner, F., Speck, R., Göktoğan, A. H., & Sukkarieh, S. (2010). Autonomous airborne wildlife tracking using radio signal strength. In *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*. <https://doi.org/10.1109/IROS.2010.5654385>
72. Cliff, O. M., Fitch, R., Sukkarieh, S., Saunders, D. L., & Heinsohn, R. (2015). Online localization of radio-tagged wildlife with an autonomous aerial robot system. In *Robotics: Science and Systems* (Vol. 11). <https://doi.org/10.15607/RSS.2015.XI.042>
73. Dressel, L. K., & Kochenderfer, M. J. (2018). Efficient and low-cost localization of radio signals with a multirotor UAV. *2018 AIAA Guidance, Navigation, and Control Conference*.
74. Nguyen, H. Van, Chesser, M., Koh, L. P., Rezatofghi, S. H., & Ranasinghe, D. C. (2019). TrackerBots: Autonomous unmanned aerial vehicle for real-time localization and tracking of multiple radio-tagged animals. *Journal of Field*

- Robotics*, 36(3). <https://doi.org/10.1002/rob.21857>
75. Shafer, M. W., Vega, G., Rothfus, K., & Flikkema, P. (2019). UAV wildlife radiotelemetry: System and methods of localization. *Methods in Ecology and Evolution*, 10(10). <https://doi.org/10.1111/2041-210X.13261>
 76. Patton, D. R., Beaty, D. W., & Smith, R. H. (1973). Solar Panels: An Energy Source for Radio Transmitters on Wildlife. *The Journal of Wildlife Management*, 37(2). <https://doi.org/10.2307/3798910>
 77. Silva, R., Afán, I., Gil, J. A., & Bustamante, J. (2017). Seasonal and circadian biases in bird tracking with solar GPS-tags. *PLoS ONE*, 12(10). <https://doi.org/10.1371/journal.pone.0185344>
 78. Lu, Y., Marty, F., Galayko, D., Laheurte, J.-M., & Basset, P. (2017). A MEMS EVEH-Assisted Long-Range RFID Tag System for Applications with Low-Frequency Vibrations. <https://doi.org/10.3390/proceedings1040582>
 79. Chen, Y. L., Liu, D., Wang, S., Li, Y. F., & Zhang, X. S. (2019). Self-powered smart active RFID tag integrated with wearable hybrid nanogenerator. *Nano Energy*, 64. <https://doi.org/10.1016/j.nanoen.2019.103911>
 80. Umeda, M., Nakamura, K., & Ueha, S. (1996). Analysis of the transformation of mechanical impact energy to electric energy using piezoelectric vibrator. *Japanese Journal of Applied Physics, Part 1: Regular Papers and Short Notes and Review Papers*, 35(5 B). <https://doi.org/10.1143/jjap.35.3267>
 81. Takeuchi, M., Matsuzawa, S., Tairaku, K., & Takatsu, C. (2007). Piezoelectric generator as power supply for RFID-tags and applications. In *Proceedings - IEEE Ultrasonics Symposium*. <https://doi.org/10.1109/ULTSYM.2007.644>
 82. Li, H., Tian, C., Lu, J., Myjak, M. J., Martinez, J. J., Brown, R. S., & Deng, Z. D. (2016). An Energy Harvesting Underwater Acoustic Transmitter for Aquatic Animals. *Scientific Reports*, 6. <https://doi.org/10.1038/srep33804>
 83. Snowdon, M. M., Horne, J., Gyr, B., & Jia, Y. (2018). Feasibility of vibration

- energy harvesting powered wireless tracking of falcons in flight. In *Journal of Physics: Conference Series* (Vol. 1052). <https://doi.org/10.1088/1742-6596/1052/1/012049>
84. Chang, S. C., Yaul, F. M., Sullivan, F. O., Otten, D. M., Lang, J. H., & Initiative, E. (2009). Harvesting Energy from moth vibrations during flight. *International Workshop on Micro and Nanotechnologies for Power Generation and Energy Conversion Applications*.
 85. Aktakka, E. E., Kim, H., & Najafi, K. (2011). Energy scavenging from insect flight. *Journal of Micromechanics and Microengineering*, 21(9). <https://doi.org/10.1088/0960-1317/21/9/095016>
 86. Shearwood, J., Hung, D. M. Y., Cross, P., Preston, S., & Palego, C. (2018). Honey-Bee Localization Using an Energy Harvesting Device and Power Based Angle of Arrival Estimation. In *IEEE MTT-S International Microwave Symposium Digest* (Vol. 2018-June). <https://doi.org/10.1109/MWSYM.2018.8439173>
 87. Soulsbury, C. D., Gray, H. E., Smith, L. M., Braithwaite, V., Cotter, S. C., Elwood, R. W., ... Collins, L. M. (2020). The welfare and ethics of research involving wild animals: A primer. *Methods in Ecology and Evolution*. <https://doi.org/10.1111/2041-210X.13435>
 88. Drinkwater, E., Robinson, E. J. H., & Hart, A. G. (2019). Keeping invertebrate research ethical in a landscape of shifting public opinion. *Methods in Ecology and Evolution*. <https://doi.org/10.1111/2041-210X.13208>
 89. Godfrey, J. D., & Bryant, D. M. (2003). Effects of radio transmitters: Review of recent radio-tracking studies. *Science for Conservation*, (214).
 90. Aldridge, H. D. J. N., & Brigham, R. M. (1988). Load Carrying and Maneuverability in an Insectivorous Bat: a Test of the 5% "Rule" of Radio-Telemetry. *Journal of Mammalogy*, 69(2). <https://doi.org/10.2307/1381393>
 91. Geen, G. R., Robinson, R. A., & Baillie, S. R. (2019). Effects of tracking

- devices on individual birds – a review of the evidence. *Journal of Avian Biology*. <https://doi.org/10.1111/jav.01823>
92. Barron, D. G., Brawn, J. D., & Weatherhead, P. J. (2010). Meta-analysis of transmitter effects on avian behaviour and ecology. *Methods in Ecology and Evolution*, 1(2). <https://doi.org/10.1111/j.2041-210x.2010.00013.x>
93. Bodey, T. W., Cleasby, I. R., Bell, F., Parr, N., Schultz, A., Votier, S. C., & Bearhop, S. (2018). A phylogenetically controlled meta-analysis of biologging device effects on birds: Deleterious effects and a call for more standardized reporting of study data. *Methods in Ecology and Evolution*, 9(4). <https://doi.org/10.1111/2041-210X.12934>
94. Bowlin, M. S., Henningsson, P., Muijres, F. T., Vleugels, R. H. E., Liechti, F., & Hedenström, A. (2010). The effects of geolocator drag and weight on the flight ranges of small migrants. *Methods in Ecology and Evolution*, 1(4). <https://doi.org/10.1111/j.2041-210x.2010.00043.x>
95. Portugal, S. J., & White, C. R. (2018). Miniaturization of biologgers is not alleviating the 5% rule. *Methods in Ecology and Evolution*. <https://doi.org/10.1111/2041-210X.13013>
96. White, C. R., Cassey, P., Schimpf, N. G., Halsey, L. G., Green, J. A., & Portugal, S. J. (2013). Methods & techniques: Implantation reduces the negative effects of bio-logging devices on birds. *Journal of Experimental Biology*, 216(4). <https://doi.org/10.1242/jeb.076554>
97. Boiteau, G., & Colpitts, B. (2001). Electronic tags for the tracking of insects in flight: Effect of weight on flight performance of adult Colorado potato beetles. *Entomologia Experimentalis et Applicata*, 100(2). <https://doi.org/10.1046/j.1570-7458.2001.00863.x>
98. Boiteau, G., Meloche, F., Vincent, C., & Leskey, T. C. (2009). Effectiveness of glues used for harmonic radar tag attachment and impact on survival and behavior of three insect pests. *Environmental Entomology*, 38(1).

<https://doi.org/10.1603/022.038.0121>

99. Gui, L. Y., Huang, X. Q., Li, C. R., & Boiteau, G. (2011). Validation of harmonic radar tags to study movement of Chinese citrus fly. *Canadian Entomologist*, 143(4). <https://doi.org/10.4039/n11-017>
100. Feuerbacher, E., Fewell, J. H., Roberts, S. P., Smith, E. F., & Harrison, J. F. (2003). Effects of load type (pollen or nectar) and load mass on hovering metabolic rate and mechanical power output in the honey bee *Apis mellifera*. *Journal of Experimental Biology*, 206(11). <https://doi.org/10.1242/jeb.00347>
101. Kim, J., Jung, M., Kim, H. G., & Lee, D. H. (2016). Potential of harmonic radar system for use on five economically important insects: Radar tag attachment on insects and its impact on flight capacity. *Journal of Asia-Pacific Entomology*, 19(2). <https://doi.org/10.1016/j.aspen.2016.03.013>
102. Switzer, C. M., & Combes, S. A. (2016). *Bombus impatiens* (Hymenoptera: Apidae) display reduced pollen foraging behavior when marked with bee tags vs. paint. *Journal of Melittology*, (62). <https://doi.org/10.17161/jom.v0i62.5679>
103. Lach, L., Kratz, M., & Baer, B. (2015). Parasitized honey bees are less likely to forage and carry less pollen. *Journal of Invertebrate Pathology*, 130. <https://doi.org/10.1016/j.jip.2015.06.003>
104. Poissonnier, L. A., Jackson, A. L., & Tanner, C. J. (2015). Cold and CO₂ narcosis have long-lasting and dissimilar effects on *Bombus terrestris*. *Insectes Sociaux*, 62(3). <https://doi.org/10.1007/s00040-015-0404-8>
105. Cucurachi, S., Tamis, W. L. M., Vijver, M. G., Peijnenburg, W. J. G. M., Bolte, J. F. B., & de Snoo, G. R. (2013). A review of the ecological effects of radiofrequency electromagnetic fields (RF-EMF). *Environment International*. <https://doi.org/10.1016/j.envint.2012.10.009>
106. Darney, K., Giraudin, A., Joseph, R., Abadie, P., Aupinel, P., Decourtye, A., ... Gauthier, M. (2016). Effect of high-frequency radiations on survival of the honeybee (*Apis mellifera* L.). *Apidologie*, 47(5).

<https://doi.org/10.1007/s13592-015-0421-7>

107. Souza Cunha, A. E., Rose, J., Prior, J., Aumann, H. M., Emanetoglu, N. W., & Drummond, F. A. (2020). A novel non-invasive radar to monitor honey bee colony health. *Computers and Electronics in Agriculture*, 170. <https://doi.org/10.1016/j.compag.2020.105241>
108. IR 2030 - UK Interface Requirements 2030: License Exempt Short Range Devices (SRDs). (2023). Ofcom UK.
109. Aumann, H., Payal, B., Emanetoglu, N. W., & Drummond, F. (2017). An index for assessing the foraging activities of honeybees with a Doppler sensor. In *SAS 2017 - 2017 IEEE Sensors Applications Symposium, Proceedings*. <https://doi.org/10.1109/SAS.2017.7894090>
110. Aumann, H. M. (2018). A technique for measuring the RCS of free-flying honeybees with a 24 GHz CW Doppler radar. In *IET Conference Publications* (Vol. 2018). <https://doi.org/10.1049/cp.2018.0540>
111. Aumann, H. M., & Emanetoglu, N. W. (2017). The radar microphone: A new way of monitoring honey bee sounds. In *Proceedings of IEEE Sensors*. <https://doi.org/10.1109/ICSENS.2016.7808865>
112. Zacepins, A., Brusbardis, V., Meitalovs, J., & Stalidzans, E. (2015). Challenges in the development of Precision Beekeeping. *Biosystems Engineering*. <https://doi.org/10.1016/j.biosystemseng.2014.12.001>
113. Meikle, W. G., & Holst, N. (2015). Application of continuous monitoring of honeybee colonies. *Apidologie*. <https://doi.org/10.1007/s13592-014-0298-x>
114. Zacepins, A., Kviesis, A., Pecka, A., & Osadcuks, V. (2017). Development of Internet of Things concept for Precision Beekeeping. In *2017 18th International Carpathian Control Conference, ICC 2017*. <https://doi.org/10.1109/CarpathianCC.2017.7970365>
115. Human, H., Brodschneider, R., Dietemann, V., Dively, G., Ellis, J. D.,

- Forsgren, E., ... Zheng, H. Q. (2013). Miscellaneous standard methods for *Apis mellifera* research. *Journal of Apicultural Research*.
<https://doi.org/10.3896/IBRA.1.52.4.10>
116. Zacepins, A., & Stalidzans, E. (2013). Information processing for remote recognition of the state of bee colonies and apiaries in precision beekeeping (apiculture). *Biosystems and Information technology*, 2(1).
<https://doi.org/10.11592/bit.130502>
117. Dineva, K., & Atanasova, T. (2018). Applying machine learning against beehives dataset. In *International Multidisciplinary Scientific GeoConference Surveying Geology and Mining Ecology Management, SGEM* (Vol. 18).
<https://doi.org/10.5593/sgem2018/6.2/S25.005>
118. Kridi, D. S., de Carvalho, C. G. N., & Gomes, D. G. (2016). Application of wireless sensor networks for beehive monitoring and in-hive thermal patterns detection. *Computers and Electronics in Agriculture*, 127.
<https://doi.org/10.1016/j.compag.2016.05.013>
119. Freitas, B. M., Sousa, R. M., & Bomfim, I. G. A. (2007). Absconding and migratory behaviors of feral Africanized honey bee (*Apis mellifera* L.) colonies in NE Brazil. *Acta Scientiarum - Biological Sciences*, 29(4).
120. Edwards-Murphy, F., Magno, M., Whelan, P. M., O'Halloran, J., & Popovici, E. M. (2016). B+WSN: Smart beehive with preliminary decision tree analysis for agriculture and honey bee health monitoring. *Computers and Electronics in Agriculture*, 124. <https://doi.org/10.1016/j.compag.2016.04.008>
121. Robles-Guerrero, A., Saucedo-Anaya, T., González-Ramírez, E., & De la Rosa-Vargas, J. I. (2019). Analysis of a multiclass classification problem by Lasso Logistic Regression and Singular Value Decomposition to identify sound patterns in queenless bee colonies. *Computers and Electronics in Agriculture*, 159. <https://doi.org/10.1016/j.compag.2019.02.024>
122. Rafael Braga, A., G. Gomes, D., Rogers, R., E. Hassler, E., M. Freitas, B., &

- A. Cazier, J. (2020). A method for mining combined data from in-hive sensors, weather and apiary inspections to forecast the health status of honey bee colonies. *Computers and Electronics in Agriculture*, 169. <https://doi.org/10.1016/j.compag.2019.105161>
123. Luke, E. P., Kollias, P., Johnson, K. L., & Clothiaux, E. E. (2008). A technique for the automatic detection of insect clutter in cloud radar returns. *Journal of Atmospheric and Oceanic Technology*, 25(9). <https://doi.org/10.1175/2007JTECHA953.1>
124. Islam, T., Rico-Ramirez, M. A., Han, D., & Srivastava, P. K. (2012). Artificial intelligence techniques for clutter identification with polarimetric radar signatures. *Atmospheric Research*, 109–110. <https://doi.org/10.1016/j.atmosres.2012.02.007>
125. Cabanes, G., Bennani, Y., Chartagnat, C., & Fresneau, D. (2008). Topographic connectionist unsupervised learning for RFID behavior data mining. In *RFID Technology - Concepts, Applications, Challenges - Proceedings of the 2nd International Workshop on RFID Technology - Concepts, Applications, Challenges, IWRT 2008; In Conjunction with ICEIS 2008*. <https://doi.org/10.5220/0001733400630072>
126. Susanto, F., Gillard, T., De Souza, P., Vincent, B., Budi, S., Almeida, A., ... He, J. (2018). Addressing RFID misreadings to better infer bee hive activity. *IEEE Access*, 6. <https://doi.org/10.1109/ACCESS.2018.2844181>
127. Hu, C., Kong, S., Wang, R., Long, T., & Fu, X. (2018). Identification of Migratory Insects from their Physical Features using a Decision-Tree Support Vector Machine and its Application to Radar Entomology. *Scientific Reports*, 8(1). <https://doi.org/10.1038/s41598-018-23825-1>
128. Hu, C., Kong, S., Wang, R., Zhang, F., & Wang, L. (2020). Insect mass estimation based on radar cross section parameters and support vector regression algorithm. *Remote Sensing*. <https://doi.org/10.3390/rs12111903>

129. Arruda, H., Imperatriz-Fonseca, V., De Souza, P., & Pessin, G. (2018). Identifying Bee Species by Means of the Foraging Pattern Using Machine Learning. In *Proceedings of the International Joint Conference on Neural Networks* (Vol. 2018-July). <https://doi.org/10.1109/IJCNN.2018.8489608>
130. Gomes, P. A. B., de Carvalho, E. C., Arruda, H. M., de Souza, P., & Pessin, G. (2017). Exploiting recurrent neural networks in the forecasting of bees' level of activity. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 10613 LNCS). https://doi.org/10.1007/978-3-319-68600-4_30
131. Gama, F., Arruda, H. M., Carvalho, H. V., de Souza, P., & Pessin, G. (2017). Improving our understanding of the behavior of bees through anomaly detection techniques. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 10614 LNCS). https://doi.org/10.1007/978-3-319-68612-7_59
132. Yang, C., & Collins, J. (2016). A model for honey bee tracking on 2D video. In *International Conference Image and Vision Computing New Zealand* (Vol. 2016-November). <https://doi.org/10.1109/IVCNZ.2015.7761542>
133. Magnier, B., Ekszterowicz, G., Laurent, J., Rival, M., & Pfister, F. (2018). Bee hive traffic monitoring by tracking bee flight paths. In *VISIGRAPP 2018 - Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications* (Vol. 5). <https://doi.org/10.5220/0006628205630571>
134. Westwanska, W. W., & Respondek, J. S. (2019). Counting instances of objects in color images using u-net network on example of honey bees. In *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, FedCSIS 2019*. <https://doi.org/10.15439/2019F94>
135. Kulyukin, V., & Mukherjee, S. (2019). On video analysis of omnidirectional bee traffic: Counting bee motions with motion detection and image classification.

- Applied Sciences (Switzerland)*, 9(18). <https://doi.org/10.3390/app9183743>
136. Urteaga-Reyesvera, J. C., & Possani-Espinosa, A. (2016). Scorpions: Classification of poisonous species using shape features. In *2016 International Conference on Electronics, Communications and Computers, CONIELECOMP 2016*. <https://doi.org/10.1109/CONIELECOMP.2016.7438563>
 137. Puig, E., Gonzalez, F., Hamilton, G., & Grundy, P. (2015). Assessment of crop insect damage using unmanned aerial systems: A machine learning approach. In *Proceedings - 21st International Congress on Modelling and Simulation, MODSIM 2015*. <https://doi.org/10.36334/modsim.2015.f12.puig>
 138. Alves, T. S., Pinto, M. A., Ventura, P., Neves, C. J., Biron, D. G., Junior, A. C., ... Rodrigues, P. J. (2020). Automatic detection and classification of honey bee comb cells using deep learning. *Computers and Electronics in Agriculture*, 170. <https://doi.org/10.1016/j.compag.2020.105244>
 139. Veeraraghavan, A., Chellappa, R., & Srinivasan, M. (2008). Shape-and-behavior-encoded tracking of bee dances. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3). <https://doi.org/10.1109/TPAMI.2007.70707>
 140. Blut, C., Crespi, A., Mersch, D., Keller, L., Zhao, L., Kollmann, M., ... Beye, M. (2017). Automated computer-based detection of encounter behaviours in groups of honeybees. *Scientific Reports*, 7(1). <https://doi.org/10.1038/s41598-017-17863-4>
 141. Boenisch, F., Rosemann, B., Wild, B., Dormagen, D., Wario, F., & Landgraf, T. (2018). Tracking all members of a honey bee colony over their lifetime using learned models of correspondence. *Frontiers Robotics AI*, 5(APR). <https://doi.org/10.3389/frobt.2018.00035>
 142. Schmaranzer, A., & Stabentheiner, S. (1987). Thermographic determination of body temperatures in honey bees and hornets: Calibration and applications. *Thermology*, 2.

143. MARTIN, S. J. (1990). Nest thermoregulation in *Vespa simillima*, *V. tropica* and *V. analis*. *Ecological Entomology*, 15(3). <https://doi.org/10.1111/j.1365-2311.1990.tb00812.x>
144. Roberts, B. R., & Osborne, J. L. (2019). Testing the efficacy of a thermal camera as a search tool for locating wild bumble bee nests. *Journal of Apicultural Research*, 58(4). <https://doi.org/10.1080/00218839.2019.1614724>
145. Liroy, S., Bianchi, E., Biglia, A., Bessone, M., Laurino, D., & Porporato, M. (2021). Viability of thermal imaging in detecting nests of the invasive hornet *Vespa velutina*. *Insect Science*, 28(1). <https://doi.org/10.1111/1744-7917.12760>
146. Koziarski, M., & Cyganek, B. (2018). Impact of low resolution on image recognition with deep neural networks: An experimental study. *International Journal of Applied Mathematics and Computer Science*, 28(4). <https://doi.org/10.2478/amcs-2018-0056>
147. Mei, L., Guan, Z. G., Zhou, H. J., Lv, J., Zhu, Z. R., Cheng, J. A., ... Somesfalean, G. (2012). Agricultural pest monitoring using fluorescence lidar techniques Feasibility study. In *Applied Physics B: Lasers and Optics* (Vol. 106). <https://doi.org/10.1007/s00340-011-4785-8>
148. Adelabu, S., Mutanga, O., & Adam, E. (2014). Evaluating the impact of red-edge band from Rapideye image for classifying insect defoliation levels. *ISPRS Journal of Photogrammetry and Remote Sensing*, 95. <https://doi.org/10.1016/j.isprsjprs.2014.05.013>
149. Adelabu, S., Mutanga, O., Adam, E., & Sebego, R. (2014). Spectral discrimination of insect defoliation levels in mopane woodland using hyperspectral data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(1). <https://doi.org/10.1109/JSTARS.2013.2258329>
150. Mullen, E. R., Rutschman, P., Pegram, N., Patt, J. M., Adamczyk, J. J., &

- Johanson. (2016). Laser system for identification, tracking, and control of flying insects. *Optics Express*, 24(11). <https://doi.org/10.1364/oe.24.011828>
151. Silva, D. F., De Souza, V. M. A., Batista, G. E. A. P. A., Keogh, E., & Ellis, D. P. W. (2013). Applying machine learning and audio analysis techniques to insect recognition in intelligent traps. In *Proceedings - 2013 12th International Conference on Machine Learning and Applications, ICMLA 2013* (Vol. 1). <https://doi.org/10.1109/ICMLA.2013.24>
152. Silva, D. F., Souza, V. M. A., Ellis, D. P. W., Keogh, E. J., & Batista, G. E. A. P. A. (2015). Exploring Low Cost Laser Sensors to Identify Flying Insect Species. *Journal of Intelligent & Robotic Systems*, 80(S1). <https://doi.org/10.1007/s10846-014-0168-9>
153. Heupel, M. R., Semmens, J. M., & Hobday, A. J. (2006). Automated acoustic tracking of aquatic animals: Scales, design and deployment of listening station arrays. *Marine and Freshwater Research*. <https://doi.org/10.1071/MF05091>
154. Lehmann, E. A., & Johansson, A. M. (2007). Particle filter with integrated voice activity detection for acoustic source tracking. *Eurasip Journal on Advances in Signal Processing*, 2007. <https://doi.org/10.1155/2007/50870>
155. Birchfield, S. T., & Gangishetty, R. (2005). Acoustic localization by interaural level difference. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* (Vol. IV). <https://doi.org/10.1109/ICASSP.2005.1416207>
156. Valin, J. M., Michaud, F., Rouat, J., & Létourneau, D. (2003). Robust Sound Source Localization Using a Microphone Array on a Mobile Robot. In *IEEE International Conference on Intelligent Robots and Systems* (Vol. 2). <https://doi.org/10.1109/iros.2003.1248813>
157. Argentieri, S., Danès, P., & Souères, P. (2015). A survey on sound source localization in robotics: From binaural to array processing methods. *Computer Speech and Language*, 34(1). <https://doi.org/10.1016/j.csl.2015.03.003>

158. Zhang, C., Florencio, D., Ba, D. E., & Zhang, Z. (2008). Maximum likelihood sound source localization and beamforming for directional microphone arrays in distributed meetings. In *IEEE Transactions on Multimedia* (Vol. 10). <https://doi.org/10.1109/TMM.2008.917406>
159. Pavlidi, D., Griffin, A., Puigt, M., & Mouchtaris, A. (2013). Real-time multiple sound source localization and counting using a circular microphone array. *IEEE Transactions on Audio, Speech and Language Processing*, 21(10). <https://doi.org/10.1109/TASL.2013.2272524>
160. Nakadai, K., Matsuura, D., Okuno, H. G., & Kitano, H. (2003). Applying Scattering Theory to Robot Audition System: Robust Sound Source Localization and Extraction. In *IEEE International Conference on Intelligent Robots and Systems* (Vol. 2). <https://doi.org/10.1109/iros.2003.1248800>
161. Nakamura, K., Nakadai, K., Asano, F., Hasegawa, Y., & Tsujino, H. (2009). Intelligent sound source localization for dynamic environments. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*. <https://doi.org/10.1109/IROS.2009.5354419>
162. Hoshiya, K., Washizaki, K., Wakabayashi, M., Ishiki, T., Kumon, M., Bando, Y., ... Okuno, H. G. (2017). Design of UAV-embedded microphone array system for sound source localization in outdoor environments. *Sensors (Switzerland)*, 17(11). <https://doi.org/10.3390/s17112535>
163. Strauss, M., Mordel, P., Miguet, V., & Deleforge, A. (2018). DREGON: Dataset and Methods for UAV-Embedded Sound Source Localization. In *IEEE International Conference on Intelligent Robots and Systems*. <https://doi.org/10.1109/IROS.2018.8593581>
164. Wang, L., & Cavallaro, A. (2018). Acoustic sensing from a multi-rotor drone. *IEEE Sensors Journal*, 18(11). <https://doi.org/10.1109/JSEN.2018.2825879>
165. Ruiz-Espitia, O., Martinez-Carranza, J., & Rascon, C. (2018). AIRA-UAS: An Evaluation Corpus for Audio Processing in Unmanned Aerial System. In *2018*

International Conference on Unmanned Aircraft Systems, ICUAS 2018.

<https://doi.org/10.1109/ICUAS.2018.8453466>

166. Michelsen, A., Towne, W. F., Kirchner, W. H., & Kryger, P. (1987). The acoustic near field of a dancing honeybee. *Journal of Comparative Physiology A*, 161(5). <https://doi.org/10.1007/BF00605005>
167. LARSEN, O. N., GLEFFE, G., & TENGÖ, J. (1986). Vibration and sound communication in solitary bees and wasps. *Physiological Entomology*, 11(3). <https://doi.org/10.1111/j.1365-3032.1986.tb00416.x>
168. Islam, R., Stimpson, A., & Cummings, M. (2017). *Small UAV Noise Analysis. Humans and Autonomy Laboratory, Duke University, Durham, NC, USA.*
169. Coggins, K. M., & Principe, J. (1998). Detection and classification of insect sounds in a grain silo using a neural network. In *IEEE International Conference on Neural Networks - Conference Proceedings (Vol. 3)*. <https://doi.org/10.1109/ijcnn.1998.687123>
170. Wang, Y., Huang, Q., Zhang, X., & Ren, B. (2014). Application of insect songs in monitoring population density level of *Locusta migratoria migratoria* (Orthoptera: Acrididae). *Zoological Studies*, 53(1). <https://doi.org/10.1186/s40555-014-0055-x>
171. Dong, X., Yan, N., & Wei, Y. (2018). Insect Sound Recognition Based on Convolutional Neural Network. In *2018 3rd IEEE International Conference on Image, Vision and Computing, ICIVC 2018*. <https://doi.org/10.1109/ICIVC.2018.8492871>
172. Kulyukin, V., Mukherjee, S., & Amlathe, P. (2018). Toward audio beehive monitoring: Deep learning vs. standard machine learning in classifying beehive audio samples. *Applied Sciences (Switzerland)*, 8(9). <https://doi.org/10.3390/app8091573>
173. Kawakita, S., & Ichikawa, K. (2019). Automated classification of bees and hornet using acoustic analysis of their flight sounds. *Apidologie*, 50(1).

<https://doi.org/10.1007/s13592-018-0619-6>

174. Heise, D., Miller, Z., Harrison, E., Gradisek, A., Grad, J., & Galen, C. (2019). Acoustically Tracking the Comings and Goings of Bumblebees. In *SAS 2019 - 2019 IEEE Sensors Applications Symposium, Conference Proceedings*. <https://doi.org/10.1109/SAS.2019.8705973>
175. Van Goethem, S., Verwulgen, S., Goethijn, F., & Steckel, J. (2019). An IoT solution for measuring bee pollination efficacy. In *IEEE 5th World Forum on Internet of Things, WF-IoT 2019 - Conference Proceedings*. <https://doi.org/10.1109/WF-IoT.2019.8767298>
176. Mehta, R., Sharifzadeh, S., Palade, V., Tan, B., Daneshkhah, A., & Karayaneva, Y. (2023). Deep Learning Techniques for Radar-Based Continuous Human Activity Recognition. *Machine Learning and Knowledge Extraction*, 5(4), 1493–1518. <https://doi.org/10.3390/make5040075>
177. Zgank, A. (2018). Acoustic monitoring and classification of bee swarm activity using MFCC feature extraction and HMM acoustic modeling. In *12th International Conference ELEKTRO 2018, 2018 ELEKTRO Conference Proceedings*. <https://doi.org/10.1109/ELEKTRO.2018.8398253>
178. Feldman, A., & Balch, T. (2003). Automatic identification of bee movement. *Proceedings of the 2nd International Workshop on the Mathematics and Algorithms of Social Insects*.
179. Feldman, A., & Balch, T. (2003). Automatic Identification of Bee Movement Using Human Trainable Models of Behavior. *Mathematics and Algorithms of Social Insects*.
180. Arif Abdul Rahuman, S., Veerappan, J., & Rajesh, R. V. (2016). Classification of flying insects with high performance using improved DTW algorithm based on hidden Markov model. *Brazilian Archives of Biology and Technology*, 59(Specialissue2). <https://doi.org/10.1590/1678-4324-2016161054>
181. Polyniak, Y., Fedasyuk, D., & Marusenkova, T. (2019). Identification of Bee

- Colony Acoustic Signals using the Dynamic Time Warping Algorithm. *ECONTECHMOD: An International Quarterly Journal on Economics of Technology and Modelling Processes*, 8.
182. Ruvinga, S., Hunter, G. J. A., Duran, O., & Nebel, J. C. (2021). Use of LSTM Networks to Identify “Queenlessness” in Honeybee Hives from Audio Signals. In *2021 17th International Conference on Intelligent Environments, IE 2021 - Proceedings*. <https://doi.org/10.1109/IE51775.2021.9486575>
 183. Gomes, P. A. B., Suhara, Y., Nunes-Silva, P., Costa, L., Arruda, H., Venturieri, G., ... Pessin, G. (2020). An Amazon stingless bee foraging activity predicted using recurrent artificial neural networks and attribute selection. *Scientific Reports*, 10(1). <https://doi.org/10.1038/s41598-019-56352-8>
 184. Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6). <https://doi.org/10.1037/h0042519>
 185. Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.
 186. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (Vol. 2017-December).
 187. Drucker, H., Surges, C. J. C., Kaufman, L., Smola, A., & Vapnik, V. (1997). Support vector regression machines. In *Advances in Neural Information Processing Systems*.
 188. Breiman, L. (2001). Random forests. *Machine Learning*. <https://doi.org/10.1023/A:1010933404324>
 189. Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H., & Deng, S. H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology*, 17(1).

<https://doi.org/10.11989/JEST.1674-862X.80904120>

190. Martinez-Cantin, R. (2015). BayesOpt: A Bayesian optimization library for nonlinear optimization, experimental design and bandits. *Journal of Machine Learning Research*, 15.
191. Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. 2006. The MIT Press, Cambridge, MA, USA (Vol. 38).
192. Stauffer, C., & Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2.
<https://doi.org/10.1109/cvpr.1999.784637>
193. Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering, Transactions of the ASME*, 82(1).
<https://doi.org/10.1115/1.3662552>
194. Fazlali, B., & Eshghi, M. (2011). A Pipeline design for implementation of LPC Feature Extraction System based on Levinson-Durbin Algorithm. In *2011 19th Iranian Conference on Electrical Engineering, ICEE 2011*.
195. Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, 300.
<https://doi.org/10.1016/j.neucom.2017.11.077>
196. Javier, R. J., & Kim, Y. (2014). Application of linear predictive coding for human activity classification based on micro-doppler signatures. *IEEE Geoscience and Remote Sensing Letters*, 11(10).
<https://doi.org/10.1109/LGRS.2014.2311819>
197. McLoughlin, I. V. (2008). Line spectral pairs. *Signal Processing*.
<https://doi.org/10.1016/j.sigpro.2007.09.003>
198. Makhoul, J. (1975). Linear Prediction: A Tutorial Review. *Proceedings of the IEEE*, 63(4). <https://doi.org/10.1109/PROC.1975.9792>

199. Sigurdsson, S., Petersen, K. B., & Lehn-Schiøler, T. (2006). Mel frequency cepstral coefficients: An evaluation of robustness of MP3 encoded music. In *ISMIR 2006 - 7th International Conference on Music Information Retrieval*.
200. Munkres, J. (1957). Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1). <https://doi.org/10.1137/0105003>
201. Williams, S. M., Aldabashi, N., Palego, C., Woodgate, J. L., Makinson, J. C., & Cross, P. (2021). Early prediction of bumblebee flight task using machine learning. *Computers and Electronics in Agriculture*, 184. <https://doi.org/10.1016/j.compag.2021.106065>
202. Shearwood, J., Williams, S., Aldabashi, N., Cross, P., Freitas, B. M., Zhang, C., & Palego, C. (2020). Localization and tracking bees using a battery-less transmitter and an autonomous unmanned aerial vehicle. In *IEEE MTT-S International Microwave Symposium Digest (Vol. 2020-August)*. <https://doi.org/10.1109/IMS30576.2020.9223950>
203. Woodgate, J. L., Makinson, J. C., Lim, K. S., Reynolds, A. M., & Chittka, L. (2016). Life-long radar tracking of bumblebees. *PLoS ONE*. <https://doi.org/10.1371/journal.pone.0160333>
204. Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*.
205. Platt, J., & others. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*.
206. Ward, J. H. (1963). Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58(301). <https://doi.org/10.1080/01621459.1963.10500845>
207. Williams, S. M., Bariselli, S., Palego, C., Holland, R., & Cross, P. (2022). A

- comparison of machine-learning assisted optical and thermal camera systems for beehive activity counting. *Smart Agricultural Technology*, 2. <https://doi.org/10.1016/j.atech.2022.100038>
208. Odemer, R. (2022). Approaches, challenges and recent advances in automated bee counting devices: A review. *Annals of Applied Biology*. <https://doi.org/10.1111/aab.12727>
209. Struye, M. H., Mortier, H. J., Arnold, G., Miniggio, C., & Borneck, R. (1994). Microprocessor-controlled monitoring of honeybee flight activity at the hive entrance. *Apidologie*, 25(4). <https://doi.org/10.1051/apido:19940405>
210. Welch, G., & Bishop, G. (2006). An Introduction to the Kalman Filter. *In Practice*, 7(1). <https://doi.org/10.1.1.117.6808>
211. Mockus, J. (1977). ON BAYESIAN METHODS FOR SEEKING THE EXTREMUM AND THEIR APPLICATION.
212. Nair, V., & Hinton, G. E. (2010). Rectified linear units improve Restricted Boltzmann machines. In *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*.
213. Donovan, B. J. (1984). Occurrence of the common wasp, *vespula vulgaris* (L.) (hymenoptera: Vespidae) in new zealand. *New Zealand Journal of Zoology*, 11(4). <https://doi.org/10.1080/03014223.1984.10428256>
214. Williams, S. M., Aldabashi, N., Cross, P., & Palego, C. (2023). Challenges in Developing a Real-Time Bee-Counting Radar. *Sensors*, 23(11). <https://doi.org/10.3390/s23115250>
215. Aldabashi, N., Williams, S., Eltokhy, A., Palmer, E., Cross, P., & Palego, C. (2021). Integration of 5.8GHz Doppler Radar and Machine Learning for Automated Honeybee Hive Surveillance and Logging. In *IEEE MTT-S International Microwave Symposium Digest (Vol. 2021-June)*. <https://doi.org/10.1109/IMS19712.2021.9574826>

216. Aldabashi, N., Morton Williams, S., Eltokhy, A., Palmer, E., Cross, P., & Palego, C. (2023). A Machine Learning Integrated 5.8-GHz Continuous-Wave Radar for Honeybee Monitoring and Behavior Classification. *IEEE Transactions on Microwave Theory and Techniques*.
217. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. <https://doi.org/10.1109/CVPR.2018.00474>
218. Shaughnessy, D. O. (1988). Linear predictive coding. *IEEE Potentials*. <https://doi.org/10.1109/45.1890>
219. Itakura, F. (1975). Line spectrum representation of linear predictor coefficients of speech signals. *The Journal of the Acoustical Society of America*. <https://doi.org/10.1121/1.1995189>
220. Vergin, R., O'Shaughnessy, D., & Farhat, A. (1999). Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition. *IEEE Transactions on Speech and Audio Processing*. <https://doi.org/10.1109/89.784104>
221. Capaldi, E. A., Smith, A. D., Osborne, J. L., Fahrbach, S. E., Farris, S. M., Reynolds, D. R., ... Riley, J. R. (2000). Ontogeny of orientation flight in the honeybee revealed by harmonic radar. *Nature*. <https://doi.org/10.1038/35000564>
222. Thielens, A., Greco, M. K., Verloock, L., Martens, L., & Joseph, W. (2020). Radio-Frequency Electromagnetic Field Exposure of Western Honey Bees. *Scientific Reports*, *10*(1). <https://doi.org/10.1038/s41598-019-56948-0>
223. Favre, D. (2011). Mobile phone-induced honeybee worker piping. *Apidologie*, *42*(3). <https://doi.org/10.1007/s13592-011-0016-x>
224. Khan, M. I., Jan, M. A., Muhammad, Y., Do, D. T., Rehman, A. ur, Mavromoustakis, C. X., & Pallis, E. (2021). Tracking vital signs of a patient

- using channel state information and machine learning for a smart healthcare system. *Neural Computing and Applications*. <https://doi.org/10.1007/s00521-020-05631-x>
225. Mathur, A., & Foody, G. M. (2008). Multiclass and binary SVM classification: Implications for training and classification users. *IEEE Geoscience and Remote Sensing Letters*, 5(2). <https://doi.org/10.1109/LGRS.2008.915597>
226. Duan, K. B., & Keerthi, S. S. (2005). Which is the best multiclass SVM method? An empirical study. In *Lecture Notes in Computer Science*. https://doi.org/10.1007/11494683_28
227. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017 (Vol. 2017-January)*. <https://doi.org/10.1109/CVPR.2017.243>
228. Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.
229. Schwarz, D., & Rodet, X. (1999). Spectral Envelope Estimation and Representation for Sound Analysis-Synthesis. In *International Computer Music Conference, ICMC Proceedings*.
230. Isbell, F., Gonzalez, A., Loreau, M., Cowles, J., Díaz, S., Hector, A., ... Larigauderie, A. (2017). Linking the influence and dependence of people on biodiversity across scales. *Nature*. <https://doi.org/10.1038/nature22899>
231. Riley, J. R., Smith, A. D., Reynolds, D. R., Edwards, A. S., Osborne, J. L., Williams, I. H., ... Poppy, G. M. (1996). Tracking bees with harmonic radar. *Nature*, 379(6560). <https://doi.org/10.1038/379029b0>
232. Gurbuz, O. D., & Rebeiz, G. M. (2015). A 1.6-2.3-GHz RF MEMS reconfigurable quadrature coupler and its application to a 360° reflective-type phase shifter. *IEEE Transactions on Microwave Theory and Techniques*,

- 63(2). <https://doi.org/10.1109/TMTT.2014.2379258>
233. Braga, A. R., Hassler, E. E., Gomes, D. G., Freitas, B. M., & Cazier, J. A. (2019). IoT for development: Building a classification algorithm to help beekeepers detect honeybee health problems early. In *25th Americas Conference on Information Systems, AMCIS 2019*.
234. Shearwood, J., Aldabashi, N., Eltokhy, A., Franklin, E. L., Raine, N. E., Zhang, C., ... Palego, C. (2021). C-Band Telemetry of Insect Pollinators Using a Miniature Transmitter and a Self-Piloted Drone. *IEEE Transactions on Microwave Theory and Techniques*, 69(1).
<https://doi.org/10.1109/TMTT.2020.3034323>
235. Hajovsky, R. G., Deam, A. P., & Lagrone, A. H. (1966). Radar Reflections from Insects in the Lower Atmosphere. *IEEE Transactions on Antennas and Propagation*. <https://doi.org/10.1109/TAP.1966.1138665>
236. Wolf, W. W., Vaughn, C. R., Harris, R., & Loper, G. M. (1993). Insect radar cross-sections for aerial density measurements and target classification. *Transactions of the American Society of Agricultural Engineers*, 36(3).
<https://doi.org/10.13031/2013.28420>
237. Morse, D. H. (1971). The Insectivorous Bird as an Adaptive Strategy. *Annual Review of Ecology and Systematics*, 2(1).
<https://doi.org/10.1146/annurev.es.02.110171.001141>
238. Öckinger, E., & Smith, H. G. (2007). Semi-natural grasslands as population sources for pollinating insects in agricultural landscapes. *Journal of Applied Ecology*, 44(1). <https://doi.org/10.1111/j.1365-2664.2006.01250.x>
239. Yang, L. H., & Gratton, C. (2014). Insects as drivers of ecosystem processes. *Current Opinion in Insect Science*. <https://doi.org/10.1016/j.cois.2014.06.004>
240. The Regional Report for Africa on Pollinators and Pollination and Food Production. (2016). In *IPBES and FAO* (pp. 1–49). Bonn, Germany.

241. Carreck, N., & Williams, I. (1998). The economic value of bees in the UK. *Bee World*, 79(3). <https://doi.org/10.1080/0005772X.1998.11099393>
242. Kairo, G., Biron, D. G., Ben Abdelkader, F., Bonnet, M., Tchamitchian, S., Cousin, M., ... Brunet, J. L. (2017). Nosema ceranae, Fipronil and their combination compromise honey bee reproduction via changes in male physiology. *Scientific Reports*, 7(1). <https://doi.org/10.1038/s41598-017-08380-5>
243. Schneider, S. S., & McNally, L. C. (1992). Factors influencing seasonal absconding in colonies of the African honey bee, *Apis mellifera scutellata*. *Insectes Sociaux*, 39(4). <https://doi.org/10.1007/BF01240624>
244. Woodgate, J. L., Makinson, J. C., Rossi, N., Lim, K. S., Reynolds, A. M., Rawlings, C. J., & Chittka, L. (2021). Harmonic radar tracking reveals that honeybee drones navigate between multiple aerial leks. *iScience*, 24(6). <https://doi.org/10.1016/j.isci.2021.102499>
245. Reynolds, A. M., Swain, J. L., Smith, A. D., Martin, A. P., & Osborne, J. L. (2009). Honeybees use a Lévy flight search strategy and odour-mediated anemotaxis to relocate food sources. *Behavioral Ecology and Sociobiology*, 64(1). <https://doi.org/10.1007/s00265-009-0826-2>
246. King, M. J., Buchmann, S. L., & Spangler, H. (1996). Activity of asynchronous flight muscle from two bee families during sonication (buzzing). *Journal of Experimental Biology*. <https://doi.org/10.1242/jeb.199.10.2317>
247. Jankauski, M. A. (2020). Measuring the frequency response of the honeybee thorax. *Bioinspiration and Biomimetics*, 15(4). <https://doi.org/10.1088/1748-3190/ab835b>
248. Carlson, A. B. (2000). *Communications Systems* (3rd Editio.). New York, NY, USA: McGraw-Hill.

Appendices

A - Localization and Tracking Bees Using a Battery-less Transmitter and an Autonomous Unmanned Aerial Vehicle*

1. Introduction

Novel technologies are required to understand the recent declines in pollinators and particularly bee species [230]. Although state of the art telemetry systems have provided some insights [63], [231], they rely on transmitters that exceed a bee's body weight or are unable to cover the entire foraging range.

Recent studies have achieved considerable size and weight reduction of the transmitters by replacing the battery with a compact energy harvester, whilst demonstrating localization and near hive monitoring [86].

A radio telemetry system designed for the localization and long range tracking of honeybees (*Apis mellifera*) and bumblebees (*Bombus* spp.) is presented herein. The system comprises a miniaturized energy harvester to power a 5.8 GHz bee-wearable transmitter. A compact phased-array antenna is combined with a logarithmic detector and processing unit to compute the AOA. A built-in WiFi module supports autonomous logging for Low Range (LoRa) sensor networks and unobtrusive monitoring applications. The fabrication of a custom made housing unit allows for the integration of the tracking system to an unmanned aerial vehicle (UAV), which is programmed to locate and autonomously follow the emitted signal.

*Published as J. Shearwood, S Williams, N Aldabashi, P Cross, B M Freitas, C Zhang and C Palego., "Localization and Tracking Bees Using a Battery-less Transmitter and an Autonomous Unmanned Aerial Vehicle," *2020 IEEE/MTT-S International Microwave Symposium (IMS)*, Los Angeles, CA, USA, 2020, pp. 1263-1266, doi: 10.1109/IMS30576.2020.9223950.

2. Miniature Self-powered Radio Telemetry Tag

The self-powered transmitter tag is shown with its main components in Figure 1 (a). To determine the maximum tag weight which honeybees can support, dummy tags were 3D printed with various weights and aspect ratios and attached to insects as shown in Figure 1 (b). Table 1 reports the results of tests performed to assess flight capability which allowed us to determine a maximum tag weight of 35mg.

This corresponds to approximately a third of *A. mellifera*'s mass (~95mg). By contrast, a ~80mg tag matching the one in previous research [86] could be carried by large insect such as carpenter bees (*Xylocopa* spp.) (~240mg) and bumblebees (~200mg), but inhibited honeybees flight.

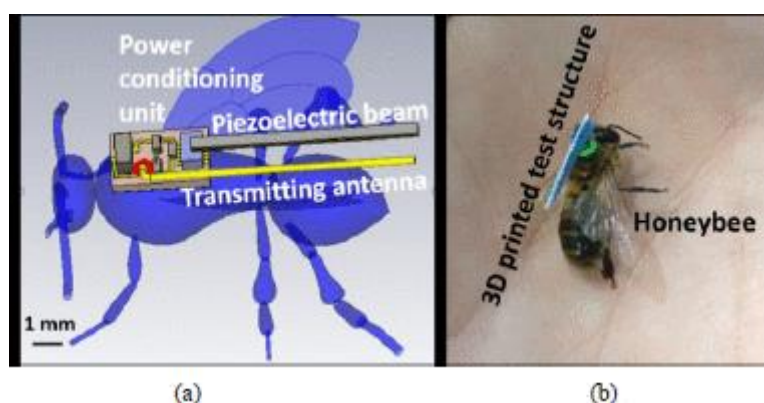


Figure 1 (a) Transmitter tag component breakdown. (b) Fabricated 3D printed test structure to evaluate flight performance under various tag weights.

Table 1 Comparative assessment of various bees' ability to fly depending on tag weight

Bee	With 80mg tag	With 35mg tag
<i>Honeybee</i>	0%	90%
<i>Bumblebee</i>	90%	100%
<i>Carpenter Bee</i>	100%	100%

As a bee's wing-beat frequency varies depending on environmental variables [63], a non-resonant energy harvester design was pursued with direct excitation from the

insect's thoracic vibration. This eliminated the need for increased tip mass and allowed straightforward fabrication of the piezoelectric beam with a uniform section.

For tag miniaturization a thin (50 μm) and flexible FR4-substrate was chosen, which reduced weight and physical hindrance whilst enabling easier attachment of the tag to the bee's thorax (Figure 2). Bees were temporarily immobilized and placed in the sponge slot visible in Figure 2 (a) before the tag was attached using removable glue.

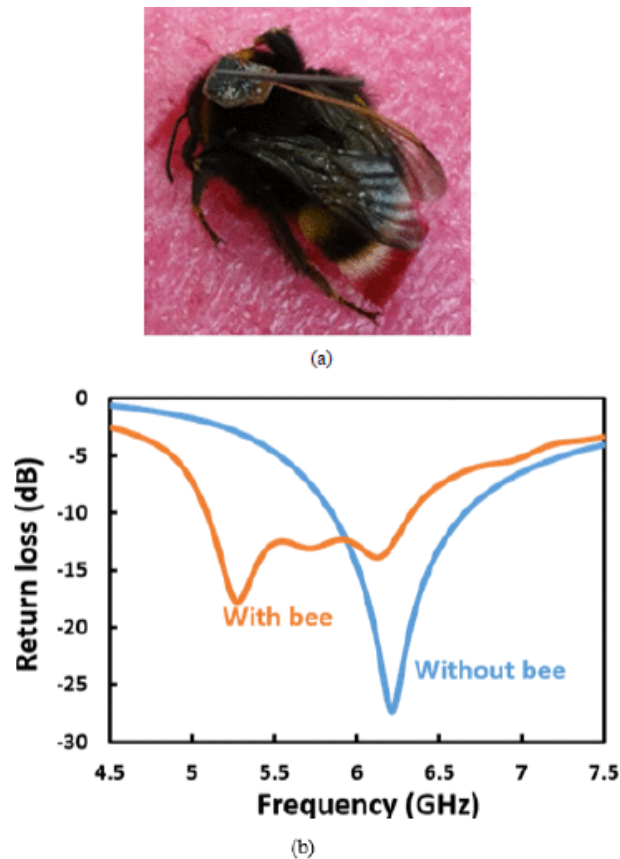


Figure 2 (a) Attachment of the self-powered transmitter tag to a bumblebee (b) simulated return loss for transmitting antenna with and without bee.

The use of a flexible material decreased circuit volume from 9.6mm² to 0.6mm², whilst reducing tag weight to ~30mg. Multi-physics optimization ensured the energy harvester provided 3.7 μW maximum power. Schottky diodes were employed for rectification of the harvester's output AC waveform since their low voltage operation yields the highest output voltage. Energy was then stored within a 1 μF ceramic capacitor, specifically chosen for higher energy output within short (~1s) transmission periods before being converted into an RF signal and radiated through the transmitting antenna.

The antenna was designed to be as small, lightweight and isotropic as possible in order to 1) reduce physical hindrance to the bee; 2) minimize transmitter circuitry complexity to cope with continuous direction changes during flight.

The possibility of integrating the antenna and energy harvester was explored for weight minimization but ruled out since the high dielectric constant of the energy harvester leads to low antenna efficiency and very small bandwidth. A simple monopole configuration with a thin wire was considered sufficiently lightweight and omnidirectional for the application. The simulated impedance match of the transmitting antenna is shown in Figure 2 (b). It can be observed that despite the proximity of the bee's body and the piezoelectric material, the transmitting antenna offers an acceptable return loss performance at 5.8 GHz.

3. Phased Array and Feeding Network

For AOA determination we designed and fabricated a multilayer microstrip antenna array with phase shifters integrated into the feeding network. Figure 3 shows the fabricated antenna, along with the measured and simulated results for the microstrip array impedance match. The slight shift in antenna resonance was attributable to fabrication tolerances.

To allow for continuous steering of the antenna radiation pattern, phase shifters were integrated into the feed for each element.

The phase shifter consists of a quadrature coupler loaded by controllable reactive loads (varactors) to provide a tuneable 360° phase shift [232].

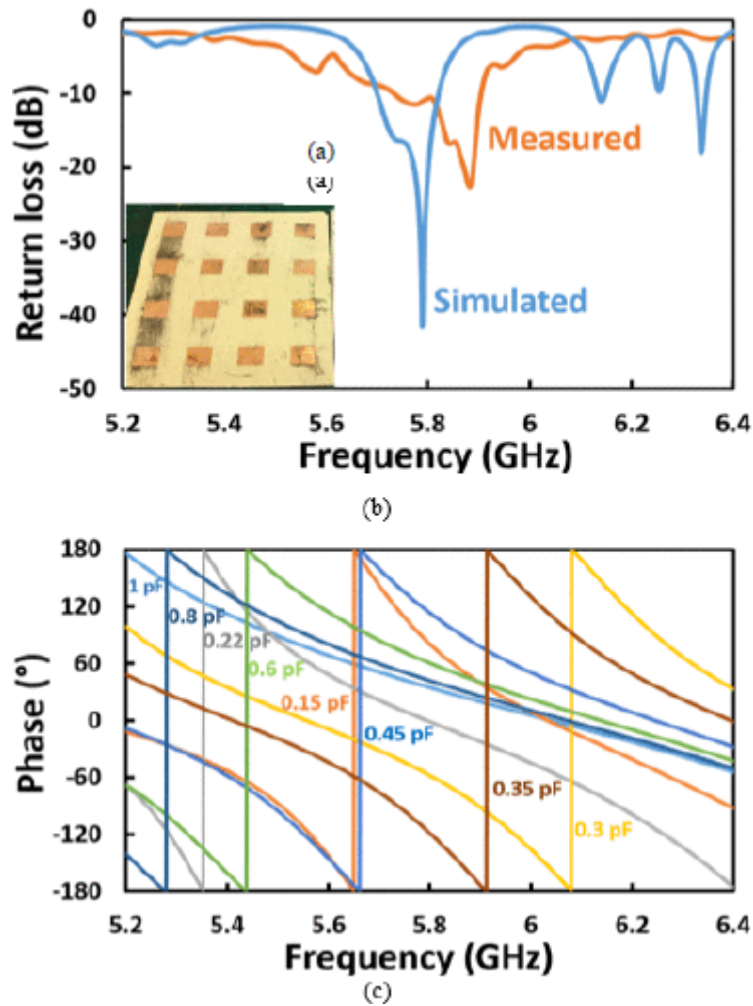


Figure 3 Fabricated phased array antenna (a), and (b) measured vs simulated return loss for microstrip antenna array. (c) Phase shift between input and output port as reactance of circuit is changed.

In order to maintain a compact structure, the feeding network substrate material was selected with a much higher dielectric constant (Rogers TMM 10i $\epsilon_r=9.8$) than the antenna substrate (Rogers TMM 4 $\epsilon_r=4.5$). The smaller feed network wavelength also allowed the antenna array to maintain appropriate spacing between elements ($D=0.5\lambda$), thus eliminating any scan blindness and grating lobes.

In order to achieve reactive load tuning diode varactors were selected based on their small size and the ease to solder. The change in active element impedance was evaluated for all scan angles of interest to ensure a return loss of < -10 dB for $\pm 50^\circ$ in azimuth and elevation.

The feeding network and simulated phase shift for the integrated phase shifters can be observed in Figure 3 (c) confirming the $\pm 50^\circ$ scanning range with load variation 0.15-1 pF.

4. Autonomous Tracking

a. Monitoring of Bumblebee Nest through a Stationary Receiver

A logarithmic detector was used to transduce the received signal power into an output voltage. The voltage was fed into a processing unit where the RSSI was recorded for each scan angle of the beam. With a receiver sensitivity of -70dBm corresponding to an output voltage of 2V, the receiver system was capable of operating within a 20m range. To eliminate false readings any AOA estimate with a $RSSI > 2V$, which corresponds to $P < -70dBm$, was deemed to be erroneous.

The antenna beam was scanned every 1° in the azimuth plane. Five readings were taken per scan angle, an example of which is reported in the inset of Figure 4, and the average RSSI reading was computed for AOA estimation.

In the preliminary tests, the system was mounted on a static support 10m from the nest, whilst tagged bees continuously left and returned to the nest (Figure 4). A typical flight from and to the hive allowed us to record the azimuth/elevation coordinates and record the flight duration (Table 2) along with the associated RSSI (not reported). This could be used for: indirect estimation of the foraging time; to verify if the recorded flight direction matched the known location of food supplies and assess number of trips a bee takes outside the hive each day. Tagged bumblebees were able to roam within the nest unhindered by the tag, in contrast with harmonic radar telemetry [63] which requires tag removal before the bee can re-enter the nest.

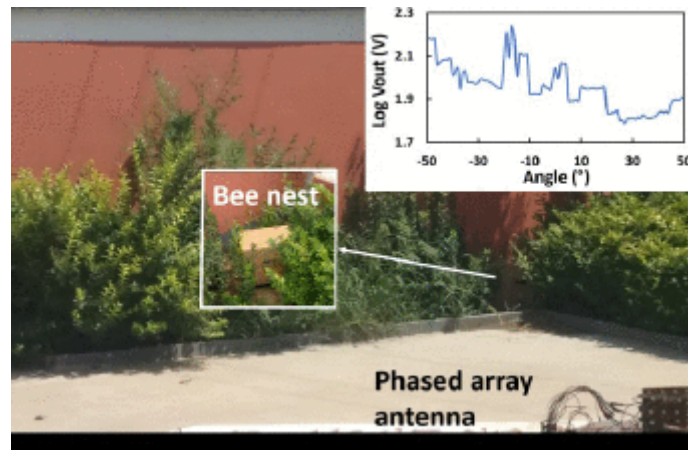


Figure 4 Experimental setup to monitor bumblebees entering and leaving the nest with 10m separation from the antenna array. Inset figure reports the received signal strength displayed as an output voltage vs azimuth scan angle for an estimated AOA of 30°.

Table 2 Recorded angular coordinates and time for tagged bee leaving and returning to hive.

<i>Take off</i>			<i>Landing</i>		
<i>Time</i>	<i>Azimuth</i>	<i>Elevation</i>	<i>Time</i>	<i>Azimuth</i>	<i>Elevation</i>
0s	5°	5°	1425s	35°	30°
3s	-12°	17°	1428s	12°	10°
6s	25°	19°	1431	-20°	22°
9s	N/A	N/A	1434s	-10°	5°
Total Flight Time				23.9 minutes	

By overcoming the cm-range limitation and required transponder-nest proximity in typical radio frequency identification (RFID) data loggers [54], the present system also demonstrated its potential for unobtrusive data logging. Remote monitoring of flower to flower bee-foraging was achieved in a greenhouse environment by accessing the static receiver via a LoRa board, which will be presented later. This could be additionally correlated to ambient hive (or greenhouse) parameters (temperature, humidity, and lighting) for autonomous monitoring of colony health [233].

b. Integration to UAV for Autonomous Dynamic Tracking

A compact housing unit was designed and fabricated using additive manufacturing to allow for the integration of the receiver unit onto a drone as seen in Figure 5. The inbuilt software development kit (SDK) allowed for connection of additional sensors and microcontrollers via serial universal asynchronous receiver/transmitter (UART) communication. This was utilised for sending all captured data from the AOA estimation to the drone's processing unit. The data was then stored into a frame, encrypted for easy identification of the start and end of a data sequence, and transmitted to a remote base station via the drone inbuilt communication system. The AOA estimation was determined from decoding the incoming data thus allowing determination of the bee's bearing and location relative to the drone. The bearing was transmitted back to the drone as a command to update its position. The RSSI was also condensed into a heat map, which displayed the estimated AOA in real time (Figure 5).

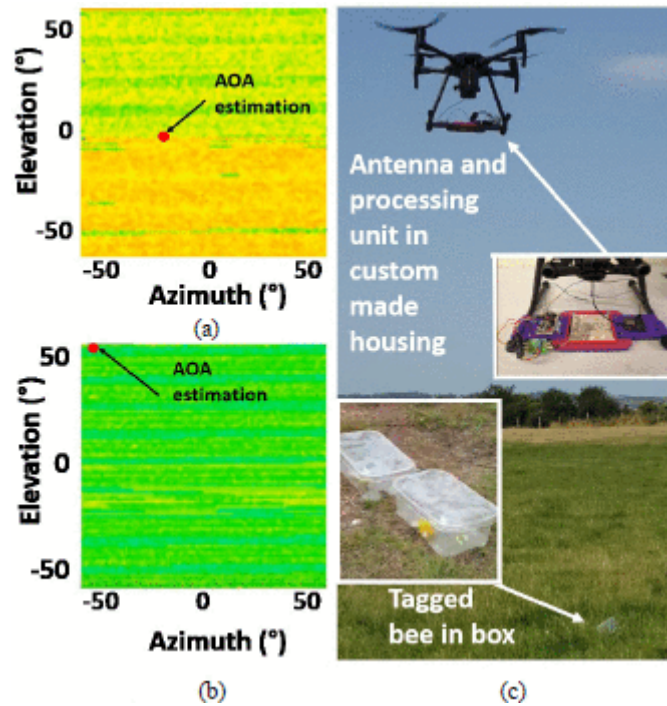


Figure 5 (a) And (b). Heat map obtained in real time outlining AOA estimates for the bee's position. (c) Antenna array, logarithmic detector and processing unit attached to drone approaching the target.

c. Performance of the Autonomous Tracking System

The autonomous tracking performance involved releasing tagged bees in a plastic, semitransparent container $25 \times 15 \times 10 \text{ cm}^3$. The drone hovered at 10m altitude at several locations with horizontal projection $\lesssim 20\text{m}$ from the tagged bee. When the bee flapped its wings inside the box the system was able to locate the target and autonomously move towards the location of the tagged bee. The drone autonomously updated its bearing and the internal log readings were subsequently plotted for two representative experiments (Figure 6 (a)). Further tests were conducted to demonstrate the long-range autonomous tracking potential of unconstrained targets. Tracking of freely moving tags (borne by an assisting student) were achieved over a 50 m range with battery-powered and handheld replicas of the tags. As shown by Figure 6 (b) the drone's flight path followed the tag emitter and recorded both linear and a zig zag trajectories with an acceptable margin of error ($+1-5^\circ$). The margin of error can be explained by the drone being programmed to

simultaneously update its roll, pitch and yaw, generating a deviation between the target and drone's path.

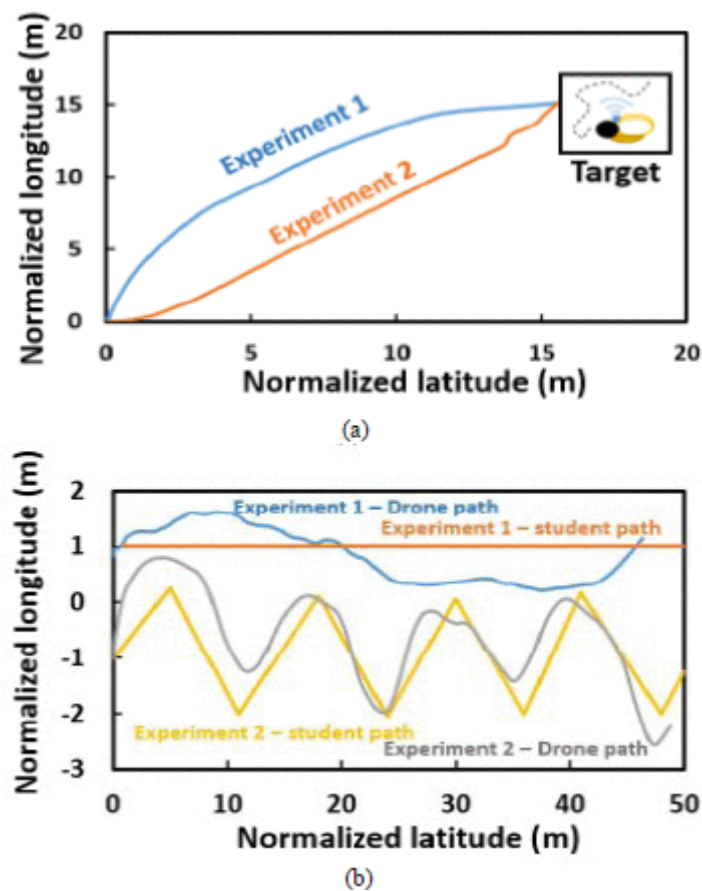


Figure 6 Update in drone's position over time while tracking (a) tagged bee flying in a plastic container. (b) Student walking a different paths with a handheld transmitter.

Whilst the preliminary tests were limited to a 50m range and up to 25% of typical bee flight speed, they suggest that a substantially increased range could be achieved. They were also instrumental in implementing additional capabilities such as adaptive speed and roll in control into later versions of the tracking code.

5. Discussion

Besides the moderate tracking error observed within each experiment (Figure 6), the tracking performance was observed to vary over repetitions of the same experiment

due to AOA estimation errors. However, the target was successfully tracked even when the earliest versions of the tracking code resulted in processing and response times as high as 2s. The current self-piloting code ensures response times of 1s whilst adding adaptive speed and path smoothing control. Therefore, the soon to be deployed enhanced system is predicted to enable more efficient tracking of unconstrained and nimble bee motion.

As the system relies upon harvesting energy from the insect's thoracic vibration during flight, localization is only possible when the targets are actively moving, both in the static and dynamic receiver scenarios. For dynamic tracking, the drone can be programmed to store, or hover above, the last recorded motion location.

6. Conclusion

A novel solution is herein presented to empower telemetry studies on a variety of bee species. The system consists of a battery-less transmitter, a phased array antenna operating at 5.8 GHz, and a self-piloting drone acting on the received RSSI signal while harvesting real time localization data. Experimental results have demonstrated the system capability to track near hive bee coordinates through a static receiver approach which can be used for foraging time estimation; autonomously converge towards flying bees in a container from up to 20m distance; autonomously track freely moving targets over a 50m range. Further work is ongoing to improve drone's response and test freely flying bees.

B - Integration of 5.8GHz Doppler Radar and Machine Learning for Automated Honeybee Hive Surveillance and Logging*

1. Introduction

The loss of biomass, which includes the decline of honeybee colonies, strongly affects global industry, especially agriculture [1]. Building systems able to assess colony activity in real time offers a path to monitor and assess colony health and potentially enact stress mitigation procedures.

Radio telemetry, harmonic radars, and RFID tags have been employed to track insects and study their behavior over their entire forage range [63, 203, 234]. Nevertheless, there has been growing concern that tagging insects may affect behavior negatively. Nectar and pollen load accounts for 35% and 20% of the body's weight respectively, and rarely reaches 80% [100]. In addition to the tag's weight, other parameters such as its balance, size, drag and even the glue used for attachment may affect an insects' take-off ability and overall behavior [63, 101].

Radars have been employed from frequency ranges of 5.8GHz to 24GHz for tag-less monitoring of honeybees [109, 110, 235, 236]. Incoming and outgoing bees at the hive entrance generate a Doppler frequency whose magnitude and duration can be correlated to insect motion and activity. However, to the authors knowledge there has been no assessment of honeybee (*Apis mellifera*) RCS at 5.8GHz. This work aims at 1) filling this gap considering the relevance in WLAN and 5G applications and 2) empowering 5.8GHz doppler radar with machine learning (ML) to facilitate honeybee hive unobtrusive and autonomous surveillance.

*Published as N. Aldabashi, S. Williams, A. Eltokhy, E. Palmer, P. Cross and C. Palego, "Integration of 5.8GHz Doppler Radar and Machine Learning for Automated Honeybee Hive Surveillance and Logging," 2021 IEEE MTT-S International Microwave Symposium (IMS), Atlanta, GA, USA, 2021, pp. 625-628, doi: 10.1109/IMS19712.2021.9574826.

2. 5.8 Ghz Doppler Radar

A block diagram of the 5.8GHz Doppler radar is shown in Figure 1 (a). The Doppler radar was built using commercial radio frequency (RF) modules. The radar operated with a transmitted power of 30 dBm, while comprising a transmitting (Tx) and receiving antenna (Rx) with gains of 12dBi and 14dBi respectively. The received signal was down-converted and amplified by an 80dB amplifier with variable gain. Radar output was sampled at 44.1 KHz in 16-bit wave format and fed to a laptop for processing using an external USB sound card. The choice of 5.8GHz was due to the wide availability of off-shelf components, which allows the future integration of the components into a single board. The experimental setup shown in Figure 1 (b) outlines the hive, the portable radar unit, a data acquisition laptop and a video camera to complement the captured radar signal with visual recording of the hive's activity.

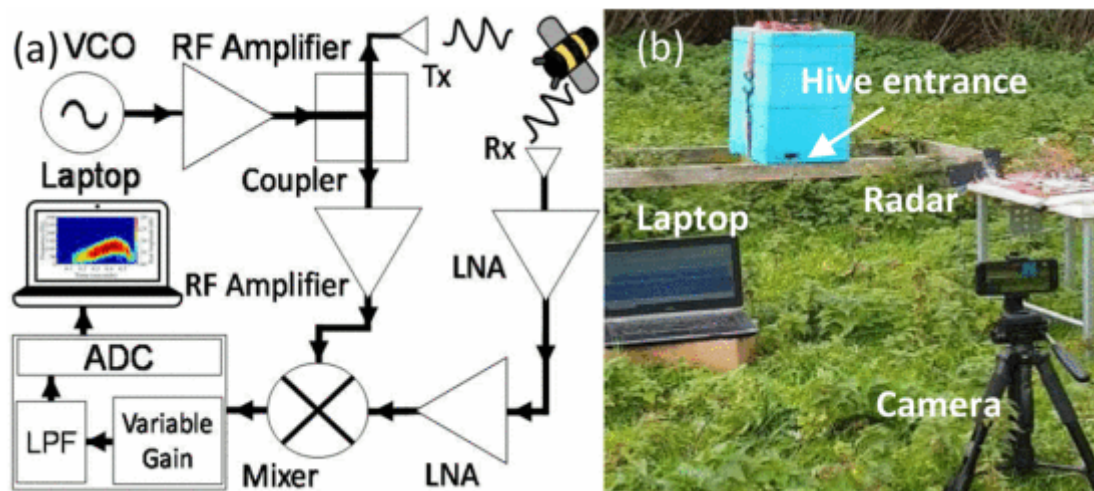


Figure 1 (a) Block diagram of the 5.8ghz doppler radar (b) experimental setup outlining the honeybee hive, the portable radar unit, a laptop for data acquisition and the video camera.

3. Honeybee RCS Approximation

Hajovsky reported honeybee RCS at 9.4GHz from -40 to -45 dBsm [235]. Wolf et al. [236] assessed the average honeybee RCS to be -65.9 dBsm and -61.5 dBsm for horizontal and vertical polarization, respectively, at 8-12GHz. Since the insects were anesthetized, the RCS measurements were not affected by honeybee movements and wing beats. Free flying honeybee RCS at 24GHz averaged at -50 dBsm for 11 bees in [110]. On the modeling side, insect RCS calculation is complex due to the involvement of multiple shapes, layers and dielectric constants. Since spheres are a common radar calibration standard, steel and water spheres were parametrically simulated in this study with increasing diameters at 5.8GHz. Additionally, an ellipsoid with a fixed height and width of 2mm and 3mm, respectively, was simulated, while changing the ellipsoid length to match different honeybee body sizes. Finally, a complete honeybee model was simulated to obtain its RCS value. The resulting RCS values are shown for different targets in Figure 2.

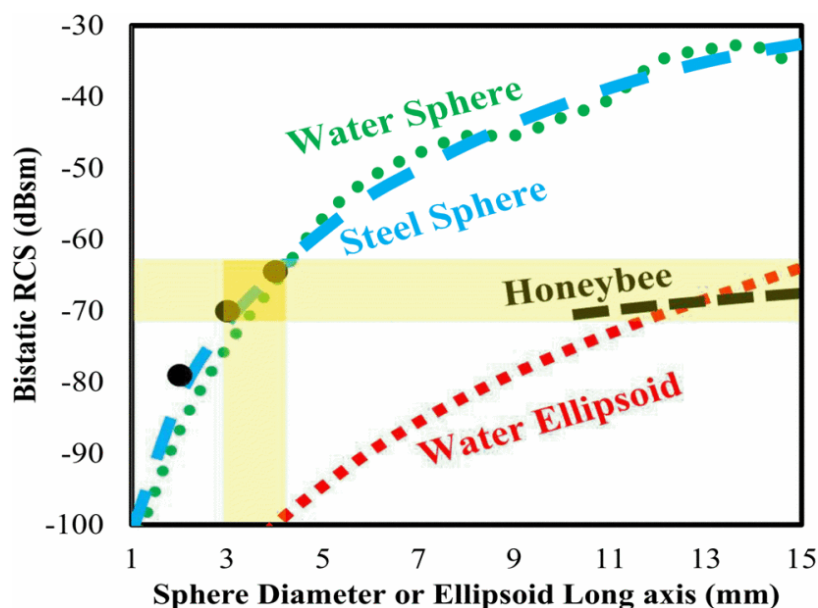


Figure 2 Simulated (dashed) and measured (symbols) RCS trends for different targets in the study.

The simulation results indicated a relatively close value of the water sphere and steel sphere RCS at 5.8GHz. The simulated RCS of the 4mm, 3mm and 2mm steel spheres were -64.6 dBsm, -71.6 dBsm and -82.1 dBsm respectively. Critically, the

RCS of the 3-4mm diameter steel spheres matched the 10-15mm ellipsoid's RCS (-75.8 to -63.9dBsm respectively) with a 0.7dBsm difference between the 4mm steel sphere and 15mm ellipsoid.

This suggests the validity of 4mm steel spheres as a model to approximate a honeybee's RCS. For further comparison, the simulated honeybee RCS are additionally shown in Figure 2 ranging between -70.7 and -67.5dBsm, for honeybee length of 10-15mm, respectively. This matches the ellipsoid and steel sphere RCS, hence further supporting the adoption of 4mm steel sphere for calibration.

4. Calibration Procedure

Radar calibration was performed by dropping a 4mm steel sphere vertically from a height of 2m and capturing the radar signature of free falling calibration standards under the gravitational force. Both Tx and Rx were set to point upwards to measure the entire free fall duration of the 4mm steel sphere, as illustrated in Figure 3 (c). Figure 3 (a) shows the 4mm steel sphere in free fall using the Short Time Fourier Transform (STFT). The speed of the 4mm steel sphere in free fall was found to be 6.26m/s corresponding to a calculated free fall time of 0.64 seconds. The peak Doppler frequency was 240Hz. This results in a maximum speed of 6.20m/s, where the total time of free fall corresponds to 0.64 seconds. The peak magnitude of the free falling 4mm steel sphere was normalized to the value obtained from the simulations of the 4mm steel sphere (-64.6dBsm). Consequently, 4mm (n=9), 3mm (n=3) and 2mm steel spheres (n=3) RCS values were measured to validate the RCS measurement and repeatability. The average values were -64.7, -70 and -79dBsm for 4mm, 3mm and 2mm steel spheres. These values were found to be in accordance with the simulated RCS values, and were additionally plotted in Figure 2 as symbols.

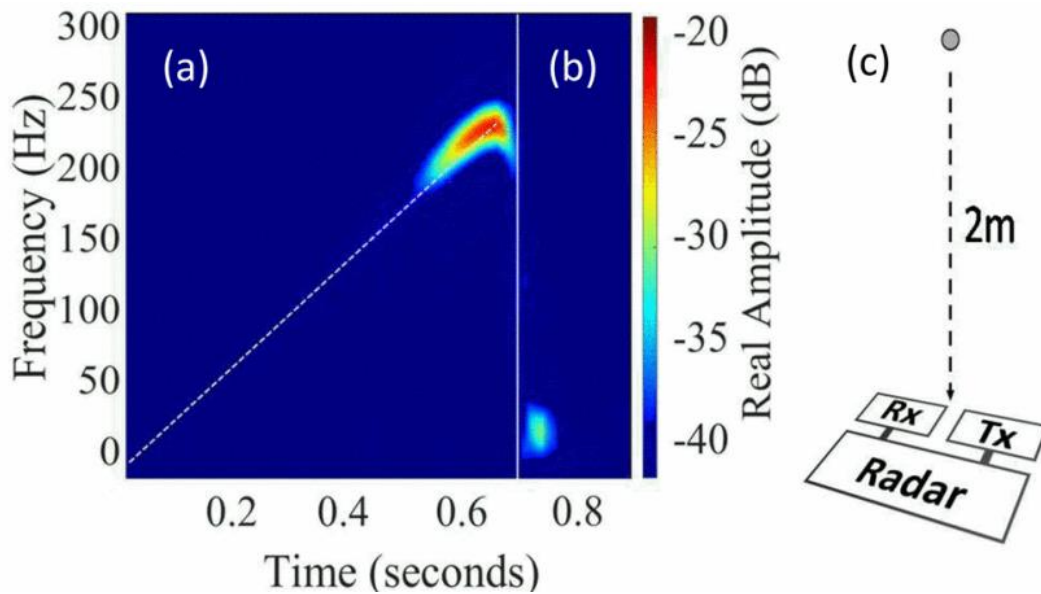


Figure 3 (a) 4mm steel sphere drop showing a comparison along gravitational acceleration. (b) Effect of steel sphere bounce after the initial drop.(c) setup for the free-falling sphere experiment.

5. Free Flying Honeybee RCS

The Doppler radar was placed at a 2m distance facing the entrance of the honeybee hive as visible in Figure 1 (b). This allowed monitoring bee behavior that occurred at the entrance of the hive. The video recordings permitted visual cross-validation of honeybee activity, which were later used to extract honeybee behavior as separate wave files.

Outgoing and incoming events were clearly discriminated through Doppler measurements. Bees leaving the hive rapidly flew out accelerating in an ascending trajectory, which was observed as an increasing Doppler shift. By contrast, bees returning resulted in a decreasing Doppler shift due to deceleration as they approached for landing. Figure 4 (a) represents a departing bee event with an increasing doppler frequency peaking at 170Hz, which indicates a flying speed of 15.81km/h. This bee was initially measured to fly at a speed of 1.1m/s as it accelerated to 4.39m/s (15.81km/h) with an acceleration of 10.2m/s^2 . The bee's acceleration approached the gravitational acceleration (9.8m/s^2) observed when dropping the 4mm sphere in Figure 3 (a) (dotted line). By contrast, Figure 4 (b)

shows a returning bee measurement, with a decreasing doppler frequency peaking at 96Hz, indicating an initial flying speed of 8.93km/h. This bee was observed to decelerate from its initial peak speed of 2.48m/s (8.93km/h) to a final 0.7m/s before entering the hive. The RCS values measured for both the departing and arriving bees (n=164) ranged between -55 to $-60\text{dBsm} \pm 3\text{dBsm}$, which is close to the simulated ellipsoid range and honeybee RCS.

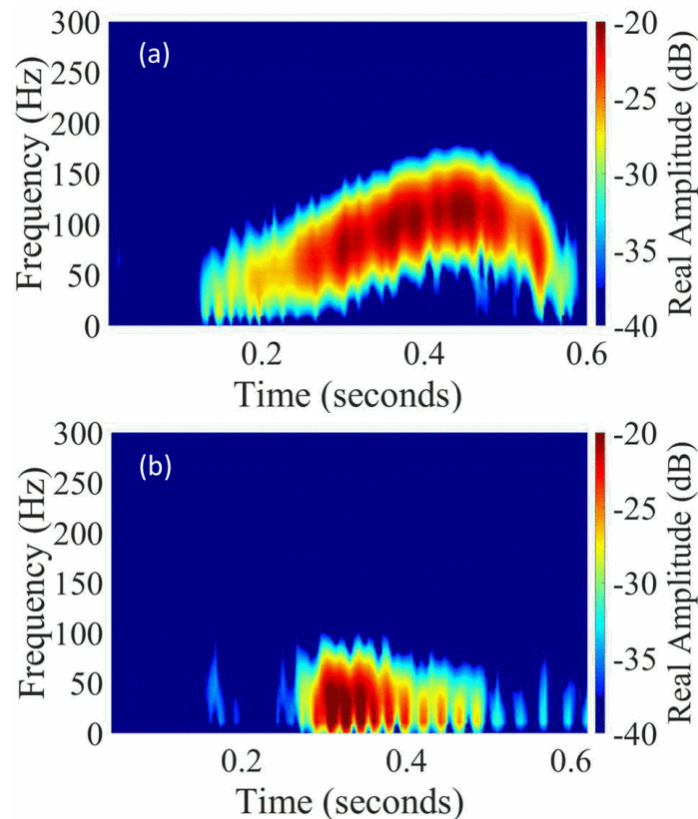


Figure 4 Radar signature for (a) honeybee departing from the hive in a straight ascending line. (b) Honeybee returning to the hive entrance in a more erratic pattern.

6. Doppler Radar Integration with Machine Learning

The radar system allowed gathering of consistent data that could be used for hive access logging and indirect estimation of foraging time. However, manual correlation of doppler signatures with video footage was onerous, prone to human error, and impractical for assessing bee activity in the field. Therefore, an automated approach based on artificial intelligence was investigated. Experiments were undertaken on a small set of labelled audio data files (n = 600). The goal was to identify the hive's

outgoing/incoming bees, using the signal intensity and signature as a means of identification. A window was used matching the size of the smallest audio file of approximately 0.4s to splice the data further into 1250 pieces. A side effect of this process was the generation of samples containing only background noise, therefore a filter was produced to remove these samples. Additionally, the data was normalized based on the maximum signal intensity across files. Image prediction based on a neural network containing both pretrained and naïve elements was undertaken using spectrograms of the audio, with the aim to predict between incoming and outgoing bees [171]. This technique of predicting over visual representations of audio signatures is becoming more common, due to the substantial improvements in recent image prediction neural networks. Prebuilt elements of the well-known MobileNet V2 architecture were not further trained, as this was detrimental to results [217]. This can be explained by differences in the data used and the real-world object data that such networks are trained upon. More typical audio-based machine learning techniques were also explored. The most efficient approach was to use random forest, processing the data as Line Spectral Pairs (LSPs) [152]. The data was split to a 4:1:1 ratio of training, validation, and test data in both cases. The neural network achieved a maximum accuracy of 87.8% with a Binary Cross Entropy loss of 0.43 with the learning curve demonstrated in Figure 5, where an epoch is one full round of training over the entire dataset. The random forest approach achieved 81.4% with a loss of 0.45. Losses in these examples represent a penalty associated to the degree at which the algorithm was incorrect, with the target being zero. The most significant barrier to refining the results were the signals containing overlapping behaviours as these were labelled according to the strongest element.

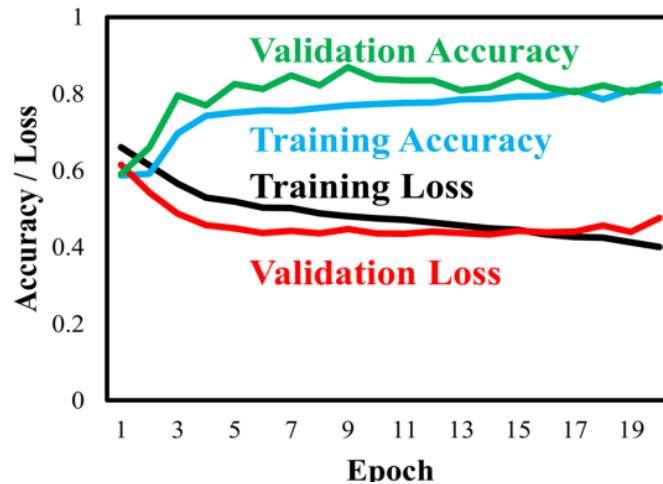


Figure 5 Learning curve of the fitted model showing a consistent accuracy limit and eventual overfitting being at epoch 16.

7. Discussion

The average RCS measurements of the departing and arriving bees was -59.06 and -59.5 dBsm respectively. Variations in the bee's body size, wingbeats and their aspect angle with respect to the doppler radar might affect measured RCS values [235]. Additionally, factoring in the bee's complex dielectric constant, it is expected that the measured RCS deviates from the simulated RCS. Radar integration with ML techniques can enhance the probability of correctly identifying outgoing/incoming bees from a hive, automate unobtrusive logging, and potentially allow monitoring of more complex behaviors.

8. Conclusion

Free flying honeybee RCS was measured using a 5.8GHz doppler radar. Simulation results supported using 4mm steel spheres calibration targets, which were measured to confirm accurate RCS measurements, and further verified with 3mm and 2mm steel spheres measurements. The RCS measurements of free flying bees were found to be in the range of -55 to -60 dBsm \pm 3dBsm. The integration of the radar with ML techniques allowed interpretation of bees leaving/entering the hive with a maximum accuracy of 87.8%. These early results show high promise for future developments. Further work is undertaken to miniaturize the design and to develop future classification algorithms.

C - A Machine Learning Integrated 5.8-GHz Continuous-Wave Radar for Honeybee Monitoring and Behavior Classification*

1. Introduction

The decrease of insect biomass has impacted global industries and agriculture [3, 10] as insects are relied upon for pollination, nutrients cycling, and derivatory functions [7, 237–239]. The significant annual contribution of insects to ecosystem services is estimated to range between U.S. \$235 and U.S. \$557 billion in value [240]. Among such insects are bees (*Apidae*)—known as the most important pollinating insects—which include the European honeybee (*Apis mellifera*) and the buff-tailed bumblebee (*Bombus terrestris*) [6]. Approximately, 80% of global pollination services are attributed to honeybees, making them a key component of the ecosystem [1, 24, 241].

Recent findings show that the combinational exposure of stressors inflicts detrimental effects on colony health, hive performance, and their population level [242]. For example, honeybee absconding, a significant cause of colony loss in pollinators, has been linked to intra-colony food pattern changes and foraging stresses [243]. The development of monitoring technologies for real-time assessment of insect activity is a key element in studies to further increase our understanding [86, 234, 244] and manage populations of *Apidae*.

Multiple sensors can be used for real-time probing of hive internal or in situ parameters, such as temperature, humidity, weight, and activity [112]. Radio telemetry, harmonic radars, and radio-frequency identification (RFID) tags have also enabled insect tracking and potentially support statistical analysis of their behavior over the entire forage range [63, 86, 203]. Nevertheless, there has been concern that tagging of *Apidae* for tracking purposes hinders natural movement and considerably affects insect response.

*Published as Aldabashi, N., Williams, S.M., Eltokhy, A., Palmer, E., Cross, P. and Palego, C. (2023). A Machine Learning Integrated 5.8-GHz Continuous-Wave Radar for Honeybee Monitoring and Behavior Classification. *IEEE Transactions on Microwave Theory and Techniques*, [online] pp.1–11. doi:<https://doi.org/10.1109/TMTT.2023.3248785>.

A generally adopted guideline prescribes that the tag's weight should not exceed the maximum nectar and pollen load which typically achieves 59% of a honeybee's body weight, but can reach 80% [245]. Thus, even placing tags as light as 30%–50% of bee bodyweight can potentially alter their take-off ability, foraging inclination, and overall performance [63, 100, 101].

Moreover, bee catching, optimum tag positioning, and attachment are inherently stressful and time-consuming processes in tag-based tracking approaches.

Tag-free continuous-wave (CW) radar systems have been designed, ranging from 5.8 to 24 GHz [109, 110, 215, 235, 236] and based on Doppler shift correlation with bee's speed and acceleration. However, such systems were relatively invasive in that the radar was placed at the hive's entrance.

A primary aim of this work is to develop a comparatively unobtrusive 5.8-GHz CW radar to monitor free-flying honeybees at a 2 m range from the hive and facing its entrance/exit. The 5.8-GHz-frequency band was chosen as a compromise between signal quality, monitoring range, and the wide availability of off-shelf components. This allowed expanding our original work [215] by integrating the CW radar into the single portable printed circuit board (PCB) in Figure 1 (b). The final aim of this work is create an autonomous machine learning (ML)-based radar for long-term data collection, supporting continuous hive health monitoring.

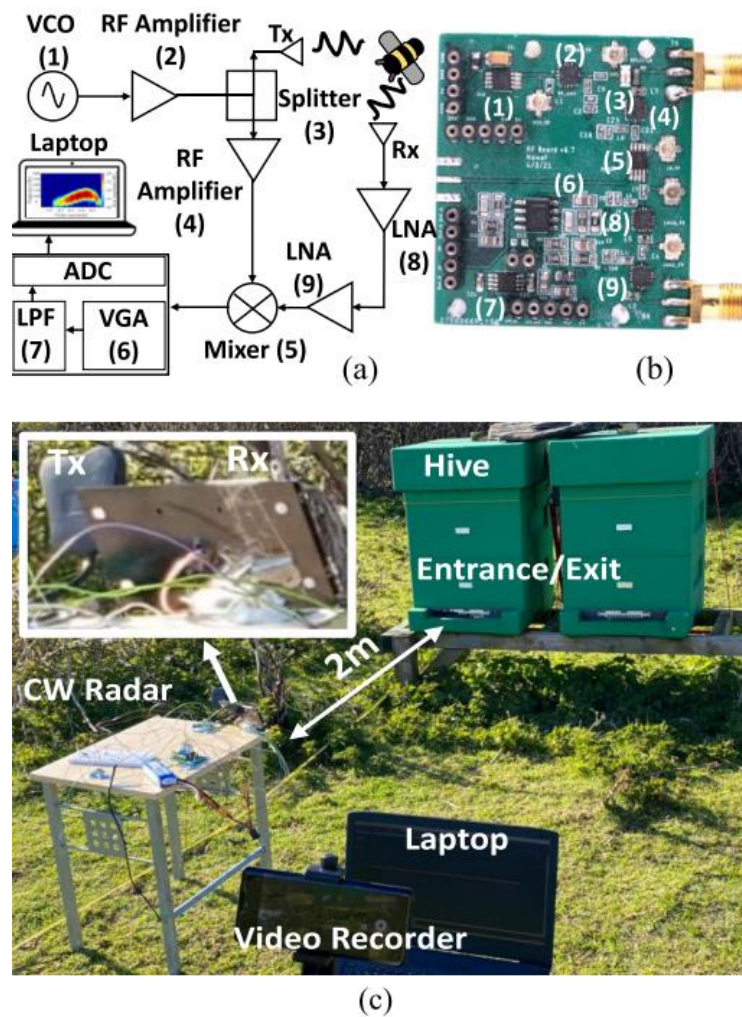


Figure 1 (a) Block diagram of the 5.8-GHz CW radar and (b) PCB implementation. (c) Setup for the hive monitoring experiment.

2. 5.8-GHz CW Radar Design

The setup of the 5.8-GHz CW radar is shown in Figure 1 (c). An important requirement that dictates the design of the CW radar is the radar cross section (RCS) of the target. The RCS of honeybees has been experimentally investigated at different frequency bands between 5.8 and 24 GHz [109, 110, 215, 235, 236]. At 9.4 GHz, the RCS of honeybees ranged from -40 to -45 dBsm [235], whereas at 24 GHz, free-flying honeybees RCS averaged at -50 dBsm for 11 bees [110]. Wolf et al. [236] mentioned honeybee RCS at 8–12 GHz to average between -65.9 and -61.5 dBsm for horizontal and vertical polarization, respectively. Finally, at 5.8 GHz, the RCS of honeybees averaged between -55 and -60 dBsm [215].

The 5.8-GHz CW radar was designed with such parameters in mind, while factoring in a 2–3 m range from the hive. It was based upon inexpensive commercial radio frequency (RF) modules powered by a LiPo battery with switch mode power supplies (SMPS) for voltage regulation. This increased monitoring flexibility and allowed identification of a wider range of behaviors (e.g., hovering, flight patterns, and missing the entrance) than mere entry/exit.

The IF signal of the double-balanced mixer (DBM) was amplified by a 60-dB custom-built variable gain amplifier (VGA) with 100-dB common-mode rejection ratio (CMRR) and integrated a low pass filter covering our frequency range of interest up to 408 Hz. The VGA’s output was recorded in 16-bit wave formats and fed to a laptop using an external USB sound card sampled at 44.1 kHz. The recorded wave files were processed on a laptop equipped with MATLAB. The 5.8-GHz CW radar parameters are summarized in Table 1.

Table 1 5.8-GHz Radar Parameters

<i>Parameter</i>	<i>Parameter Value</i>
<i>Frequency</i>	5.8 GHz
<i>Peak transmitted power</i>	28 dBm
<i>Radar range</i>	2 meter
<i>Transmitter gain</i>	12 dBi
<i>Transmitter beamwidth</i>	42°H and V
<i>Receiver gain</i>	17 dBi
<i>Receiver beamwidth</i>	32°H and 76°V
<i>RF Receiver gain</i>	38.40 dB
<i>Receiver noise figure</i>	2.22 dB

Circuit optimization enabled integrating the functional system in [215] into the 45 × 40 mm four-layer PCB in Figure 1 (b) with an approximate cost of U.S. \$75 and the same specifications as the original system.

While millimeter-wave modules afford superior radar resolution, the sensitivity-cost trade-off achievable at 5.8 GHz is critical in greenhouse scenarios requiring multiple receiver deployment.

3. Predicted Versus Measured Radar Signatures

A. Doppler Signatures: DBM Receiver

Although bee flight may involve composite/heterogeneous motion patterns, identification of uniform motion segments was a valuable abstraction to guide further analysis. Simplified Doppler shift models were first analytically derived by considering: ideal point scatterers; elementary motion segments with either uniform speed and uniform acceleration or sinusoidally varying acceleration (pendulum motion); and representative bee range, speed, and acceleration as extracted from previous experiments [215].

MATLAB was then used for accurate spectrogram calculation accounting for: target range and angular deviation; spherical targets that matched the honeybee's extracted RCS at 5.8 GHz; and power spectral density variation from path loss and TX/RX antenna directivity. The availability of measurement data for either real honeybees or spherical calibration targets resulted in the spectrogram pairs visible in Figure 2 (a) and (b) through (e) and (f) for each elementary motion segment. Notice that every modeled motion modulates the backscattered signal phase as a function of the motion basic parameters and transmitted signal wavelength λ . Since the models presented in Eq. 1–4 express Doppler frequency shift as a proxy for target radial speed $v_R(t)$, they effectively describe demodulation of frequency-modulated waveforms with $\beta = \frac{2}{\lambda}$ modulation index and $v_R(t)$ modulating signal. Thus, from CW radar theory, the expected Doppler shift signature for a target moving at uniform radial velocity v_R and a DBM receiver architecture is of the type

$$f_{d1}(t) = \frac{2v_R}{\lambda} = \frac{2f_0 v_r}{c} \quad \text{Eq. 1}$$

where λ is dictated by the transmitter signal frequency f_0 and the speed of light c , resulting in the predicted spectrogram in Figure 2 (a). The predicted Doppler signature for a uniform accelerated motion with radial acceleration a_R is

$$f_{d2}(t) = \frac{2a_R t}{\lambda} \quad \text{Eq. 2}$$

leading to theoretical and measured signatures in Figure 2 (c) and (d), respectively. The spectrograms in Figure 2 (b) and (d) represent rare examples of bee free flights at constant speed (acceleration) in front of the radar over appreciable time spans. The predicted Doppler shift for a simple pendulum is as follows, neglecting friction over small angle swings:

$$f_{d3}(t) = \frac{2l\dot{\theta}}{\lambda} \cos \theta \quad \text{Eq. 3}$$

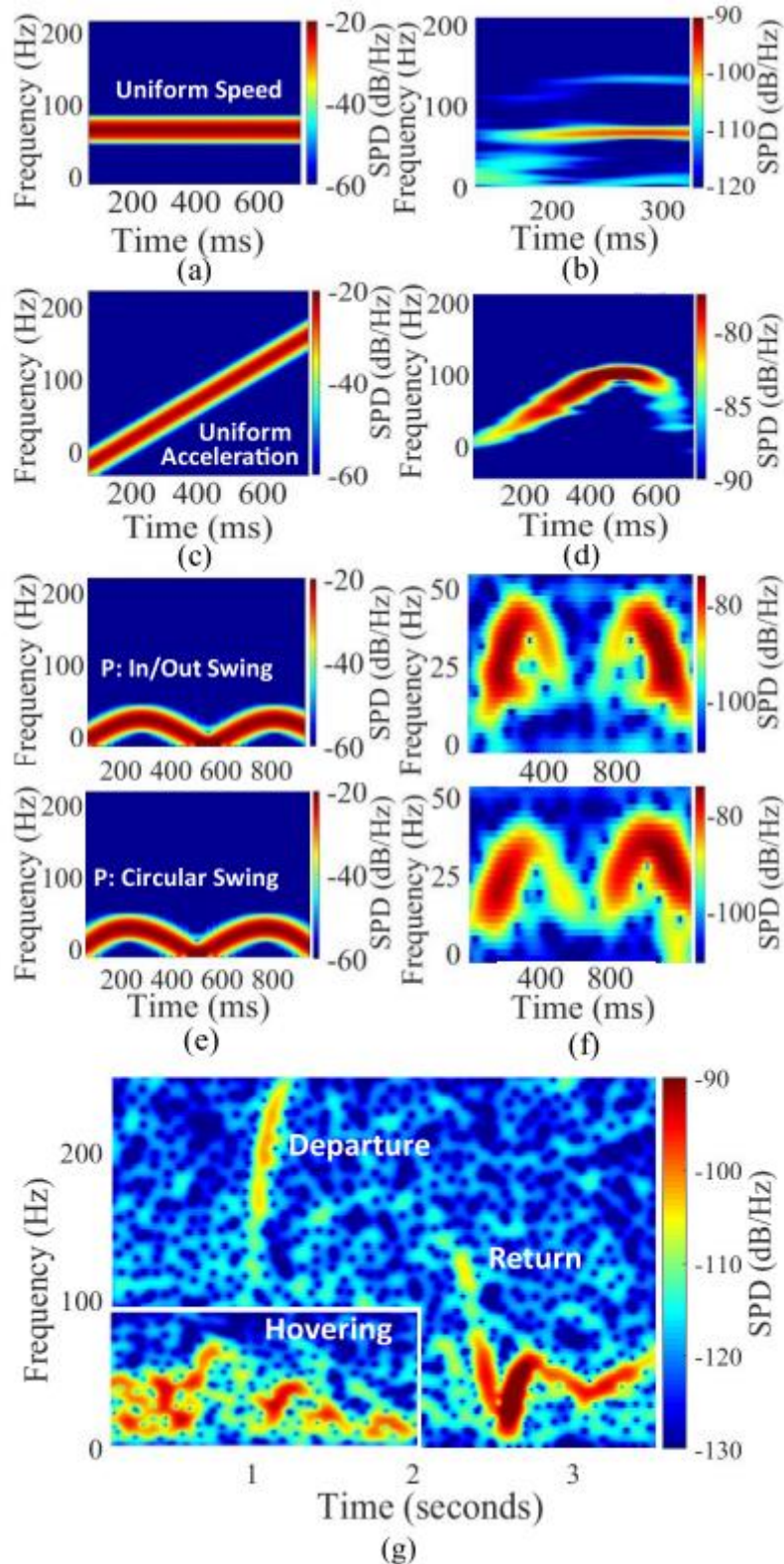


Figure 2 Doppler signatures: (a), (c), and (e) for target motion across elementary segments as predicted from CW radar theory + MATLAB models and (b), (d), and (f) for measured honeybee metal sphere target. (g) Measured near-hive bee signatures reproducing some of the modeled elementary motions.

where l is the pendulum length and θ is the swing angle coordinate. While not directly applied in bee flight analysis, such motion type was key to radar calibration and RCS extraction. It also supported the analysis of more articulate periodical motion patterns which were involved in bee hovering behavior. One such pattern is circular motion which was experimentally characterized by forcing pendulum circular swings at some distance from the observer. For a target rotating at angular speed ω_o over a circumference of radius r centered at distance R_0 and in the same plane as the radar, the predicted Doppler signal is

$$f_{d4}(t) = 2\omega_o r \frac{\sin\left(\theta + \tan^{-1}\frac{\sin\theta}{R_0 - \cos\theta}\right)}{\lambda} \quad \text{Eq. 4}$$

The resulting simulated and measured signatures also appear in Figure 2 (e) and (f), respectively, which outlines the resemblance with a simple pendulum motion.

Field experiments demonstrated the radar's capability to detect free-flying honeybees from a 2 m range and for motion segments comparable to the idealized models. Representative instances of flight detection are shown in Figure 2 (g). A leaving flight is shown to result in a red shift of 254 Hz corresponding to a speed of 6.46 m/s (23.25 km/h), which slightly deviated from the linear trend described by Eq. 1 due to saturation of the leaving speed. Similarly, a returning flight signature outlined blue shift from the initial 130-Hz value, corresponding to a speed of 3.35 m/s (12.09 km/h), to the final 60-Hz value, corresponding to a speed of 1.55 m/s (5.52 km/h). The returning bee signal also featured a higher power spectral density indicating closer proximity to the hive.

Notice that the initial linear deceleration trend was followed by bee speed oscillation before landing. The stationary, winding, or periodical aspect in similar segments was visually confirmed through the camera data and generally branded as "hovering." An even more prominent instance of hovering appears in the inset of Figure 2 (g), evoking a spiraling down circular motion signature. Hovering was frequently observed both during take-off and landing events, possibly in response to a busy hive entrance. Bees were also observed briefly flying out to instantly return to the

hive's entrance, resulting in net zero in/out logging. This spurred the hovering classification effort described later.

B. Micro-Doppler Signatures: IQ Receiver

Micro-Doppler analysis was additionally afforded to disentangle minor motion signatures, such as bee thorax vibration and wingbeat, from gross body translation. This enabled identification of different insect species (e.g., honeybee versus bumblebee), while potentially supporting disambiguation of more complex patterns with similar Doppler signals. In order to gain approaching and receding micromotion discrimination, a micro-Doppler module was also developed to include a quadrature receiver.

Micro-Doppler signatures of uniform speed flights were well-approximated through linear scatterer combination of: constant speed body translation as ruled by Eq. 1 and harmonic (spring) motion to additionally model wing flapping

$$f_{d5}(t) = \frac{2}{\lambda} [v_r + A_w \omega_w \cos \omega_w t] \quad \text{Eq. 5}$$

where A_w and ω_w are the wingbeat amplitude (~ 1 mm) and angular speed ($\sim 2\pi \cdot 200$ Hz), respectively. The ratio of honeybee wings to main body RCS used for micro-doppler signature prediction was of the order of 1:5, and both motions were radially directed in the approximate model. More accurate spectrogram calculation was obtained by modeling bee body as a 15-mm-long, 3-mm-wide ellipsoid and wing length of 9.7 mm while also allowing both positive and negative Doppler shifts for in-phase/quadrature (IQ) mixing response.

Calculations were specifically carried out for a bumblebee, flapping wings at lower frequency than typical honeybee (~ 175 versus 220 Hz [246, 247]) and approaching the IQ receiver at low body speed of $v_R = 0.05$ m/s. Overlapping to uniform speed translation in Eq. 1 of the harmonic motion in Eq. 5 was more conveniently analyzed

through phase modulation (PM) formalism. Hence, a quadrature-carrier description of the received micro-Doppler signal had the form

$$x_r(t) = A[\cos(\beta \cos \omega_w t) - \sin(\beta \cos \omega_w t)] \quad \text{Eq. 6}$$

where A is an RCS-dependent amplitude term and $\beta = A_w \omega_w \sim 2$ for $A_w \sim 1$ cm is the modulation index, thus indicating departure from a narrowband tone modulation ($\beta < 1$) scenario. Therefore, standard PM analysis [248] yielded

$$x_r(t) = A \sum_{\substack{n=-\infty \\ \text{Even}}}^{\infty} J_n(\beta) (-1)^{\frac{n}{2}} \cos(2\pi(f_B + nf_w)) \\ + - A \sum_{\substack{n=-\infty \\ \text{Odd}}}^{\infty} J_{|n|}(\beta) (-1)^{\frac{n-1}{2}} \sin(2\pi(f_B + nf_w)) \quad \text{Eq. 7}$$

where f_B is the frequency shift for the main body translation, and $J_n(\beta)$ is the coefficient for the Bessel function of the first kind. Eq. 7 only slightly deviates from standard PM formulas since Eq. 5 forces a $\beta \cos \omega_w t$, rather than the conventional $\beta \sin \omega_w t$ dependence, and is seen to result in spectral lines at $f_w \pm f_B$ in Figure 3 (a). Slow body speed also forces $f_B \ll f_w$ and $f_w \pm f_B \sim f_w$, unlike $f_B \gg f_w$ in narrowband tone modulation, which underpins the following experimental scenario and allows direct wingbeat frequency extraction.

For experimental validation, a bumblebee was placed in a transparent container allowing restrained flight range along with simultaneous video recording of both the insect motion and the radar output. Video segments were selected for comparison with the radar readouts where the insect was either dashing and rubbing its limbs against the box walls without flapping its wings or fluttering/flying through wing flapping. The corresponding radar recorded files were processed as a short-time Fourier transform (STFT) to extract micro-Doppler signatures with a window length of 256 samples out of the 44.1k samples and an overlap of 250. Figure 3 (b) represents the recorded micro-Doppler signature and clearly outlines the expected features from theoretical calculations.

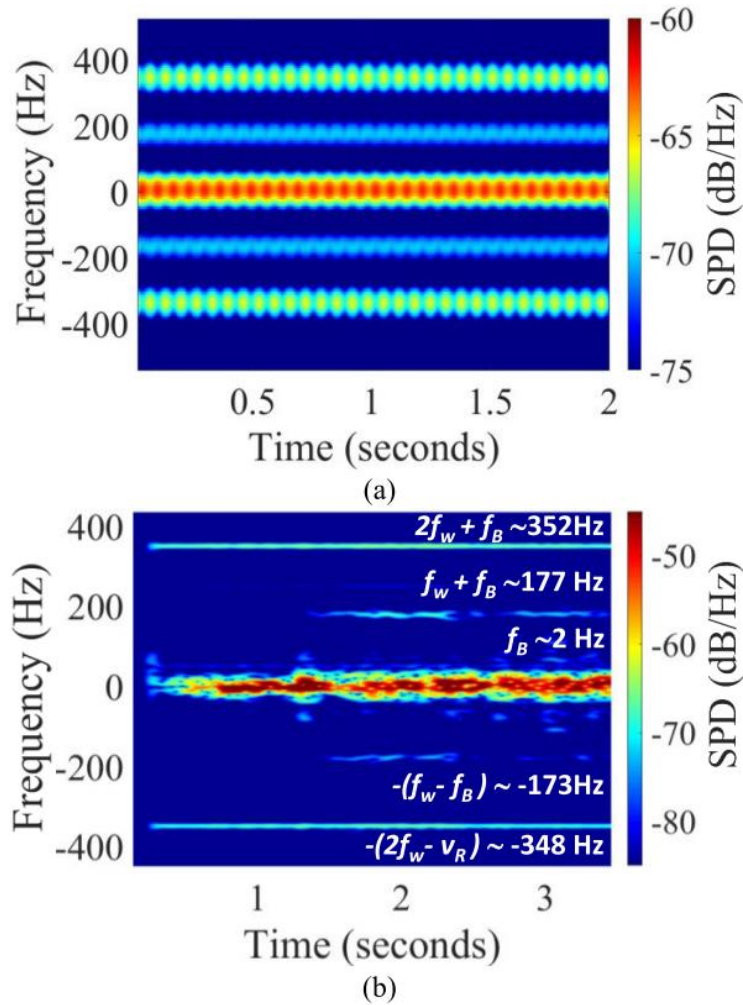


Figure 3 (a) Predicted micro-Doppler signal for bumblebee approaching the radar at 0.05-m/s constant speed and using IQ receivers. (b) Micro-Doppler signal recorded though IQ mixer for mostly fluttering bumblebee motion in a transparent container.

The spectral power density mainly concentrated around the expected 2-Hz component corresponding to body fluttering at 0.05 m/s. Although this was somewhat cluttered by coexistence of additional motion types (e.g., rubbing of limbs against box walls), fluttering was captured in the video footage and precisely timed by audio recording of “buzzing bursts.”

Weaker but clear horizontal bands at $\pm(nf_w \pm v_R)$ for $n = 1, 2$ confirmed coexistence and mixing of the translation and harmonic motions components, as from the model in Eq. 5–7. Furthermore, the extracted wingbeat frequency $f_w = 175$ Hz matched the typical bumblebee range [246].

This confirmed the approach applicability to classify insect species, or even micro-Doppler profiling of individuals within a species. However, due to the late adoption of the IQ mixing approach, the ML analysis in later sections could be only supported through double-balanced mixing.

C. Classification of More Complex Free-Flight Patterns

While the elementary motion segments described so far assist identification and logging of near-hive activities, free-flying bee patterns are markedly different because: uniform motion patterns represent a minority of all recorded motions; direction and motion type changes occur frequently and at unpredictable rate; individual bee flights and their radar image are heavily influenced by other insect flights within the detection cone; nonflying nontargets such as bees crawling (and still flapping wings, or buzzing) on the hive walls in the detection cone result in significant clutter; and in outdoor experiments, wind is also a source of noise forcing flight path alteration, hive and radar setup shaking, and electromagnetic (EM) background (e.g., grass, bushes, and branches) fluctuation.

The specificity of honeybee behavior unpredictability and interference from collective dynamics add to the general challenge in radar detection, for example, from warping of the received signal due to the antenna radiation pattern. As such, extraction of microwave features, from a traditional signal processing standpoint, is a challenging task in most outdoor bee detection scenarios. This explains why the ML algorithms described in later sections focused on behavioral classifications more than, or independently of, direct microwave features extraction.

Building an ML model to classify simple and separable away/toward and circular motion segments results in accuracy in excess of 95%. The availability of analytical models to tune upon enables the extraction of Doppler radar features such as effective speed, and periodical features with accuracy between 85% (when circular motion is included) and 97% (without circular motion). However, the key point is that ML is largely unnecessary for such “well-behaved” flight instances where direct spectrogram readout allows efficient feature extraction.

Pursuing too strict a comparison between traditional signal processing and ML-extracted features might be misleading and overlook potential opportunities in new approaches. ML has the capacity to enable correct behavioral classification even when microwave features extraction, whether through traditional processing or ML, is impossible. This is largely due to the fact that confirmation from video footage and expert beekeeper insight is used for algorithm training even for instances where distinct motion types produce virtually identical spectrograms (e.g., radial versus circular oscillations).

Assessing where ML-feature extraction breaks down due to overlapping of multiple signals or sheer clutter is a compelling research question. However, the topic appears conveniently explorable in a more forgiving scenario than outdoor free-flying bee detection.

4. RCS and Range Increase Using Silver Coating

Additional investigations were directed at increasing the honeybee's RCS as a means to enhance detection range without drastic hardware changes. While higher RCS could be achieved at 10.5 or 24 GHz [110], frequency increase was ruled out to maintain the low cost and commercial availability of the present components, along with coherence with earlier pollinators telemetry systems [86, 215, 234].

The RCS for spherical targets of different material was first simulated as a function of the target diameter using CST Microwave Studio. The simulated RCSs are shown in Figure 4 (a) for steel spheres and wooden spheres. The simulated RCS for a full-wave model of a honeybee is reported in Figure 4 (a). The honeybee's model length was varied between 10 and 15 mm. Interestingly, the bee model matched both the amplitude and slope of the 4-mm steel sphere RCS versus size curve. The 4-mm water sphere dispersive model (DC $\epsilon' = 78$ and $\sigma = 1.59$ S/m) in Figure 4 (a) was also extrapolated to match the RCS calculation at 10.5 and 24 GHz in [110].

Although the simulated RCS of the 4-mm wood sphere was -83.9 dBsm, which is below the radar detection threshold (-80 dBsm), coating the wood sphere with a

~100- μm silver layer ($\sigma = 6.30 \times 10^7 \text{ S/m}$) was predicted to increase the RCS by an average of 11.7 dBsm and to a detectable -72.06 dBsm .

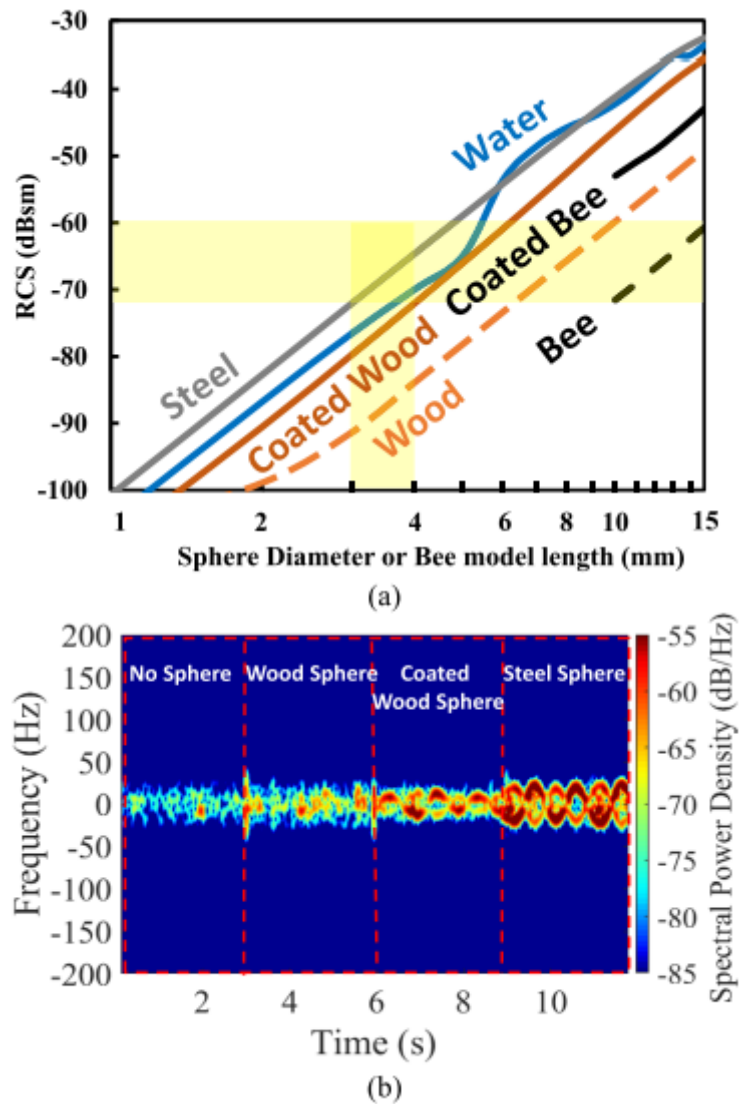


Figure 4 (a) Simulated RCS values of steel, water, wooded spheres, and a bee. (b) Signal amplitude increasing as pendulum target increases in RCS. The four wave files were combined to demonstrate the increase in amplitude.

The ~100- μm coating was obtained through a small brush stroke and validated by profilometer measurements to guide later application of silver nanoparticle layers.

The simulation results in Figure 4 (a) also show that while the coated wood sphere achieved higher RCS compared to the uncoated wood sphere, it featured a decisively lower RCS than the 4-mm steel or water sphere.

Experimental validation of the RCS increase through silver coating was achieved through IQ radar detection of target oscillations in a pendulum setup [215]. Targets were suspended through a non-metallic support and a thin 15-cm thread placed 0.5 m from the radar. Oscillations of 30° were allowed perpendicularly to the radar antenna's main beam axis and alternatively suspending: no target; a 4-mm wood sphere; a 4-mm coated wood sphere; and a 4-mm steel sphere. The results shown in Figure 4 (b) indicate that the uncoated wood sphere was poorly detected as its RCS was below the minimum detectable signal, resulting in a barely distinct signature from the unloaded wire case.

By contrast, a clear oscillating behavior was detected for the coated wood sphere. Finally, the steel sphere showed an even stronger signature in accordance with the higher RCS, confirming the simulated trend shown in Figure 4 (a).

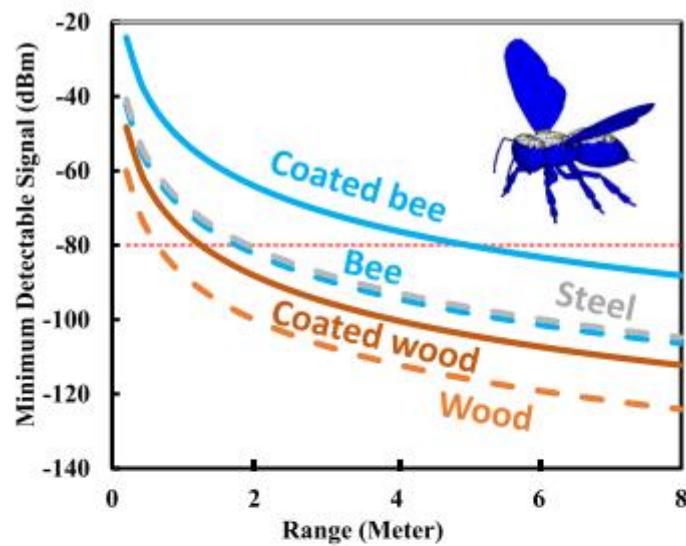
Silver coating was expected to enhance the bee detection range in accordance with the modeled RCS increase, also through no hardware modifications. The predicted radar detection range for uncoated and coated targets is reported in Figure 5 (a). This study informed a theoretical and experimental cost-benefit analysis of using coating to increase bee radar detection range. Notice the uncoated bee detection is within the approximation of the 4-mm steel sphere, which agrees with [110] and [215].

A coated bee model was developed as suggested in Figure 4 (a) and used to plot the curves visible in Figure 4 (a) and Figure 5 (a). Such model aimed at maximizing the bee's coated surface while not impairing its flight capacity nor obstructing the breathing ability, by avoiding coating of bee wings and spiracles.

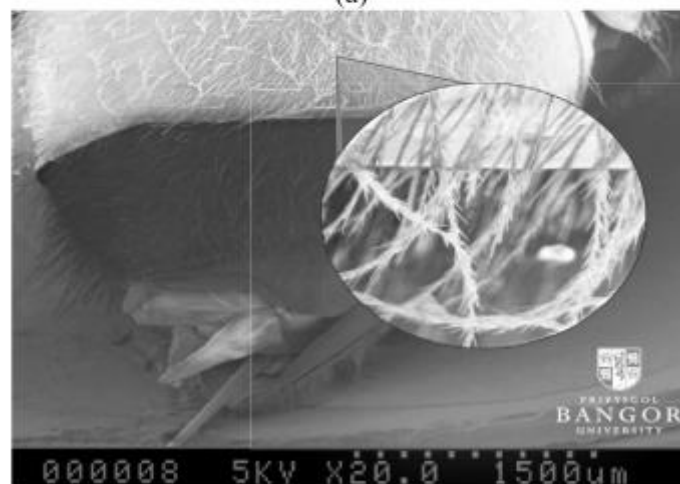
Feasibility of applying silver coating through a single small brush stroke was experimentally tested using (dead) specimens. The scanning electron microscope (SEM) photograph shown in Figure 5 (b) outlines a honeybee's layer of coated hair (seta) with length ranging from 300 to 900 μm . The bee's surface available for

coating favors distribution of silver coating in a closely packed arrangement that could approach the performance of the coating layers used for the RCS study. As shown in Figure 4 (a) and Figure 5 (a), the RCS and detection range improvement for a partially coated honeybee could achieve 18 dBsm and ~4 m, respectively. Such enhancement seems potentially interesting to dynamic tracking applications, drastically reducing tag load along with the challenges in higher frequency modules.

Nevertheless, for near-hive monitoring via stationary transceivers, the ~2-m range achieved while avoiding the invasiveness/inconvenience of the coating process might be preferable.



(a)



(b)

Figure 5 (a) Increased detection range of both coated wooden sphere and coated bee (displayed in the top right corner). (b) SEM image of honeybee thorax displaying the coated honeybee seta.

5. Machine Learning

The volume of data collected using the DBM was suitable for ML. Many data quantification techniques are available for audio signals derived from the RF Doppler signature, such as line spectral frequencies (LSFs), linear predictive codes (LPCs), and Mel-frequency Cepstral coefficients (MFCCs) [218–220]. These were chosen to specifically address the lack of directional information in the DBM data.

Simple temporal features of the signal were additional candidates for ML including mean amplitude, rms, zero-crossing rate, short-time energy, spectral centroid, kurtosis, skew, standard deviation, mean, variance, and energy. These were combined as the temporal data.

A. Binary Classification

Initial concern was the classification of the signals into two separate classes of bees entering the hive and bees leaving. This was not as simple as looking at the direction (toward/away from the radar) of flight as insects could move in a completely free environment irrespective of the radar. Our previous work looked at using spectrograms of the signatures as a means of classifying the radar signals, building from the MobileNet V2 architecture [217]. This achieved 88.7% accuracy but was a computationally expensive approach. In addition, it was shown that such neural networks (NNs) maladapted to this data due to their training on real-world objects rather than radar signatures. Spectrograms are a 2-D representation of a 1-D signal, whereas image-processing NN is designed primarily to classify 3-D objects represented in 2-D.

To retrain such a network requires a large quantity of data on par with MobileNet's original work. Audio (rather than spectrogram) processing was a feasible alternative as all the 5.8-GHz output radar signals contained relevant information at sub-1-kHz frequencies. Audio files were subdivided into 0.4-s segments, with each being vetted for a minimum signal.

The window of 0.4 s was deemed a suitable trade-off in view of the system application as a real-time monitor. This window matched our initial observation of the smallest complete event in the dataset. Subdividing this event further would risk failing to supply complete information to the ML. Larger windows did not offer sufficient improvements in accuracy to combat the additional cost in terms of samples. For rapid in-and-out signals, the window of 0.4 s emerged as a threshold between clear separation and overlapped, unusable samples. Some windows without sufficient information, such as tail ends of a signal (representing less than 5% of the length of the sample), were discarded. This created approximately 700 bee-in to 600 bee-out signals.

To generate more data, augmentation approaches were investigated. These included adding artificial noise, time shifting, and pitch shifting. The sensitive, RF nature of our data meant that noise and pitch alteration could not be enacted without compromising the ability for predictions, and thus, time-shifting was used.

As each signal was 0.4 s long, shifting could be affected at 0.1, 0.2, and 0.3 s, allowing for a fourfold volume increase, in addition to balancing the dataset. Support vector machine (SVM), random forest, and NN were learning algorithms to generate the augmented dataset [187, 188, 204]. A standard train-test split ratio of 4:1 was used, with the NN taking an additional 10% of the training data as a validation set.

Data quantized by the above methodologies were much smaller than spectrogram images and therefore did not need the raw predictive power associated with MobileNet's capabilities. A smaller sequential model was chosen, using two densely connected layers of 32 neurons each, activated by a scaled exponential linear unit (SELU) function. Although rectified linear unit (RELU) was first used as an activation function, we noted that SELU performed slightly better across all datasets. This difference was small (<2% accuracy) but significant enough to warrant the change. The final layer had a dropout rate of 0.5 to minimize risk of overfitting from the smaller than ideal number of samples.

The model itself was optimized using the Adam algorithm due to the number of parameters (~1400 in the case of the MFCC/bee-frequency Cepstral coefficient (BFCC) approach) [185]. Larger networks were also tried, both increasing the layer

and neuron count, but it was found that the network quickly reached a point of overfitting even when increasing the dropout rate significantly.

For comparison, the random forest and SVM used runtime hyperparameter tuning. For each data quantization method, the hyperparameters were chosen via Bayesian optimization, which was done on fourfold cross-validation with each cross-validator running five times. The random forest used values from the following possible hyperparameters:

1. estimator count between 100 and 1000, in increments of 100;
2. split criterion of either Gini impurities or entropy for information gain;
3. the number of features to consider for a split being either the square root or \log_2 of the total number of features.

Additionally, the SVM hyperparameters were from the following possibilities:

1. a regularization parameter between 1e-6 and 100;
2. kernel coefficient of 1e-6 to 100;
3. polynomial kernel function degree between 1 and 5;
4. a kernel choice of either linear, polynomial, radial basis function, or sigmoid.

Initial experiments showed that the MFCC and LSF approaches were the strongest predictors with both achieving approximately 85% accuracy. Mel-frequency in MFCC refers to the melodic scale used to attune audio to match how it is processed by the human ear. The simplest form of this is expressed as

$$F_m = C \log_{10} \left(1 + \frac{F}{D} \right) \quad \text{Eq. 8}$$

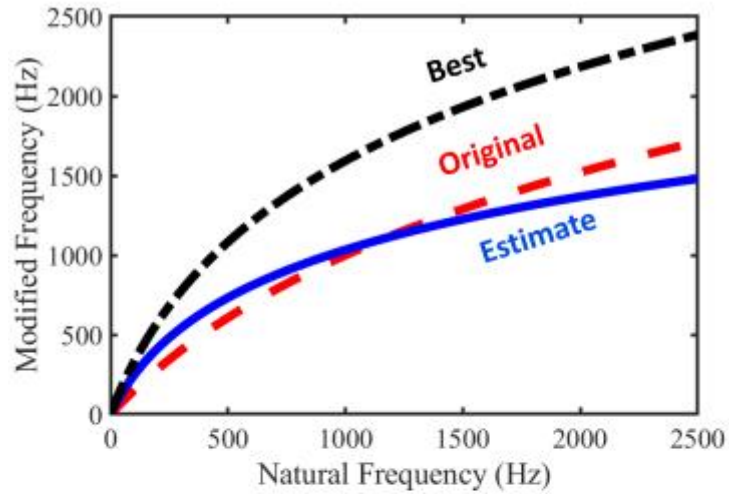
where F_m is the Mel-frequency of natural frequency F , the constant (C) is 2595, and the denominator (D) is 700 in the original equation.

Though this gave relatively reliable results, it has little to do with MFCCs original use of approximating human ear perceived frequency. Our data were from the radar and, while stored as audio, did not represent physical sound. The two primary parameters

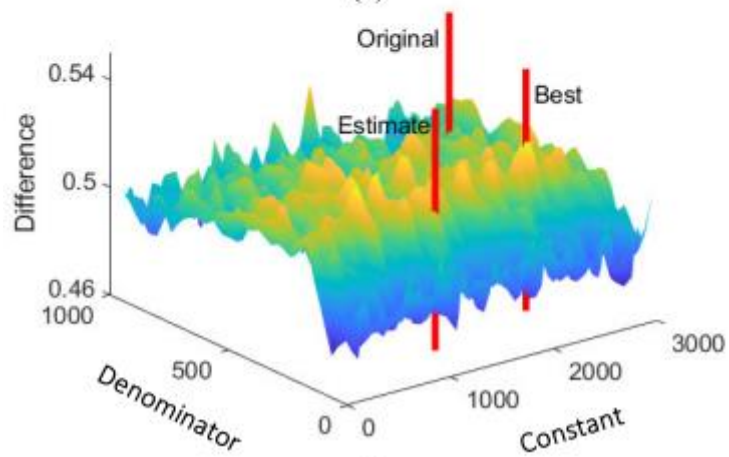
of the equation (the constant and the denominator) allowed for tuning and subsequent impact assessment on the final algorithm precision.

The parameters were changed to 1250 for C and 175 for D as a test point. These were chosen to maintain the new curve close to the original MFCC curve but emphasized frequencies below 500 Hz, where most activity was observed. The resulting estimate is demonstrated in Figure 6 (a) and resulted in a 2% accuracy improvement for the random forest model. At each point, the random forest was retrained and tested on the data five times, with each iteration having a bootstrapped randomized copy of the data. This ensured that only a mean accuracy was used, and the results would be replicable. The generated grid gave a clear indication of the optimal parameters to be used as demonstrated in Figure 6 (b), though it is noted that Figure 6 (b) has been smoothed via a Gaussian filter to allow improved parsing. The results suggest that the best parameters in our case were a constant (C) of 2325 and a denominator of 260 (D).

When these parameters were returned into the equation for testing, it showed an increased accuracy of 91.1%, an improvement even on the more computationally expensive spectrogram approach. The hyperparameters found to achieve these results were 900 estimators, entropy as a split criterion, and the square root of total feature count being used as a baseline when looking for the best split. A full breakdown of results including all algorithms used alongside each quantization method is presented in Table 2. The strength of this BFCC technique is that the original experiment can now serve as a testbed for when the dataset size is increased. Sample points can be taken from the original, computationally expensive, modeling and compared to later results to test for deviation, which would indicate that the constant and denominator need further refinement. This can be achieved without remodeling the parameters entirely.



(a)



(b)

Figure 6 (a) Natural versus modified frequency values as produced by altering the standard Mel-frequency algorithm, showing the original difference, the first estimate, and the eventual best version. (b) Plot of the two parameters of the MFCC algorithm (constant and denominator) and the effect that their change has on the accuracy of results. The z-axis shows the absolute difference between the accuracy and loss of the trained random forest.

Table 2 Accuracy and Loss breakdown for different learning algorithms and classification approaches.

<i>Algorithm</i>	<i>Approach</i>	<i>Binary Classification</i>		<i>Ternary Classification</i>	
		<i>Accuracy (%)</i>	<i>Loss</i>	<i>Accuracy (%)</i>	<i>Loss</i>
<i>Neural Network</i>	LPC	69.02	0.4787	66.54	0.5896
	LSF	85.48	0.3647	89.22	0.2933
	MFCC	79.01	0.4718	58.36	0.6636
	BFCC	79.77	0.4528	58.18	0.6498
	Temporal	72.05	0.5602	52.23	0.6293
<i>Random Forest</i>	LPC	73.42	0.5979	65.43	0.9348
	LSF	84.66	0.4634	88.85	0.7117
	MFCC	85.23	0.4255	63.20	0.9091
	BFCC	91.13	0.3693	63.75	0.8943
	Temporal	70.68	0.5750	65.80	0.8645
<i>Support Vector Machine</i>	LPC	55.61	0.6590	65.80	0.9149
	LSF	67.67	0.5066	93.37	0.2667
	MFCC	63.56	0.5566	58.36	0.9126
	BFCC	71.23	0.5145	58.29	0.8930
	Temporal	64.38	0.5806	52.42	0.9927

B. Ternary Classification

Adding hovering to entry and exit movement was deemed a critical improvement to the system. Hovering is defined as all behavior where the bee might fly close to the entrance of the hive but make no attempt to enter or leave the area. The bee might move closer, or further away, from the radar. It might also move in and out of the detection cone rapidly or stay in view for prolonged periods. In essence, the bee moves freely causing signals that resemble those of bees entering and leaving as in Figure 7 (a). Classifying this behavior is valuable in both a commercial and research

setting, first by removing the potential for these hovering flights to be falsely classified as entry and exit. In addition, it may prove that standalone hovering flights, or hovering before leaving, can be attributed to bee orientation flights, which can be a good indicator of growth, measured by the rate of young bees first leaving the hive [221].

Following the previous procedure of windowing, 200 samples were recovered of strictly hovering behavior. It should be emphasized that we also revisited the original data and split any samples that contained hovering and either of the other types. To balance the dataset, samples of hovering were given additional time shifts of 0.15 and 0.25 s creating an approximate equilibrium between all three classes. With log loss being unsuitable for ternary predictions, and to find comparable loss values between ML models, hinge loss was chosen as a suitable replacement [226]. It was observed from the results that both MFCC and BFCC algorithm lost almost all predictive power, in contrast to the binary results. At best, these techniques were closely matched at 64% accuracy.

Full accuracy and loss values are provided in Table 2 for each learning algorithm alongside each quantization method.

However, LSFs improved significantly over binary results. An improvement was seen in SVM prediction, with an achieved accuracy of 93.4%. Not only is this the best result in ternary results, but it is also a significant improvement in prediction across all models trained.

The hyperparameters found for this accuracy were a regularization parameter of 13.9, a polynomial degree of 2, and a kernel coefficient of 25.2, using a polynomial kernel.

These results appear as an outlier for SVM, both from the binary (best result 71.2%) and ternary (second-best result 65.8%) classifications. The SVM was trained in 15 separate instances with separate permutations of test and train data. This 15-fold cross-validation returned an average accuracy of 91.02% with a loss of 0.3046. Even when moving 20% additional random data from the training set into the testing set, for a 3:2 ratio of train and test, the accuracy only dropped to 89.5% with a loss of

0.3774. The results are also supported by the NN that achieved 89.2% accuracy and 0.2933 loss, only trailing the SVM results slightly.

The random forest also achieved 88.9% accuracy, though with significantly higher loss. The final step for ML was to compare with the potential of the previous spectrogram approach. The spectrogram system was expanded to incorporate three classes with no other changes with respect to the previous work [215]. This three-class NN system achieved an accuracy of 75.5%, much below its original two-class success of 88.7%.

A full visual breakdown of the results across both works is demonstrated in Figure 7 (b). The spectrograms accuracy reduction for three-way classification is likely due to the limitations in image resolution and lack of visual difference between the three classes.

In theory, the availability of both positive and negative frequency shifts enabled by IQ data would likely increase spectrogram diversification and classification accuracy. However, this would come at the expense of increased computational costs associated with image processing NNs.

Finally, a higher resolution analog-to-digital converter (ADC) and a higher frequency radar providing a higher RCS for honeybees are expected to overcome the limitations in image resolution.

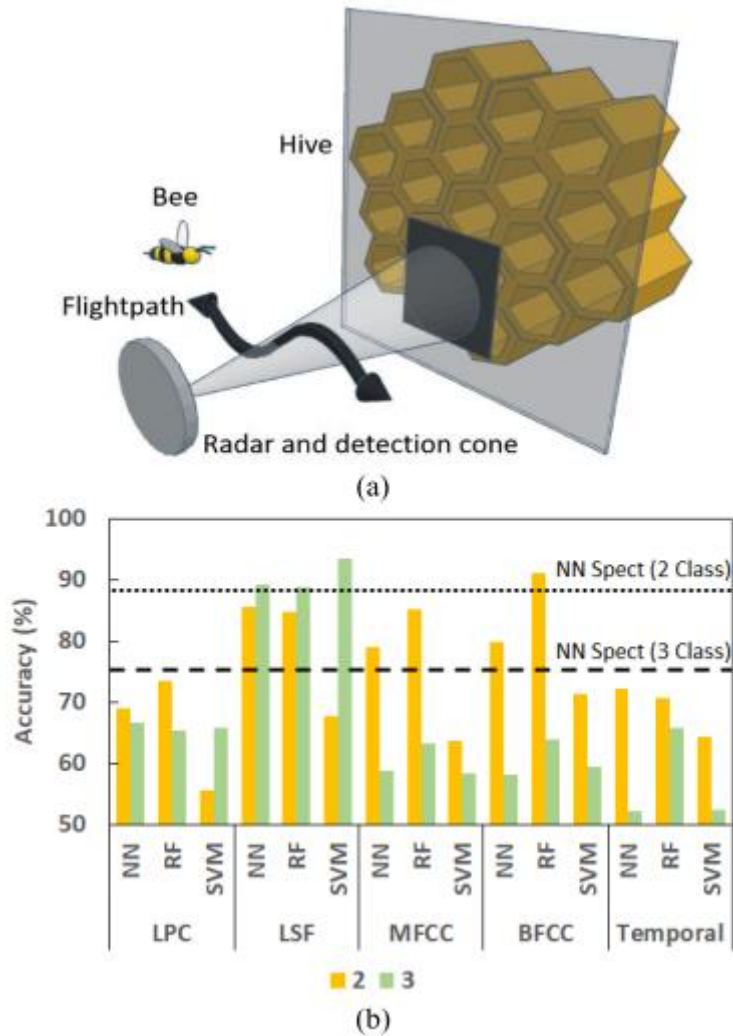


Figure 7 (a) Visualization of a bee hovering in front of a hive. The bee might come closer or move further away in addition to its other movements. (b) Visual plot of the progress from two to three classes across all included audio quantization methods, in addition to two benchmark lines for comparison to the NN, spectrogram method featured in the previous work.

C. ML Summary

Results from both binary and ternary (audio-based) classifications show that the dataset has potential to be used in ML applications. In particular, the processed Doppler data have the potential to make predictions more accurate enabling species identification and monitoring of traffic in a more diverse area, such as wild woodland.

A real-time implementation of either binary or ternary classification to monitor hive entrance activity overextended periods of time would make for an interesting research topic, especially correlating activity to several other metrics such as hive health, pollination success, and reproductive success. The present algorithms

reliance on audio-frequency rather than image/spectrogram processing is a key enabler for edge computing architectures. Similarly, the cost-capacity trade off achieved at 5.8 GHz benefits from the requirement for processing boards with lower data compression than in higher frequency systems.

Commercially, being able to monitor traffic would provide information on which hives are inactive and in need of replacement for maximum pollination efficacy.

The future availability of IQ data could potentially support a comparative study of our ML-based ternary classification approach to classical signal processing techniques such as hidden Markov model (HMM).

It should be noticed that the application of HMM to the DBM data to detect entering, leaving, and hovering motion resulted in very poor accuracy of 53%. By providing insight on bee's direction, IQ data would allow for more robust classification algorithms ultimately ushering higher prediction accuracy.

6. Conclusion

This article demonstrates that the current 5.8-GHz CW radar can be deployed to monitor free-flying honeybee activity. This allows long-term data collection that facilitates hive surveillance. The radar system was able to identify different near-hive behavior such as leaving, entering, and hovering. In addition to the ability to record free-flying honeybees, the radar was also capable of detecting micro-Doppler signals associated with bee wing/limb motion, using both a DBM and an IQ mixer. The developed simulation model accurately predicted calibration target RCS and detection range increase when adding silver coating whose applicability was both theoretically and practically explored.

The volume and quality of the data collected by the radar using the DBM setup were suitable for ML analysis, which was investigated in depth. The BFCC algorithm resulted in the highest classification accuracy of 91.13% and loss of 0.3693 when using the random forest algorithm in binary classification. In ternary classification, the LSF approach exhibited the highest accuracy rate of 93.37% and the lowest loss of

0.2667 when using the SVM algorithm. It is believed that IQ data, higher ADC resolution, and advanced signal processing techniques could further improve the already considerable accuracy rate.

Due to the potential of both real-time hive status monitoring and complex behavior classification, the system can support the extraction of behavioral data as a proxy for important hive health metrics. Real-time and automated monitoring could additionally provide beehive owners with data on inactive hives or potential need for human mitigation. This would not only assist bee farms in hive surveillance tasks; it would also empower soft fruit industry stakeholders with innovative means to monitor biological pollinator behavior and efficiency at relevant locations and/or the impact of specific flower patches and polytunnels settings to increase the overall yield.