Classifying Dangerous Species Of Mosquito Using Machine Learning

Michael Flynn

School of Computing Sciences

University of East Anglia

This dissertation is submitted for the degree of

Doctor of Philosophy

May 2022

Acknowledgements

I would like to thank: Professor Anthony Bagnall, whose guidance, understanding and patience throughout the past four years has been greatly appreciated. Without the opportunities afforded to me by him, I would not be where I am today; Dr Aaron Bostrom, whose contagious enthusiasm for his work helped cultivate my own passion for research; Dr Jason Lines, for all the advice, particularly at the beginning of my PhD; and the BBSRC, for all the training and support, especially in the height of the pandemic.

I would also like to thank all the friends I have made throughout my time at UEA, particularly Michael Price, Daniel Ling, William Vickers, James Large, Joshua Thody, Benjamin Cheshire, Oliver Wagg, Warren Reynolds and those whom I shared the long PhD lunches with. The laughter and discussions have been a highlight.

A special thanks go to Angela and Nigel Hodder for putting up with me for the last two years. I am extremely grateful for all that you have done.

Finally, I would like to thank my partner, Abigail Hodder - your unwavering belief and support over the last four years has been invaluable; and my family, who have been an endless source of encouragement.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Michael Flynn

May 2022

Publications

As first author

- Flynn M., Bagnall A. (2019) Classifying Flies Based on Reconstructed Audio Signals. In Intelligent Data Engineering and Automated Learning – IDEAL 2019. IDEAL 2019. Lecture Notes in Computer Science, vol 11872. Springer, Cham.

- Flynn M., Large J., Bagnall A. (2019) The Contract Random Interval Spectral Ensemble (c-RISE): The Effect of Contracting a Classifier on Accuracy. In Hybrid Artificial Intelligent Systems. HAIS 2019. Lecture Notes in Computer Science, vol 11734. Springer, Cham.

As contributing author

- Middlehurst M., Large J., Flynn M., Lines J., Bostrom A., and Bagnall A.(2021). Hive-cote 2.0: a new meta ensemble for time series classification. arXiv preprint arXiv:2104.07551.

- Ruiz A.P., Flynn M., Large J. et al. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min Knowl Disc 35, 401–449 (2021).

- Bagnall A., Flynn M., Large J., Lines J., and Middlehurst M. (2020). On the usage and performance of the hierarchical vote collective of transformation based ensembles version 1.0 (hive-cote v1. 0). In International Workshop on Advanced Analytics and Learning on Temporal Data, pages 3–18. Springer, Cham.

- Bagnall A., Dau H. A., Lines J., Flynn M., Large J., Bostrom A., Southam P., and Keogh E. (2018a). The uea multivariate timeseries classification archive, 2018. arXiv preprint arXiv:1811.00075.
- Bagnall A., Flynn M., Large J., Lines J., Bostrom A., and Cawley, G. (2018b). Is rotation forest the best classifier for problems with continuous features? arXiv preprint arXiv:1809.06705.

Abstract

This thesis begins by presenting the performance of modern Time Series Classification (TSC) approaches, including HIVE-COTEv2 & InceptionTime, on 4 new insect wing-beat datasets. The experiments throughout this thesis endeavour to explore whether it is possible to classify flying insects into their respective species and into group based on their sex. Furthermore, it is hypothesised that a hierarchical approach to classifying flying insects is possible via filtering "easy" cases using cheap to obtain features, reducing the number of times processing intensive approaches are utilised. Experiments are undertaken on 3 representations of the data: Harmonic Spectral Product (HSP), the raw data and spectral data. HSP is a method of extracting the fundamental frequency of a signal. It represents a logical benchmark for comparison and, is easy and quick to extract. In one dataset, InsectSounds, species are separated into sex. Evaluation of the results achieved with the HSP representation showed that despite a relatively poor overall accuracy this feature produces a low type II error with respect to female mosquitoes. It is shown that classes of mosquitoes species that are female were more likely to be miss-classified as other female mosquito classes and, where fly classes are miss-classified as mosquito classes, they are typically classified as male mosquitoes. Previous work had shown that transformation into the frequency domain has a positive effect on performance. Audio data is typically recorded at a high sample rate, which results in high spectral resolution. As a result, approaches from the literature have used truncation of high and low frequency data to reduce runtime. It is hypothesised that inclusion of low frequency data will aid classification. This is because low frequency data is likely caused by the body of the mosquito and morphological differences, such as size, are strongly correlated to sex. The results show that the performance of all approaches was improved by the use of spectral data. The results also showed that spectral data that included low frequency information resulted in a higher overall accuracy than transformations that discarded it.

Formative experiments showed that HIVE-COTEv1 was the most accurate approach at classifying flying insects. HIVE-COTEv1 is a heterogeneous approach that consists of 4 modules, Random Interval Spectral Ensemble (RISE), Bag Of SFA Symbols (BOSS), Shapelet Transform Classifier (STC) and Time Series Forest (TSF). The predictive power of these modules are combined via Cross-validation Accuracy Weighted Probabilistic Ensemble (CAWPE). The RISE approach was chosen as the spectral component as it was "best in class" at the inception of HIVE-COTEv1. It is suggested that a significant improvement to the usability and accuracy of RISE, would translate as an improvement in the performance of HIVE-COTEv1. The introduction of contracting provided a method through witch the training time of RISE could be effectively controlled, improving its usability. A review of the interval selection procedure led to improvements that had a significant positive effect on accuracy. A review of spectral transforms and the method of combining them led to a further improvement to accuracy, and an architecture in which multiple transformations are applied.

In order for smart traps to be effective they are required to work for extended periods in rural locations. Implementations of hierarchical approaches show that two expert features, HSP and time of flight (TOF) are effective in reducing test time and therefore the amount of processing required. This is achieved via first classifying the test case using simple approaches, such as BayesNet, and only if the confidence in the prediction does not meet a parameterised threshold using a more powerful approach.

In an evaluation of several methods of combination, the most efficient of these is shown to increase classification accuracy by 0.6%, increase the TPR of female mosquitoes by 48/10,000, decrease the FNR of female mosquitoes by 83/15,000 and reduce test time by 1.5 hours over 25,000 instances, when compared to the single best approach InceptionTime. Furthermore, a cumulative approach to combining the expert features with the InceptionTime approach resulted in a 4.14% increase in accuracy, an increase in the TPR of female mosquitoes of 139/10,000 and a decrease in the FNR of female mosquitoes of 45/15,000.

Table of contents

List of figures

List of tables

List of Symbols

Acronyms

ACF	Autocorrelation Function
AF	Audio Features
ANN	Artificial Neural Network
AR	Autoregressive
AUROC	Area Under the Receiver Operator Curve
BN	Bayesian Network
CAWPE	Cross-validation Accuracy Weighted Probabilistic Ensemble
cBOSS	contracted Bag of SFA Symbols
CNN	Convolutional Neural Network
DC	Direct Current
DFT	Discrete Fourier Transform
DrCIF	Diverse Representation Canonical Interval Forest
DTW	Dynamic Time Warping

ED	Euclidean Distance
FFT	Fast Fourier Transform
FNR	False Negative Rate
FPR	False Positive Rate
GPU	Graphical Processing Unit
HC1	HIVE-COTEv1.0
HC2	HIVE-COTEv2.0
HIVE-COTE	Hierarchical Vote Collective of Transformation-Based Ensembles
HSP	Harmonic Spectral Product
IDFT	Inverse Discrete Fourier Transforms
IT	InceptionTime
LED	Light Emitting Diode
MFCCS	Mel-frequency Cepstral Coefficients
MLP	Multilayer Perceptron
NB	Naive Bayes
NLL	Negative Log Likelihood
NN	Nearest Neighbour
PACF	Partial Autocorelation Function
PCB	Printed Circuit Board

PS	Power Spectrum
PSD	Power Spectral Density
RESNET	Residual Network
RISE	Random Interval Spectral Ensemble
ROCKET	Random Convolutional Kernel Transform
SAX	Symbolic Aggregate Approximation
SFA	Symbolic Fourier Approximation
SPEC	Spectrogram
STC	Shapelet Transform Classifier
SVM	Support Vector Machine
TDE	Temporal Dictionary Ensemble
TEIC	Technological Educational Institute of Crete
TOF	Time Of Flight
TPR	True Positive Rate
TSC	Time Series Classification
TSF	Time Series Forest
UCR	University Riverside California
WHO	World Health Organisation

Chapter 1

Introduction

Over the last century there have been many attempts at solving the problem of automatically classifying insects, based on audio, or audio like features. Increased interest in classifying insects has been fuelled by a number of factors. Insects are responsible for the pollination of the majority of crop species, but are also vectors for disease and responsible for a substantial number of fatalities. Mosquitoes (members of the Diptera order) can be found throughout the world. Their ability to thrive in a location is tightly coupled to factors such as temperature and humidity. Changes in climate have led to an increase in the number of locations where dangerous mosquitoes can be found and mechanistic models have shown that the area in which mosquito species capable of transmitting blood borne diseases can survive, will extend into the European continent by 2030 [8].

Over the last two decades, the annual number of deaths caused by malaria has reduced by approximately 300,000. However, the number of cases has not seen the same level of decline and in 2019 was estimated as 229 million. The majority of these cases were reported in the WHO defined African region. The absence in the decline of cases is conflated by large increases in population size, particularly in the sub-Saharan region, where the population has increased by 435 million in the period between 2000 and 2019. Efforts to stem the flow of deaths caused by blood borne disease have been effective. This is in part due to the funding for resources such as: mosquito nets, which as of 2015 were available to 68% of children in endemic areas; and antimalarial medication, which at the same point was available to 13% of infected children [69]. Despite this, there remains some cause for concern, as the incidence rate in the West Pacific, Europe and the Americas has been increasing since 2015.

As the number of countries with pre-eradicated status continues to move toward being 'malaria-free', the WHO has highlighted the requirement for a change in strategy. This new strategy outlined in the Global Technical Strategy for Malaria 2016-2030 [70], postulates that emerging technologies, such as smart traps, will play a pivotal role in preventing the reintroduction of blood born disease, as they represent the potential to mount effective and cost efficient surveillance. Currently, quantifying the abundance of an insect species in a natural setting is challenging. Typically, expert entomologists are required to manually identify species using morphological differences. This can result in a lengthy delay in the relaying of useful information to authorities and the amount of data that can be collected is limited. Monitoring the presence and abundance of mosquitoes is crucial in understanding the population dynamics and effectiveness of interventions. Advances in the sensor technology that would be required for these proposed smart traps has recently made the collection of large datasets more feasible [16] [75] [94] [79]. These approaches, described in more detail in Chapter 2, record data as an object passes through a target area. The movement of insects can be detected and recorded using infrared light emitting diodes (LEDs) and optical transistors. The voltage or current at the base of the optical transistor fluctuates as a function of occlusion from the LED. As an insect flies between the two components, the majority of the occlusion is caused by its wings. As the wings are also the mechanical source of a mosquitoes characteristic buzz, the resultant data is often interpreted as audio, and as such, can be framed as a time series classification problem.

Time-series datasets are so called, because of the relationship between attributes. Where a traditional classification datasets may represent measurements or characteristics collected arbitrarily with respect to either order or time, a fundamental assumption with time-series data is that the attributes are in some sense ordered. For example, attributes might correspond to the height of tide at one point along a coast line over time, but equally might correspond to the height of tide at one point in time along a coast line. This facilitates the implementation of classification approaches that use this assumption, to somehow extract informative features from the data. In Chapter 3, a number of well known time-series approaches are described, including the state-of-the-art approaches HIVE-COTEv2.0, and InceptionTime (IT). These are later applied to the "insect classification problem" - is it possible to determine an insects specie? Is it possible to determine an insects sex? Is it possible to determine an insects genus?

1.1 Research Hypothesis

This thesis sets out the results of a number of experiments designed to help further the field of automatic insect classification using features derived from wingbeats. The primary hypothesis is that it is possible to classify multiple species of mosquitoes into their respective sex using wingbeat features. Following this, experiments were devised to asses whether it is possible to determine a mosquitoes genera, which approaches produced the best results when classifying insects into their respective species and sex, and finally, is it possible to devise a hierarchical approach to classifying mosquitoes that reduced processing time whilst maintaining or improving accuracy?

1.2 Contributions

The contributions presented in this thesis are summarised as follows:

1. **Benchmarking algorithms on 4 new insect datasets.**

   The results of experimentation across 4 recently curated insect-centric datasets: InsectSound, MosquitoSound, Aphids & FruitFlies. These experiments, for the first time, provide a performance baseline established by state-of-the-art time-series approaches. The results from experiments on the InsectSounds and MosquitoSounds datasets were first published in [34] and the results for the Aphids and FruitFlies datasets, as well as all results from deep learning approaches, are first presented in this thesis.

2. **A novel approach to controlling train time**

   An adaptive approach which employs regression as a mechanism to intelligently control the space from which intervals, a subset of the dataset consisting of contiguous attributes, can be drawn. The approach was first published in [35]. An experimental review undertaken on 112 datasets from the UCR archive compares a new approach to an obvious baseline where it demonstrates a superior ability to adhere to a limit on the duration of training, whilst maintaining a comparable accuracy.

3. **A novel interval selection policy that significantly improves accuracy**

   The Random Interval Spectral Ensemble (RISE) is the spectral component of the The Hierarchical Vote Collective of Transformation-Based Ensembles version 1 (HIVE-COTEv1). HIVE-COTEv1 is a heterogeneous ensemble formed of best-in-class from 4 archetypes of approach, presented in section 3.3, and was designed with oscillatory data, such as audio, in mind. As such, RISE represents a sensible starting point for audio classification problems. Furthermore, any improvements to the performance of RISE will potentially improve the HIVE-COTEv1 approach. An investigation into the interval selection process of the RISE algorithm resulted in a second algorithmic contribution. The proposed interval selection implementation produces a significantly better performance with regards to accuracy than 3 alternative methods, including the default method used in the published version [5]. This is achieved via an novel method of selecting start point, end point and length, that hinges on a deterministic number of anchor points.

4. **A new spectral approach cRISE$_{\mathbf{All}}$**

   The novel cRISE$_{\mathrm{All}}$ algorithm. The configuration of this approach is the outcome of an extensive review of spectral transformations and methods of combination. It represents the most performant cRISE variation of those presented in this Thesis. The motivation of which was to improve the performance of HIVE-COTEv1 with respect to audio classification: e.g. given the audio signal, is it predict the species of insect?

5. **The presentation of a hierarchical approach to insect classification**

   Validated by the findings in Chapter 5, 7 methods of combining the classification test distributions from multiple approaches were implemented. These distributions represent the predicted probability of a test instances class membership. In these experiments methods combine the distributions of approaches built on expert features and the raw data. The objective was to increase test accuracy and decrease the overall processing time. The methods are split by their hierarchical or cumulative nature and results showed that expert features are effective as a filter against unlikely female mosquito cases.

1.3 Thesis structure

The remainder of this thesis is laid out as follows. In Chapter 2, *The Classification of Flies*, the mosquito condition is described. This includes: a brief introduction on the taxonomy of flying insects and the structure of the Culicidae family; the mosquito life cycle, providing context to the difficulties in controlling mosquito populations; the feeding habits of adult female mosquitoes, including an explanation of why they require a blood meal and information on the cost of mosquito transmitted disease, both in terms of life and money. A review into the relevant literature is then undertaken, highlighting the performance of approaches applied to the classification of flying insects prior to this thesis. Throughout this review, the historic lack of high quality data that has in effect, curtailed progress in this area s discussed. Following this, detail of two recently developed pieces of hardware that have been used to create large insect wingbeat datasets is presented. In Section 2.4 comparisons between the designs of the hardware are drawn, with commentary on the effects this has on the resultant data. Finally, 4 large insect-centric pseudo-acoustic wingbeat datasets, curated via the aforementioned hardware are described. This includes a discussion of dataset characteristics, such as distributions of intra-class wingbeat frequencies and time-of-flight information.

In Chapter 3, *Time-series Classification*, the idea of time-series data is introduced and contrasted to the structure of traditional data. This includes introducing the descriptive notation used throughout the remainder of this thesis. The way in which classification algorithms are divided based on core characteristics is explained, before introducing and defining each approach that features throughout the experiments in Chapters 4, 5 and 6. Finally, the tools and measures used to compare and quantify the approaches are presented.

Chapter 4, *The Random Spectral Interval Ensemble (RISE)*, presents the efforts made to improve both the usability and the accuracy of the RISE algorithm. RISE represents the spectral module of the HIVE-COTEv1 algorithm, which is shown in Chapter 5 to perform best on 2 insect wingbeat datasets. The hypothesis was that improvements to RISE would in turn improve the performance of HIVE-COTEv1. Previous work showed that the runtime complexity of RISE made the train time prohibitive for large problems. In response, the development of a novel and effective method of controlling runtime, detailed in (contribution 2), was undertaken. Furthermore, through a review of the interval selection process a significant improvement to accuracy via a novel selection policy, detailed in (contribution 3), was made. Finally, an extensive survey investigating spectral transforms, training set tuning and methods of feature combination led to a reinvigorated approach cRISE$_{\mathrm{All}}$, detailed in (contribution 4).

Chapter 5, *Insect Classification I*, presents a more robust method of extracting the wingbeat frequency feature from raw data and a new approach to converting raw data to the frequency domain. In Chapter 5 the benefits of this approach is shown through experimentation and in the process, a new benchmark is established on 2 large insect wingbeat datasets, highlighted in contribution 1. This work revolves around two ideas. Firstly, that wingbeat frequency could be used as an effective method for screening candidate female mosquito cases and secondly; low frequency data, which is likely to inform insect body size, contains useful information.

Chapter 6, *Insect Classification II*, presents experimental results on current state-of-the-art approaches, such as HIVE-COTEv2 and IT, over 4 large insect wingbeat datasets, before proposing a hierarchical method that incorporates expert features to improve accuracy, whilst decreasing overall processing time. The Chapter begins with comment on the effectiveness and reliability of expert features over these datasets, highlighting both the strengths and the weaknesses. A review of the current, best in domain approaches, on both raw and spectral representations is then undertaken.

Contrary to expectations, it is found that the best performance does not come from the spectral domain and that approaches that derive features from convolutions perform best. Methods of combining the IT approach, which is found to perform best, with expert feature approaches are then presented. The outcome, detailed in contribution 5, is an improvement in predictive accuracy and a decrease in test time.

Finally, in Chapter 7, A discussion of our contributions and thoughts is presented, before some potential avenues for future work, based on the findings and observations of this thesis are detailed.

Chapter 2

The Classification Of Flies

2.1 Introduction

The classification of insects using features derived from the characteristics of wingbeats has been an active area of research for over 80 years. However, recently there has been a renewed interest in the topic. This interest is a result of many factors, but reflects the changing attitude and focus on the relationship between insects and human health. This relationship is most commonly visible in the contexts of disease, where fly species are particularly likely to be vectors for transmission; and crop pollination, where the health of specific flying insect populations are directly related to the yearly yield. In many settings, the ability to monitor the size, positioning and movement of insect populations is desirable. In the context of mosquitoes this is particularly evident, as the decision to request aid and direct resources is uniquely tied to information on local population emergence, abundance and positioning. Enhanced surveillance solutions now form an integral part of the World Health Organisation's (WHO) global technical strategy for malaria 2016 - 2030 [69]. The report highlights the importance of accurate and timely data, and describes it as key in helping stakeholders make informed decisions.

Traditionally, insects are classified via their morphological differences by expert examination or, when this is not possible, by gene sequencing. This involves trapping and recovering insects, a labour intensive task that is further complicated by the need for highly skilled entomologists to perform the manual classification. The process is often slow and can result in some delay before the finalised population data reaches the appropriate authorities. As a result, local hospitals are often caught unaware during outbreaks of disease and efforts to combat the outbreak can be poorly targeted. Automated in situ classification is now seen as a vital tool in controlling malaria. Recently, multiple studies have shown that a novel use of low cost sensors is capable of producing large quantities of seemingly high quality flying insect data [16] [75] [79] [94]. These approaches, discussed in more detail in Section 2.5, record the movement of an insect whilst in flight. The nature of this recording technique has allowed the creation of detailed datasets that offer the opportunity to attempt the classification of sex, species and genus. An additional benefit in the adoption of hardware based data collection is the inclusion of spatial and temporal information.

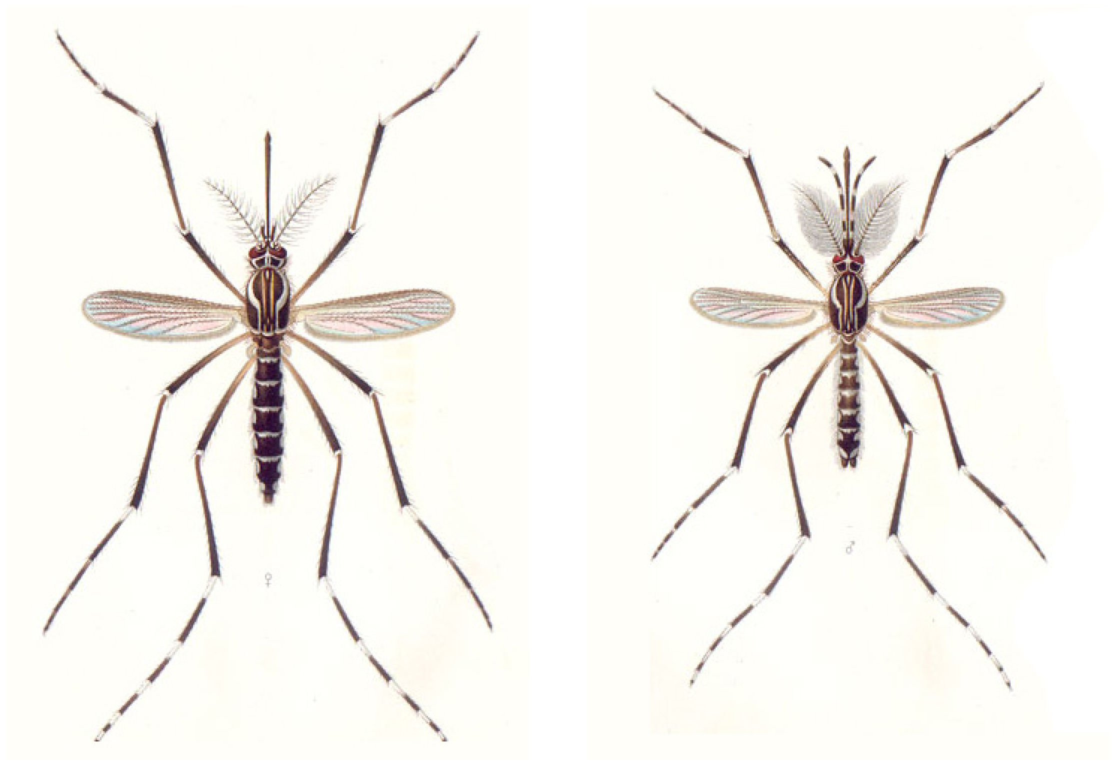
The remainder of this chapter is laid out as follows. In Section 2.2, clarity on the colloquial phrase mosquito is provided and some background as to the structure of the Culicidae family is given; this includes a brief overview of the mosquito life cycle and the challenges associated with predicting the emergence of adult mosquitoes; a discussion of the feeding habits of mosquitoes, including how female mosquitoes locate their hosts and why they need to feed on blood; identification of the areas in which these vectors for disease can be found, illustrating their near global domination; and a discussion of the costs associated with mosquito transmitted disease, particularly malaria, both in terms of loss of life and money spent trying to eradicate, treat and prevent its spread. The entomological information presented in this section is a summary of several publications. For a further, more in depth discussion and explanation of the mosquito condition, A. N. Clements: The Physiology of Mosquitoes [17] is recommended. In Section 2.3 a summary of the literature on classifying flying insects is presented, providing an overview of approaches that have been applied to the classification of flying insects. Throughout the section, the historic problem of data quality and abundance is highlighted. This is followed by a discussion on how a lack of high quality data has led to a failure to demonstrate the extent to which approaches are able to generalise. In Section 2.4, a description of the state-of-the-art data collection processes used to produce large insect-centric datasets is provided, before an explanation of the recording process. Finally, in Section 2.5, a detailed description of four flying insect datasets used in Chapters 5 and 6 is given.

2.2 The Culicidae Family

Mosquitoes are members of the Dipetra order of the Insecta class of animals. The Dipetra order is estimated to contain 155,477 species [59], making it the third largest order in this measure, beaten only by the Coleoptera (beetle) and Lepidoptera (butterfly and moth) orders. The Dipetra order is commonly known as the 'true flies order'. All insects that fall into this order have: heads capable of rotation through three axis; compound eyes; mouth pieces that either pierce and suck or lap and suck; and feet adapted to hold onto smooth surfaces via claws and pads. Another shared trait amongst flies is their life cycle stages. Flies are laid as eggs, typically on a suitable food source for the emerging larvae. The larvae, having hatched and consumed adequate sustenance forms a pupa, from which it later emerges in its adult form. However, the key characteristic of the Diptera order is that flight is only achieved via the use of one set of wings. In the location where a second set would be found in other orders, such as Hymenopttera, which boast two sets of wings joined via hooks, there are instead specialised sensors known as halters, that aid flight and provide information on wind speed, pitch and roll [28].

The phrase Mosquito is the colloquial name for the Culicidae family of the Diptera order. This family contains roughly 3,500 species split across 112 genera [53] and 3 sub-families. Often displaying species and genera level morphological differences, mosquitoes are arthropods. Their bodies are typically slender and house three sets of long slim legs. They also exhibit the characteristic proboscis mouth piece. They can be identified via their long slim wings, that present with a scaly effect on the surface. They typically measure between 3mm and 6mm in length and will be black or have black markings. Another defining behavioural characteristic is the way in which they position their hind legs whilst at rest. Mosquito species will hold their hind legs upwards and laterally outward from the body. Whereas other Diptera familes, such as the Chironomidae (midges), tend to hold their front legs outwardly ahead of them whilst at rest.

As previously mentioned, all fly species go through the same stages in their life-cycle [41]. The first three of these (egg, larvae and pupa) typically occur in water. Depending on species and ambient temperature, these first stages are expected to last between 5 - 14 days. However, there are some specific adaptations that species which live in areas that experience extended periods of drought or freezing have evolved, that allow their development to be temporarily paused. This process, diapause, is similar to hibernation and effectively allows the insect to wait for the environment to become suitable before continuing their life cycle. One example of this is in the Pitcher Plant mosquito (Wy. smithii) [76]. Found in Northern America, the eggs of this species are commonly laid in late August. Once the length of day declines adequately, the mosquito larvae enters a state of diapause until the days begin to lengthen again. During this period, the larvae can withstand being frozen solid. Another example of



(a) *Ae. aegypti* female

(b) *Ae. aegypti* male

Fig. 2.1 Male and female *Ae. aegypti* mosquitoes [39].

diapause in mosquito species can be seen in the Asian Bush mosquito (Ae. japonicus). The eggs of this species are resistant to desiccation and continue their life-cycle once re-sumerged in water [46].

Typically, mosquito eggs are laid in stagnant water by females (although, there are examples of some species opting to lay eggs either at the waters edge or directly onto aquatic plants [2]). Species from one genera do not gravitate toward one uniting water source. Instead, they select a breeding site based on the local environment and any specie specific adaptations. These sites include lakes, puddles, marshes, salt water [21] [95] and phytotelmata, such as the reservoirs found in Bromeliads, hollow tree trunks, and even the liquid found in pitcher plants. Some species, including the Pitcher Plant mosquito is exceptionally selective about its breeding site and will seldom lay eggs anywhere but in the pitchers of the Purple Pitcher plant (Sa. purpurea).

Other species that favour phytotelmata breeding grounds quickly adapt to artificial sites common in human settlements, such as buckets or tyres [84]. This group of adaptable mosquitoes includes the *Aedes* genus, which are an important vector for disease [93]. This fact has led to extensive education campaigns in countries affected, aimed at increasing awareness of the risk standing water presents. Cheap and accurate automatic traps could provide an early warning to authorities and as a result, aid in the targeting of resources to communities, in turn, improving the effectiveness of these campaigns and the overall cost effectiveness of national vector mitigation efforts.

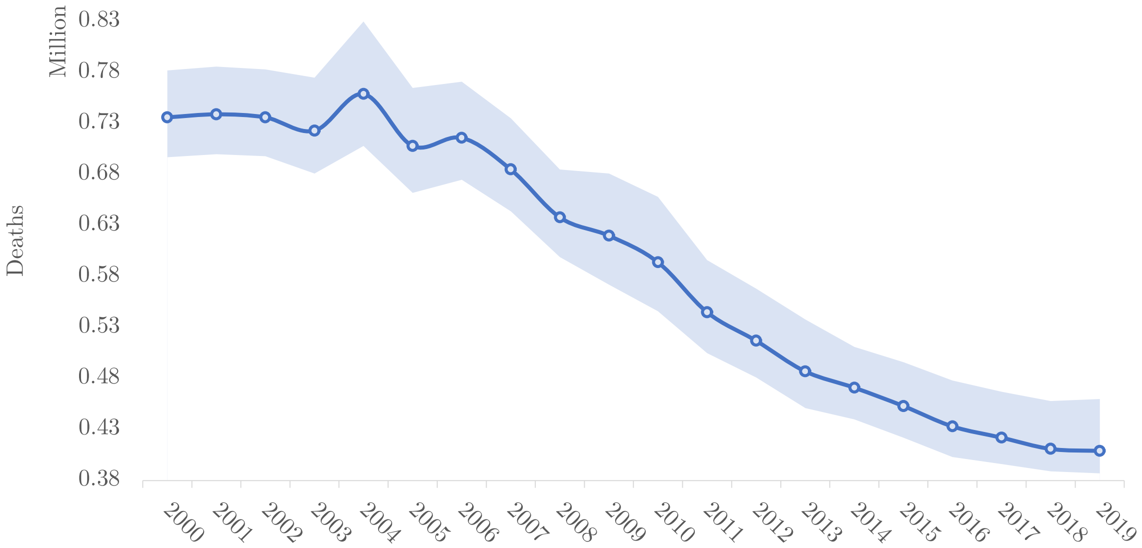


Fig. 2.2 Estimated annual global number of deaths from malaria shown with a 95% confidence interval [71]

Mosquitoes exhibit four main procedures of oviposition [23]. The method of oviposition are not consistent within each genus and it is common to find multiple oviposition methods within the species of a genus. A widely adopted method, often referred to as dapping, sees the female mosquito skipping along the water's surface, dropping single eggs at each point of contact. This approach is commonly seen amongst species of the Anopheles genus, as well as other flying insect species in the Ephemeroptera (Mayflies) and Ephemeroptera (dragonflies) orders. Many species of the Mansonia genus opt for a vastly different approach to dapping and instead lay eggs in small raft-like structures on the underside of aquatic plants such as lilly pads, whereas most species of the *Culex* genus lay eggs in a similar configuration, but instead opt to position these on the waters surface [24]. The final common oviposition behaviour found in mosquitoes is similar to the aforementioned dapping technique and is commonly found in species of the *Adeas* genera. Females will drop singular eggs in the manner described above. However, these eggs are deposited in damp areas close to the waters edge, typically locations such as gutters, buckets or tyres. The eggs then develop slightly before entering a period of diapause, triggered by desiccation, with development restarting after submersion at the next rainfall. An additional trait specific to Adeas eggs is their tendency to hatch in a semi-random manner within a single clutch. This makes the species from the Adeas genus very difficult to control, compared to species whose eggs hatch in a more predictable fashion. Applying insecticides in regular intervals is prohibitive for three main reasons: it is costly; it has a significant detrimental effect on wider entomology and it increases the chances of insecticide resistance. The future of vector control through insecticides will undoubtedly be more expensive as the cost of developing new chemicals is increasing whilst the available funding has decreased [51]. As a result, improved targeting strategies could be key in maintaining an effective level of control and emerging technologies are seen as a key in fulfilling this role.

Both male and female mosquitoes predominantly feed off liquids high in sugars, salts and amino acids. Commonly, these meals are nectar produced by plants (in some cases mosquitoes are pollinators) and honeydew [72]. The Culicidae family is organised into three sub-familes: Anophelinae, Culicinae and Toxorhynchitinae. Species from the Toxorhynchitinae sub-family are exclusively nectarivores and as a result, have developed long curved proboscis that are well suited to feeding on plants. Species from the remaining 2 sub-families are anautogenous. These species each have a favourable host, but will feed off a variety of vertebrates if necessary. In some cases, mosquito species are capable of producing eggs both anautogenously and autogenously. Typically, these partially anautogenous species will produce and lay their first clutch of eggs without the need for a blood meal, but require the proteins available in blood to produce subsequent clutches [40]. Furthermore, many species of autogenous mosquitoes that do not require a blood meal to produce eggs will produce larger clutches if they have one. This anautogenous behaviour is commonly seen in species of the Anopheles, Aedes and Culex genera, all of which are important vectors for disease. For example: An. gambiae and An arabiensis are both vectors for malaria and lymphatic filariasis; Cx. pipiens, Cx. quinquefasciatus, Cx. globocoxitus and Cx. australicus are vectors for a number of diseases including encephalitis and lymphatic filariasis; Ae. aegypti (shown in Figure 2.1), Ae. formosus and Ae. albopictus are the primary vector of yellow fever throughout the world and are also known to transmit dengue fever. Additionally, there are a handful of other species from these genera that are responsible for disease transmission in more remote areas, such as the South Pacific Islands [36]. Just the aforementioned 9 species cover almost the entire globe and in many places, 2 or 3 of them co-exist.

Typically, mosquito species are crepuscular or nocturnal feeders [48]. They spend the remainder of their time in the shade, although females will still bite if disturbed. The antennae of a blood sucking mosquito species houses its olfactory system. In species that prey on humans for their blood meal, females hunt their hosts via the presence of carbon dioxide (CO_2). However, the extent to which a mosquito is compelled to choose a host is more complicated. In the species Cu. quinquefasciatus, it is the presence of nonanal in the perspiration that most impacts the likelihood of selection [20]. In the Ae. aegypti species, it is the presence of sulcatone, a keytone commonly characterised as citrus [87]. Furthermore, the hunting behaviour of female mosquitoes is two fold. Firstly, they employ a non-specific search pattern in order to locate a host that emits the desired chemical triggers [22]. Once located, they perform a more specialised search of the host to identify a suitable site for feeding. Most often, the mosquito will feed shortly after landing. Sometimes, the mosquito will wander, using its labium to search for a site to feed. Occasionally, after wandering for some time, the mosquito may move on without feeding. The exact mechanism for host selection is not entirely understood, but there is mounting evidence that skin micro-flora play a role [11].

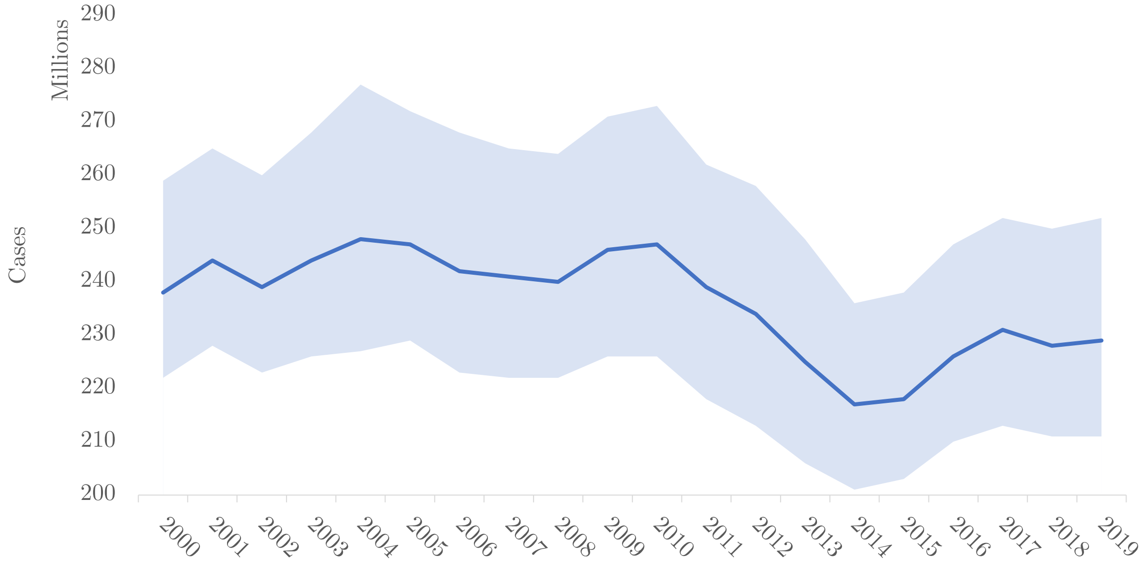


Fig. 2.3 Estimated global annual number of cases from malaria shown with a 95% confidence interval [71]

Mosquitoes are found in almost every landmass on the planet. The exceptions being Antartica and a handful of islands with polar climates [7]. The criteria for exception is down to quirks of the weather patterns, rather than the harsh temperatures. For example, there are no mosquitoes in Iceland, where the temperature can fluctuate significantly as seasons change [78]. These fluctuations interrupt the diapause of mosquito eggs, but do not last long enough for a full life cycle to complete. There are however mosquitoes in the arctic regions of Alaska where they emerge in vast numbers as the sun melts the frozen tundra. With an abundance of food in the biolayer and very few predators, swarms containing millions of mosquitoes emerge to feed on Caribou [38]. However, the window to reproduce in the arctic is small, as on average, the summer lasts approximately 8 weeks from mid June to mid August. This is a stark contrast to the active period of mosquitoes in more temperate regions, where it is not uncommon in tropical climates for mosquitoes to operate thought the entire year.

There is an established relationship between environmental factors and the abundance of mosquitoes. These factors are also tightly coupled to the effectiveness of viral reproduction. The ambient temperature, humidity and precipitation, all significantly affect the prevalence for disease in the local human population [6] [58]. This transmission seasonality is closely monitored and has been effective in explaining the location and number of outbreaks throughout the southern hemisphere during El Nino [50]. An obvious application of this information is in modelling the effect climate will have on mosquito dispersion. Commonly, the spread of disease vectors such as mosquitoes is modelled via a correlative or mechanistic model. Correlative models are based on taking existing information on the climates in which the vector is already prevalent and uses predictions on how climate is expected to change to project the areas likely to be capable of sustaining the vectors in the future. Mechanistic models are broader and incorporate information on how viruses and hosts change behaviour as the climate changes. Mechanistic models have successfully been used to model historical outbreaks and predict how risk might change with forecasted weather. However, the introduction of widespread intelligent trap network, provides the opportunity to develop a data driven approach to modelling. Typically, areas which are most affected by disease vectors are areas with tropical climates. These include large parts of the African and Asian continents, as well as South America. However, cases of viral infection via vectors such as mosquitoes have been recorded in Europe and are becoming more frequent.

At least one mechanistic model has predicted that by 2030, southern England will be climatically hospitable for the transmission of malaria to take place [8].

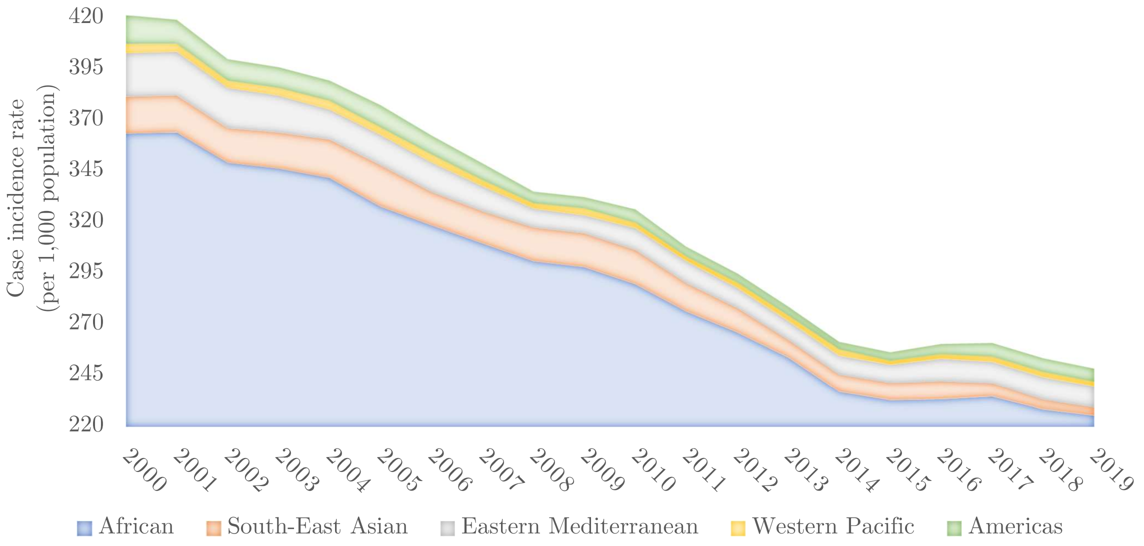


Fig. 2.4 Case incidence rate displayed per WHO region for the period between 2000 - 2019 [71]

Since the year 2000, the annual number of deaths from malaria has been in decline, see Figure 2.2. Furthermore, there has been a decline in deaths over the same period in each of the regions defined by the WHO. However, the annual number of cases, shown by Figure 2.3, has not seen the same level of decline. Estimated to be 238 million in 2000, the number of cases reached a low of 217 million in 2014, after a period of rapid decline. Cases have steadily risen since then, and in 2019 there were estimated to be 229 million.

According to data from the latest World Malaria Report published by the WHO [71], the African region saw an increase in cases in real terms from 204 million in 2000 to 215 million in 2019. However, the incidence rate, shown in figure 2.4, reduced in the same period from 363 to 225 cases per 1,000 people. This contradiction between the number of cases and the case incident rate, is the result of an increase in population from 665 million to 1.1 billion in the sub-Saharan area. The number of deaths caused by malaria in the African region decreased by 44% over this period, falling from 680,000 in 2000 to 384,000 in 2019. Despite these reductions, the African region still accounts for roughly 94% of both cases and deaths globally, see Figure 2.5. The Southeast Asia region has seen a dramatic reduction in both cases and deaths over the period between 2000 and 2019. The number of cases reduced by 74%, from 23 million to 6.3 million and the number of deaths also reduced by 74%, from 35,000 to 9,000. Over the same period, the incidence rate reduced by 78%, from 18 to 4 per 1000 people. The burden in the region falls mostly on India. It accounted for both the largest reduction in cases, 20 million in 2000 to 5.6 million in 2019, and the largest contribution, 88% of the region's cases and 86% of the region's deaths in 2019. The Eastern Mediterranean region recorded a decrease of 2 million cases, from 7 million to 5 million and, a decrease of 1,100 deaths, from 12,000 to 10,100, in the period between 2000 and 2019. In 2019, Sudan was the largest contributor of cases in the region, contributing 46%. The Western Pacific region contributed 1.7 million cases in 2019, a reduction of 43% from the 3 million cases recorded in 2000. The region also saw a 52% reduction in deaths over the same period, from 6,600 to 3,200. Papua New Guinea shoulders most of the burden in the region and accounted for 80% of the cases recorded in 2019. Lastly, the Americas region saw a 40% reduction in cases, from 1.5 million to 0.9 million. The region also saw a 39% reduction in deaths, from 909 to 551. However, over the same period the number of cases in Venezuela increased from 35,500 to 467,000.

Despite all the reductions throughout the period of 2000 to 2019, there remains cause for concern. Figure 2.4 shows the changes in case incidence rate for each of the WHO defined regions. The graph shows that both the total and individual incidence rates have significantly reduced overall. However, since 2015, only the African and Southeast Asian regions have seen a decrease in the case and death incidence rate and the Americas, Eastern Mediterranean and Western Pacific and all experienced increases in both case and death incidence rates.

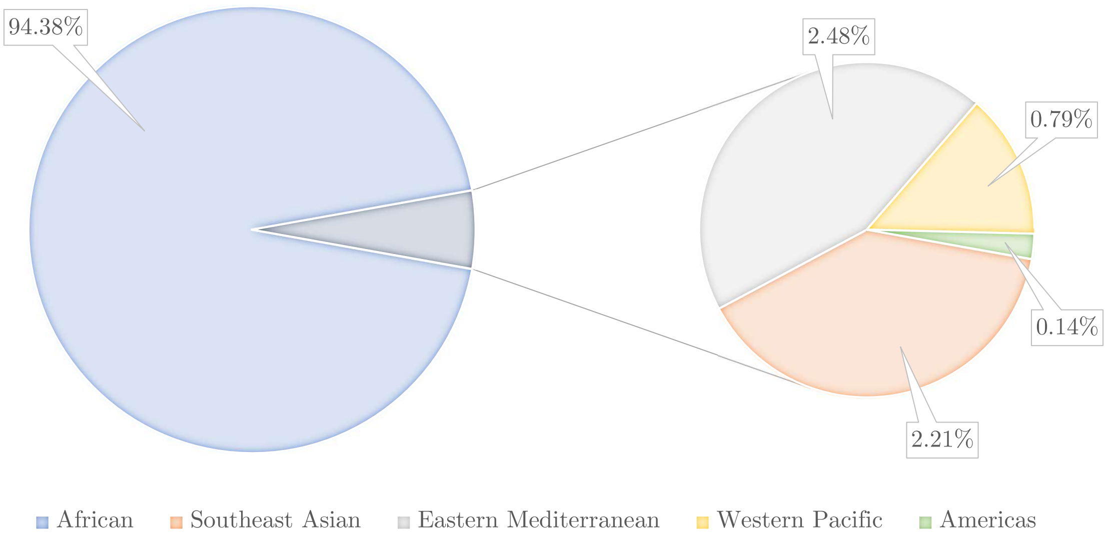


Fig. 2.5 Distribution between WHO defined regions of deaths from Malaria in 2019 [71]

In response to the devastating impact of Malaria, the WHO have published The Global Technical Strategy for Malaria 2016 - 2030 [69], a strategy which has now been adopted by the World Health Assembly. The strategy goals are to: reduce Malaria cases by 90%; reduce mortality rates by 90%; eliminate Malaria in 35 countries and prevent a resurgence of Malaria in all 'Malaria free' countries. The strategy is divided into 3 pillars: ensuring universal access to malaria prevention, diagnosis and treatment; accelerating efforts towards attainment of malaria-free status, and transforming malaria surveillance into a core intervention. It is intended that recent advancements in technology should be utilised to help enact the plan and in some cases the extent to which the plan can be executed is tightly coupled to technology. For example, an important part of pillar 1's Malaria prevention interventions hinge on vector control. In order to control mosquito populations, effective entomological surveillance is required, including a periodical review of: abundance; seasonality; time and place of biting; resting and host preference. Also, the intervention strategy highlights the value that recorded data has, when making decisions on the timing and location of applying chemical vector control techniques. Historically, any information gathered is done so manually. The process is typically lengthy, as it requires personnel with highly specialised skills to classify insects. The traditional techniques also provide relatively course data, as they are unable to provide information on the frequency of insects with respect to time of day. As a result, there is an obvious opportunity to improve the quality, quantity and speed at which information can be provided to local authorities coordinating vector control programs across the world.

Table 2.1 A table showing the chronology of literature discussed in section 2.3.

Year	Author	Title	Feature	Approach
1942	Reed et al. [77]	Frequency of wingbeat as a characteristic for separating species races and geographic varieties of drosophila	• Wingbeat frequency	• Statistical analysis
1981	Farmery et al. [32]	Optical studies of insect flight at low altitude	• Wingbeat frequency	• Statistical analysis
1986	Moore et al. [66]	Automated identification of flying insects by analysis of wingbeat frequencies	• Wingbeat frequency • Harmonics	• Statistical analysis
1991	Moore et al. [64]	Artificial neural network trained to identify mosquitoes in flight	• Frequency spectrum	• Single layer network
2002	Moore et al. [67]	Automated identification of optically sensed aphid (homoptera: Aphidae) wingbeat waveforms.	• Middle of series • Frequency spectrum • Wingbeat frequency • Harmonics	• Single layer network • 1-NN ED
2005	Li et al. [54]	Automated identification of mosquito (diptera: Culicidae) wingbeat waveform by artificial neural network.	• Raw series • Wingbeat frequency • 3 harmonics	• ANN
2014	Chen et al. [16]	Flying insect classification with inexpensive sensors.	• Wingbeat frequency • Frequency spectrum	• Naive Bayes • 8-NN ED • Single layer network
2018	Fanioudakis et. al. [31]	Mosquito wingbeat analysis and classification using deep learning.	• Spectrogram • PSD • Raw series	• DenseNet121 • 5 layer CNN • InceptionV3 • MobileNet • XGBoost • LightGBM • NASNetMobile

2.3 Classifying Fly Species

One of the earliest investigations regarding the classification of mosquitoes with respect to their wingbeat frequency was, undertaken in 1942 by Reed et al. [77]. A technique outlined by Chadwick et al. in 1939 [15] allowed the precise frequency of an insect's wingbeat to be recorded via the use of a stroboscope. The research was primarily investigative and provided evidence that the frequency of an insects' wingbeat may be sufficient to differentiate between species of fruit flies from within the same genus. The experiments were undertaken using a relatively small sample size of 332 insects, split disproportionately over three classes, representing two variants of the D. pseudoobscura specie and a single variation of the D. miranda specie. The average difference in wingbeat frequency between the two variants of D. pseudoobscura was measured to be 7.34 times the standard error, leading to the conclusion that it might serve as an adequate feature to differentiate between multiple species of fruit fly more generally. The study went on to document that the longitude and latitude at which the insects were collected, had no bearing on wingbeat frequency. However, a positive correlation between ambient temperature and the average wingbeat frequency of each specie was well noted, a relationship that is now well documented [80] [85] [89]. Furthermore, it was shown that the differences in wingbeat frequency corresponded to an insect's morphological features. The study theorised and experimentally proved that the wingbeat frequency of each insect was proportional to the ratio between the thorax volume and wing area, where the thorax volume represents an approximation of muscle volume.

In 1986, Moore et al. [66] used several combinations of wingbeat frequency and the amplitudes of the first four harmonics to devise discriminant features. These features were then used to classify the *Ae. aegypti* and *Ae. triseriatus* species into four classes, defined by species and sex. The wingbeat data was recorded using an approach developed by Unwin et. al. [90]. The approach made use of the way in which photodiodes react with respect to the intensity of light. The photodiode absorbs photons and produces current. The relationship between the intensity of photons and produced current is linear. The experimental set-up consisted of housing each group of mosquitoes in an opaque vessel, between an array of photodiodes and a light source. A device then records the fluctuations in current as the mosquitoes move between the light source and photodiode. The majority of the information recorded whilst the mosquito occludes the light source is produced by the wings. Therefore, when interpreting the fluctuations in current with respect to time, the recordings give an good approximation of the corresponding wingbeat sound. Each group consisted of 15 individual insects and recordings were made on the 1^{st}, 2^{nd}, 4^{th}, 6^{th}, 8^{th}, and 10^{th} day after emergence. Each recording consisted of 512 samples and was recorded with a 10,000Hz sample rate. On each occasion, at least 12 recordings were made. An analysis of the recorded data showed that throughout the experiments the wingbeat frequencies of the classes were ranked in the same order, with no overlap in variance. The largest insects, *Ae. triseriatus* consistently had the lowest wingbeat frequency, whilst the *Ae. aegypti* consistently had the highest. Also, the wingbeat frequencies all appeared to increase in the days immediately following emergence, before either levelling out or falling prior to climbing again. Interestingly, the evolution of average wingbeat frequency within the sample was not consistent between species. Furthermore, upon pooling the data there was found to be significant variation between all groups ($n = 468$, $df = 3$, $P < 0.0001$) and, although there was an overlap between some groups, the groups were found to have a significantly different mean wingbeat frequency ($\alpha = 0.05$, Duncan's multiple range test [29]). Discrimination functions were derived for each of the aforementioned feature combinations via the DISCRIM procedure of the Statistical Analysis System [86]. Accuracies were produced using a different collection of mosquitoes of the same classes. It was found that wingbeat frequency produced the highest accuracy of 84%, followed by wingbeat frequency plus the absolute amplitude of the first harmonic and the relative amplitudes of the 2^{nd}, 3^{rd} and 4^{th} harmonics, which produced an accuracy of 82% and lastly, the wingbeat frequency plus the relative amplitudes of all four harmonics produced an accuracy of 81%.

Although this work does draw conclusions that are supportive of the notion that wingbeat frequency is an adequate attribute to classify mosquitoes by race, it also highlights that there is an overlap of wingbeat frequencies between species; there is variability introduced by temperature that, although measurable, may make the general classification problem difficult; the wingbeat frequency of a mosquito does vary throughout adulthood and in some cases overlaps with the expected frequency of other species.

Until a study in 1991 by Moore et al. [64], the majority of research has focused on whether specific attributes such as fundamental frequency and harmonics provide enough information to accurately classify mosquito genus and specie. Exploiting the Artificial Neural Network's (ANN) ability to accept an array of values, Moore et al. [64] discovered that valuable information is contained within the spectra. A three-layer network was developed using Brain Maker Professional v1.5, consisting of an input layer containing 256 nodes a hidden layer containing 127 nodes and the output layer. The signal was normalised prior to classification. Identification based on the output with the highest likelihood was correct 92% of the time. A threshold was then applied, stipulating that if the highest output value was less than 90%, or less than 90% of the remaining three values combined, the case was classified as unidentified. This lead to a reduction in accuracy, to 75%, but all six cases were unidentified rather than misidentified. To quantify what impact the additional information present in the spectra had, all frequency bins other than the one representing wingbeat frequency were set to 0 and the network was retrained. The effect was a drop in accuracy to 88%, when only considering the output of highest likelihood. Applying the decision rule changed the classification output such that there was one misclassified and five unclassified instances. Although these results were promising, the two groups of mosquito used were morphologically very dissimilar. Furthermore, these experiments utilised a very small number of insects.

In a more analytical paper [67] Moore et al. performed five experiments in which a 1-NN classifier in conjunction with the Euclidian Distance function and an ANN, consisting of a hidden layer of 4,983 nodes and an output node representing each species, are directly compared. The dataset consists of classes representing the five most common Aphids found in Guam and was derived using a technique adapted from Unwin and Ellington (1979) outlined in [65]. For these experiments, the five data representations used were: the middle portion of the time series; frequency spectrum; signature, a concatenation of fundamental frequency and harmonic amplitudes; harmonic amplitude alone and fundamental frequency. The findings in this paper support much of the previous literature on the classification of winged insects. The mean wingbeat frequency was found to differ significantly between all classes, but variance within classes negated its ability to act as a consistent distinguishing attribute, and the wingbeat frequency also maintained a positive correlation to ambient temperature. The ANN out-performed the 1-NN on all variations of the data, achieving a high of 69% accuracy on the frequency spectrum series. Farmery et al. [32] showed that the angle of an insect's flight path relative to the light sensor, has a profound effect on the frequency spectrum of the wingbeat, although no effect on the fundamental frequency. However, in the 2002 paper [67], Moore et al. hypothesised that within class variance was also increased by insects buzzing their wings without taking-off, a behaviour witnessed multiple times throughout the data recording process. Critically, this paper highlights two things:

firstly, winged insect classification may be a generic problem. That is, a system in which mosquito classification is trivial, may also be a system in which aphid classification is trivial; secondly, that although an automated classification application could be viable from an algorithmic point of view, the challenge of building a system which is robust enough to deal with the variability introduced by unrestricted insect flight paths and behaviours, is non-trivial.

In 2005 Li et al. [55] used a neural network trained via back propagation to classify 9 groups of mosquitoes representing 5 species. In this case, there were 100 input nodes, 5 hidden layers of ten nodes and 9 output nodes. Three data sets were created: the raw time series $x_1 \ldots x_{300}$; spectral information f_1 f_2 f_3 f_4, where f_1 represents the wing beat frequency and f_2 f_3 f_4 represent the second, third and fourth harmonics and a concatenation of the previous two $x_1 \ldots x_{300}$, $f_1 \ldots f_4$, which was then normalised before classification. The results show that classification using just the frequency spectrum was most effective at 72.67% accuracy, followed by the third data set at 59.77% accuracy. These results appear to further bolster the idea that automatic insect classification using spectral information is feasible.

In the culmination of multiple iterations of research into data collection and classification techniques, Chen et. al. [16], represents the current state of the art approach with regards to the Insect wingbeat data, described in Section 2.5.3. Some subtle but effective changes to the raw data collection method presented by Unwin et. al. [90] allowed Chen et. al. [16] to collect many more recordings of insects in flight than previous experiments. A 9 step experiment was devised, whereby an additional insect group, consisting of 5,000 instances, was added at each step. This led to a classification problem containing four species of mosquito separated by sex and two species of fly. The insect wingbeat dataset has now been made public and at present remains the most varied insect wingbeat dataset available.

The slight adjustment to Unwin's recording method sees an infra-red beam, aimed at a photo transistor, rather than a DC halogen lamp. This provides a reduction in the recorded area, which in turn eliminates anomalies related to non-flight recordings. The output from the photo transistor is then passed through a PCB, before being recorded by a Zoom H2 Handy digital recorder at a sample rate of 32,000Hz and saved as an MP3 file.

The first step in the experimental process consisted of both setting a benchmark accuracy using the relatively simple one dimensional Naive Bayes approach and illustrating what benefit a multidimensional Bayesian approach brings. This was achieved with a two class experiment, *Ae. aegypti* male vs *Ae. aegypti* female. As previously mentioned, the first configuration used a commonly evaluated feature [85] [80] [81], wingbeat frequency. It had previously been shown that the distribution of wingbeat frequencies mirrored a Gaussian distribution. Although the Bayesian classifier does not have to assume Gaussian Distribution, it is less computationally taxing. The second approach utilised the Nearest neighbour (kNN) classifier's ability to process multi-dimensional data. In this case, the conditional probabilities were calculated based on the frequency of each class within the $k = 8$ nearest neighbours. This approach was tuned using a validation process outlined in [49], in which part of the training set is kept separate and used to evaluate the error rate at multiple k values. The value of k that minimises the error on the training set is then selected. The data used was a randomly selected subset from a pool of over 20,000 examples. The number of each sex present in the experiment was incremented in steps of 100 instances, between 100 instances – 1,000 instances. Each step was rerun 100 times to produce an average accuracy for both configurations. Configuration one produced an accuracy of 97.47%, if 1,000 training examples are present for each sex. Whereas, the second configuration reduces the error rate by more than two thirds to 0.78%, from 2.53%, achieving 99.22% accuracy.

This equates to roughly 8 miss-classifications per 1,000, providing conformation that increasing the amount of data produces an increase in accuracy [42].

Undoubtedly the introduction of more data increases accuracy. However, the classes used in this experiment are very different morphologically. Females are in general much larger than males of the same specie and this is reflected in both the wingbeat frequency and harmonic signature. The second stage of the experimental process endeavoured to test the assumption that an ANN would perform traditional approaches. This was tested via simple comparison with the second configuration from the preceding experiment. A Fourier transform was applied to the raw data and the power coefficients were retained for the experiments. The species used were: *Cx. stigmatosoma* female, *Ae. aegypti* female and *Cx. tarsalis* male. The training and test data were both randomly sampled from disjointed sets. The experiment was run on over 1,000 random re-samples, with training set sizes incremented in size between 5 and 50 instances. The neural network consisted of one hidden layer of 10 nodes. It was found that the neural network accuracy converges with that of the Bayesian classifier, after a relatively small number of instances are introduced. However, it perform consistently worse for small datasets and even a with large numbers of instances, still maintains a large uncertainty value. Chen et. al. [16] note that the number of instances used in this example, contradicts the claim of being able to produce large data sets stating that:-

> 'in some cases it may be necessary to carry out semi-supervised learning, using only a small number of labelled instances. This model can then be used to classify examples either from achieved data or in the field if necessary.'

Other characteristics of a Bayesian classifier that were noted as potentially beneficial in terms of the application were:

- Very low CPU and memory requirements. Although in a laboratory environment this may give little benefit, in the field, this represents a substantial advantage.
- Small number of parameters; this makes them relatively simple to implement and set-up, when compared to neural networks, which often have many parameters that need careful tuning [67]; [55].
- They are capable of integrating additional information very efficiently, allowing new information, perhaps expert, or based on trends in the area to be incorporated in the classification. This ability allows for the modular improvement of the algorithm over time.
- Trivial to produce an unknown classification class. Allowing the ability to monitor the number of these instances for further investigation.

Chen et. al.[16] then propose a further improvement based on the time of intercept (TOF) information collected, whilst producing their dataset. For each class in the training set, a histogram of frequency with regards to time of day was produced. The posterior probability is now calculated via class-conditioned probability of the insect sound, using the nearest neighbour method outlined previously and the class-conditioned probability based on when the sound was produced. This incorporation of Circadian rhythm information is reported to provide the significant increase in accuracy from 87.57% to 95.23% for the classification of *Cx. tarsalis* female in classification between *Cx. stigmatosoma* male, *Ae. aegypti* male and *Cx. tarsalis* female.

This led to the development of a broader experiment, utilising the complete 50,000 instance data set. The experiment began with the classification of just two classes. At each subsequent step, an additional species consisting of 5,000 instances was added. In this experiment, the classifier also made use of additional TOF data, as well as the data present in the frequency spectrum.

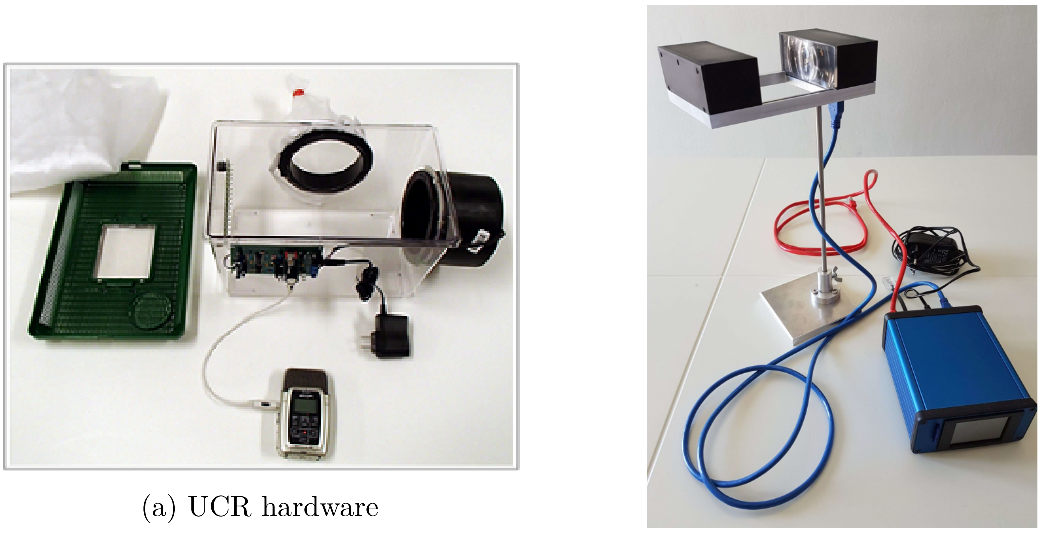
Table 2.2 A Table presenting the results of Chen. et. al's. approach, outlined in [16].

Step	Species added	Accuracy
Step 1	*Ae. aegypti* ♀	-
Step 2	*Mu. domestica*	98.99%
Step 3	*Ae. aegypti* ♂	98.27%
Step 4	*Cx. stigmatosoma* ♀	97.31%
Step 5	*Cx. tarsalis* ♂	96.10%
Step 6	*Cx. quinquefasciatus* ♀	92.69%
Step 7	*Cx. stigmatosoma* ♂	89.66%
Step 8	*Cx. tarsalis* ♀	83.54%
Step 9	*Cx. quinquefasciatus* ♂	81.04%
Step 10	*Dr. simulans*	79.44%

As shown in Table 2.2, the approach is robust against multiple species of mosquito, achieving a minimum of 79.44% accuracy. The incorporation of TOF information undoubtedly improves performance and anecdotal results show that it is likely that morphologically similar species of mosquito have evolved differing Circadian rhythms. However, it is unclear exactly how much of a positive effect it has, as Chen et. al. [16] chose to only report the effect on accuracy for *Cx. tarsalis* female.

In recent years, advancements in processing power and particularly the power of Graphical processing Units (GPUs) have facilitated the development of deep learning frameworks such as Tensorflow and Pytorch. In turn, these frameworks have accelerated the development and proliferation of deep learning approaches. In many domains, including image classification, natural language processing and automated animation, these are now considered the state-of-the-art approach. In multiple cases these approaches have been successfully applied to insect-centric image databases [13] [14] [47] [54]. In one recent study, Fanioudakis et al. [31] reviewed the performance of 7 deep learning approaches in conjunction with spectrograms, power spectral density and raw audio representations. The approaches were evaluated on the mosquito wingbeat dataset, described in Sub-Section 2.5.4. The experiments were undertaken on an 80/20 train/test split and the results showed that the DenseNet121 approach in conjunction with spectrogram images provided the highest accuracy of 96%. The 2^{nd}, 3^{rd}, 4^{th} and 5^{th} most accurate approaches were also those that made use of spectrogram images. Fanioudakis et al. [31] went onto demonstrate the clustering ability of deep networks. Clusters can be formed from the latent features used by the final layers of the network to produce classifications. It was postulated that in an application setting, this would be useful in detecting outliers which may represent an unknown insect, or mechanical malfunction with the trap. Furthermore, Fanioudakis et al. [31] highlights the opportunity to visualise the extent to which regions from the input data are being utilised via saliency maps, an important tool in ensuring models are not learning from areas which represent arbitrary information, such as padding.

2.4 Hardware, Data Extraction And Processing



(a) UCR hardware

(b) TEIC hardware

Fig. 2.6 Hardware used to record the wingbeat motion of insects during flight.

All four datasets presented in Section 2.5 were produced using the hardware shown in Figure 2.6. The systems are similar in design and at their core, the recording apparatus consists of an infra-red beam aimed at a photo transistor. When the infrared light enters the phototransistor, current is allowed to flow across the base. This current is proportional to the luminance. Monitoring the current, allows the recording of fluctuations in the laser beam's energy at the base of the phototransistor. As an insect breaks or partially occludes the beam, fluctuations in the energy are recorded. Typically, these fluctuations are produced by the up and down motion of the wings. As a result, the information recorded bares a striking resemblance to that of the incidental sound made by the wings during flight. However, the mechanisms that trigger recording and the on-chip processing that takes place differ. These differences are briefly discussed in the Sub-Sections 2.4.2 and 2.4.1.

2.4.1 UCR Hardware

Shown in Figure 2.6a, the UCR hardware was used in the production of the InsectSound dataset, described in Sub-Section 2.5.3. The output from the photo transistor is passed through a small PCB, before being recorded by a Zoom H2 Handy digital recorder, at a sample rate of 16,000 Hz and saved as an MP3 file. Each MP3 was limited to six hours in length by the recorder's firmware, rather than disc space, and files began and ended simultaneously.

The raw MP3 files were fed through a detection algorithm to automatically extract potential insect sounds. The extraction was framed as a binary classification problem. A nearest neighbour classifier in conjunction with the Euclidean distance measure was used to determine whether data intervals contained insect sound or not. The data was processed via a sliding window that assessed 100 ms of data at a time (stated to represent the average length of time an insect would take to pass through the infra red beam). Each interval was transformed to the power domain prior to classification, as shown in Figure 2.7. The training data consisted of 10 manually labelled cases of each class. The size of the training set was chosen so as not to affect the speed

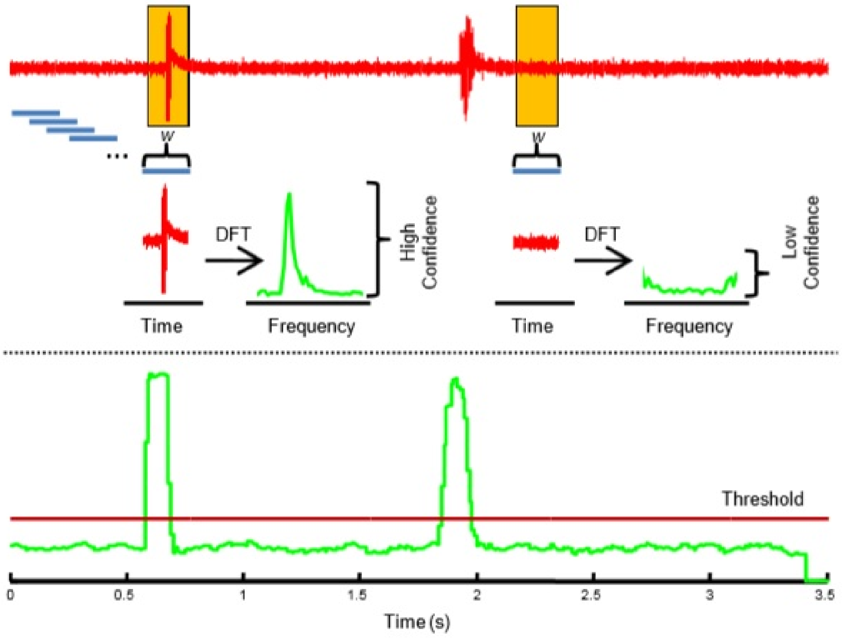


Fig. 2.7 illustration of the sliding window method used during the creation of the InsectSound dataset.[16]

of the algorithm, whilst retaining enough variability to maintain a high accuracy. The expected concept drift, due to temperature changes or the decline of battery power, was not expected to affect the performance of the system, due to a high signal to noise ratio. The UCR system also addresses the possibility of background noise contamination. American domestic electricity produces a 60 Hz signal, which can bleed into the recordings, due to inadequate filtering in power transforms. In order to negate this and obtain the best possible signal, the raw data was subjected to spectral subtraction of the background noise [9] [30]. Finally, each audio clip was centre padded with zeros in a 1 second WAV file and labelled according to its respective class.

2.4.2 TEIC Hardware

The optical components in this solution are infrared SFH4356 LEDs. They were deemed suitable for capturing data at the scale required, due to their 860 nm wavelength and raise time of 12 ns. The LEDs are arranged into four rows of six, as this combination provides equal current to LEDs, as well as providing a more consistent light distribution over a target area. The amplifier (OPA380), operational amplifier (AD8606), analogue to digital converter (ADS8863), demodulator (AD630) and output low pass filter (OPA1654), were all chosen as they were proven to be more reliable at the required sensitivity, or were found to produce the least noise in recording, when compared to other components of similar specifications. The light receiving configuration makes use of an optical light guide. The LEDs are directed towards a 2D polymer surface, commonly used in edge lit LCD screens. The surface directs the light down onto a 1D array of 22 photodiodes arranged in a parallel configuration. This arrangement provides a larger field of view (FOV) than previous iterations, consisting of either of 1D and 2D matrices of photodiodes. A large FOV is beneficial, as the insect occludes the LEDs for a greater period of time. However, it also increases the likelihood of interference from other insects and other light sources. Data recorded by this approach also produced spectrograms that were visually smoother than those of previous iterations using 1D or 2D photodiodes without the light guide. The system makes use of 2 buffers. The first buffer is used to compute the root-mean-square of every 128 sample (16 ms) window. If the RMS exceeds a predefined level, 5000 samples from the second buffer are committed to memory. The first 1000 samples precede the first buffers window and the remaining 4000 immediately proceed it. This ensures important information from the onset of the wingbeat is not lost. Samples are recorded at a 16-bit resolution at a 8,000Hz sample rate.

2.5 Wingbeat Datasets

In this section, the four publicly available pseudo-acoustic insect-centric datasets used in this Thesis are presented, these are summarised in Table 2.3. The Aphids

Table 2.3 A table summarising the insect-centric datasets used in Chapters 4, 5 and 6

Dataset	Classes	Instances	Attributes	Majority class
Aphids	*A. fabae*	2,036	5,000	0.4203
	D. platanoidis	3,192		
	M. persicae	19		
	P. testudinaceus	115		
	Pollen beetle	4,034		
	Ps. chrysocephala	194		
	R. padi	8		
FruitFlies	melanogaster	6,064	5,000	0.5305
	suzukii	10,142		
	zaprionus	18,312		
InsectSound	*Ae. aegypti* ♀	5,000	600	0.1000
	Ae. aegypti ♂	5,000		
	Dr. simulans	5,000		
	Mu. domestica	5,000		
	Cx. quinquefasciatus ♀	5,000		
	Cx. quinquefasciatus ♂	5,000		
	Cx. stigmatosoma ♀	5,000		
	Cx. stigmatosoma ♂	5,000		
	Cx. tarsalis ♀	5,000		
	Cx. tarsalis ♂	5,000		
MosquitoSound	*Ae. aegypti*	5,000	3,750	0.166
	Ae. albopictus	5,000		
	An. arabiensis	5,000		
	An. gambiae	5,000		
	Cx. pipiens	5,000		
	Cx. quinquefasciatus	5,000		

dataset was produced by Kirtsy Hassal from Rohamstead research[1] as part of ongoing work [44]; both the FruitFlies and MosquitoSound datasets were produced during the development of a low cost insect sensor at the Technological Educational Institute of Crete (TEIC) [75] by Professor Ilyas Potamitis. The MosquitoSound dataset was first published as part of a Kaggle competition[2], whereas the FruitFlies was donated directly. The InsectSound data was produced as part of an ongoing project at the

[1] https://repository.rothamsted.ac.uk/staff/841v5/kirsty-hassall
[2] https://www.kaggle.com/potamitis/wingbeats

University California Riverside (UCR) [16] and is part of the UCR TSC archive[3]. The Aphids, FruitFlies and the MosquitoSound dataset were all recorded with hardware developed at TEIC, shown in Figure 2.6b, whereas, the Insect Sounds dataset was recorded with hardware developed at UCR, shown in Figure 2.6a.

The problem, then, is to classify the sex, species and/or sub-species based on the resulting time series. This signal can then be interpreted as audio, and therefore a logical starting point is to apply approaches typically used in other audio classification problems. An additional benefit to the adoption of hardware based data collection is the inclusion of spatial and temporal information. It is well documented that the intensity of insect activity changes throughout the day and including this time of flight information has been shown to improve classification performance [16]. In the following sub-sections each dataset is discussed in detail. In doing so 3 data characteristics are provided: wingbeat frequency, wingbeat length and time of flight. The wingbeat frequency information is extracted via the Harmonic Spectral product approach, detailed in Section 5.3; the wingbeat length is defined as the number of consecutive samples that exceed a threshold value, where the threshold is set in order to negate background noise; and the time-of-flight data is a time-stamp captured by the hardware, in this case the date is discarded. In Chapters 5 and 6 an experimental evaluation of a range of classifiers on insect classification problems is provided, helping to indicate the most promising algorithmic approaches in this rapidly expanding field.

2.5.1 The Aphids Dataset

The Aphid dataset, summarised in Tables 2.3 and 2.4, was produced at Rohamstead Research by Kirsty Hassal. It is comprised of instances recorded in both laboratory and field conditions. In both cases, information regarding the time of flight was also

[3] http://www.timeseriesclassification.com

Table 2.4 Aphids dataset summary

#	Class name	Number
1	*D. platanoidis*	3,192
2	*M. persicae*	19
3	*P. testudinaceus*	115
4	*Pollen beetle*	854
5	*Ps. chrysocephala*	194
6	*R. padi*	8
7	*S. avenae*	270

recorded. Table 2.4 shows the label and number of instances in each class. Each of the 4,652 instances represent 620 ms of audio sampled at 8 Khz. In laboratory conditions, data collection was achieved in a semi-supervised fashion, where insects of each class were housed in separate perspex containers along with the recording equipment. Whereas recordings captured in the field were labelled by hand.

Figure 2.8 shows boxplots of the wingbeat-lengths of each class. The *M. persicae* and *R. padi* classes were omitted, due to their small class size. The wingbeat-length is expressed as number of samples and is defined as the interval between the first and last sample, greater than a predefined threshold. The threshold was set at 500, in order to negate background noise. All classes displayed similar minimum values, only varying by 11 samples. The *D. platanoidis*, *P. testudinaceus*, *Pollen beetle* and *S. avenae* classes also displayed similar medians, only varying by 120 samples. Overall, the plots show that there is considerable overlap between the wingbeat-length distributions of each class. The relationship between the median and mean values, show there is considerable skew towards larger values in all classes. Due to the nature of the recording setup, there are opportunities for erroneous readings. These include insects entering the beam simultaneously or in quick succession, insects flying along the beam, and insects resting on the perspex where the infrared beam intersect the housing. Undoubtedly, some of the outliers will reflect these issues. Interestingly, the *Ps. chrysocephala* displays a significantly smaller inter-quartile range than the other classes. Also, the mean and median values are 45 sample apart. This is unlikely to be entirely due to the smaller class size as it contains 79 more examples than the *P. testudinaceus* class and 76 less than *S. avenae* class. Both of which display a similar and larger skew, and significantly larger inter-quartile ranges.

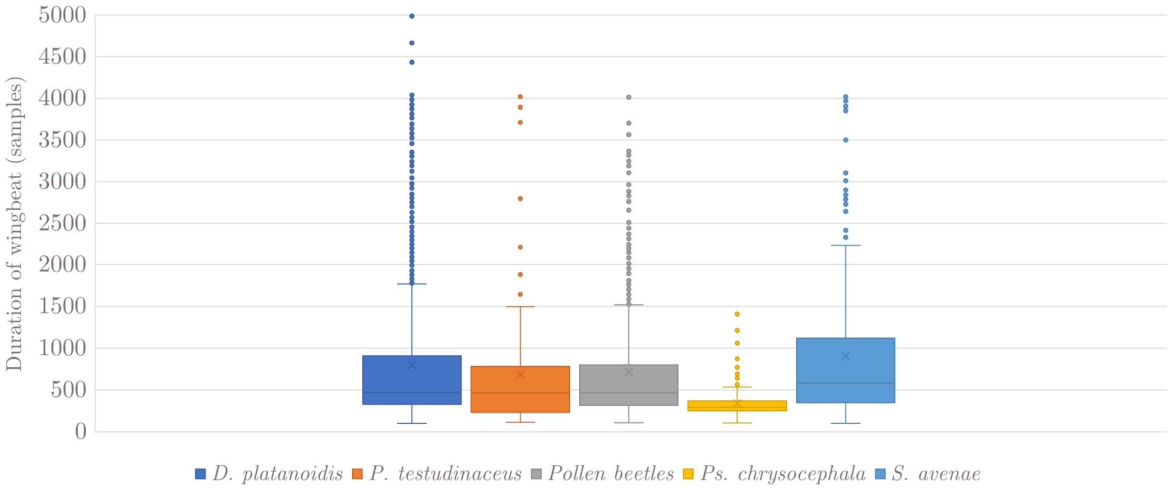


Fig. 2.8 A boxplot showing the distribution of wingbeat lengths in the Aphids dataset.

Figure 2.9 presents the time of flight information captured during recording. Again, the *M. persicae* and *R. padi* classes were omitted, as the small class size provides little meaningful information. The graph demonstrates that in all but the *Pollen beetle* class, there is activity throughout the 24 hour period, whereas, the *Pollen beetle* is inactive between 12pm and 3am. The *Ps. chrysocephala* appears to be the most consistently active insect, with one dip in activity at 6pm. In contrast, the *P. testudinaceus* and *S. avenae* insects have obvious periods of high activity. In both cases, this activity begins at 1pm and lasts until 11pm. Lastly, the *D. platanoidis* is the only insect to display two periods of higher activity. These periods are subtle and peak at 9am and 8pm. The inter-class activity throughout the day is high.

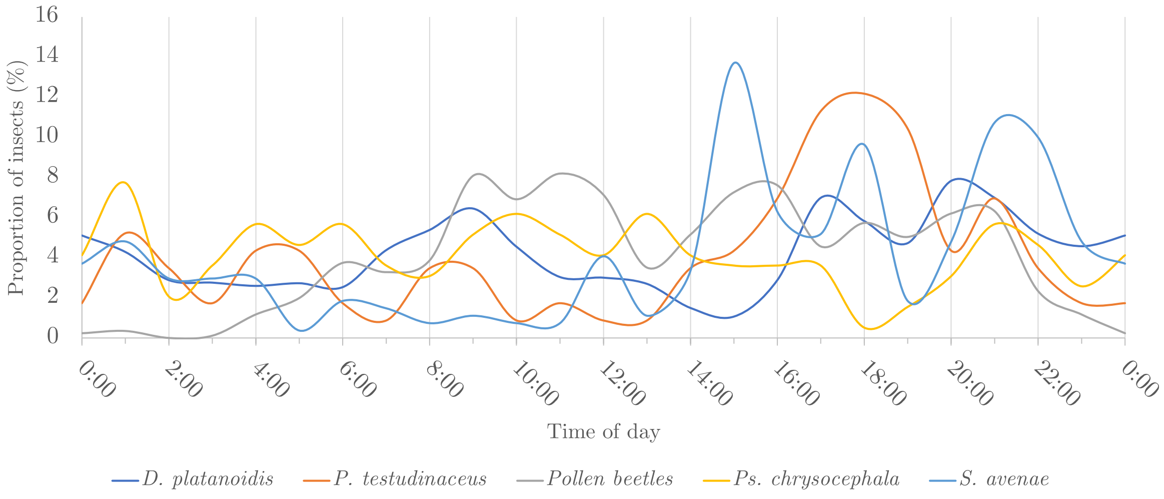


Fig. 2.9 A plot showing the activity of insects in the Aphids dataset throughout 24 hours.

Figure 2.10 presents the distribution of wingbeat frequencies from each class. The data is presented as the proportion of class size per frequency, for each class. The frequencies were extracted via the harmonic spectral product approach, detailed in Section 5.3. The graph shows there is significant overlap between the frequencies found in each class. This is most severe between the two beetle classes *Pollen beetles* and *Ps. chryocephala*. However, the difference in frequency between the peaks of two Aphid classes *D. platanoidis* and *S. avenae*, is 20 hz. Furthermore, the distribution of the third Aphid class *P. testudinaceus* is completely enveloped by the aforementioned two. Uniquely, the *P. testudinaceus* class displays a second region of activity centred around 550hz. This is unlikely to be the harmonic resonance of the primary peaks, which can be seen at 250 hz and 350 hz.

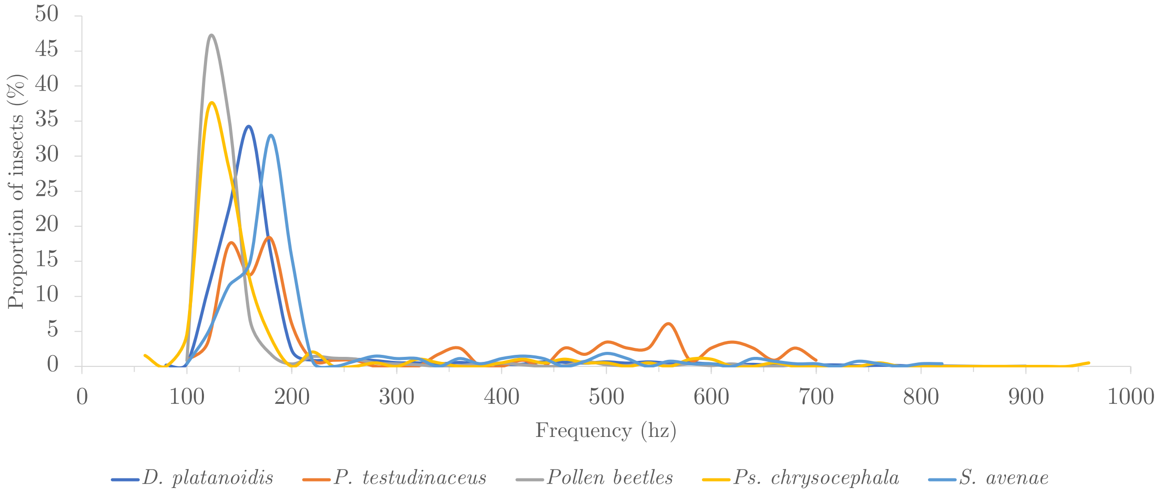


Fig. 2.10 A plot showing the distribution of wingbeat frequencies in the Aphids dataset.

Table 2.5 FruitFlies dataset summary

#	Class name	Number
1	Melanogaster	6,064
2	Suzukii	10,142
3	Zaprionus	18,312

2.5.2 The FruitFlies Dataset

The Fruit Flies dataset, summarised in Tables 2.3 and 2.5 was curated at TEIC as part of the development of a large aperture infrared sensor for capturing pseudo-acoustic insect data. The data was captured in the field. As a result, it is unknown which species of the Zaprionus genus is present. The proportion of male and females present in the dataset is also unknown. The data was captured through multiple, not necessarily contiguous, 24 hour periods. Table 2.5 shows the label and number of instances in each class. Each of the 34,518 instances represent 620 ms of audio, sampled at 8 Khz. The recording hardware automatically segments the data stream, based on a root mean square (RMS) threshold value, into 5,000 attribute intervals.

Figure 2.11 displays a summary of wingbeat lengths for each class. The wingbeat-length is defined as the size of the interval between the first and last sample that has an amplitude greater than a predefined threshold. In this case, the threshold value is set to an amplitude of 0.01 dB. Overall, the distributions of wingbeat lengths overlap significantly. The melanogaster and zaprionus classes display similar distributions: they are both slightly skewed toward longer wingbeats; the means and medians are 96 and 161 samples apart; the 3_{rd} quartiles are 75 samples apart and the interquartile ranges are 2,413 and 2,420 respectively. However, the suzukii class exhibits a more prevalent skew toward longer wingbeats, a similar 3_{rd} quartile value and a significantly smaller interquartile range of 1,996.

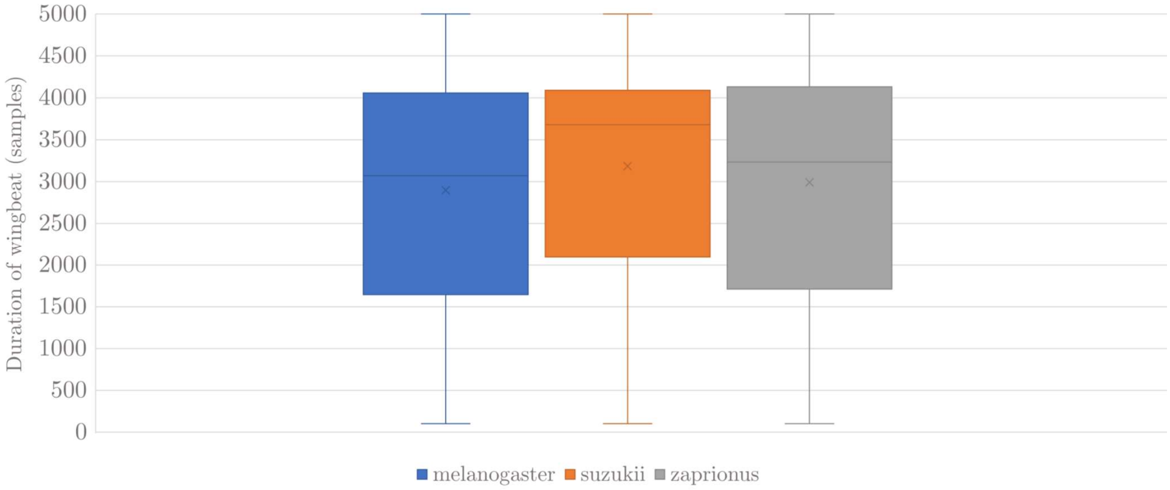


Fig. 2.11 A boxplot showing the distribution of wingbeat lengths in the FruitFlies dataset.

Figure 2.12 shows the distribution of insect activity through 24 hours for classes from the FruitFlies dataset. The data is presented as a proportion of the total class size and was derived from histogram bin counts at a 1 hour resolution. The graph shows that the melanogaster class is consistently active throughout the 24 hour period.

Potentially, there is a slight reduction in activity around 12am. The zaprionus class is most active between 7am and 12am, presumably dictated by dawn, and the suzukii class is most active between 12am and 7pm. Despite the distinct high activity periods, there is extensive overlap between the distributions. This is primarily due to the existence of baseline activity in all classes that is present throughout the 24 hour period.

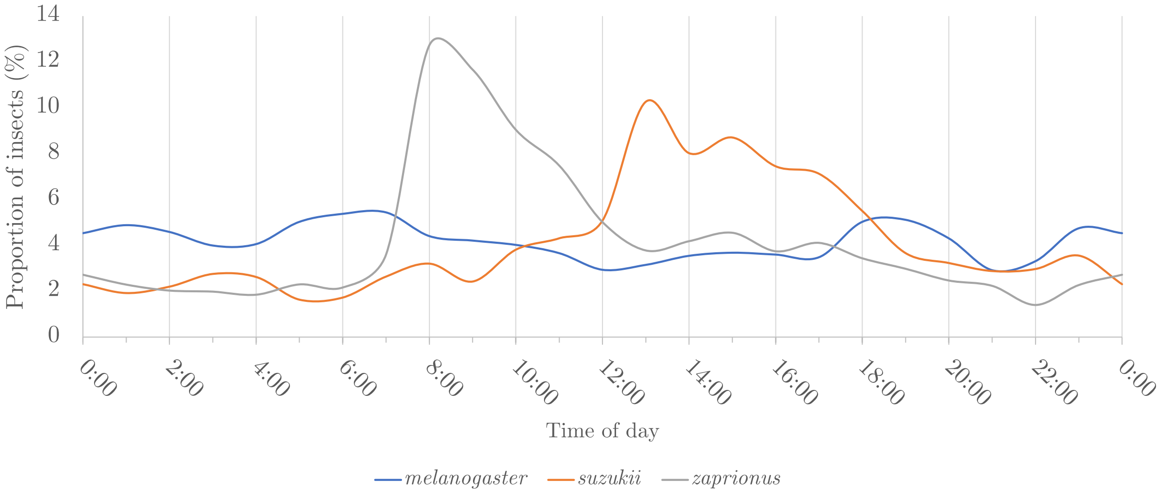


Fig. 2.12 A plot showing the activity of insects in the FruitFlies dataset throughout 24 hours.

Figure 2.13 shows there is a significant overlap between the distribution of wingbeat frequencies of each class. This particularly affects the suzukii and zaprionus classes. This overlaps suggests that the classes are morphologically similar. The graph also shows the variance in each class is low, 180 hz in the cases of melanogaster and suzukii and 70 hz in the case of zaprionus. The graph is void of the noise characteristic of erroneous recordings, examples of which can be seen in the frequencies greater than 250 hz in Figure 2.10. This indicates that the recording setup was effective at mitigating behaviours such as: multiple insect simultaneously entering the infrared beam; insects flying along the beam and insects resting on the perspex blocking the beam.

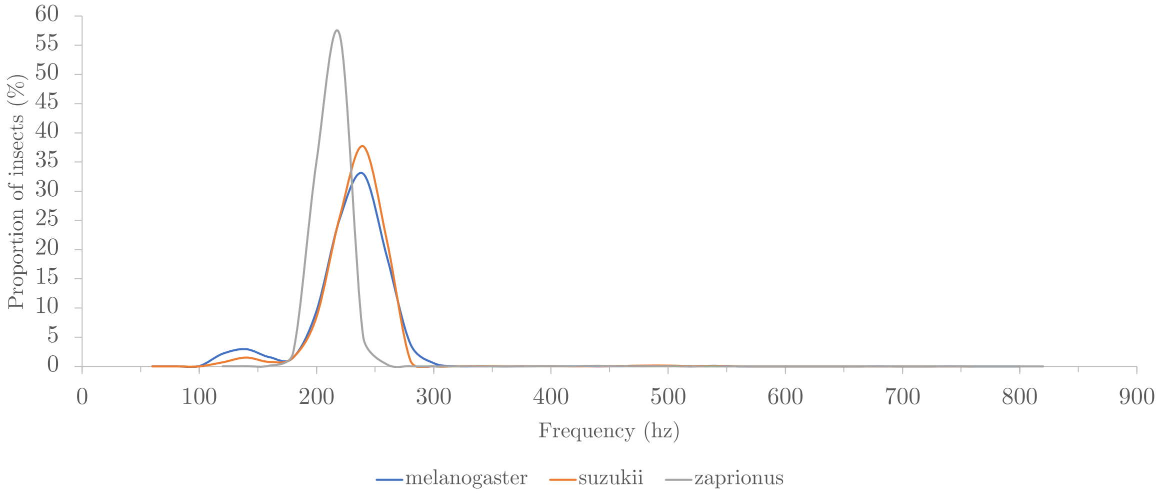


Fig. 2.13 A plot showing the distribution of wingbeat frequencies in the FruitFlies dataset.

2.5.3 The InsectSound Dataset

The InsectSound dataset was curated at UCR as part of work undertaken by Chen et. al. [16]. The dataset, described in Tables 2.3 and 2.6, consists of 10 classes of 5,000 instances. An in depth description of the recording hardware and data extraction process is presented in Sub-Section 2.4.1. This dataset consists of six species: two fly species, *Dr. simulans* and *Mu. domestica* and four mosquito species, *Ae. aegypti*, *Cx. quinquefasciatus*, *Cx. stigmatosoma* and *Cx. tarsalis* separated by sex.

All insects used for this data set were derived from wild colonies. The *Cx. tarsalis* species was collected at the Eastern Municipal Water District's treatment wetland in 2001 from San Jacinto, California. The *Cx. quinquefasciatus* species was collected

Table 2.6 InsectSound dataset summary

#	Class name	Number
1	*Ae. aegypti* ♀	5,000
2	*Ae. aegypti* ♂	5,000
3	*Dr. simulans*	5,000
4	*Mu. domestica*	5,000
5	*Cx. quinquefasciatus* ♀	5,000
6	*Cx. quinquefasciatus* ♂	5,000
7	*Cx. stigmatosoma* ♀	5,000
8	*Cx. stigmatosoma* ♂	5,000
9	*Cx. tarsalis* ♀	5,000
10	*Cx. tarsalis* ♂	5,000

in 1990 from southern California [37]. *Cx. stigmatosoma* was collected in 2012 from the Aquatic Research Facility at Riverside California. *Ae. aegypti* originated from 2,000 eggs delivered from Thailand [91]. *Mu. domestica* was collected in 2009 from San Jacinto, California and the *Dr. simulans* were collected in 2001 from Riverside, California.

The larvae of all the mosquito species used in this data set were raised in enamel pans under the same conditions. The temperature was kept at 27°C and they were exposed to a 16:8 hour light/dark cycle, with a one hour dusk/dawn period. All four species were fed the same 3:1 mixture of ground Rodent Chow and Brewer's yeast.

Both *Mu. domestica* and *Dr. simulans* larvae were subjected to 12:12 hour light/dark cycles, with no dusk/dawn period, at a consistent 26°C. However, the *Mu. domestica* larvae were fed a mixture of water, bran meal, alfalfa, yeast and powdered milk, whilst *Dr. simulans* larvae were fed a mixture of rotting fruit.

Figure 2.14 shows the distributions of wingbeat lengths for each of the 10 insect classes in the InsectSound dataset. The box plots show that there is considerable overlap between classes. In cases where species are sex separated, the mean and median is higher in the female sex. This is most clearly illustrated in the *Cx. quinquefasciatus* species. Female mosquitoes are typically larger than their male counterparts. This results in a lower wingbeat frequency and longer wingbeat length. All classes from the InsectSound dataset also exhibit a skew toward longer wingbeat lengths. This is likely to be a result of erroneous readings caused by: near simultaneous recordings, insects flying along the infrared beam, or insects resting at the point which the beam enters the enclosure.

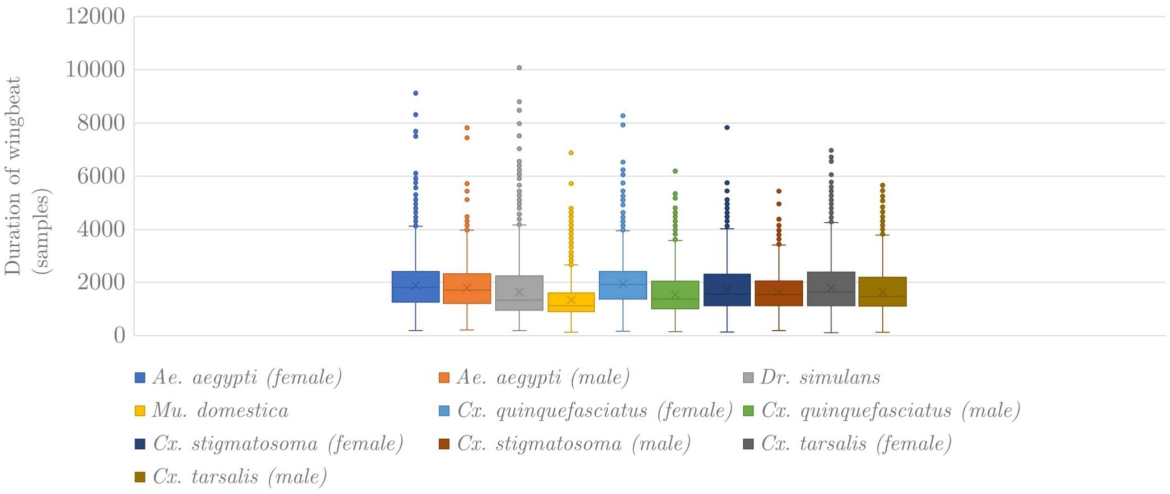


Fig. 2.14 A boxplot showing the distribution of wingbeat lengths in the InsectSound dataset.

Figure 2.15 demonstrates the flight activity throughout 24 hours for each of the mosquito classes from the InsectSound dataset. The *Dr. simulans* and *Mu. domestica* classes are both dinural, although *Dr. simulans* do peak in activity around sunset. The graph shows the classes grouped by sex and grouped by genus. Overall, the graph shows the crepuscular nature of mosquitoes with two peaks in activity during the twilight hours. The graph also highlights the difference between the typical activity of the *Culex* and *Aedes* genera. In all cases: the *Culex* genus maintains significant levels of activity throughout the night between the two twilight periods; the males are always more active than their female counterparts in the early twilight period and both males and females are almost entirely dormant throughout the day. However, the *Aedes* specie remains active throughout the day; is almost entirely dormant throughout the night and, the male and the females were equally as active in both the twilight periods.

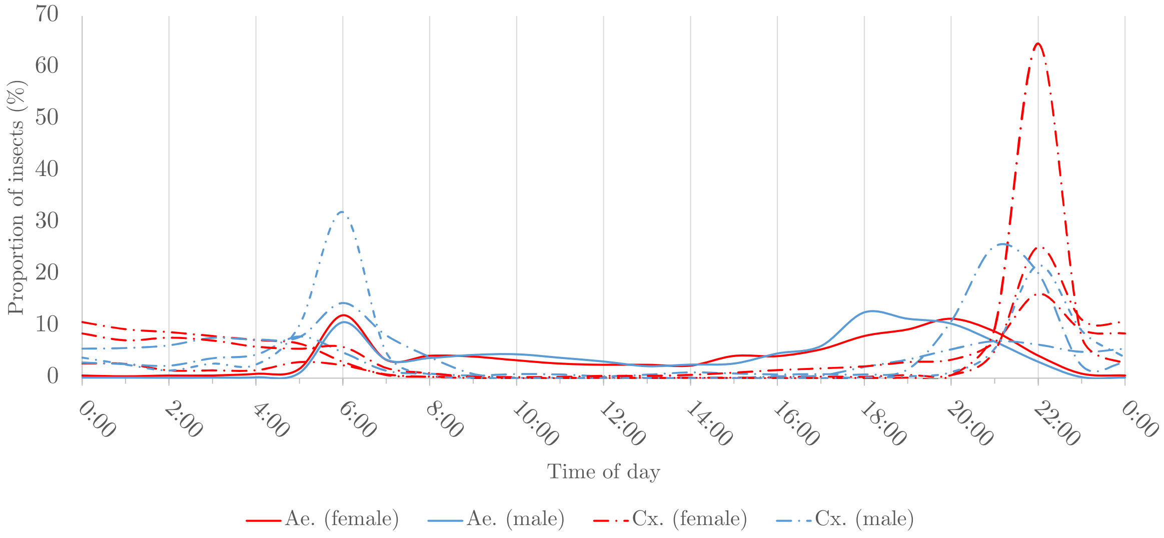


Fig. 2.15 A plot showing the activity of insects in the InsectSound dataset throughout 24 hours.

Figure 2.16 shows the distributions of wingbeat frequencies for the 10 classes in the InsectSound dataset. Classes drawn in either an orange or blue hue are mosquito classes, where the orange hue represents female classes and the blue hue represents male classes. The two genera are also visually separated, with the *Culex* genus being represented with a broken line. The graph shows that the classes fall into three distinct groups. The *Mu. domestica* and *Dr. simulans* are grouped together between 100 hz - 250 hz, with low frequency noise from the mosquito classes. The female mosquito classes are then grouped together between 250 hz - 450 hz. Finally, the male mosquito classes are grouped together between 450 hz - 750 hz. The graph does not show any significant difference in the distributions of frequency between the two genera present in the dataset. However, the split between male, female and other insects is significant. Isolation and analysis of the noise between 100 hz - 300 hz showed it was consistent with the expected frequencies of the first harmonic, or the body of the insects.

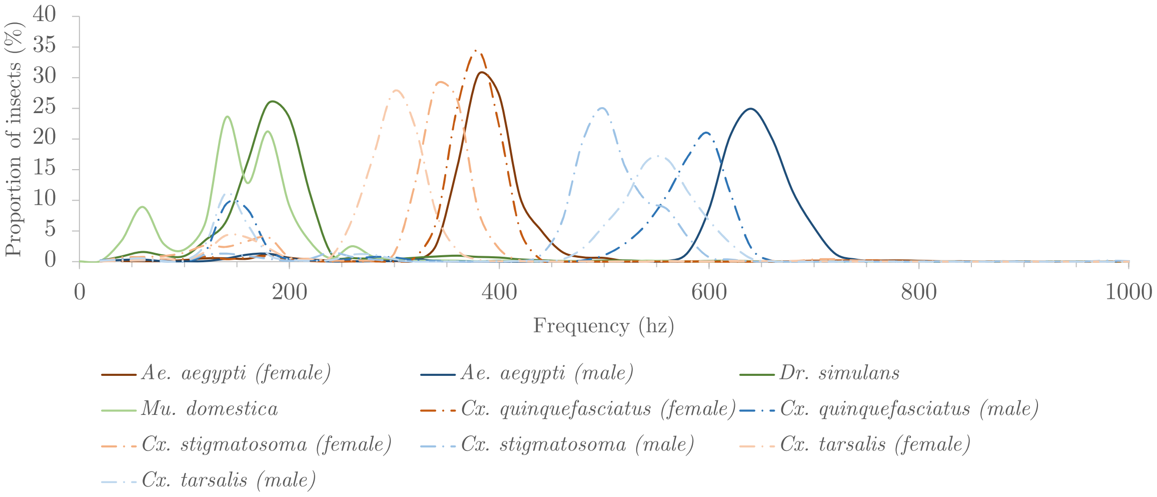


Fig. 2.16 A plot showing the distribution of wingbeat frequencies in the InsectSound dataset.

2.5.4 The MosquitoSounds Dataset

Table 2.7 MosquitoSounds dataset summary

#	Class name	Number
1	*Ae. aegypti*	5,000
2	*Ae. albopictus*	5,000
3	*An. arabiensis*	5,000
4	*An. gambiae*	5,000
5	*Cu. pipiens*	5,000
6	*Cu. quinquefasciatus*	5,000

The MosquitoSound dataset is comprised of six mosquito species from three genera. These are *Ae. aegypti*, *Ae. albopictus*, *An. arabiensis*, *An. gambiae*, *Cu. pipiens*, *Cu.*

quinquefasciatus. There is no differentiation between sexes. The raw data consists of 279,566 instances, split disproportionately between classes, where each instance represents 0.625 seconds of audio sampled at 8 kHz. A detailed description of the recording process is presented in Sub-Section 2.4.2. Tables 2.3 and 2.7 present the number of instances, attributes and sample rate of the dataset, after making these changes.

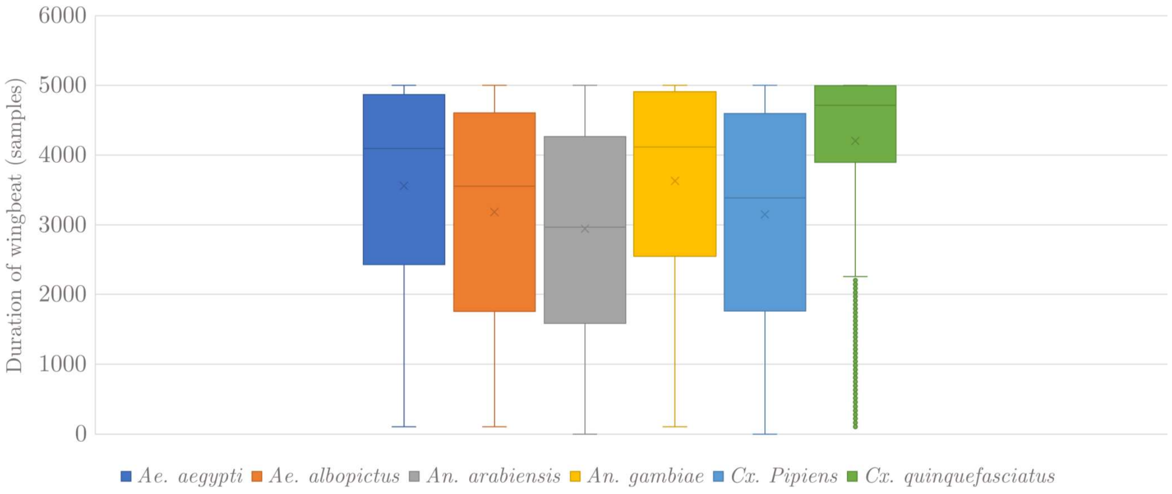


Fig. 2.17 A boxplot showing the distribution of wingbeat lengths in the MosquitoSound dataset.

Figure 2.17 shows the distribution of wingbeat lengths for each of the 6 classes from the MosquitoSound dataset. The plot shows: that there is significant overlap between the classes, for all insects but *Ae. arabiensis* the distribution of lengths is skewed toward larger wingbeat lengths, and that there is no obvious pattern between the distributions of insects from the same genus. However, the spectral information presented in Figure 2.19 suggests that both sexes are only present in the *Ae. albopictus*, *An. arabiensis* and *An. gambiae* classes.

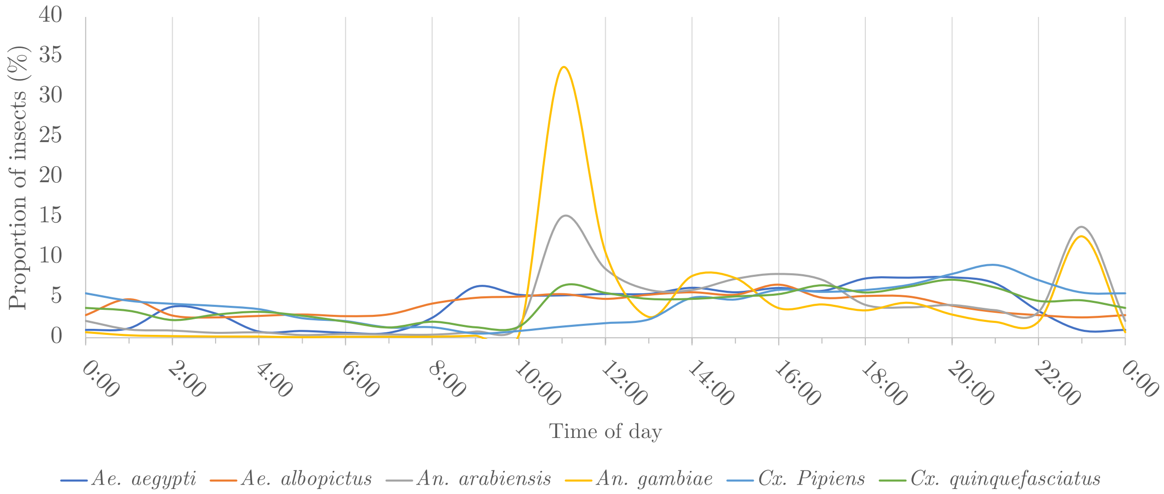


Fig. 2.18 A plot showing the activity of insects in the MosquitoSound dataset throughout 24 hours.

Figure 2.18 shows the proportion of insects from each class that were active throughout 24 hours. The data is the culmination of multiple, not necessarily contiguous days and was discretised to 1 hour bins. Similarly to Figure 2.15, the *Ae. aegypti* class is almost dormant throughout the night and active throughout the day. The Anopheles genus classes both exhibit strong crepuscular behaviour and the *Ae. albopictus* along with the classes from the Culex genus, appear to remain active throughout the entire 24 hour period, with a reduction in activity around the morning twilight period.

Figure 2.19 shows the distributions of fundamental frequencies from each of the six mosquito species in the MosquitoSound dataset. The fundamental frequencies were derived using the harmonic spectral product approach, detailed in Section 5.3. In the classes, *Ae. albopictus*, *An. gambiae* and *An. arabiensis* the graph shows strong evidence of both male and female mosquitoes being present. This is due to the presence of two sets of peaks, one at 550 hz and 800 hz and the other at 475 hz and 775 hz.

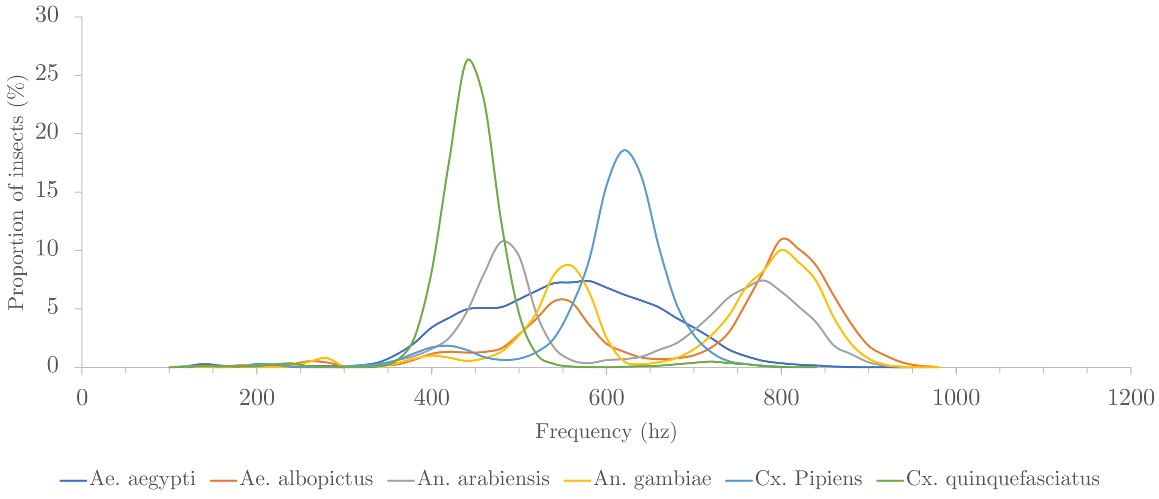


Fig. 2.19 A plot showing the distribution of wingbeat frequencies in the MosquitoSound dataset.

Interestingly, the distribution of the *Ae. aegypti* class is not consistent with the data shown in Figure 2.16. The distribution is wide, with one discernible peak at 575 hz.

2.6 Conclusion

In conclusion, this chapter introduces the "True Flies" order of insects, Culicidae, and presents some of the key characteristics of Mosquitoes. The Mosquito life cycle is briefly described the challenges associated with controlling Mosquitoes with pesticides are explained. In Section 2.3 a literature review is undertaken, where the key concepts and approaches applied to insect classification using wingbeat data are highlighted. In Section 2.4 The hardware used to record and curate large collections of insect wingbeat data are discussed. Throughout this the differences between the 2 pieces of hardware are highlighted. Lastly, in Section 2.5 the insect datasets that feature in experiments presented in Chapters 5 and 6 are described and discussed.

Chapter 3

Time Series Classification

3.1 Introduction

Time series datasets are comprised of instances consisting of naturally ordered values, commonly know as time series. A resulting assumption of this characteristic, is that the relationship between two neighbouring points may be leveraged to help form a prediction. This differs from traditional classification problems, where it is assumed that there is no information in the ordering of attributes. One common example of a time series data is audio, where the importance of the intrinsic order of attributes is obvious. However, there are less intuitive examples where continuous data is not recorded with respect to time. One example of this is in the encoding of outlines as time series, as shown in Figure 3.1. A second example is information expressed in the frequency domain, shown in Figure 2.19. Transforming audio data into the frequency domain often results in a better classification performance. This is typically down to two reasons: firstly, underlying structures in the data are often revealed and are less likely to be confounded by both high and low frequency noise and; secondly, the dimensionality of data can often be reduced. Typically the expected frequencies of the target data is significantly less than the Nyquist limit and can often be ignored. Often, transformation into the frequency domain is embedded into approaches, although it may be one part of a multi-stage process, in deriving features for training. In cases where approaches do not leverage features from the frequency spectrum, the transformation can be performed externally.

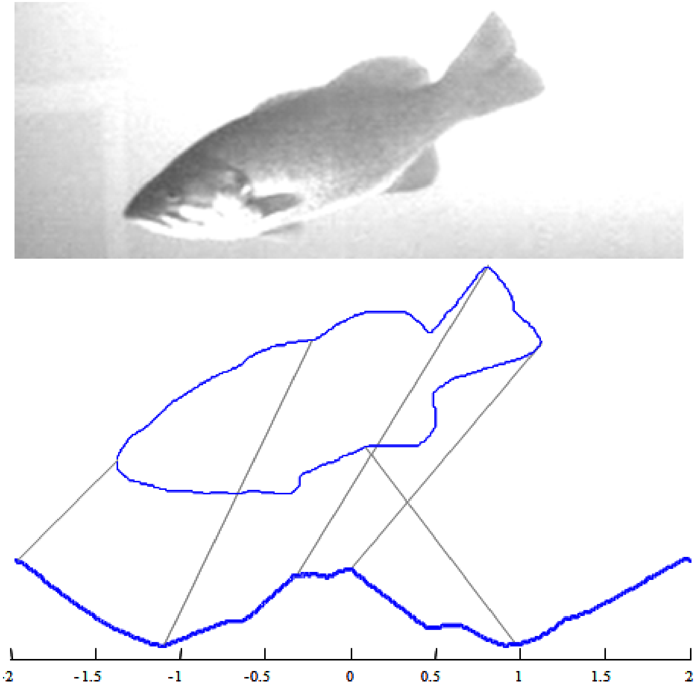


Fig. 3.1 An example of time series mapping from an image outline.

In Section 3.2, a description of time series data and the relevant notation is provided, followed by an introduction to the types of time series approach and explanation of their common characteristics. In Section 3.4 the tools used to compare the performance of approaches is presented. In Section 3.3 the approaches used throughout the experiments presented in Chapters 5, 4 and 6 are introduced and described.

3.2 Time Series Data

Given a time-series dataset, sometimes referred to as “problem”, $\mathbf{T} = \{\mathbf{t}^1, \ldots, \mathbf{t}^n\}$ each instance, $\mathbf{t} = \{x_1, \ldots, x_m, c\}$, consists of m attributes, typically consisting of real values, and a class value, $c \in \mathbf{C}$. The objective of a classification approach is to map test instances to their correct class. Typically, classification consists of two phases, training and testing. During the training phase, an internal representation of the relationship between the data and class labels is developed using the training data, the exact mechanism is unique to each approach. Once training is complete, the performance of the approach is evaluated by assessing the number of correct classification attempts using the test data. Prior to classification, a dataset, $\mathbf{T}$, is divided into non-intersecting training and testing subsets.

As mentioned previously, the training phase of each classification approach is unique. However, similarities in the underlying transformations allow us to group together approaches. In Section 3.3, approaches are separated into six categories: dictionary based, distance based, interval based, kernel based, shapelet based and hybrids. Dictionary based approaches, including the Bag Of SFA Symbols (BOSS) (see Sub-Section 3.3.1), make use of transformations such as Symbolic Fourier Approximation (SFA) and Symbolic Aggregate Approximation (SAX). Both of these transforms are forms of dimension reduction which utilise discretisation, as well as a further step in which the discretised data is represented as a combination of characters from an alphabet. Distance based approaches, such as Nearest Neighbour (NN) (see Sub-Section 3.3.2), assign class probabilities to a test instance, based on the k most similar instances from the training set. The most common measures of similarity are Euclidean distance (ED) and Dynamic Time Warping (DTW). Interval based approaches introduce variation by training each constituent classifier on a subset of the attributes available. An instance is considered a subsection of the data where, for all instances, a number of contiguous attributes are extracted. In some cases, data may be normalised in order to extenuate the importance of the location of information with respect to the instance length. In some cases, the selection of intervals is also used with boosting in order to incrementally improve the quality of the intervals. Interval approaches work best on data that is sparse and has a large number of attributes, where both the content of the interval and its location is indicative of class membership. Typically, transforms are also incorporated into interval approaches, for instance in the Time Series Forest (TSF) described in section 3.3.3 $\sqrt{m}$, intervals are selected and reduced to the mean, slope and standard deviation for each of the constituent decision trees.

In almost all cases, classification approaches employ transformations in an attempt to reveal class discriminate information. In some cases, approaches are developed with prior expertise incorporated to maximise their effectiveness in a specific domain. This is the case in the RISE classifier, which was developed specifically to be effective on audio datasets and as such, leverages transformations common in the signal processing domain.

3.3 Classification Approaches

In the following sub-sections the approaches used throughout the remainder of this thesis are described. They are organised into groups based on their internal transformation or structure, these groups include: Dictionary, Distance, Interval, Kernel, Shapelet, Hybrid and Deep Learning. The approaches included were chosen as they are thought to represent either the current ‘best in group’, such as InceptionTime and the Temporal Dictionary Ensemble (TDE) or, are well established benchmark approaches, such as Time Series Forest (TSF) or Nearest Neighbour (NN) with Dynamic Time Warping (DTW).

3.3.1 Dictionary based approaches

Dictionary based approaches extract, distretise and compress segments of data into words. The words of each instance are then expressed as a histogram of frequency allowing for comparison, typically via the Nearest Neighbour approach. A sliding window is used to parse instances and a word is formed at each window. The resolution of discretisation informs the word length, and the value at each bin is assigned a letter from a set of symbols of fixed size. These approaches work well on problems where the discriminant factor between classes is the frequency of patterns, as changes in the pattern frequency is reflected in the histograms.

The contracted Bag of SFA Symbols (cBOSS) [63]

cBOSS is a dictionary based approach. This approach summarises the frequency of words in each instance of a dataset, this is done by sliding a window of size w across each instance. cBOSS compresses each instance into a histogram of word frequency. Data within each window is transformed into the spectral domain before the first l terms are discretised at α resolution. A non-symmetric distance function and Nearest Neighbour classifier are then used to determine class probability. cBOSS is an ensemble, where the w, l and α parameters are set randomly for each repetition. This approach also sports the 'c' prefix denoting that a contracting mechanism is implemented, allowing for the control of training time.

The Temporal Dictionary Ensemble (TDE) [61]

Similarly to the cBOSS algorithm the TDE approach is also an ensemble of 1-NN classifiers, that employs a dictionary style approach derived from the SFA transform. However, the constituent classifiers within TDE are themselves an improved iteration of the BOSS approach, known as Improved Base BOSS (IBB) classifiers. These improvements include: using the histogram intersection measure in the place of the custom distance measure used by BOSS and including the frequencies of bi-grams, using the last non-overlapping window of each word additionally to the traditional words formed via the SFA transform. TDE also makes use of the spatial pyramid structure to better describe the temporal location of features, and provides a parameter to control their depth. Features are then weighted according to the depth from which they are derived, with features derived from the whole series weighted less. Information Gain binning is also introduced as an alternative method for generating breakpoints for discretisation in the SFA transform. The ensemble is generated via randomly selecting the parameters for the first 50 constituents. Thereafter, a Gaussian processes regresor is employed to the select combinations of parameters that might provide the best accuracy, based on the leave one out cross validation accuracies of previous constituents. This process is repeated until 250 constituents are built.

3.3.2 Distance Based Approaches

Distance based approaches aim to quantify the similarity between instances. Unlike other approaches the whole series is used. Typically they are used in conjunction with the Nearest Neighbour. The most common distance measures used are Euclidean Distance and Dynamic Time Warping, although there are other approaches, such as Manhatten Distance. These approaches perform well on problems where discriminant features span the whole series or where features exist out of phase.

Nearest Neighbour (NN)

The NN classifier is an example of a whole series approach. Here, a test instance is compared to each instance in the training set. Each training instance is assigned a distance that represents its similarity to the test instance. Class probabilities are assigned based on the frequency of each class appearing in the set of k most similar matches. The Euclidean Distance function (ED) and Dynamic Time Warping (DTW) are two common distance measures used in conjunction within the Nearest Neighbour construct to asses similarity between training and test instances.

Euclidean Distance (ED): Given a training instance, $\mathbf{t}^1$, and test instance, $\mathbf{t}^2$, of length, n. The Euclidean distance between them, d, is then squared sum of per-element differences and can be summarised as, (3.1)

$$d = \sum_{i=1}^{n} |\mathbf{t}_i^1 - \mathbf{t}_i^2| \tag{3.1}$$

Dynamic Time Warping (DTW): Given a training instance, $\mathbf{t}^1$, and a test instance, $\mathbf{t}^2$, of length, n. Let $\mathbf{M}$, be an $n \times n$ matrix for which each index represents the point wise distance between $\mathbf{t}^1$ and $\mathbf{t}^2$, such that $M_{i,j} = (\mathbf{t}_i^1 - \mathbf{t}_j^2)^2$. A valid warping path, $\mathbf{P} = (e_{i,j}^1, \ldots, e_{i,j}^s)$, traverses the matrix such that, $e_{i,j}^1 = M_{1,1}$ and $e_{i,j}^s = M_{n,n}$, whilst ensuring that, $0 \leq e_i^{k+1} - e_i^k \leq 1$ and $0 \leq e_j^{k+1} - e_j^k \leq 1$, for all $k < s$. The distance of a path, P_D, is defined as the sum of all elements in $\mathbf{P}$. Of all possible paths through the matrix, the one which minimises the total distance between $\mathbf{t}^1$ and $\mathbf{t}^2$ is returned.

The permitted deviation from the diagonal through the matrix $\mathbf{M}$ that the path can take, is typically parameterised. This warping window, r, is set to proportion such that $r = 1$ equates to there being no constraint on the amount on deviation and $r = 0$ enforces the path to follow the diagonal, in effect mimicking the Euclidean Distance measure.

3.3.3 Interval Based Approaches

Interval approaches use sub-sections of the dataset to build many weak learners. The size and location of intervals are typically chosen at random. The selected interval is extracted from all instances from the set, as a result the process can be interpreted as a form of attribute selection. In some cases, such as Time Series Forest, multiple intervals are selected and combined. These approaches work well on problems that have many attributes or feature lots of noise. This is because the selection of intervals is typically quick and the resultant feature vector is often smaller than the original series and, the collective effect of using many intervals minimises the chance of noise dominating classification.

Time Series Forest (TSF) [27]

TSF is another tree based interval ensemble. It is specifically designed to perform well on time series problems. For each tree in the TSF approach $\sqrt{m}$ random intervals are selected from the data. The mean, standard deviation and slope are then computed from each interval before being concatenated to form a new training set. These Time Series Trees (TST) also use a novel splitting criteria, namely Entrance Equation (3.2).

$$E = \Delta Entropy + \alpha \cdot Margin \tag{3.2}$$

The Entrance criterion was intended to improve on the standard ΔEntropy splitting criteria by incorporating a $Margin$ value. The $Margin$ is defined as the distance from a proposed split value and neighbouring instances. Additionally, TSF sets the number of threshold values considered for splitting, k. For each attribute at each node, the range between the minimum and maximum value is divided into k intervals on which to test. This removes the need to sort instances at each node, drastically improving the runtime complexity.

The Contracted Random Interval Spectral Ensemble (RISE) [56]

RISE is a tree based interval ensemble. Each tree in the ensemble is grown on spectral features derived from random Intervals. For each tree a random interval is selected from the training set. The interval is then transformed using a number of spectral approaches, including the Fast Fourier Transform (FFT) and Auto Correlation Function (ACF) independently. Class probabilities are then assigned as a proportion of base classifier votes. As the spectral component in the powerful HIVE-COTEv1 approach RISE was considered a sensible approach to focus on, the results of which are detailed in Chapter 4.

The Diverse Representation Canonical Interval Forest (DrCIF) [62]

The DrCIF classifier is an improvement on the CIF classifier introduced by Middlehurst et. al. [60]. The approach grows a forest of Time Series Trees [27] on a representative set of unique features. First order differences and the frequency spectrum are derived from the training set. This results in 3 training set representations. From each representation k random intervals are chosen. Then, a summary statistics from a pool of 29 are then randomly selected and applied to each of the k intervals. The features are then concatenated into a $3 \times k \times a$ series. This new training set is used to grow each Time Series Tree. Variation is injected into the forest by way of randomness in the position and length of the intervals selected. In order to ensure test instances are transformed correctly for each tree, the index and length of the intervals are recorded, along with which summary statistics were applied. The 29 statistics found in the pool are a combination of those found in the Catch22 [57] approach, along with the: mean; standard-deviation; slope; median; inter-quartile range; min and max.

3.3.4 Kernel based approaches

Kernel based approaches produce features from convolutions between generated kernels and instances. Typically, kernels are generated randomly from parameterised functions, but can also be predefined, if prior knowledge is available. The speed of the convolutions and base classifiers used, typically a regressor, enable the extraction of many features. In the case of The Random Convolutional Kernel Transform (ROCKET) approach, convolutions are compressed and expressed as 3 values. Kernel approaches are a relatively new type of approach, but have been shown to be effective across all types of problems [62].

The Random Convolutional Kernel Transform (ROCKET) [25]

ROCKET is an example of a pipeline. Where the features from one transformation process are used to train one classifier. This differs from approaches such as RISE, which is an ensemble of many classifiers and distinct transforms. ROCKET uses a novel feature derived from the output of convolving 10,000 randomly generated kernels on the training data, before then learning class boundaries using a ridge regression classifier. Each of the 10,000 kernels is applied to the training data. The maximum value and a novel feature, proportion of positive values (PPV), are then derived from each of the resulting feature maps. Although the kernels are random, the parameters used to create them are selected from the following spaces: the length, l, is selected such that, $l \in \{7, 9, 11\}$; the value of each weight, w_i, in the kernel is selected such that, $w_i \sim N(\mu, \sigma 2)$, where $\mu = 0$ and $\sigma^2 = 1$; dilation, d, is sampled from an exponential scale up to input length and the decision to apply padding to the series is made at random. If true, the series is zero padded, such that the centre value of the kernel is applied to every value in the series. The feature spaces for parameters were learnt on a ‘development’ subset of 40 randomly selected datasets from the UCR univariate time series classification archive. The PPV summarises the proportion of the series correlated to the kernel and was found to significantly improve classification accuracy. In effect, each instance in the dataset is transformed in to a 20,000 attribute series, consisting of max and PPV values. This transformed dataset is then used to train the ridge regression classifier.

Arsenal [62]

Arsenal is an ensemble of ROCKET classifiers. During development of HIVE-COTEv2.0, described in Sub-Section 3.3.6, it was established that the ridge regressor classifier used in ROCKET produced poor probabilities, despite the high accuracy. This was problematic as the CAWPE ensemble structure, described in section 3.3.6, leverages weighted probabilities, to assess the extent to which each classifier is confident in the prediction made. In order to mitigate this, an ensemble of smaller ROCKETs is used. The number of kernels used in each constituent is reduced from 10,000 to 2,000 and the number of ROCKETs in the ensemble is by default 25. A majority vote system is then employed to derive a probability distribution.

3.3.5 Shapelet Based Approaches

Shapelets are Sub-Sections of the data that best describe class memberships. Unlike Intervals, they are phase independent. Approaches evaluate many shapelets for each instance, evaluating their ability to split the data into the correct classes. Shapelet approaches work best on problems in which classes are defined by the existence of characteristic features, rather than the frequency of features.

Shapelet Transform Classifier (STC) [10]

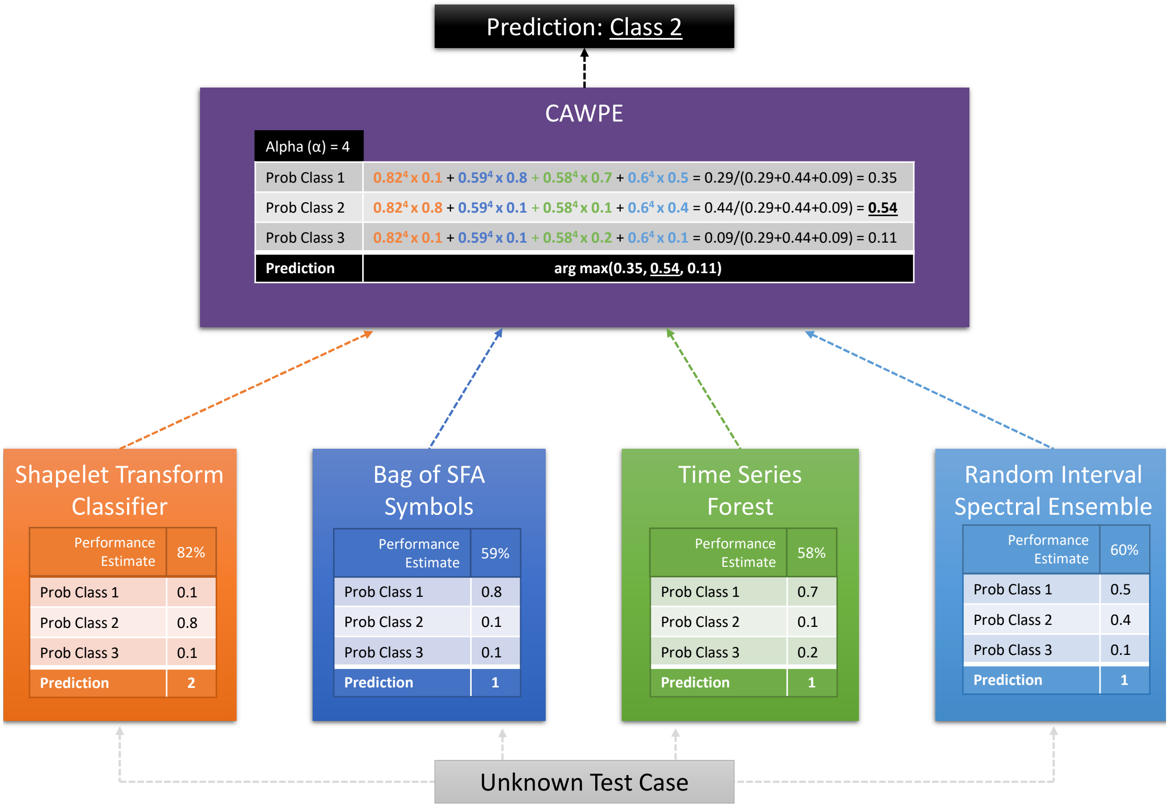
Shapelets are subsequences extracted from single instances within a training set. Selected randomly, the minimum distance between a shapelet and each training instance is stored and its ability to split classes effectively is assessed via information gain. The *k* best shapelets are retained in an ordered list. In order to find the minimum distance between a shapelet and an instance, the shapelet is slid along the instance, and the Euclidean Distance is computed before the shapelet is shifted by one attribute. The minimum distance is then retained. This approach is effective in determining phase independent features. The search space is fully enumerated if it is determined to take less that 1 hour, otherwise shapelets are chosen randomly. After the top *k* shapelets are determined, the training set is transformed into the shapelet space. The transformation expresses instances as the minimum distances from each of the *k* shapelets. As a result, the ith attribute of the jth instance in the transformed training set is the minimum distance between the ith shapelet and the jth instance from the raw training set. The benefit of this transformation process is that the features are agnostic of any classification approach.

3.3.6 Hybrid Approaches

Hybrid approaches ensemble over many different representations. The most well known hybrid approaches are the heterogeneous Hierarchical Vote Collective of Transformation-Based Ensembles (HIVE-COTE), which incorporate the predictions of approaches from different domains. However, there are homogeneous approaches, such as The Time Series Combination of Heterogeneous and Integrated Embedding Forest (TS-CHIEF) [83]. The mechanism of incorporating the predictive power of constituents differs for each approach. In HIVE-COTE the predictions of constituents are combined via the Cross-validation Accuracy Weighted Probabilistic Ensemble (CAWPE), whereas

TS-CHEIF embeds features in a decision tree. These approaches are designed to perform well in general. They are particularly useful when there is no prior knowledge of the target data's structure.

The Hierarchical Vote Collective of Transformation-Based Ensembles version 1 (HIVE-COTEv1) [56]



Prediction: Class 2

CAWPE

Alpha (α) = 4	
Prob Class 1	0.82^4 x 0.1 + 0.59^4 x 0.8 + 0.58^4 x 0.7 + 0.6^4 x 0.5 = 0.29/(0.29+0.44+0.09) = 0.35
Prob Class 2	0.82^4 x 0.8 + 0.59^4 x 0.1 + 0.58^4 x 0.1 + 0.6^4 x 0.4 = 0.44/(0.29+0.44+0.09) = **0.54**
Prob Class 3	0.82^4 x 0.1 + 0.59^4 x 0.1 + 0.58^4 x 0.2 + 0.6^4 x 0.1 = 0.09/(0.29+0.44+0.09) = 0.11
Prediction	**arg max(0.35, 0.54, 0.11)**

Shapelet Transform Classifier

Performance Estimate	82%
Prob Class 1	0.1
Prob Class 2	0.8
Prob Class 3	0.1
Prediction	**2**

Bag of SFA Symbols

Performance Estimate	59%
Prob Class 1	0.8
Prob Class 2	0.1
Prob Class 3	0.1
Prediction	**1**

Time Series Forest

Performance Estimate	58%
Prob Class 1	0.7
Prob Class 2	0.1
Prob Class 3	0.2
Prediction	**1**

Random Interval Spectral Ensemble

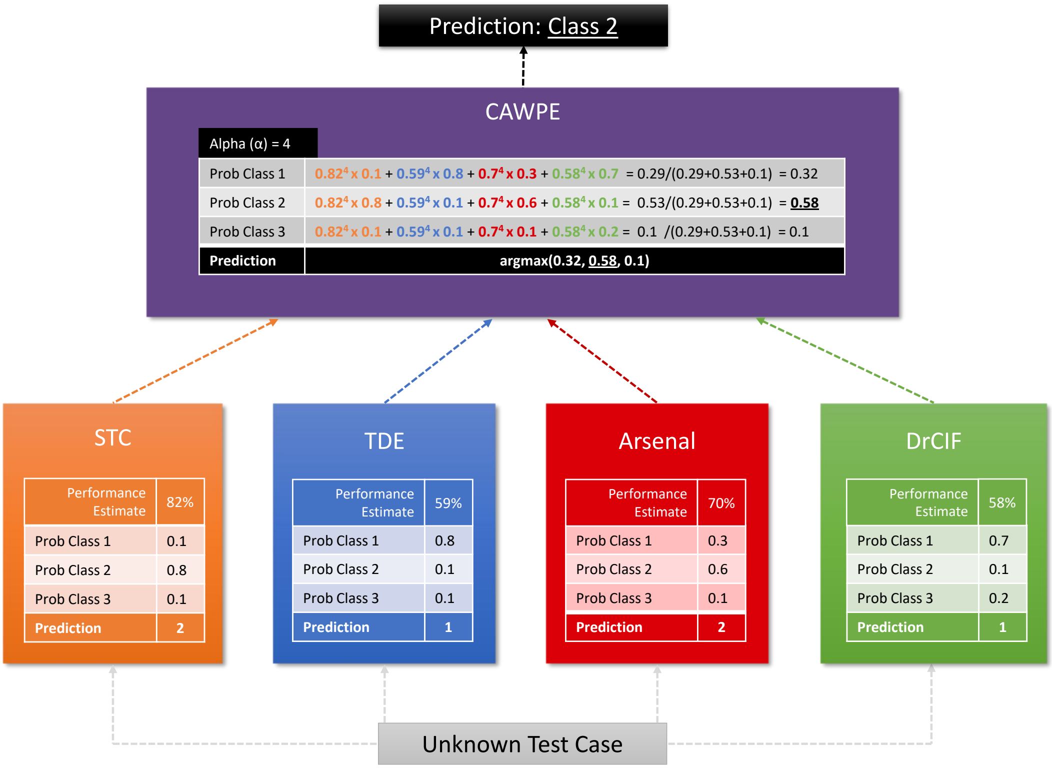
Performance Estimate	60%
Prob Class 1	0.5
Prob Class 2	0.4
Prob Class 3	0.1
Prediction	**1**

Unknown Test Case

Fig. 3.2 HIVE-COTEv1

HIVE-COTEv1, depicted in Figure 3.2 capitalises on the idea that the best approach for a problem is often found when considering the underlying patterns in the data. It was concluded that in order to produce an unsupervised approach, a heterogeneous group of classification algorithms should be selected. Furthermore, the approaches should, as much as possible, produce fundamentally different internal representations of the data. The HIVE-COTEv1 algorithm is formed of 4 modules: STC, cBOSS, TSF and RISE and utilises the Cross-validation Accuracy Weighted Probabilistic Ensemble (CAWPE) control structure, to combine the respective probability distributions. During training, each module derives a training accuracy. If the module is capable of providing this via some internal mechanism, it does so, otherwise the training accuracy is found through a 10-fold cross validation process. When passed a test instance HIVE-COTEv1's class prediction is a function of the internal modules. For each module, the respective train accuracy is raised to the power of, α, a parameter of HIVE-COTEv1. Its probability distribution is then multiplied by this value. The probability distribution of HIVE-COTE then becomes the summation of these values, represented as a percentage.

The Hierarchical Vote Collective of Transformation-Based Ensembles version 2 (HIVE-COTEv2) [62]



Alpha (α) = 4	
Prob Class 1	0.82^4 x 0.1 + 0.59^4 x 0.8 + 0.7^4 x 0.3 + 0.58^4 x 0.7 = 0.29/(0.29+0.53+0.1) = 0.32
Prob Class 2	0.82^4 x 0.8 + 0.59^4 x 0.1 + 0.7^4 x 0.6 + 0.58^4 x 0.1 = 0.53/(0.29+0.53+0.1) = **0.58**
Prob Class 3	0.82^4 x 0.1 + 0.59^4 x 0.1 + 0.7^4 x 0.1 + 0.58^4 x 0.2 = 0.1 /(0.29+0.53+0.1) = 0.1
Prediction	**argmax(0.32, 0.58, 0.1)**

STC

Performance Estimate	82%
Prob Class 1	0.1
Prob Class 2	0.8
Prob Class 3	0.1
Prediction	**2**

TDE

Performance Estimate	59%
Prob Class 1	0.8
Prob Class 2	0.1
Prob Class 3	0.1
Prediction	**1**

Arsenal

Performance Estimate	70%
Prob Class 1	0.3
Prob Class 2	0.6
Prob Class 3	0.1
Prediction	**2**

DrCIF

Performance Estimate	58%
Prob Class 1	0.7
Prob Class 2	0.1
Prob Class 3	0.2
Prediction	**1**

Fig. 3.3 HIVE-COTEv2

HIVE-COTEv2, depicted in Figure 3.3 is an updated configuration of the HIVE-COTEv1 ensemble approach consisting of the ‘best in class’ for the: shapelet; dictionary; kernel and interval type approaches. Of the four approaches included in HIVE-COTEv1, 3 have been replaced, with only the Shapelet transform classifier remaining. The dictionary approach cBOSS has been superseded by the TDE, described in Sub-Section 3.3.1. The interval approach TSF has been removed and the kernel approach Arsenal, described in Sub-Section 3.3.4, has been included. The interval approach RISE, has been superseded by the interval approach DrCIF, described in Sub-Section 3.3.3. The CAWPE control structure remains the same.

3.3.7 Deep Learning Approaches

Deep learning approaches consist of interconnected units called neurons. Each neuron sums its input, applying some bias, and maps the output of the sum to an activation function. Most network architectures are complex, consisting of multiple layers of many neurons. Some common archetypes include: Fully Connected Networks (FCNs) in which all neurons from one layer are connected to all neurons in the proceeding layer; Convolutional Neural Networks (CNNs) in which the multiple layers are fed the input at different resolutions. This is typically followed by a pooling layers to combine the resultant features before a number of fully connected layers; and Recurrent Neural Networks (RNNs) in which connections between neurons from a directed graph, this is done in order to extract temporal features. In all cases the weights between connected neurons are assigned randomly and updated via back propagation during training. The updates accentuate the importance of connections that positively effect training accuracy. These approaches perform best on large datasets where the variation amongst the instances of each class is sufficient to achieve a robust model.

Convolutional Neural Network (CNN) [96]

Convolutional networks are often used in image recognition, where kernels of different sizes or shapes are used to extract specific features from images. In the time series domain the approach is the same but 1-dimensional. For each layer the kernel is slid over the input Consisting of 4 layers in total, the code used to implement the CNN in this work can be found at the sktime-dl repository[1]. With respect to the work in this Thesis, the kernel size for both convolutional layers was set to 7 and the filter step size were 6 and 12 respectively. After each layer the convolution layer output is passed to a pooling layer, before being flattened prior to final output. In all cases a sigmoid activation function is used. The model was trained using the Adam optimiser and the mean squared error was used in the loss function.

Multilayer Perceptron (MLP) [92]

The MLP used in this work consists of three fully connected layers (1 input layer, 1 hidden layer and 1 output layer) consisting of 500 neurons. Each of these are coupled with a rectified linear unit to facilitate the MLPs non linearity and the dropout rates are 0.1, 0.2, 0.3 respectively. The code used can be found at the sktime-dl[2] repo.

Residual Network (RESNET) [45]

Technically defined as a deep neural network, the RESNET architecture is comprised of three blocks of three layers (1 convolutional layer, 1 batch normalization layer and 1 global pooling layer). The defining characteristic of this architecture is the way in which the blocks are interconnected. The residual connections between each block allow the gradients of previous layers directly through to later layers, reducing the

[1] https://github.com/sktime-dl/classifiers/deeplearning/_cnn.py

[2] https://github.com/sktime-dl/classifiers/deeplearning/_mlp.py

vanishing gradient effect. The default parameters were used in this work. The code used can be found at the sktime-dl[3] repo.

InceptionTime [33]

InceptionTime is a homogeneous ensemble of residual networks, each of which incorporate inception modules [88]. Each network in the ensemble consists of 2 blocks of 3 inception modules, followed by a global average pooling layer and finally a softmax layer. As shown if Figure 3.4 residual information is passed from the raw signal between the final layers of each block. Each network is initialised with random weights. In work undertaken by Fazwa et. al. [33] and Middlehurst et. al. [62], InceptionTimes performance with respect to accuracy was shown not to differ significantly from that of HIVE-COTEv1.

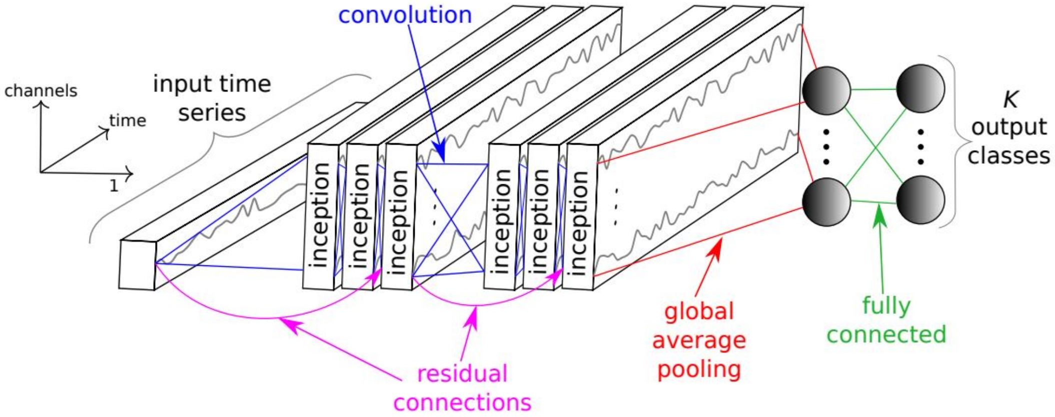


Fig. 3.4 Inception module [33]

3.4 Comparing Classifiers

In order to asses and compare the performance of many classifiers over multiple datasets, it is important to consider a range of descriptive statistics and statistical

[3] https://github.com/sktime-dl/classifiers/deeplearning/_resnet.py

tests. Typically, results are presented as simple statistics, such as the accuracy of a single experiment. In this Thesis the procedure outlined in [3] is adopted, allowing a deeper analysis of performance. Additionally to single statistics, significance testing, presented as critical difference diagrams, is also used. Finally, train and test duration of experiments and memory consumption is presented, As well as the derivations from contingency tables, such as F_1 score, sensitivity and specificity.

3.4.1 Critical Difference Diagrams

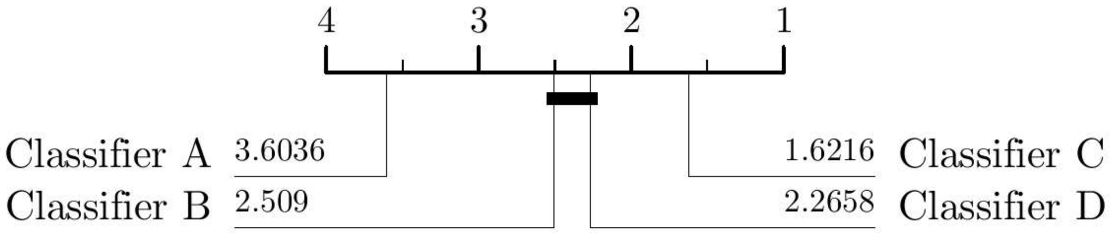


Fig. 3.5 An example of a critical difference diagram with 4 classifiers and 1 clique.

The approach adopted for producing critical difference diagrams is an adaptation of the process outlined by Demsar et. al. in [26]. Classifiers in the diagram are often displayed with their average ranks. Classifiers which do not differ significantly from one another are joined via a thick black line, and are said to be of the same clique. This is depicted in Figure 3.5 where classifier A is significantly worse than all others classifiers; classifiers B and D are significantly better than classifier A, significantly worse than classifier C but not significantly different from each other, and classifier C is significantly better than all other classifiers. Although this process is often undertaken to asses classifier test accuracy it is agnostic of the metric used.

The process is a two stage rank-sum style test, whereby a modified Friedman test is undertaken to establish if there are any significant differences between classifiers, followed by the Wilcoxon signed-rank tests with the Holm correction to establish where the differences are. In the first stage, the Friedman test is used to test the null hypothesis that the mean ranks of the classifiers are not significantly different. In this case, the alternative hypothesis states that in at least one case the mean rank of a classifier is significantly different from another classifier's mean rank.

Given M, a k by n matrix of classification accuracies, where k is the number of classifiers, n is the number of datasets and the notation $m_{i,j}$ corresponds to the accuracy of classifier i on problem j. The first step of stage one is to derive the k by n matrix R where $R_{i,j}$ corresponds to the rank of classifier i on problem j. Ranks are assigned according to the order of accuracies and in the case of ties, average ranks are assigned. In the second step, a vector of mean ranks $\hat{r}$ of length k is computed, where $\hat{r}_i$ corresponds to the mean rank of classifier i over all n problems.

To test the null hypothesis, that there is no significant difference between the average rank of classifiers, the Friedman statistic, (3.3) is used.

$$\chi_F^2 = \frac{12n}{k(k+1)} \left[\sum_{i=1}^{k} \hat{r}_i^2 - \frac{k(k+1)^2}{4} \right] \tag{3.3}$$

The Friedman statistic assumes a chi-squared distribution, with $k-1$ degrees of freedom. However, Demsar notes that the assumed chi-squared distribution results in a undesirably conservative outcome. As a result, the recommendation is to use a statistic presented in [18], which assumes an F-distribution with $k-1$ and $(k-1)(n-1)$ degrees of freedom, shown in equation (3.4).

$$F_F = \frac{(n-1)\chi_F^2}{n(k-1) - \chi_F^2} \tag{3.4}$$

Assuming the null hypothesis is rejected, the second stage of the process consists of multiple Wilcoxon signed-rank tests. This test requires that all classifiers are compared to each other. For each of the n comparisons the Wilcoxon's test statistic, w, is derived. Firstly, the absolute difference between and the associated sign are calculated for each problem, these differences are then ordered by rank. The test statistic, w, is the sum of products between the corresponding ranks and signs where the absolute difference is greater than 0. The z score for the ith comparison can then be calculated as, (3.5), where N_r denotes the number of non-zero rank values.

$$z_i = \frac{w_i}{\sqrt{\frac{N_r(N_r+1)(2N_r+1)}{6}}} \tag{3.5}$$

Finally, the Holm correction is used to adjust the α value, taking into account the number of tests being undertaken. This process controls the probability of family-wise errors occurring, ensuring that the probability of falsely rejecting the null hypothesis is less than α. As a result the test for rejection of the null hypothesis becomes (3.6).

$$z_i < \frac{\alpha}{m+1-i} \tag{3.6}$$

Where m represents the total number of comparisons and i represents the number of comparisons made so far.

3.4.2 Timing and Memory

In order to comment on the potential viability of approaches in the context of real world deployment, it is important to understand their behaviour entirely. The analysis of training time, test time and memory usage will inform how approaches may be used in an application setting. For example, it is more difficult to ensure a model reflects the environment if it takes many days to incorporate new data in the training set. Issues around memory consumption often impact the viability of deploying battery operated devices and the relationship between test time and test instance length informs the question of ‘real time’ classification. Often these characteristics are overlooked in favour of more traditional measures of performance, such as accuracy. Although they do not inform the usefulness of an approach in the same way, they do assist in providing context when discussing suitability.

3.4.3 Performance Statistics

The use of multiple performance statistics allows us to better understand how classification approaches perform on the application focused datasets. As previously mentioned, measures additional to accuracy, such as sensitivity, specificity and the F_1 score, allow us to provide a well rounded analysis of approaches. It becomes apparent why this information, in addition to accuracy is important, when considering that False positives (Type I errors), where cases are mistakenly classified as positive when that are in fact negative, and False Positives (Type II errors), where instances are mistakenly classified as negative, when they are in fact positive, may not carry an equal real world cost.

An example of common statistics and their derivation is shown in Table 3.1. Conveniently, it is possible to create contingency tables directly from the respective classifier/problem confusion matrix. This allows us to make multiple analysis, without the need to reformat datasets. For example, it is possible to asses a classifiers ability to accurately predict female mosquitoes in the InsectSounds dataset without re-formatting the dataset into a binary problem, by leveraging the confusion matrix of the full multi-class experiment.

Table 3.1 Common statistics found using a contingency table.[1]

		True condition			
		Positive	Negative	Prevalence $= \frac{\textit{True condition positive}}{\textit{Total population}}$	Accuracy $= \frac{\textit{True positive} + \textit{True negative}}{\textit{Total population}}$
Predicted condition	Positive	True positive	False positive Type I error	Positive prediction value $= \frac{\textit{True positive}}{\textit{Predicted condition positive}}$	False discovery rate $= \frac{\textit{False positive}}{\textit{Predicted condition positive}}$
	Negative	False negative Type II error	True negative	False omission rate $= \frac{\textit{False negative}}{\textit{Predicted condition negative}}$	Negative predictive value $= \frac{\textit{True negative}}{\textit{Predicted condition negative}}$
		True positive rate $= \frac{\textit{True positive}}{\textit{Condition positive}}$ False negative rate $= \frac{\textit{False negative}}{\textit{Condition positive}}$	False positive rate $= \frac{\textit{False positive}}{\textit{Condition negative}}$ True negative rate $= \frac{\textit{True negative}}{\textit{Condition negative}}$	Positive likelihood ratio $= \frac{TPR}{FPR}$ Negative likelihood ratio $= \frac{FNR}{TNR}$	F_1 score $= 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR}$

3.5 Conclusion

In this Chapter, Time Series data was described and defined, along with the notation used throughout the remainder of this thesis. Furthermore, the idea of an interval was also defined. This is particularly relevant as the work in the Chapter 4, focuses on improvements to the Interval based approach RISE. In Section 3.3 the approaches relevant to the experiments presented in Chapters 4 5 & 6 are described. Finally, in Section 3.4 the method of comparing classifiers based on the results they produce is outlined.

Chapter 4

The Random Spectral Interval Ensemble (RISE)

Contributing publications

- Flynn M., Large J., Bagnall T. (2019) The Contract Random Interval Spectral Ensemble (c-RISE): The Effect of Contracting a Classifier on Accuracy. In Hybrid Artificial Intelligent Systems. HAIS 2019. Lecture Notes in Computer Science, vol 11734. Springer, Cham.

4.1 Introduction

HIVE-COTEv1, is a heterogeneous ensemble. It consists of four modules: TSF, an interval approach that leverages summary statistics; RISE, an interval approach, that makes use of spectral transforms; BOSS, a dictionary approach and STC, a shapelet approach. These modules were deliberately chosen, as they represented the state of the art in each respective domain. The intention of this decision was to capitalise on the extended coverage provided by multiple domain experts, by intelligently combining the module probability distributions, such that modules that are likely to perform best on the test set have a greater impact on the probability distribution of HIVE-COTEv1. The probability distributions of HIVE-COTEv1 modules are combined via the CAWPE ensemble structure. The CAWPE ensemble structure is a hierarchical voting structure, where the vote of each module is modulated by a weight, and the weight of each module is assigned via the performance on the training data set. The extent to which each module contributes, is accentuated by raising its weight by the exponent α. In the test phase, the probability distribution per instance is then the normalised sum of the module probability distributions.

The motivation for including RISE in the HIVE-COTEv1 ensemble, is to account for problems for which discriminant features are most likely to be found in the frequency domain, or uncovered via autocorrelation [3]. These transforms are typically used on oscillatory data, such as audio, with success. This is because audio often comprises many complex sine waves that in the time domain, potentially mask discriminant features.

In Chapter 5 experimental results on two new large pseudo-acoustic insect-centric datasets are presented. HIVE-COTEv1, in conjunction with spectral features is found to produce the best accuracy overall and the lowest type II errors for female mosquito classification. However, the time required to train approaches on datasets, with large numbers of both instances and attributes was in some cases prohibitive. As a result, a method of controlling the time spent in the train phase was deemed necessary, we refer to this as “contracting”. Large training times are problematic as the number and size of datasets is expected to grow, and effective use of all available data is paramount in producing the most effective models. As such, being able to control the amount of time taken during training is useful. In this chapter, the issue is addressed, presenting two methods of controlling the train time of RISE: The “Naive approach” and the “Adaptive Approach”. The performance of these approaches is assessed in terms of contract adherence, effectively a measure of predictability. Furthermore, additional small changes are also introduced that increase performance. This is followed by a review of the core element of the cRISE framework, the “interval selection policy”. The interval selection policy denotes the method in which intervals are constructed. The method chosen can inadvertently skew the distribution of selected attributes over the course of training. In order to clarify the effect this has on the accuracy of cRISE, a review is undertaken in which 3 additional interval selection policies are evaluated: Policy 2, Policy 3 & Policy 4. Since the inception of cRISE, only a small number of prospective spectral transforms have been evaluated and the question of which spectral transforms perform best in the general audio classification space is very much open. Furthermore, alternate forms of combination such as: random selection; tuning mechanisms and ensembling are interesting opportunities to improve performance. Through the implementation of: the Audio Features (AF) transform, a collection of 6 common audio descriptors; a Spectrogram transform; and Mel Frequency Cepstral Coefficients (MFCCs) these issues are addressed. Transform performance is assessed individually, via 3 tuning methods and finally through the random selection and CAWPE combination methods.

The remainder of this chapter is laid out as follows: in Section 4.2 the contracting and the contracted RISE approach, cRISE is introduced. In Section 4.3 a review of interval selection policies is undertaken and their impact on the accuracy of cRISE assessed. In Section 4.4 3 proposed spectral transforms to the cRISE approach are presented, and their individual performances, tuning methods and combination methods discussed, before presenting an improved cRISE configuration, Finally, some conclusions are presented in ection 4.5.

4.2 Improving Usability Via Contracting

Data sets of increasing size are now common within machine learning. Big data undeniably has its benefits. However, as advancements in processing capabilities begin to slow and the complexity of algorithms increase, we are often faced with more data than we are capable of processing. Even with the rise in popularity of cloud computing platforms and high performance computer facilities, it often becomes infeasible to construct a full learning model on all of the available data. A particularly common area in which the problem arises is the spectral/audio domain. This is typically down to the sample rate used to record data. Consider that the standard audio sample rate is 44.1kHz. Creating models from audio data requires either extensive bespoke preprocessing, or adaptations of the learning algorithms to compensate for the volume. We do not look to challenge or reaffirm the traditional, volume of data/increase in accuracy paradigm. Instead, we aim to investigate the relationship between reduced train time and accuracy, assuming a fixed volume of data.

All experimental processes strive to make complete use of the training set and in ideal conditions, this will always be preferable. However, experience has shown us that working with large datasets can cause extreme training and test times. When working through these problems, it has become apparent that very little research has been undertaken in understanding how reduced training time affects accuracy. Homogeneous ensembles typically require a large number of trees to be effective. The most basic way of managing training times is simply to build base models until the time has expired. However, for very large problems, this may result in very small ensembles. The Random Interval Spectral Ensemble (RISE) is a Time Series Classification (TSC) algorithm that uses spectral features. It selects a different random interval of the series for each base classifier, then calculates spectral coefficients to be used as features. For large problems, if intervals close to the full series length are selected it is possible to use all available computation on very few models. To compensate for this, an investigate into whether it is possible to predict the run time is undertaken, this prediction can then be used to guide the interval sampling, ensuring a minimum size ensemble.

RISE draws on ideas from tree-based ensembles such as random forest [12] and the TSC interval feature classifier TSF, described in Sub-Section 3.3.3. The base RISE algorithm is described in Algorithm 1. It shows that the first tree in RISE is a special case that uses the whole series for spectral transformation, this step is included for continuity with the previous spectral classifiers used in The Collective of Transformation-Based Ensembles [5] (COTE) classifier which only used the whole series. We then build r random decision trees on the spectral transform of unique random intervals selected from the data. This is similar to TSF, however, key difference is that TSF uses time domain features by calculating the mean, variance, and slope of $\sqrt{m}$ intervals, where RISE extracts spectral features.

Algorithm 1 BuildRISE(Training data *train*, number of classifiers *r*, minimum interval length *minLen*)

```
Let $\mathbf{F} \leftarrow < F_1 \dots F_r >$ be the trees in the forest.
Let $m$ be the length of series in $train$
$wholeSeriesFeatures \leftarrow$ getSpectralFeatures($train$)
buildRandomTreeClassifier($F_1$,$wholeSeriesFeatures$)
for $i \leftarrow 2$ to $r$ do
    $startPos \leftarrow$ randBetween($1, m - minLen$)
    $endPos \leftarrow$ randBetween($startPos + minLen, m$)
    $train \leftarrow$ removeAttributesOutsideOfRange($train$,$startPos$,$endPos$)
    $intervalFeatures \leftarrow$ getSpectralFeatures($train$)
    buildRandomTreeClassifier($F_i$,$intervalFeatures$)
```

RISE uses several forms of spectral features: the power spectrum, the autocorrelation function, the partial autocorrelation and the autoregressive model. New classes are classified using a simple majority vote. Further details can be found in [56]. The run time for transforming a series is quadratic in the interval length.

4.2.1 The Contracted RISE Algorithm (cRISE)

In many areas, it may be advantageous or even necessary to constrict the run time of a classification algorithm. Generally, it is not well understood how long classification algorithms take to run for any given problem. Run time is of practical importance when considering which algorithm to use, or how much preprocessing to perform. This is of particular relevance when using cloud services, where the computation is charged per time period, situations in which there is a hard deadline, or where there is a limit on how long a process is allowed to run. Two solutions to these problems are check-pointing, periodically saving a partial version of the classification model to disk; and contracting, limiting the amount of computational time an algorithm is allowed. When used together, they make a classifier more flexible and useful to the practitioner. Check-pointing RISE is simple, especially with a Java implementation; where the constructed trees can simply be serialised at certain points and RISE adapted to allow the loading from file. Contracting is also simple: trees can be built until we run out of time, or reach the maximum number. However, this simple contracting approach can result in very small ensembles, if the series are very long. In this Thesis an adaptive scheme is proposed that avoids this problem, by dynamically estimating the build time for each particular tree.

Algorithmic Improvements

cRISE is a tree based interval ensemble. Each tree in the ensemble is grown on spectral features derived from random Intervals. For each tree a random interval is selected from the training set. The interval is then transformed using the Fast Fourier Transform (FFT) and Auto Correlation Function (ACF) independently. Prior to transformation into the spectral domain the interval is padded to the next power of 2 using the mean value of the interval. The output of the FFT is then concatenated with the first 100

ACF coefficients to form a new training set. Class probabilities are then assigned as a proportion of base classifier votes.

A number of small but influential changes are implemented in cRISE. These were implemented with the aim of not significantly decreasing accuracy, whilst drastically improving runtime. These changes are outlined below and a more thorough description is provided in Algorithm 3.

RISE uses power spectrum (PS), autocorellation function (ACF), partial autocorelation function (PACF) and autoregressive model (AR) features over each interval. It was found that combining them created a more accurate classifier than just using one set [56]. However, the disadvantage is that although the PS can be found in $O(nlog(n))$ time if the series length is a power of 2, there is no easy way to do this for the PACF and AR terms. Hence cRISE does not derive PACF or AR features, and only selects intervals that are a power of 2. An interval is still selected randomly, but now it is rounded to the nearest power of two. To correct for intervals exceeding the series length, the interval is divided by 2, ensuring a valid interval and favouring shorter intervals.

Timing Models

The following Sub-Sections introduce two methods of controlling the train time of RISE. We refer to these approaches as timing models. The Naive Model, describes the obvious and simplest approach of using a timer to end training after the contract is up and the Adaptive Model introduces a Linear Regression Model to map the independent, length of interval, variable to the dependant, time taken to build tree, variable. The results of these 2 approaches are then presented and discussed in Section 4.2.2.

The Naive Model The simplest way to limit the train time of tree based ensemble, is to simply set a timer and keep adding trees until the contract is met, or a maximum number of trees have been built. This is described in Algorithm 2 where the timer is initialised and then started before any of the trees are built. Then, at each iteration of the for-loop the elapsed time is compared to the contract time. If the elapsed time it greater than the contract time the build process is ended, irrespective of whether i equals r.

Algorithm 2 Build cRISE_Naive(Training data *train*, number of classifiers *r*, minimum interval length *min*)

```
Let $F \leftarrow< F_1 \dots F_{500} >$ be the trees in the forest.
Let $m$ be the length of series in $train$
startForestTimer()
for $i \leftarrow 1$ to 500 AND queryForestTimer() do
    $validLengths \leftarrow$ getValidPowersOf2($min$, $instanceLength$)
    $randomLength \leftarrow$ randBetween(maxValue($validLengths$)/2)
    $r \leftarrow$ findClosest($validLengths, randomLength$)
    $startPos \leftarrow$ randBetween$(1, m - r)$
    $interval \leftarrow$ removeAttributesOutsideOfRange($train, startPos, r$)
    $intervalFeatures \leftarrow$ getSpectralFeatures($interval$)
    buildRandomTreeClassifier($F_i$,$intervalFeatures$)
    updateTimer()
```

Adaptive model cRISE performs two transformations: A Discrete Fourier Transform (DFT) to find the power spectrum and construction of the Autocorrelation Function (ACF). With the simplest implementation, each of these is O(r^2), where r is the interval width. The efficiency of the Fourier Transform can be improved to O($rlog(r)$) by using the FFT. To gain the full benefit, cRISE is restricted to intervals of length the power of 2. However, the best average case complexity for the ACF is O(r^2). Hence, the transformations will dominate the runtime in relation to the decision tree. Therefore, the runtime t for a single member of the ensemble of interval length r can be modelled as:

$$t = a \cdot r^2 + b \cdot r + c.$$

Given the contract time t and the constant factors a, b and c, the positive root of the quadratic can be used as the maximum allowable interval for the tree. The quadratic terms will of course be both problem and hardware dependent. Hence, an adaptive algorithm is used to learn these parameters. For each tree built, the selected interval is recorded along with the observed run time. Using this data, a least squares linear regression model can be fit and updated. For clarity, let $x_1 = r^2$ and $x_2 = r$, our dependent variable matrix is then:

$$X = \begin{bmatrix} 1, x_{11}, x_{12} \\ 1, x_{21}, x_{22} \\ \ldots \\ 1, x_{k1}, x_{k2} \end{bmatrix}$$

the estimates of the parameters are $B = (\hat{a}, \hat{b}, \hat{c})^T$ and the response variable is $Y = (y_1, y_2, \ldots, y_k)^T$. The least squares estimates are then:

$$B = (X^T X)^{-1} X^T Y.$$

Since $(X^T X)$ is based on sums of squares, there is no need to recalculate it from scratch each time. It is also possible to update $(X^T X)^{-1}$ online with the Sherman-Morrison formula [82] for further improvements in efficiency. After the construction of each tree, the remaining contracted time t is updated, the coefficients $t = \hat{a} \cdot r^2 + \hat{b} \cdot r + \hat{c}$ re-estimated, and a new maximum allowable interval r calculated. This is used as the maximum for the next iteration.

The incorporation of the ability to model the time taken to build each tree is shown in algorithm 3. The pseudocode shows that a timer is started prior to entering the main for loop where an interval is randomly chosen, transformed and used to build a random tree. The time taken to undertaken to build and the length of the interval generated is then used to update coefficients in a linear regression model. This model is then used to find the maximum interval length that could be used to build the next tree without breaching the contract.

Algorithm 3 Build c-RISE_Adaptive(Training data *train*, number of classifiers *r*, minimum interval length *min*)

```
Let $\mathbf{F} \leftarrow < F_1 \ldots F_{500} >$ be the trees in the forest.
Let $m$ be the length of series in $train$
startForestTimer()
for $i \leftarrow 1$ to 500 AND queryForestTimer() do
    startTreeTimer()
    buildAdaptiveModel()
    $max \leftarrow$ findMaxIntervalLength()
    $validLengths \leftarrow$ getValidPowersOf2($min$, $max$)
    $randomLength \leftarrow$ randBetween(maxValue($validLengths$)/2)
    $r \leftarrow$ findClosest($validLengths, randomLength$)
    $startPos \leftarrow$ randBetween($1, m - r$)
    $interval \leftarrow$ removeAttributesOutsideOfRange($train, startPos, r$)
    $intervalFeatures \leftarrow$ getSpectralFeatures($interval$)
    buildRandomTreeClassifier($\mathbf{F}_i$,$intervalFeatures$)
    $y \leftarrow$ queryTreeTimer()
    updateAdaptiveModel($r, y$)
```

4.2.2 Results

In the following Sub-Sections the performance of the: RISE and cRISE approaches, and the Naive and Adaptive timing models are presented discussed. In both cases the approaches are directly comparable and, in both cases we show that the contributions laid out in this Thesis results in a superior performance.

RISE vs c-RISE

RISE has been shown to be significantly more accurate than other spectral based approaches on the TSC archive data and on simulated data [3] and therefore was selected as the spectral component for HIVE-COTEv1. However, RISE is computationally expensive, since each transformed series is based on an O(r^2) operation (finding the

Table 4.1 Summary of the RISE and cRISE performance over all 85 UCR datasets.

	Acc.	Std. Dev. (problems)	Std. Dev (folds)	Min. Acc.	Max. Acc.	Largest win	Mean win Acc.	Wins
RISE	0.7983	0.1422	0.0262	0.3569	0.9990	0.2169	0.0327	50
cRISE	0.7996	0.1419	0.0258	0.2928	0.9946	0.4559	0.0500	34

PACF), where r is the series length. This is further impacted by the derivation of auto regressive and spectral features. In a summary of experiments over 85 datasets from the UCR archive, it was concluded that these features do not significantly affect accuracy. This is illustrated by Figure 4.1, which shows that cRISE and RISE share the same clique in the critical difference diagram and Table 4.1, which further summarises the performances of the two approaches. Table 4.1 These results show that cRISE produces a marginally better mean accuracy (denoted by Acc in the table); less variation in accuracy over problems (denoted by Std. Dev. problems) and across folds (denoted by Std. Dev. folds); and a higher mean accuracy over all the problems in which it does best (denoted by Mean win acc). The table also shows that the worst and best performances of RISE are better than cRISE. Furthermore, computation of these features represent significant complexity in the algorithm and as such they have a detrimental effect on runtime. Table 4.2 presents both the mean and median training time for RISE and cRISE over 85 datasets from the UCR archive and highlights how significant the difference in computation time is.

Fundamentally, cRISE behaves in the same manner as RISE when not under contract. This allows us to attribute any changes in runtime or accuracy to the removed transformations. The impact on runtime when deriving the PACF and AR features is unsurprising. But the result shown in the critical difference diagram of Figure 4.1 was unexpected. The outcome of these experiments is the removal of PACF and AR derivations. Moving forward, all experimental results are achieved with the

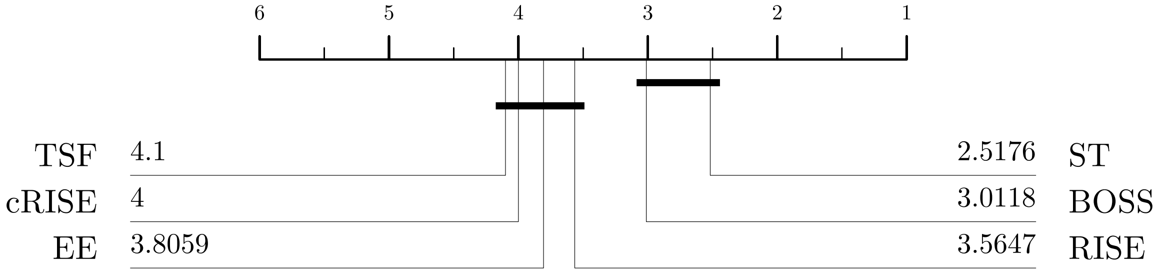


Fig. 4.1 A pairwise critical difference diagram showing the ranks of TSF, cRISE, EE [68], RISE, BOSS, and STC over the same 10 random resamples of 85 UCR datasets.

Table 4.2 Mean and median train times of RISE and cRISE over all UCR datasets, as well as average speed up.

	cRISE	RISE	Difference (%)
Mean (seconds)	144	3554	95.95
Median (seconds)	145	3794	96.18

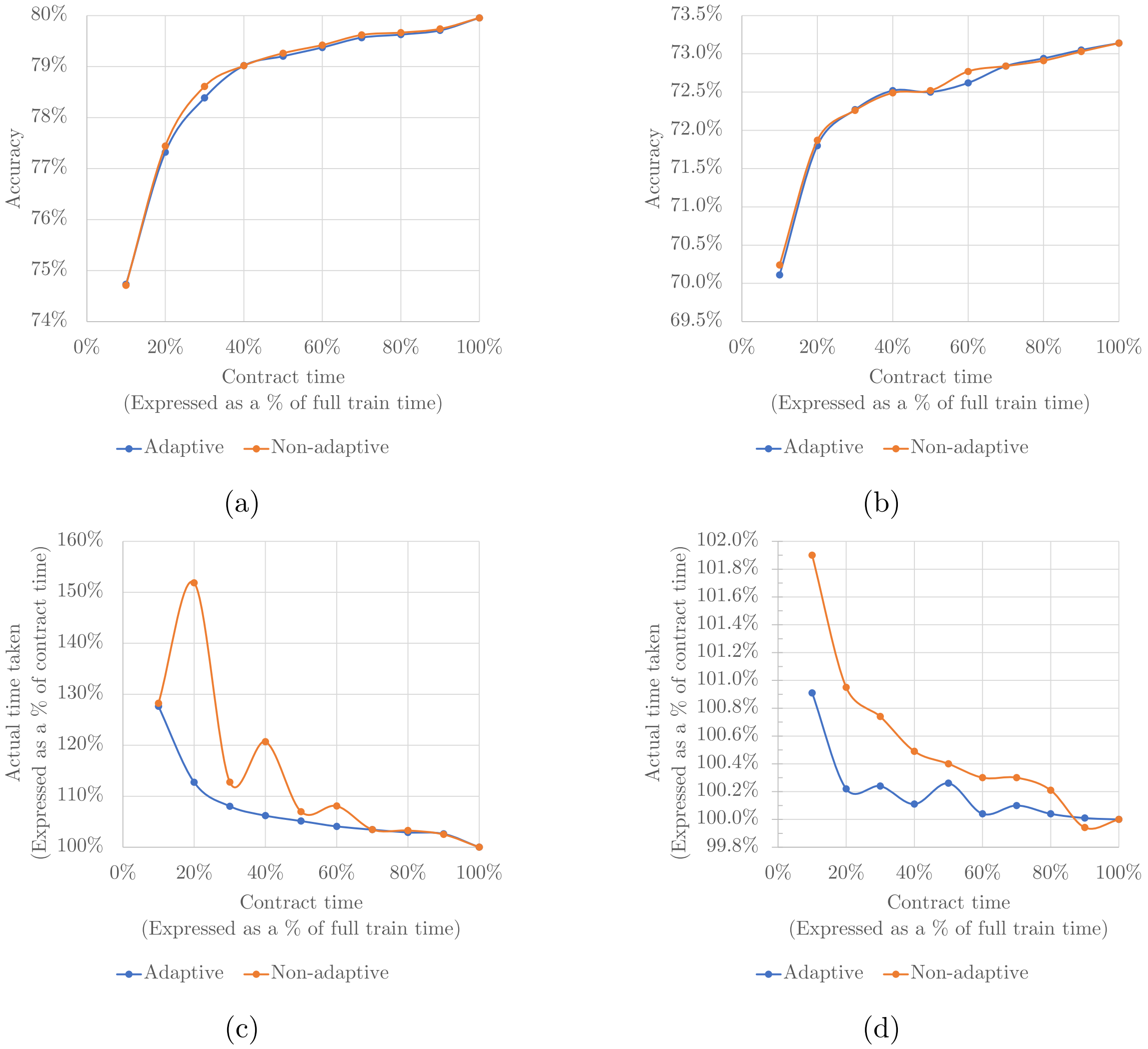
updated architecture of cRISE, in which only spectral and cross correlation features are leveraged.

Naive vs Adaptive Timing Models

In this section, An evaluation of how the accuracy of cRISE changes as a function of total training time for both the Naive and Adaptive timing approaches and, how well each approach adheres to the contract itself is undertaken.

In order to achieve this, nine pairs of experiments were carried out. For both approaches, a contract is set representing 10% - 90% total training time per dataset in, 10% increments. This allowed us to examine changes in accuracy at 10 evenly spaced points in time, as well as test the ability of each approach to stay within the contract.

Figure 4.2 (c) shows how the actual train time changes over different contracts. Each point represents the mean train time over 85 datasets and 10 folds, for each



(a) (b)

(c) (d)

Fig. 4.2 The graphs on the left show the performance of the Adaptive and Naive approaches in respect to their ability to adhere to contract time. The graphs on the right show the predictive accuracy of the Adaptive and Naive approaches over 10 contracts. The top row displays results averaged over all datasets from the UCR database. The bottom row displays results averaged over all problems in the UCR datasets with at least 700 attributes.

contract. The contracts themselves are defined as a percentage of full train time per dataset. This represents 8,500 experiments per approach, over 10 contract percentages.

Figure 4.2 (a) also shows that the Adaptive approach displays much more predictable behaviour, in the context of adhering to contract time. Adhering to relatively small contract times presents more of a challenge to both approaches. This can be explained by the limit imposed on the minimum interval size. Small contracts are better fulfilled by small intervals as each iteration represents a smaller proportion of the contract, allowing for finer control over the total time taken.

Initially, this appears as a major flaw in both approaches. However, this problem is largely exacerbated by the existence of many small problems in the UCR 85 database that without intervention take between 1 and 4 hours to complete.

In order to remove the bias introduced by smaller datasets, the same experiments were repeated with all datasets containing data over 700 attributes.

Figure 4.2 (d) demonstrates how the actual time taken changes over different contract times for problems from the UCR archive with 700 or more attributes. Figure 4.2 (d) shows that both approaches were confounded by smaller problems. It also illustrates the Adaptive approaches superior ability to adhere more closely to the contract than, the Naive approach.

Interestingly, these changes in contract accuracy have very little to no effect on accuracy. Figures 4.2 (a) and 4.2 (b) show how accuracy changes as a function of contract time for all UCR datasets and datasets over 700 attributes respectively. This is important, as it confirms that the superior ability of adhering to contract time comes at no cost to accuracy for the Adaptive approach.

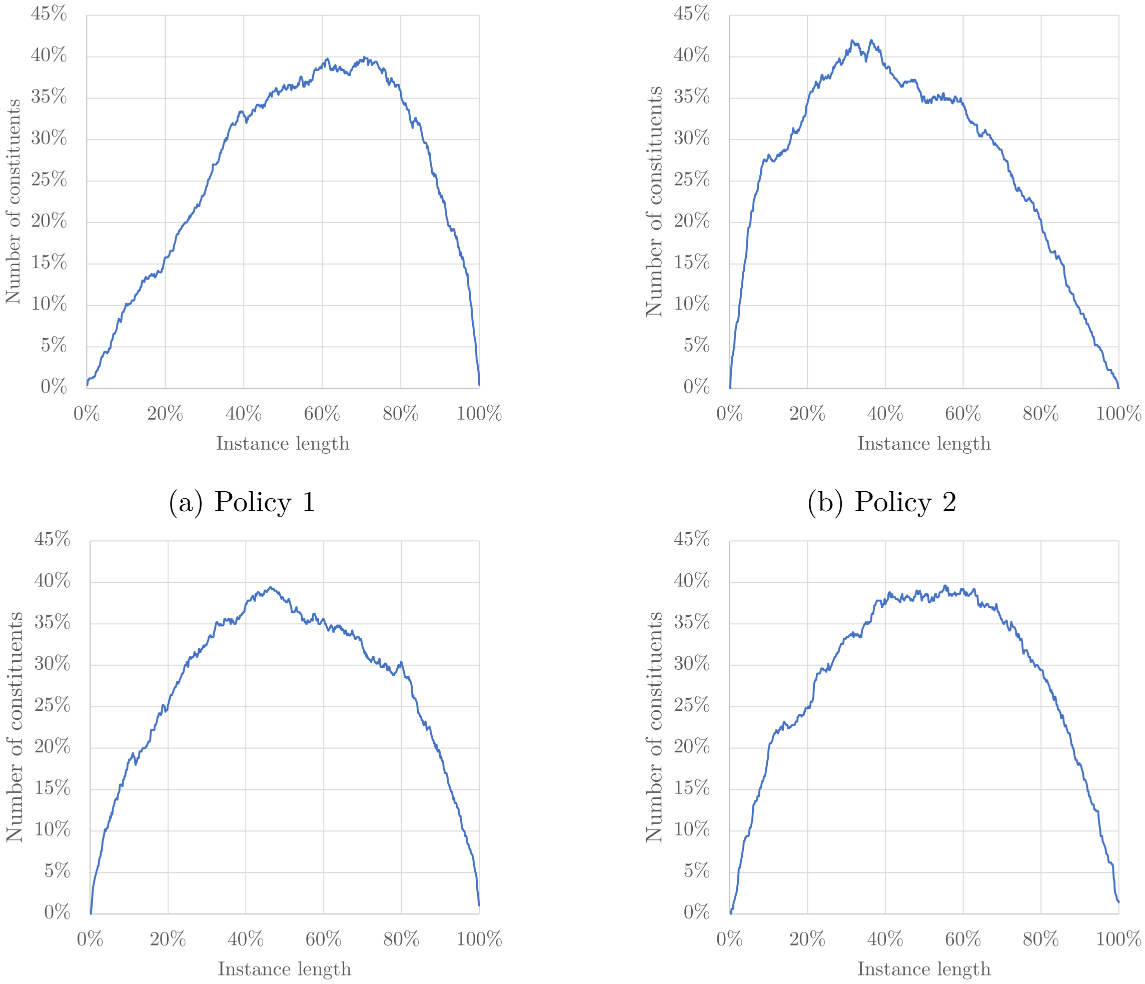
4.3 A Review Of Interval Selection Policies

As with many approaches which employ ensembles of weak learners, cRISE makes use of random intervals to introduce variance in the constituent models. Individual constituents are typically simple models, such as trees, and produce poor accuracies. The likelihood of individual trees producing a poor accuracy is increased in an interval based approach, as there is a chance that discriminant information from the original series is not present. In order to negate this, a large number of constituents are used. Interval selection is the process in which both the location and length of interval is chosen. Typically, they are both picked randomly - within the constraints of the maximum and minimum interval size. In this section we: present the current policy alongside 3 alternate selection policies; discuss the effect that each policy has on the distribution of selected attributes and go on to present the effect that these policies have on classification accuracy. The experimental results presented are derived from 112 datasets from UCR database and represent 30 stratified resamples.

4.3.1 Interval Selection Policies

Policy 1

Policy 1 represents the original interval selection implementation carried over from RISE into the cRISE algorithm. At the inception of cRISE, little thought was given to the selection policy governing the length and position of intervals. As a result, the most obvious procedure was implemented, shown in Algorithm 4. For the fist constituent the entire series is used for transformation (line 6). For all other constituents, the start index is selected first (line 9), followed by length (line 13), and finally the end index is set (line 14). The selection process does not have any tunable parameters.



(a) Policy 1

(b) Policy 2

(c) Policy 3

(d) Policy 4

Fig. 4.3 4 plots showing the distribution of attribute selection for the cRISE classifier using different interval selection policy's. The y-axis shows how often each attribute is selected as a proportion of the number of constituents, the x-axis represents the length of a problem. The plots were produced from fold 0 of the MosquitoSounds problem.

Algorithm 4 Policy 1 - The default interval selection process for cRISE

```
function SELECTINTERVAL($min$, $max$, $r$)
    Let $min$ be the minimum interval length.
    Let $max$ be the maximum interval length.
    Let $r$ be the index of the current weak learner.
    $startIndex \leftarrow 0$
    if $r == 1$ then
        $endIndex \leftarrow max$
    else
        $startIndex \leftarrow randBetween(0, (max - min))$
        if $startIndex == (max - min)$ then
            $endIndex \leftarrow max$
        else
            $length \leftarrow randBetween(min, (max - startIndex))$
            $endIndex \leftarrow startIndex + length$
    return $startIndex, endIndex$
```

The resulting distribution of selected attributes over all constituents is shown in Figure 4.3 (a). The selection of the start index first produces a negative skew as the start index is only constrained by the maximum interval length. As a result, attributes in the first 20% of the problems length are significantly less likely to be included in intervals, when compared to the more symmetrical distributions of Policies 3 and 4.

Policy 2

The second policy, shown in Algorithm 5, is an equally straight forward approach to the first. As previously, the first constituent receives an interval containing all attributes (line 6). However, the end index is selected first (line 9), then the length (line 13), before the start position is set (line 14). There are no tunable parameters.

The resulting distribution, shown in Figure 4.3 (b), is positively skewed. As expected, the skew displays similar characteristics to the distribution produced by Policy 1. The attribute at 80% of the problem length is selected 10% less frequently when compared to the more symmetrical distributions of policies 3 and 4. The most

Algorithm 5 Policy 2

```
function SELECTINTERVAL($min$, $max$, $r$)
    Let $min$ be the minimum interval length.
    Let $max$ be the maximum interval length.
    Let $r$ be the index of the current weak learner.
    $endIndex \leftarrow max$
    if $r == 1$ then
        $startIndex \leftarrow 0$
    else
        $endIndex \leftarrow randBetween(0, (max - min)) + min$
        if $(endIndex - min) == 0$ then
            $length \leftarrow min$
        else
            $length \leftarrow randBetween(0, (endIndex - min)) + min$
        $startIndex \leftarrow (endIndex - length)$
    Return $startIndex, endIndex$
```

commonly selected attributes are included in 40% of constituents and the peak of the distribution is at the 30% mark, with respect to number of attributes.

Policy 3

Policy 3, defined in algorithm 6, is an implementation of the selection process found in the cBOSS approach, described in Sub-Section 3.3.1. In contrast to policies 1 and 2, this implementation does not enforce the use of a problem length interval to be used in conjunction with the first constituent. Instead, the policy randomly changes the order in which the start index, and end index is chosen. The order is selected by a random boolean (line 5). If this is true the start index is chosen (line 6), then the length is chosen (line 8), before the end index is assigned. If the boolean is false, the end index is selected (line 11), then the length (line 13 or 15) and finally the start index is assigned (line 16).

As shown in Figure 4.3 (c) this policy results in a roughly symmetrical distribution of selected attributes. The most commonly selected attribute is at 45% of the problem length and was present in 39% of constituents.

Algorithm 6 Policy 3

```
function SELECTINTERVAL($min$, $max$)
    Let $min$ be the minimum interval length.
    Let $max$ be the maximum interval length.
    if randBoolean() then
        $startIndex \leftarrow randBetween(0, (max - min))$
        $range \leftarrow max - startIndex$
        $length \leftarrow randbetween(0, (range - min)) + min$
        $endIndex \leftarrow startIndex + length$
    else
        $endIndex \leftarrow randBetween(0, (max - min)) + min$
        if $(endIndex - min) == 0$ then
            $length \leftarrow 3$
        else
            $length \leftarrow randBetween(0, (endIndex - min)) + min$
        $startIndex \leftarrow (endIndex - length)$
    Return $startIndex, endIndex$
```

Policy 4

In contrast to the 3 previous policies discussed, policy 4, described in Algorithm 7, does not randomly select the position of each interval. Instead, a number of equally spaced anchor points are defined. The anchor points are used sequentially, based on the constituent index. The location in the interval that corresponds to the anchor point depends on the location of the anchor relative to the problem length. For example, if 5 anchor points are used in conjunction with a problem of length 100, the anchor points would correspond to attributes 10, 30, 50, 70, 90. If the interval length chosen for constituent 1 is 30, the interval would contain attributes 7 - 37, where 10% of the interval length is taken from the left of the anchor point and 90% from the right. If the interval length chosen for constituent 3 is also 30, the interval would contain attributes 35 - 65. Lengths are randomly sampled from a uniform distribution before being adjusted via an exponential mapping. An example of the function used is shown in Figure 4.4. The adjustment favours smaller intervals, which in turn produces more variation between constituents.

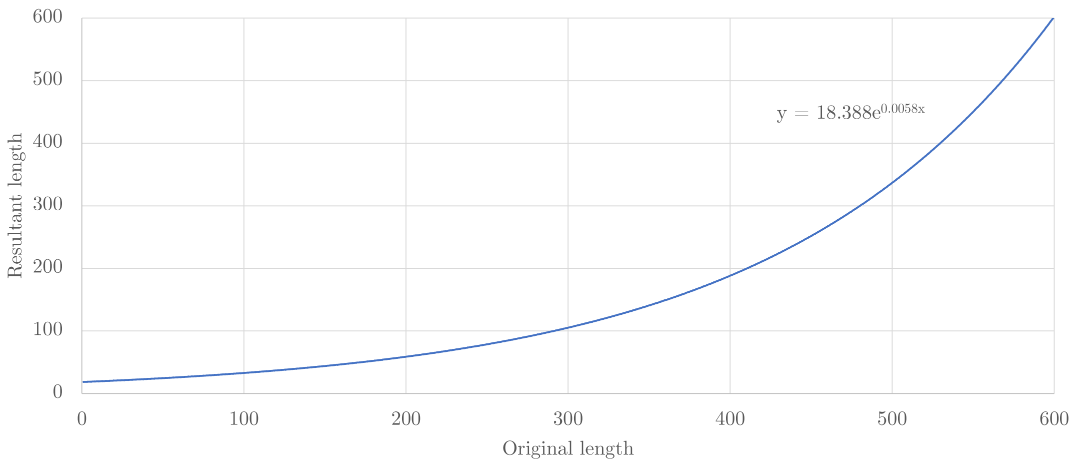


Fig. 4.4 An example of the exponential function used in interval selection policy 4 to adjust intervals from the InsectSound dataset.

By default, the number of anchor points is set to the square root of the problem length (line 6), m, although it can be tuned. The values required to produce the correct mapping in the exponential function are then derived (Lines 7 & 8) before a length is selected (line 9) and converted (line 10). The percentage of the interval found to the left of the anchor point is then derived (lines 11 - 13), before the anchor point is set (lines 15 - 18). If the number of partitions is set to 1, the anchor point is set to half the problem length. Otherwise, the anchor point and proportion of attributes found to the left of it is derived using the constituent index. Finally, the start index and end index are set (lines 19 - 20).

4.3.2 Results

The critical difference diagram in Figure 4.5 summarises the experimental results of 30 stratified resamples from the 112 datasets in the UCR archive. The diagram shows that the four policies presented in Sub-Section 4.3.1 form 3 cliques. Policies 3 and 4 form the top clique, with means ranks of 1.996 and 2.4516 respectively. Despite the lower mean rank, policy 4 produces a marginally higher accuracy of 0.8096, an increase

Algorithm 7 Policy 4

```
function SELECTINTERVAL($min$, $max$, $r$, $m$)
    Let $min$ be the minimum interval length.
    Let $max$ be the maximum interval length.
    Let $r$ be the index of the current weak learner.
    Let $m$ be the number of attributes in an instance.
    $numPartitions \leftarrow ceil(\sqrt{m})$
    $b \leftarrow log(max \div min) \div (max - min)$
    $a \leftarrow max \div e^{(b \times max)}$
    $x \leftarrow randBetween(0, (max - min)) + min$
    $y \leftarrow a \times e^{(b \times x)}$
    $temp \leftarrow r \div numPartitions$
    $temp \leftarrow temp - floor(temp)$
    $temp \leftarrow temp + ((1 \div numPartitions) \div 2)$
    $anchorPoint \leftarrow 0$
    if numPartitions == 1 then
        $anchorPoint \leftarrow (m \div 2)$
    else
        $anchorPoint \leftarrow floor((m \div 1) \times temp)$
    $startIndex \leftarrow ceil(anchorPoint - (y \times temp))$
    $endIndex \leftarrow (startIndex + length)$
    Return $startIndex, endIndex$
```

of 0.0020 over policy 3. However, policy 3 produces a lower variance over dataset folds, lower variance in rank over all datasets and boasts a lower mean train time. Whereas, policy 4 is ranked 4th for all datasets with respect to train time.

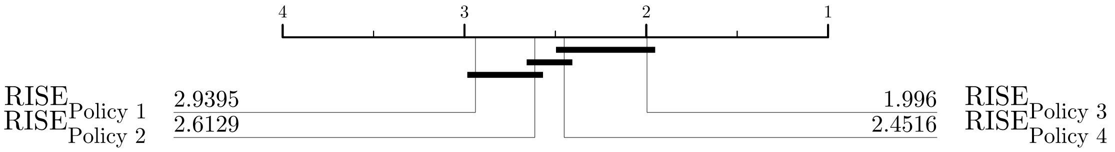


Fig. 4.5 Critical difference diagram showing cRISE (RISE) policies on 112 datasets from the UCR archive.

Cross-referencing the results in the critical difference diagram with the distributions shown in Figure 4.3 reveals a correlation between performance and the distribution of attribute selection. The normally distributed policy 3 is significantly better than both the positively and negatively skewed policies 1 and 2, possibly revealing patterns common throughout multiple datasets.

Experiments undertaken on 13 audio datasets, including those presented in section 2.5, showed that there was no significant difference between the policies. Within the single clique, the policies were ordered 4, 3, 2, 1. Policy 4 produced the highest accuracy of 78.11%, an improvement of 0.46% over policy 3. As the cRISE is intended as an approach which excels at classifying oscillatory data, such as audio, policy 4 is favoured as the interval selection method in experiments moving forward.

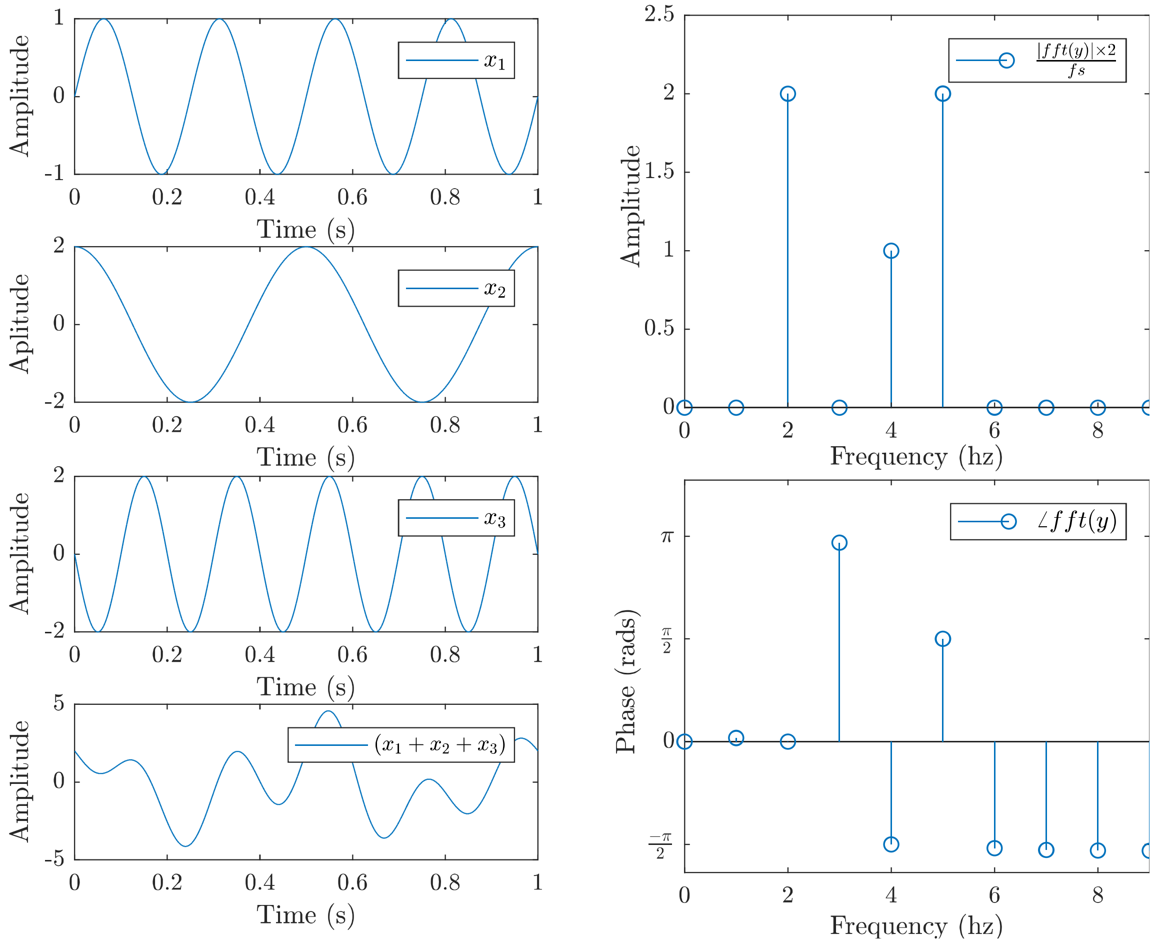
4.4 Improving Accuracy Using Transforms

One obvious area of improvement for the cRISE classifier might be the inclusion of additional, or altogether alternative transforms. In the following section, alternative transforms are presented that are new to the cRISE framework. These transforms were chosen, as they are often used in signal processing. Furthermore, the question of tuning is also addressed. cRISE is presented in conjunction with each transform separately, through selection based on the training set and via multiple combination mechanisms. An ablation study is then undertaken with the best performing approaches in order to provide a deeper understanding of their performance.

4.4.1 Transforms

The following subsections describe the spectral transformations evaluated in Section 4.4.2. Each of the transforms were evaluated in conjunction with the cRISE approach in an effort to increase accuracy

Fig. 4.6 Decomposing complicated signal



Spectral Series

The Discrete Fourier Transform (DFT), shown in equation (4.1), is probably the most important tool in time-series analysis. The transform facilitates the decomposition of a complicated signal, **x**, into its constituent sine waves. For oscillatory signals expressed as a function of time, the transform will output a complex vector, **X**, that is expressed as a function of frequency. The resultant vector can be used to extract either the power, or phase, associated with each constituent frequency of the complicated signal. This is illustrated in Figure 4.6, where plots on the left show a complicated signal and its three constituent sine waves, and plots on the right show the resultant vector from the DFT, interpreted as amplitude per frequency and phase per frequency. Furthermore, via the Inverse Discrete Fourier Transform (IDFT) shown in Equation (4.2), it is possible to reverse the decomposition and return the signal back into the time domain.

$$\begin{gathered}\boldsymbol{X}(k) = \sum_{n=0}^{N-1} \boldsymbol{x}(n) \cdot e^{-i\omega_k n} \\ k = \{0, 1, \dots, N-1\}\end{gathered} \tag{4.1}$$

$$\begin{gathered}\boldsymbol{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \boldsymbol{X}(k) \cdot e^{i\omega_k n} \\ n = \{0, 1, \dots, N-1\}\end{gathered} \tag{4.2}$$

The Fourier Transform is a powerful tool that is used in signal processing, as well as quantum mechanics and many forms of spectroscopy. It also forms the basis for many additional transforms, such as Morlet wavelet transforms, Spectrograms and Mel-Frequency Cepstral Coefficients (MFCCs). The transform makes use of the fact that any complicated signal can be expressed as a combination of sine waves. A complex signal can be expressed in terms of sine waves from 0 hz to half of the recording sampling rate. The extent to which each of these constituent sine-waves contributes to the complicated signal, is expressed as the similarity between the two, where the similarity is expressed as the sum of per element dot products. Typically, the two signals are expressed as complex vectors. This allows per sample phase information to be encoded into the imaginary component of the respective sine-wave, where the amplitude is represented by the real component. The length of the resulting vector is equal to the length of the target signal. The power of each constituent frequency bin in the resulting vector, is the absolute value of the element divided by the number of elements and the phase is the angle of the element.

In the form shown in Equation (4.1) the best, worst and average time complexity is $O(n^2)$. This is prohibitive for longer signals. As a result, the Fast Fourier Transform (FFT) is often implemented in practice. Popularised by Cooley and Tukey [19] the FFT algorithm applies a divide and conquer approach to the transform. As a result, the time complexity of this approach is $O(nlog(n))$, providing the interval length is a power of 2.

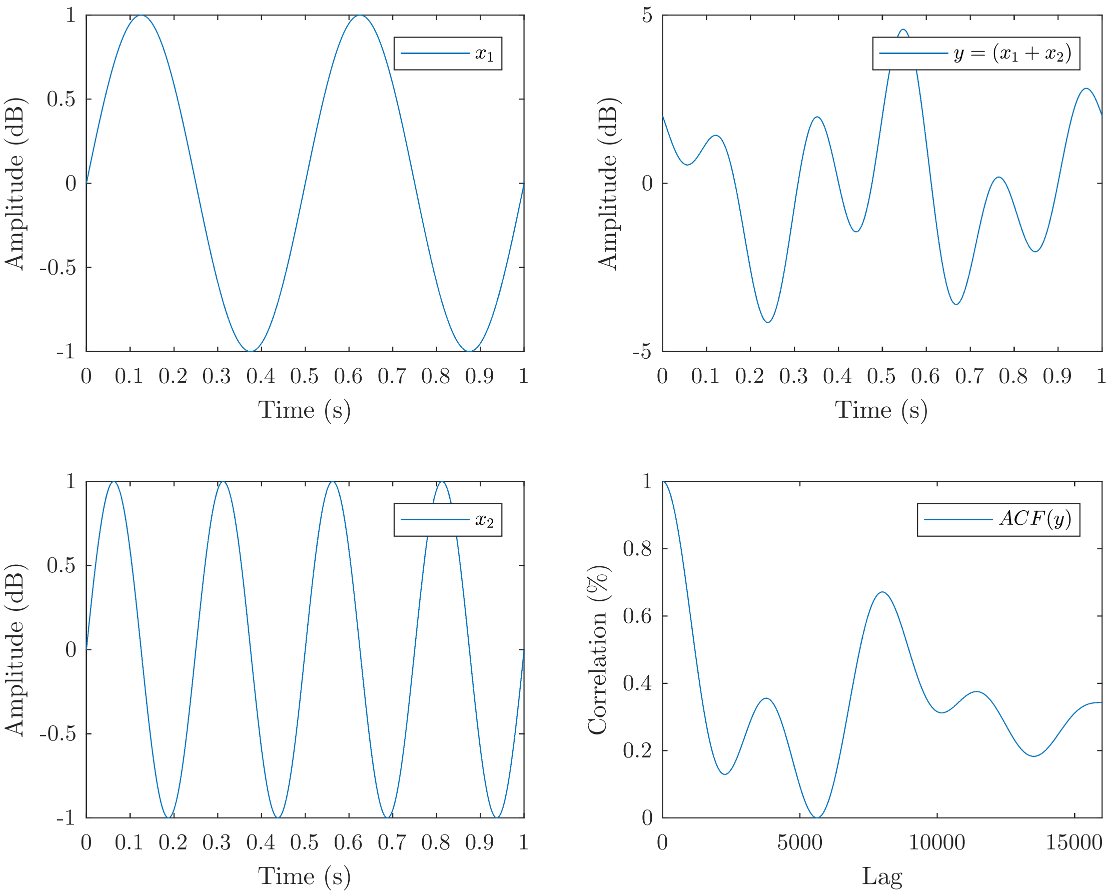
Autocorrelation

Frequently refereed to as serial correlation, autocorrelation is a quantified measure of the relationship between a series and itself over multiple lag values, k.

$$\begin{gathered} r_k = \frac{\sum_{i=1}^{n-k}(x_i - \bar{X})(x_{i+k} - \bar{X})}{\sum_{i=1}^{n}(x_i - \bar{X})^2} \qquad (4.3) \\ X = \{x_1, x_2 ... x_n\} \\ \bar{X} = \frac{\sum_{i=1}^{n} x_i}{n} \end{gathered}$$

This function, (4.3), is often used in detecting non-randomness in data and helping identify appropriate times series models. Consider that an attribute, x_i, from a time series, $\mathbf{t}$, is directly influenced by the neighbouring attribute, x_{i-1}. Whereas, in a series of variables randomly selected from a distribution matching that of the time series, $\mathbf{t}$, there is no meaningful relationship between neighbouring attributes. Thus, the autocorrelation of an oscillating signal such as a simple sinusoid would, at a lag value representing a phase shift of 2π radians, produce a high autocorrelation. Furthermore, the analysis of the relationship between k and r_k can inform the periodicity of the signal.

Fig. 4.7 Auto correlation



This is demonstrated in Figure 4.7. The top right plot shows a complicated signal formed from the sine waves shown in the left hand plots. The top left plot shows a 2 hz sine wave and the bottom left plot shows a 4 hz sine wave. Both generated with a sample rate of 16,000 hz. The autocorrelation of the complicated signal is displayed in the bottom right plot. It shows peaks in correlation at lags of 4,000 and 8,000, aligning with a 2π radians phase shift of the 2 hz and 4 hz signals at 0.5 and 0.25 seconds. In this case, it was possible to use prior knowledge of the sine waves sample rate to show the corresponding frequencies. Typically, audio is a significantly more complicated combination of both sine and cosine waves and information regarding the sample rate of recorded sounds is not available. As a result, determining individual constituents is not feasible. However, the autocorrelation coefficients still provide valuable insights, from which effective features can be derived.

Audio Features

Spectral Centroid The spectral centroid is notionally defined as the ‘centre of mass’ of an audio signal. As shown in Equation (4.4) [73], it denotes the weighted mean. The feature provides a quantitative value, that describes the qualitative idea of brightness. The feature is often used as both a global and local feature.

$$\mu_1 = \frac{\sum_{i=0}^{m} f_i x_i}{\sum_{i=0}^{m} x_i} \tag{4.4}$$

Derived from the power spectrum, f_i denotes the central frequency represented by the i^{th} bin and x_n its value.

Spectral spread The spectral spread, shown in Equation (4.5) [73], describes to what extent to which the energy in the series deviates from the spectral centroid. As such, a signal consisting of a single tone will exhibit low spectral spread, centred around the spectral centroid, whereas a noisy signal consisting of many sine and cosine constituents will exhibit a high spectral spread.

$$\mu_2 = \sqrt{\frac{\sum_{1=0}^{m} (f_i - \mu_1)^2 x_i}{\sum_{i=0}^{m} x_i}} \tag{4.5}$$

Spectral Flatness Spectral flatness provides a quantitative value for expressing how tonal a signal is. A high value, indicates a signal in which the total power is evenly distributed throughout the spectrum. A signal with high spectral flatness would sound like white noise. However, a sound with low spectral flatness indicates that the spectrum contains peaks and likely sounds like a collection of either sine or cosine waves.

$$\text{Spectral flatness} = \frac{\sqrt[m]{\prod_{i=0}^{m} x_i}}{\frac{1}{m}\sum_{i=0}^{m} x_i} \tag{4.6}$$

Defined in Equation (4.6) [73], the flatness is defined as the geometric mean divided by the arithmetic mean.

Spectral Skewness The spectral skewness describes the distribution of power around the centroid. In positive skews, the arithmetic mean is greater than the median and the majority of the power will be found in frequencies lower than the centroid. For negative skews, the distribution is flipped.

$$\mu_3 = \frac{\sum_{i=1}^{m}(f_i - \mu_1)^3 x_i}{(\mu_2)^3 \sum_{i=1}^{m} x_i} \tag{4.7}$$

Defined in equation (4.7) [73], the skewness is often defined as the $3^r d$ moment in context of quantitative features used to define the shape of a signal.

Spectral Kurtosis Kurtosis, shown in Equation (4.8) [73], provides information on the shape of a distribution. Values greater than 0 denote a leptokurtic distribution. In comparison to a normal distribution, there is expected to be a higher peak and more values in the tails. In extreme cases, there can be more values at the tails of the distribution, than around the mean. Values less than one denote a platykurtic distribution. At the mean of this distribution, the peak will be lower then when compared to a normal distribution. There will also be less pronounced tails.

$$\mu_4 = \frac{\sum_{i=1}^{m}(f_i - \mu_1)^4 x_i}{(\mu_2)^4 \sum_{i=1}^{m} x_i} \tag{4.8}$$

Generally interpreted as a measure of how likely outliers are to occur compared to a normal distribution of the same variance, Kurtosis is considered the fourth moment behind mean, variance and skewness.

Zero-crossing Rate The zero-crossing rate, shown in Equation (4.9) [73], is an important measurement in multiple fields. It is used in pitch detection, voice activity recognition, image processing and analogue to digital conversion.

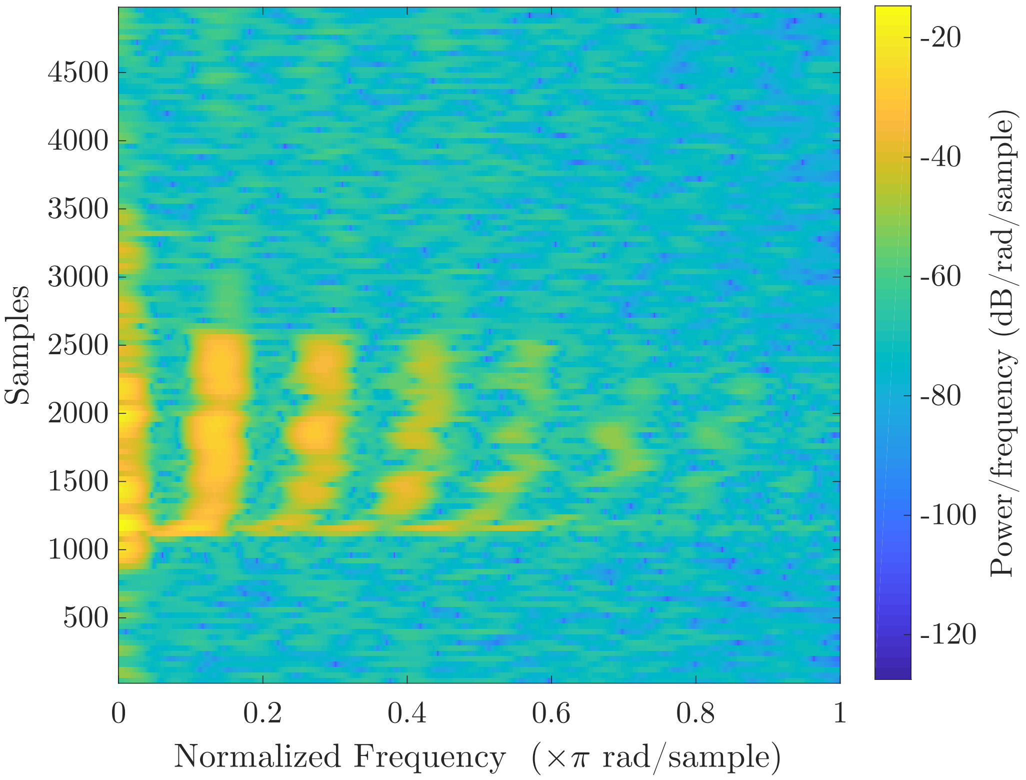
$$\begin{aligned} I_A(x) &:= \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{if } x < 0. \end{cases} \\ ZCR &= \frac{1}{m}\sum_{i=2}^{m} |I_A(i) - I_A(x-1)| \end{aligned} \tag{4.9}$$

The measurement determines the number of times a signal transitions from negative to positive over some time frame. As a result, 1 second of audio at a sample rate of $200hz$ containing a $100hz$ signal would have a zero crossing rate of 0.5 and a $50hz$ signal, a rate of 0.25.

Spectrogram

A spectrogram, shown in Figure 4.8, provides information on the power at each frequency band at discrete points in time. Given a signal, many short form FFTs are

Fig. 4.8 A spectrogram of a *Ae. agypti* recording from the MosquitoSound dataset.



taken and combined to create a spectrogram. Typically, there is an overlap between each window used for FFTs, in order to provide more temporal resolution. Figure 4.8 shows an example using a raw case form the MosquitoSound dataset. Each short form FFT was computed on 80 samples and the overlap between each window was set to 40 samples. These recordings are sampled at 8,000 hz so the spectrogram shows information up to 4,000 hz. The spectrogram clearly shows the low frequency noise, probably caused by the body of the insect. The second band is assumed to be the dominant wingbeat frequency, at roughly 600 hz, and those following it are the harmonics.

Mel Frequency Cepstral Coefficients (MFCCs)

MFCCs are typically used in speech recognition and genre classification systems. They are often computed from short, sometimes overlapping intervals, to improve temporal resolution, similar to Spectrograms. The Quefrency domain expresses the rate of change in frequency bands. The Cepstrum exists in the quefrency domain and is computed by taking the log of the frequency spectrum, before taking the inverse fourier transform.

The Mel-scale represents an alternate interpretation of pitch than the Hertz (Hz) scale. The relationship between Hz and mels is not 1 to 1. The mel scale attempts to represent pitch on a scale which is perceived by a listener to be linear. For example, a sound at 600 mels and 1000 mels would have the same perceived change in pitch as a sound at 2000 mels and 2400 mels. However, the difference in hertz between 600 mels and 1000 mels is 500 Hz and the difference between 2000 and 2400 is 1500 hz. In effect, the scale approximates the way in which the ear interprets sounds of different pitch.

In order to derive Mel-frequency cepstral coefficients, the frequency spectrum should be convolved with a series of filters, a Mel filter bank. The log of the filter output is then used to produce a cepstrum.

4.4.2 Results

As detailed in Sub-Section 4.2.1, the current cRISE configuration consists of concatenating features from the spectral domain with auto-correlation coefficients. In this Sub-Section: an evaluation of each of the transformations described in Sub-section 4.4.1 undertaken separately and before a discussion on their usefulness is presented. The performance of individual transforms, along with the performance of cRISE, provides a baseline to which results can be compared. The experiments also provide the opportunity to show that the combination of the FFT and ACF transforms produces a performance greater than either individually; Experience tells us that, typically, what works for one dataset does not always work best for another. This lead to experiments designed to assess the viability of using tuning mechanisms to select the best transform for a dataset based on the training data. However, the implemented approaches were not able to out perform cRISE; This lead to an investigation assessing additional mechanisms for combining multiple transforms, namely concatenation - an extension of the approach used in cRISE to combine FFT and ACF coefficient, and ensembling using the CAWPE methodology utilised in HIVE-COTEv1. Both of which outperformed the cRISE approach. Finally, a discussion of the results from an ablation study which attempts to assess the importance of individual transforms to the performance of both combination approaches which outperform cRISE is performed.

Individual Transforms

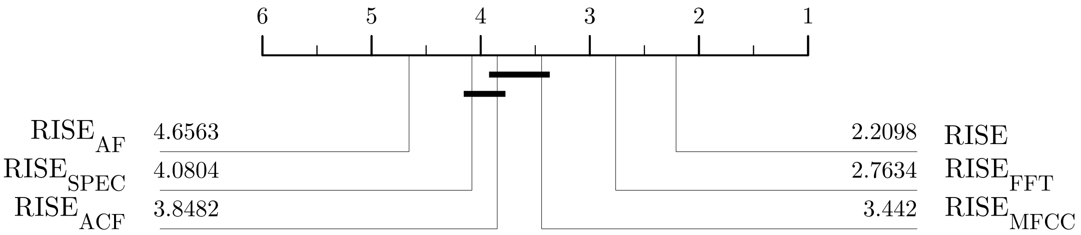


Fig. 4.9 Critical difference diagram showing cRISE (RISE) and combinations of the cRISE approach with only 1 transform used at a time on 112 datasets from the UCR archive.

The critical difference diagram summarising the performance of the individual transforms along with the cRISE approach is presented in Figure 4.9. The transformation included in the approach is donated by the subscript and in all cases, other than cRISE, only 1 transform was included. The figure shows that the transforms form 4 groups. The AF transform performs worst, the Spectrogram (SPEC) and MFCC transforms both share a clique with the ACF transform, but not with each other, and the FFT features perform best. However, all individual transforms perform significantly worse than cRISE.

	FFT	MFCC	ACF	SPEC	AF	Total
FFT		12	15	13	5	45
MFCC	9		7	9	0	25
ACF	8	4		3	3	18
SPEC	7	3	1		1	12
AF	4	1	5	0		10
Total	28	30	28	25	9	

Table 4.3 A table presenting the relationship between 1^{st} rank and 2^{nd} rank for all transformations as number of datasets.

	FFT	MFCC	ACF	SPEC	AF
FFT		0.0225	0.0222	0.0205	0.0058
MFCC	0.0143		0.0225	0.0123	-
ACF	0.0291	0.0273		0.0228	0.0263
SPEC	0.0199	0.1215	0.0166		0.0063
AF	0.0067	0.0016	0.0238	-	

Table 4.4 A table presenting the relationship between 1^{st} rank and 2^{nd} rank for all transformations as the mean difference in accuracy.

The total of each row in Table 4.3 demonstrates the number of times each transform was ranked 1^{st} and the columns represent the number of times each respective approach ranked second. For example, FFT was ranked first on 45 datasets and of those MFCC ranked second 12 times. Table 4.4 presents the same relationship, expressed in terms of the average difference in accuracy. The tables show that the AF transform performs very similarly to FFT, MFCC and SPEC transforms and on datasets which it ranks 1^{st} it is typically by a very small margin. This suggests that as well as having the lowest average accuracy and the least number of wins, it also brings little diversity to the collection of transforms.

Tuning Mechanisms

One way of potentially improving accuracy, is to implement a tuning mechanism in the training phase. The goal of which would be to select the transform which would produce the best test accuracy, based only on the training data. In order to ascertain the potential improvement a good tuning mechanism could make, an 'Oracle' approach is also included. The Oracle approach represents the maximum performance, where the best transform for each dataset has been chosen based on test accuracy. The difference in accuracy between the cRISE and Oracle approach is 0.0114. Alongside the cRISE and Oracle approaches three tuning mechanisms were implemented for evaluation. These are:

cRISE$_{\textbf{Fisher}}$ - For each transformation, the dataset is summarised by the mean, median, standard deviation, slope, inter-quartile range, minimum and max values. The Fisher score is then used as measure of ability to split the dataset into its respective classes. The transformation producing the best Fisher score is selected as the transformation in the subsequent build phase of cRISE.

cRISE$_{\textbf{KNN}}$ - For each transform the mean accuracy of a 5 fold stratified re-sample on the training data is computed. The accuracies are produced via a 1-NN$_{\text{ED}}$ approach. The transform that produces the highest accuracy is selected for cRISE.

cRISE$_{\textbf{cRISE}}$ - For each transform the mean accuracy of a 5 fold stratified re-sample on the training data is computed. The accuracies are produced via a newly instantiated cRISE object with the appropriate transform selected. The transform that produces the highest accuracy is then selected for the parent cRISE approach.

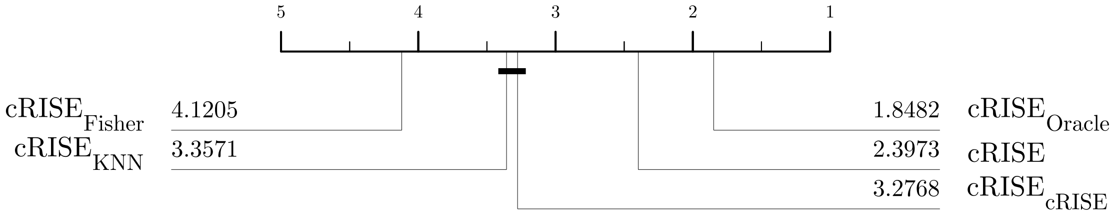


Fig. 4.10 Critical difference diagram showing tuning strategies on 112 datasets from the UCR archive.

Figure 4.10 shows the critical difference diagram comparing the results of cRISE$_{\text{Fisher}}$, cRISE$_{\text{KNN}}$, cRISE$_{\text{cRISE}}$, cRISE and cRISE$_{\text{Oracle}}$. Despite a small difference of 0.0010 accuracy between the cRISE and cRISE$_{\text{cRISE}}$ approaches, they are found to be significantly different. A review of the transforms chosen in the $\text{cRISE}_{\text{cRISE}}$ approach shows that the variance in cases included in the training set affects the selection. For example, on the ArrowHead dataset the optimal transform, MFCC, is selected on 26 of the 30 folds. This accounts for a difference of 0.008 and on no datasets does the $\text{cRISE}_{\text{cRISE}}$ approach select the optimal transform on all 30 folds.

Combining Transforms

Further analysis shows that for 40 of the 112 datasets from the UCR archive, the cRISE approach outperforms the most accurate single transform. This enhances the idea that mechanisms which combine multiple transforms could outperform even a perfect tuning approach. In order to further investigate the effect that combining multiple transforms have on accuracy, three methods of combination were implemented and evaluated. These were: the CAWPE approach, detailed in Sub-Section 3.3.6; the cRISE approach of concatenating the transform features, $\text{cRISE}_{\text{All}}$, and the RANDOM approach, where a transform is picked at random for each tree.

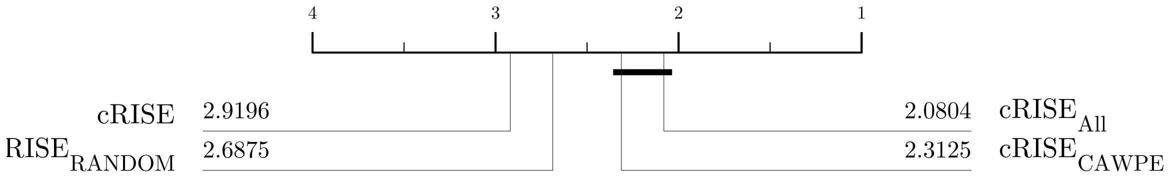


Fig. 4.11 Critical difference diagram showing combination strategies on 112 datasets from the UCR archive.

The critical difference diagram shown in Figure 4.11 summarises the accuracy and rank of the three approaches alongside cRISE. All of the approaches are found to be significantly different to cRISE and produce superior accuracies. Of the three combination approaches, cRISE$_{\text{All}}$ and cRISE$_{\text{CAWPE}}$ are grouped together in the top clique, with accuracies of 0.8262 and 0.8212 respectively.

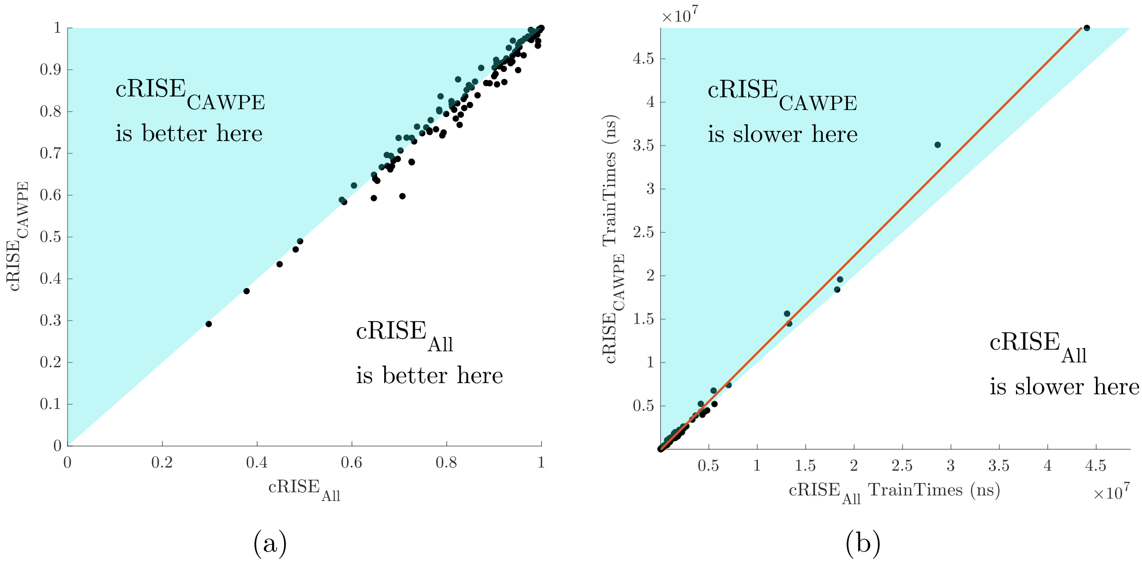


Fig. 4.12 Pairwise scatter diagrams plotting the accuracies and train times of the cRISE$_{\text{CAWPE}}$ and cRISE$_{\text{All}}$ combination approaches.

The scatter diagrams presented in Figure 4.12 show the relative performance of cRISE$_{\text{CAWPE}}$ and cRISE$_{\text{All}}$ in relation to accuracy (a) and train time (b). The cRISE$_{\text{All}}$ approach produces a convincing performance in terms of accuracies, and ranks above the cRISE$_{\text{CAWPE}}$ approach on 69.44% of the 112 datasets. As well as this, cRISE$_{\text{All}}$ produces: a lower mean test accuracy standard deviation over dataset folds, showing it is less susceptible to variations in data; the lowest mean test accuracy rank and a lower standard deviation of test accuracy rank over all the datasets. Furthermore, as cRISE$_{\text{CAWPE}}$ weights the ensemble members via the respective train accuracies, the train time of cRISE$_{\text{All}}$ is also favourable, despite performing costly transformations.

Transform Ablation

As shown in previous experiments, the performance of each transformation varies widely over the UCR archive. In order to assess the extent to which each of the transformations impact the performance of the cRISE$_{\text{All}}$ and cRISE$_{\text{CAWPE}}$ approaches, an ablation experiment was undertaken. Table 4.5 provides a key indicating which transforms are included for each configuration included in the experiment. Based, on Tables 4.3 and 4.4 the expectation is that omission of the AF transform is unlikely to have a significantly negative effect in terms of accuracy and, as the features produced appear to have a similar predictive power to the FFT transform.

	FFT	ACF	AF	MFCC	SPEC
All	x	x	x	x	x
1		x	x	x	x
2	x		x	x	x
3	x	x		x	x
4	x	x	x		x
5	x	x	x	x	

Table 4.5 Table showing which cRISE variants are included in each CAWPE configuration.

The critical difference diagram in Figure 4.13 presents the performance of all transform configurations from Table 4.5 in the CAWPE combination mechanism. It shows that the removal of the FFT, ACF, MFCC and SPEC transforms all result in a significantly worse performance. However, the performance of configurations 1, 2, 4 and 5 are not significantly different from each other or the cRISE approach. Furthermore, the omission of the FFT and SPEC transforms result in an extremely similar performance despite the FFT transform being found to significantly out perform the SPEC transform individually. The removal of the AF transform, results in a significantly better performance. This is primarily reflected as the difference in mean rank as the CAWPE$_3$ approach reports higher variance in rank and accuracy over all datasets with a small real difference in accuracy of 0.0010.

Figure 4.14 presents the critical difference diagram of all the configurations from Table 4.5 in the cRISE concatenation combination mechanism. The diagram shows that the configurations form 3 cliques. Clique 1 is formed of cRISE$_{\text{All}}$ and cRISE$_5$,

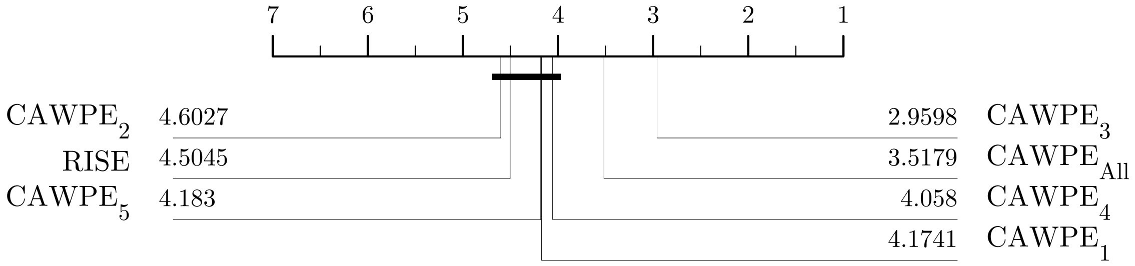


Fig. 4.13 Critical difference diagram showing CAWPE variations and cRISE (RISE) on 112 datasets from the UCR archive.

clique 2 is formed of $cRISE_5$ and $cRISE_3$, whilst clique 3 is formed of $cRISE_1$, $cRISE_2$ and $cRISE_4$. All the cRISE configurations are found to be significantly different from the cRISE approach and in all cases they produce superior accuracy. The results show that omitting the FFT, AF or MFCC transform results in a significantly worse performance. Also, the diagram shows that omitting the FFT or MFCC transforms, produce results that are significantly worse than those in which the AF transform was omitted.

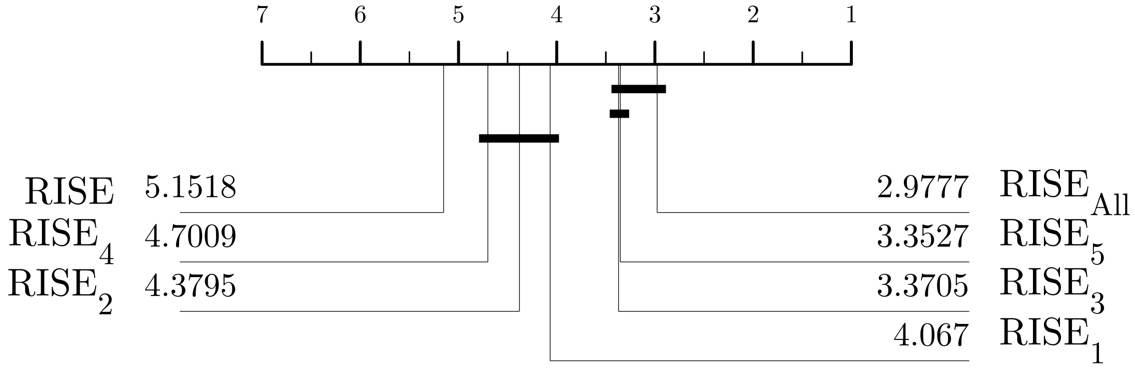


Fig. 4.14 Critical difference diagram showing cRISE (RISE) variations on 112 datasets from the UCR archive.

A comparison of the $cRISE_{All}$ and $CAWPE_3$ approaches, shows that there is no significant difference between them. The $cRISE_{All}$ approach was found to be more accurate at 0.8262% over 30 folds on 112 datasets, with a difference of 0.004%. It ranked first on 57 of 112 datasets and was found to produce a marginally lower standard deviation in test accuracy over folds and across all datasets. The train time of the cRISE$_{\mathrm{All}}$ approach was found to be lower on 62 of the 112 datasets. An investigation into datasets for which the train time was higher, revealed that for smaller datasets the CAWPE$_3$ benefited from the reduction in the number of FFTs performed with the removal of the AF transform, despite the need to produce train accuracies.

4.5 Conclusion

In conclusion, this chapter presents changes to the RISE algorithm that significantly improves its accuracy, usability and robustness. The presented changes to RISE result in a significantly faster train time. This result was achieved using an established experimental design on the well known UCR database[1] [4]. These changes remove two costly derivations, making cRISE at least twice as fast whilst the cost to test accuracy is shown to be insignificant. The Adaptive and Naive timing models were also presented and compared in the context of cRISE. On 85 problems of the UCR database the experimental design does not show any significant difference in accuracy between models, although one approach did show a superior ability to adhere to contract. This superior ability to adhere to a contract time, was made further evident when considering a problem consisting of significantly longer series. It was shown that the Adaptive approach is robust to scaling of series length, whereas the Naive approach is bound by series length, number of cases, or both.

Through a review of the interval selection policy implemented in the cRISE approach, it was demonstrated that there was a selection bias in the interval generation implementation that had a significant effect on performance. The evaluation of multiple

[1] www.timeseriesclassification.com

alternate selection polices facilitated the adoption of an all together improved approach, in which the entire instance length is given equal weighting. This leads to a more normal distribution of attribute selection, rather than the negative skew of the original policy and an improvement in mean accuracy over the UCR database.

Through the introduction of additional transformations, an investigation into alternate cRISE configurations against the traditional concatenated FFT and ACF transforms was undertaken. The evaluation and comparison of the new transforms, allowed us to comment on their apparent usefulness and performance, with respect to the benchmark. An investigation of tuning approaches in which proxies were utilised to predict the optimal transform, proved fruitless. Of the 3 approaches implemented, it was shown that all were significantly worse than the benchmark of traditional and theoretical benchmarks cRISE and ORACLE. However, this led to an examination of combination mechanisms, including: concatenation, random selection and ensembling. Through an ablation study of the best performing combination strategies, we discuss the merits of each approach and the extent to which they are affected by transformation omission. The findings have led to a new recommended cRISE configuration and collectively, they represent a significant improvement in the usability of cRISE and consequently HIVE-COTEv1.

Chapter 5

Insect Classification I

Contributing publications

- Flynn M., Bagnall A. (2019) Classifying Flies Based on Reconstructed Audio Signals. In Intelligent Data Engineering and Automated Learning – IDEAL 2019. IDEAL 2019. Lecture Notes in Computer Science, vol 11872. Springer, Cham.

5.1 Introduction

The World Health Organisation's (WHO) global technical strategy for malaria 2016-2030 [69], discussed in Chapter 2, hinges on the development of automatic monitoring solutions for Mosquitoes. Furthermore, to be really effective these systems would need to be suited to deployment in rural settings in large numbers. In order to achieve this they will need to be power efficient and low cost. At the core of any such system is a classification problem: given a segment of audio collected as something passes through a sensor, can we classify it? We examine the case of detecting the presence of fly species, with a particular focus on mosquitoes. This gives rise to a range of problems such as: can we discriminate between species of fly? Can we detect different species of mosquito? Can we detect the sex of the insect?

The hardware and datasets presented in Chapter 2 mark a distinct step forward in the realisation of the WHO's goals. Capable of producing large datasets of seemingly high quality data, already there is a small but considerable corpus of flying insect datasets. Until recently, practitioners were only able to make use of small numbers of insects across a handful of species, and as discussed in Section 2.3, have not been able to leverage sophisticated approaches which perform best on large datasets. The result has been a stagnated period in the progress of flying insect classification application and progress has been somewhat limited.

In Section 3.2 the idea of time-series data and an explaination of its intrinsic difference to non-ordered numeric data is introduced. A common example used when discussing Time Series Classification (TSC) is audio, as it is possibly one of the most obvious examples of naturally ordered data. On this basis, the expectation is that modern TSC approaches will perform well on the datasets presented in Section 2.5, which can be interpreted as audio and represent the incidental sound produced by flying insects. Until now, many of these TSC approaches have not been applied to the insect classification problem. It is the accepted belief that those which internalise transforms into the spectral domain will perform best

In this Chapter, a range of TSC approaches are assessed, including the state-of-the-art HIVE-COTEv1 approach, on the two mosquito datasets available at the time (2019): InsectSound and MosquitoSound. These are described in Sections 2.5.3 and 2.5.4. Similarly to the publications discussed in Section 2.3, The first expeirements form an investigation into the performance of fundamental wingbeat frequency as a discriminant feature, discussing both the overall accuracy, but also the type II error rate with regards to female mosquito classes. Further experiments are then performed with a range of sophisticated TSC approaches on two spectral transformations and the raw dataset. For each of these a discussion of the impact on accuracy for each transformation is undertaken, as well as considering what impact transformation into the spectral domain has more generally. Finally, information on test time and a judgement on whether any of the approaches could be considered ‘real time’ is presented. The intention is to broaden the understanding of whether TSC approaches are an effective tool in the application of insect classification, whilst providing a robust and reproducible evaluation for future comparisons.

The rest of this chapter is structured as follows: in Section 5.2 the approaches used in the evaluation are described and details of the parameters used provided; in Section 5.3 the transformation techniques used prior to classification are described; in Section 5.4 our results are presented and in Section 5.5 we summarise and conclude, attempting to answer the previously posed questions:-

> “Is it possible to discriminate between species of fly? Is it possible to detect different species of mosquito? Is it possible detect the sex of the insect?”

.

5.2 Approaches And Parameters

The following experiments include an array of algorithms in an effort to understand which types of classification approach perform best on the insect application. As a result, the performance of many of these approaches on the two wingbeat datasets used have not previously been published. These include: STC, described in Sub-Section 3.3.5, in which the small subsections are selected at random and retained based on their ability to summarise class specific features; TSF, described in Sub-Sction 3.3.3, in which random intervals are selected and distilled into statistical features that are used to grow C4.5 decision trees; BOSS, described in Sub-Section 3.3.1, in which instances are first split and compiled into a dictionary of words represented as histograms and classification takes place via a 1-NN, used in conjunction with a bespoke distance measure and cRISE, described in Sub-Section 4.2.1, in which random intervals are selected and transformed into spectral and autocorrelation coefficients. These new representations are then combined, before being used to grow random decision trees. The HIVE-COTEv1.0 ensemble is also evaluated. This is described in Sub-Section 3.3.6 and uses the CAWPE control structure to combine the predictive power of the TSF, BOSS, cRISE an STC approaches. The cBOSS algorithm was contracted to complete training within 24 hours for the InsectSound representations, and 1 hour for the MosquitoSound representations, after it failed to produce results within the 7 day hard limit imposed on the UEA HPC.

5.3 Data Pre-processing

The work in this chapter focuses on the two wingbeat datasets which were available, InsectSound and MosquitoSound. A detailed description of these can be found in Section 2.5. Pre-processing was undertaken to reduce their sizes in order to improve the manageability. In the case of the InsectSound dataset, two processes were undertaken. Each instance contains a relatively small amount of data centre padded with zeroes in order to produce 1 second of audio at 16,000 hz. This is the result of extraction via a 100 ms sliding window from a larger recording. The first pre-processing step reverses this via extracting the central 1,600 attributes from each instance. In the the second step, the extracted data is resampled to 6,000hz. The new sample rate represents a 2.6 times reduction in the number of attributes for each instance whilst maintaining spectral fidelity up to 3,000 hz, which is roughly 3 times the maximum expected wingbeat frequency of mosquitoes. In the case of the MosquitoSound dataset, two pre-processing steps are also undertaken. Firstly, the number of instances in each class was reduced to 5,000, bringing it in line with the InsectSound dataset. Secondly, to further reduce the dimensionality of the dataset, the instances were resampled to 6,000 hz. This reduced the number of attributes by 1,400 and reduced the Nyquist limit to 3,000 hz.

As previously mentioned, the prior belief was that spectral features will be most effective in maximising accuracy with regards to audio-centric datasets. In an effort to assess this, results from transformed copies of both datasets are also presented in Section 5.4. The datasets have been transformed into the spectral domain via the two methods presented in Algorithms (9) and (8). In each case, the transformation is undertaken on the raw dataset prior to reduction in problem length detailed above.

Algorithm 8 Transform 1

```
1: function TRANSFORM($M$, $samplerate$)
2:     Let $M \leftarrow M_1 \ldots M_n$ be a dataset.
3:     Let $nfft \leftarrow 6000/2$.
4:     for $i \leftarrow 1$ to $n$ do
5:         $M_i \leftarrow resample(M_i, 6000, samplerate)$
6:         $M_i \leftarrow fft(M_i, 6000)$
7:         $M_i \leftarrow getPowerSpectrum(M_i)$
8:         $M_i \leftarrow truncate(M_i, 1, nfft)$
9:     return M
```

In Algorithm (8), T1, the number of FFT bins (nfft) is set as the nyquist limit (line 3), defined as half the sample rate. Then, each instance is re-sampled to 6,000 hz (line 5) prior to transformation into the spectral domain, via the FFT (line 6). From the output, the power spectrum is extracted (line 7) and truncated (line 8). This results in each instance containing 3,000 attributes, representing the power at each frequency from 1 hz to 3,000 hz.

In algorithm (9), T2, each instance is transformed at the original sample rate. The original sample rate is also used as the number of FFT bins (line 4). From the FFTs output, the power spectrum is extracted (line 5) and truncated at 100 hz and 2,000 hz (line 6). This results in each instance containing 1,900 attributes, representing the power at each frequency between 100 hz and 2,000 hz. This method of transformation

Algorithm 9 Transform 2

```
function TRANSFORM($M$, $samplerate$)
    Let $M \leftarrow M_1 \ldots M_n$ be a dataset.
    for $i \leftarrow 1$ to $n$ do
        $M_i \leftarrow fft(M_i, samplerate)$
        $M_i \leftarrow getPowerSpectrum(M_i)$
        $M_i \leftarrow truncate(M_i, 100, 2000)$
    return M
```

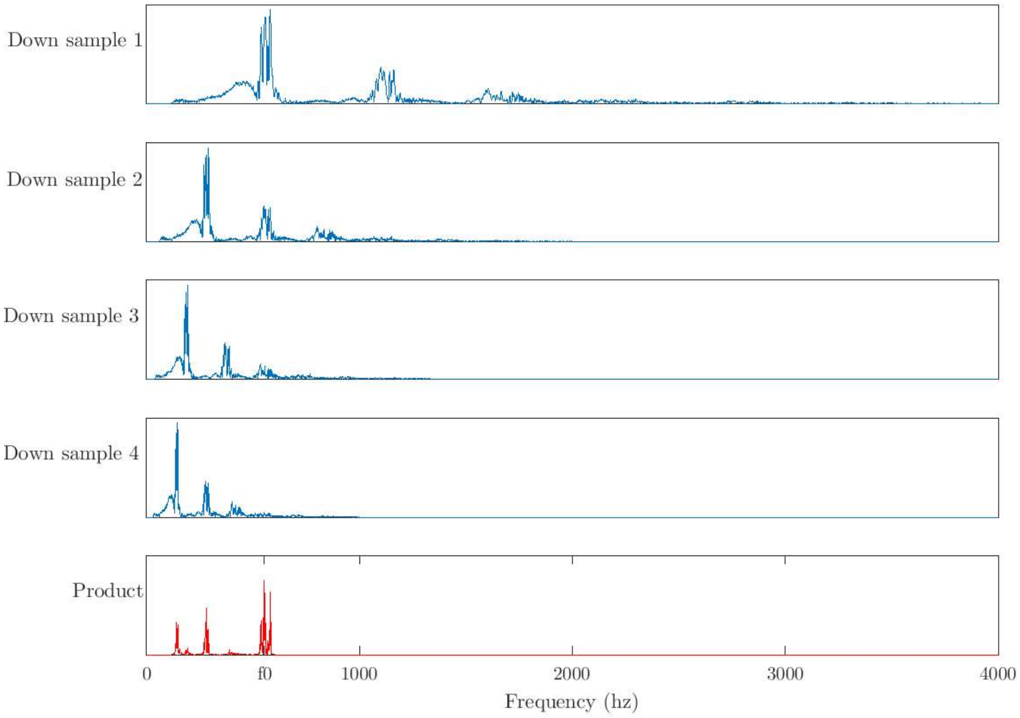
was used in [16] by Chen. et. al. and forms part of the approach used in the evaluation undertaken during the curation of the InsectSound dataset. Moving forward this approach is also referred to as the UCR transform.

As discussed in Section 2.3, investigations into the classification of flying insects typically start with wingbeat frequency. In conjunction with results from the raw signal, the fundamental frequency feature is a logical point for comparison when discussing the performance of classification approaches. However, classification accuracy achieved with this feature is often significantly worse than that gained from the whole spectra, although it is often significantly faster in real terms. Until recently, psudeo-acoustic insect-centric datasets were relatively small, often not in excess of ten thousand instances in total. The results of the studies discussed in Chapter 2 were often inconclusive and they commonly concluded that wingbeat frequency alone is not an adequate predictor of class.

The fundamental frequency of an insect's wingbeat whilst flying follows a normal distribution across a population and there is often large intra-class variability. Classes which do not significantly differ morphologically, exhibit similar wingbeat motions. This often leads to substantial overlap between wingbeat frequencies of different classes, a problem that is only made worse, as the number of classes increase.

Commonly, an audio interval is transformed into the spectral domain at one resolution, in order to determine its fundamental frequency component. However, in some cases, particularly if the target signal is not pure, this approach is susceptible

Fig. 5.1 A figure showing the power spectrum of a wingbeat recording down sampled at 4 increasing rates in blue and, the harmonic spectral product of these 4 down sampled in red. f0 then indicates the dominating frequency from the series.



to errors. To combat this, the Harmonic Spectral Product approach is used. This approach is more robust to errors introduced by noise, as it takes into account the repeating nature of the harmonic property, illustrated in Figure 5.1.

5.4 Results

In order to produce robust results from which to draw our conclusions, all results are the product of experiments undertaken on 10 stratified folds of the data. In the interest of producing reproducible results, all random functions used to produce data folds were seeded with the fold index.

The rest of this section is organised as follows: In section 5.4.1 we evaluate the accuracy achieved by $1\mathrm{NN}_{\mathrm{ED}}$ and the Naive Bayes approaches, using just the fundamental frequency attribute. In section 5.4.2 we investigate the performance of approaches in conjunction with spectral features and in section 5.4.3 we present and discuss all approaches with regard to timing.

5.4.1 Fundamental Frequency

The figures presented in Section 2.5 shows the distribution of fundamental frequencies present in each of the insect-centric datasets. It is clear from these that even when dealing with a relatively small number of classes, the ability to discriminate between them using just the wingbeat frequency is likely to be limited. This is further evidenced by experiments performed using the fundamental frequency attribute.

The fundamental frequencies of the instances in both the MosquitoSound and InsectSound datasets were extracted using a peak finding algorithm, in conjunction with the harmonic product spectrum technique, described in Section 5.3. Table 5.1 displays results from experiments undertaken with these datasets. For the InsectSound dataset the accuracies of both approaches is around 55%. The problem consists of 10 equal sized classes and in the test set the number of instances in each class is 2,500. The accuracy of each approach on the MosquitoSound dataset is also around 55%. In this case the dataset consists of 6 equally sized classes, each with 2,500 instances in the test set.

However, an analysis of the confusion matrices in Figure 5.2 show that both the Naive Bayes and 1-NN$_{\mathrm{ED}}$ approaches produce low type II errors in relation to female mosquito classification. A comparison of the misclassifications pattern shows the approaches are extremely similar, although the number of mis-classifications does vary slightly. In the case of the InsectSound dataset, misclassification is strongly correlated to sex. Cross-referencing the class indices with Table 2.6, shows that the female classes: 1, 5, 7 and 9, are most commonly misclassified as a different female class. This is also

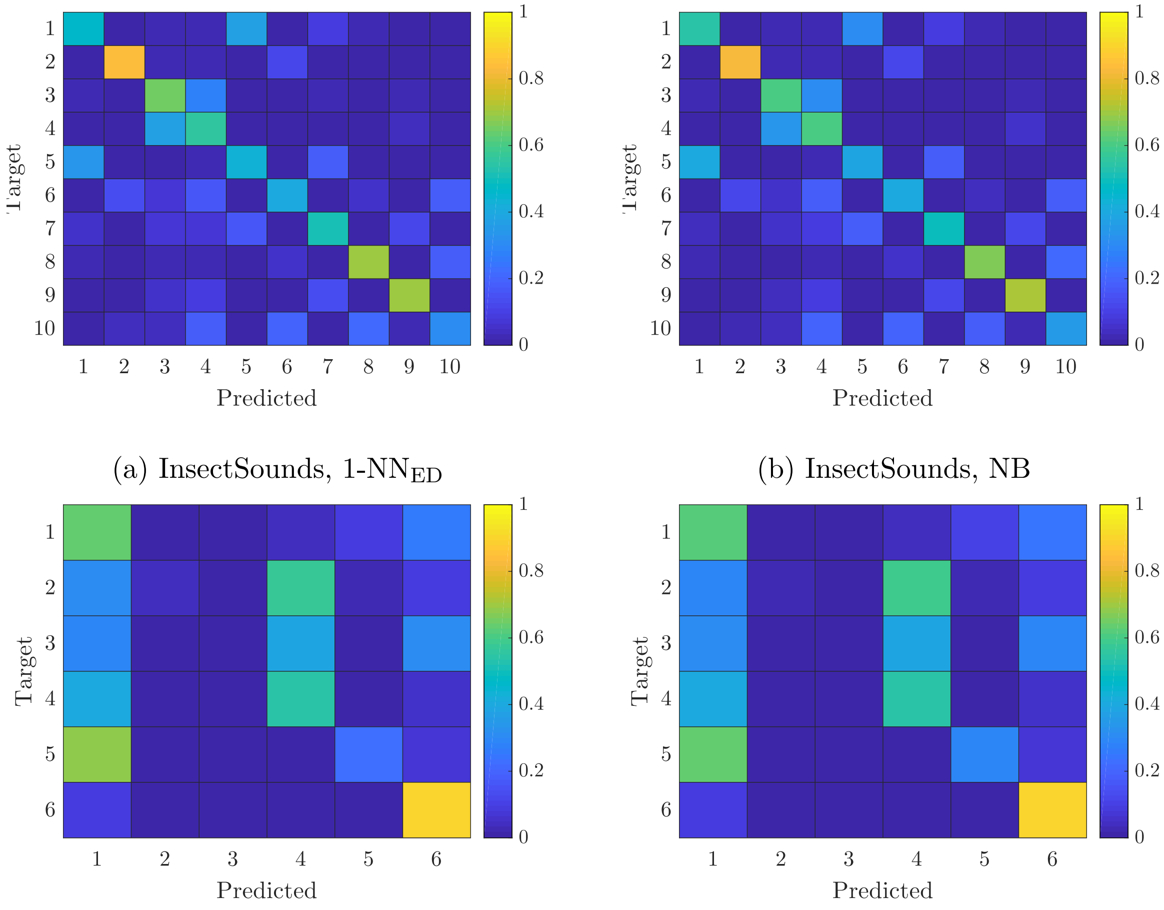
Table 5.1 Table showing mean accuracy, the Area under the reciever operator curve (AUROC) and Negative Log Likelihood (NLL) for 1 Nearest Neighbour with Euclidean Distance and Naive Bayes approaches, evaluated over 10 folds on the fundamental frequency attribute of the MosquitoSound (6 classes) and InsectSound (10 classes) datasets.

Dataset	Classifier	Accuracy	AUROC	NLL
InsectSound	1-NN$_{\mathrm{ED}}$	0.5540	0.9104	1.6184
	NB	**0.5574**	**0.9207**	**1.6006**
MosquitoSound	1-NN$_{\mathrm{ED}}$	0.5551	0.8205	1.6318
	NB	**0.5563**	**0.8240**	**1.6188**

true of the male classes: 2, 6, 8 and 10 and the fly classes, 3 and 4. This pattern is consistent with the information presented in Figure 2.16, where male and female classes are shown to form distinct groups. Furthermore, when confusion between the fly species *Dr. simulans* and *Mu. domestica* and mosquitoes does occur, it is more likely to occur with male classes. In summary, the TPR, as defined in Table 3.1, for female mosquitoes is 0.8918 for 1-NN$_{\mathrm{ED}}$ and 0.8944 for NB and the FNR is 0.1082 for 1-NN$_{\mathrm{ED}}$ and 0.1056 for NB. The confusion matrix for both approaches on the MosquitoSound dataset provides little insight. The high levels of confusion between classes, particularly the tendency not to predict class 3, is unsurprising when considering the frequency distribution shown in Figure 2.19. For example, the tendency to predict class 4 is driven by the large variance exhibited, and absence of class 3 predictions is driven by substantial inter-class overlapping.

5.4.2 Spectral Approaches

Table 5.3 shows the results of cRISE, 8-NN$_{\mathrm{ED}}$, BOSS, TSF, STC and HIVE-COTEv1. For each approach, results are presented for the raw dataset as well as the T1 and T2 transformed datasets. The results shown confirm the prior belief that, "*spectral features are most effective*". This is most obvious when looking at the results of BOSS with



(a) InsectSounds, 1-NN$_{\mathrm{ED}}$

(b) InsectSounds, NB

(c) MosquitoSounds, 1-NN$_{\mathrm{ED}}$

(d) MosquitoSounds, NB

Fig. 5.2 Figures showing the confusion matrices for 1-NN$_{ED}$ and NB on the Harmonic Spectral Product transformed InsectSounds and MosquitoSounds datasets.

respect to MosquitoSound, where there was an increase of 30% in accuracy between spectral and non-spectral features, and all but STC on the InsectSound dataset achieved a higher accuracy in conjunction with the spectral data. The data from table 5.2 summarises the impact of the transforms and presents the: minimum and maximum difference between the T1 and T2 transforms and the raw data, alongside the mean and median accuracy over all approaches on the transformed datasets. Transformation into the spectral domain has a positive effect on the results obtained on both datasets, with a maximum increase in accuracy of 28.68% on InsectSound, and a 30.09% increase on MosquitoSound. On the InsectSound dataset there is little difference in these measures between transformations. However, there is some notable differences between the two transformations on the MosquitoWingbeat dataset, where the minimum and maximum values differ by 2.17% and 3.45% respectively. The difference in transformation performance is also reflected in Table 5.3. On the InsectSound data the performance of the two transforms is largely tied. Both transforms result in a worse performance with STC. The T1 transform then performs best with the HIVE-COTEv1 and TSF. Whereas, the T2 transform performs best with the BOSS and 8-NN$_{\text{ED}}$. Furthermore, the T1 approach produces the largest increase in accuracy and T2 transform produces a marginally better mean accuracy. On the MosquitoSound dataset the T1 transform outperforms the T2 transform. The T1 transform has a positive affect on accuracy for all approaches and performs best on all but 8-NN$_{\text{ED}}$, whereas the T2 transform results in a negative affect on the HIVE-COTEv1 approach. The T1 transform also produces a greater mean and median accuracy, as well as a greater minimum and maximum difference in accuracy. The variations in performance between the two datasets is most likely the result of the physical differences in the hardware used to to produce the datasets. The TEIC hardware, used to produce the MosquitoSound dataset, and discussed in Section 2.6b, has a larger target area. This results in insects being recorded for a greater duration and ultimately results in signals containing more low energy information, information which the T-2 approach discards. As a result, the T-1 transform produces a better performance when low frequency information is present, whilst producing a similar performance to T-2, when it is not present.

Overall, HIVE-COTEv1 performed best on InsectSound with an accuracy of 0.7895, an increase of 12.69% on the approach described by Chen et al. [16], 8-NN$_{\text{ED}}$+T2. However, these results omit powerful time-of-flight information, an attribute that is reported to have increased the accuracy of the 8-NN$_{\text{ED}}$+T-2 combination by 10% on the Insect-

Table 5.2 A table summarising the results in Table 5.3, showing the minimum and maximum difference between raw and transformed data, as well as the mean and median accuracy.

		Min	Max	Mean	Median
InsectSound	T1	-0.0446	0.2868	0.5924	0.7158
	T2	-0.0371	0.2811	0.5931	0.7233
MosquitoSound	T1	0.0011	0.3009	0.6121	0.6885
	T2	-0.0228	0.2664	0.6083	0.6745

Sound dataset. cRISE was the most accurate approach on the MosquitoSound dataset with an accuracy of 0.7558, an increase of 19.61% on the 8-NN$_{\mathrm{ED}}$+T-2 combination.

Table 5.3 Table showing mean accuracy, AUROC and NLL over 10 folds for STC, TSF, cRISE, 8-NN$_{\text{ED}}$, BOSS and CAWPE ensembles, HIVE-COTEv1 (HC1) for T-1, T-2 and no spectral transformation.

Dataset	Classifier	Transform	Accuracy	AUROC	NLL
InsectSounds	HC1	T-1	0.7895±0.0150	0.9811	1.0855
	HC1	T-2	0.7800±0.0145	0.9800	1.0555
	HC1	none	0.7740±0.0021	0.9758	1.2669
	STC	none	0.7604±0.0026	0.9743	**0.9833**
	cRISE	n/a	0.7347±0.0037	0.9656	1.4938
	TSF	T-1	0.7313±0.0016	0.9701	1.0415
	TSF	T-2	0.7256±0.0029	0.9690	1.0133
	STC	T-2	0.7233±0.0041	0.9695	1.0184
	STC	T-1	0.7158±0.0041	0.9683	1.0608
	BOSS	T-2	0.6668±0.0106	0.9496	1.2531
	8-NN$_{\text{ED}}$	T-2	0.6626±0.0073	0.9308	1.9543
	BOSS	T-1	0.6620±0.0157	0.9474	1.2650
	8-NN$_{\text{ED}}$	T-1	0.6556±0.0068	0.9275	2.0613
	BOSS	none	0.5751±0.0164	0.8962	1.9126
	8-NN$_{\text{ED}}$	none	0.5639±0.0053	0.9009	2.7342
	TSF	none	0.4445±0.0100	0.8480	2.3956
MosquitoSounds	cRISE	n/a	0.7558±0.0043	0.9492	1.8608
	HC1	T-1	0.7532±0.0062	0.9512	1.0745
	HC1	none	0.7521±0.0068	0.9484	1.3210
	TSF	T-1	0.7408±0.0046	0.9455	**1.0019**
	HC1	T-2	0.7293±0.0058	0.9422	1.1294
	TSF	T-2	0.6950±0.0040	0.9272	1.1167
	STC	T-1	0.6885±0.0069	0.9271	1.1679
	STC	T-2	0.6745±0.0039	0.9197	1.2021
	STC	none	0.6020±0.0396	0.8836	1.5147
	BOSS	T-1	0.5942±0.0849	0.8534	1.8636
	BOSS	T-2	0.5597±0.0790	0.8390	1.9558
	TSF	none	0.4778±0.415	0.8067	1.9673
	8-NN$_{\text{ED}}$	T-2	0.3829±0.0119	0.7257	4.3566
	BOSS	none	0.2933±0.470	0.6208	3.3282
	8-NN$_{\text{ED}}$	T-1	0.2840±0.0069	0.6402	5.3337
	8-NN$_{\text{ED}}$	none	0.2539±0.0051	0.5885	6.1839

Figure A.1 presents the confusion matrices produced from the predictions of HIVE-COTEv1 on the InsectSound dataset and cRISE on the MosquitoSound dataset. As expected by the difference in accuracy between HIVE-COTEv1, cRISE and NB, the plots generated from the HIVE-COTEv1 and cRISE reults show significantly less confusion than those of NB, shown in Figure 5.2. A comparison to the FNR's from Sub-Section 5.4.1 show the largest difference is in the male mosquito group where, HIVE-COTEv1 achieves a FNR of 0.0112, an improvement of 0.1478. However, the FNR of the female grouping is also improved at 0.0390, a difference of 0.0666. Analysis of the cRISE MosquitoSound confusion matrix highlights the extent to which spectral information can help differentiate between classes. The confusion matrix shows that classes are most commonly misclassified as classes of the same genus. This is apparent when comparing the comparing classes 1 and 2 or 3 and 4.

5.4.3 The Relevance Of Test Time.

The successful application of classification algorithms in real world scenarios also require them to be timely. It is commonly accepted that an algorithm is 'real time' if it is able to classify an instance in less time than is represented by the data. Instances from the MosquitoSound represent 620 milliseconds and those from the InsectSound represent 100 milliseconds.

Figure 5.3 plots mean test time per instance, which is averaged over folds for each approach. The timing data was generated during experiments run on the spectral datasets, the results of which were discussed in Sub-Section 5.4.2. Results of non-spectral experiments have been omitted in the interest of brevity.

In all cases, TSF performs best and in a timely manner with respect to relative instance length. The results exhibited very little variance across folds. In respect to timing, the UCR transformation approach performs best overall. This is most

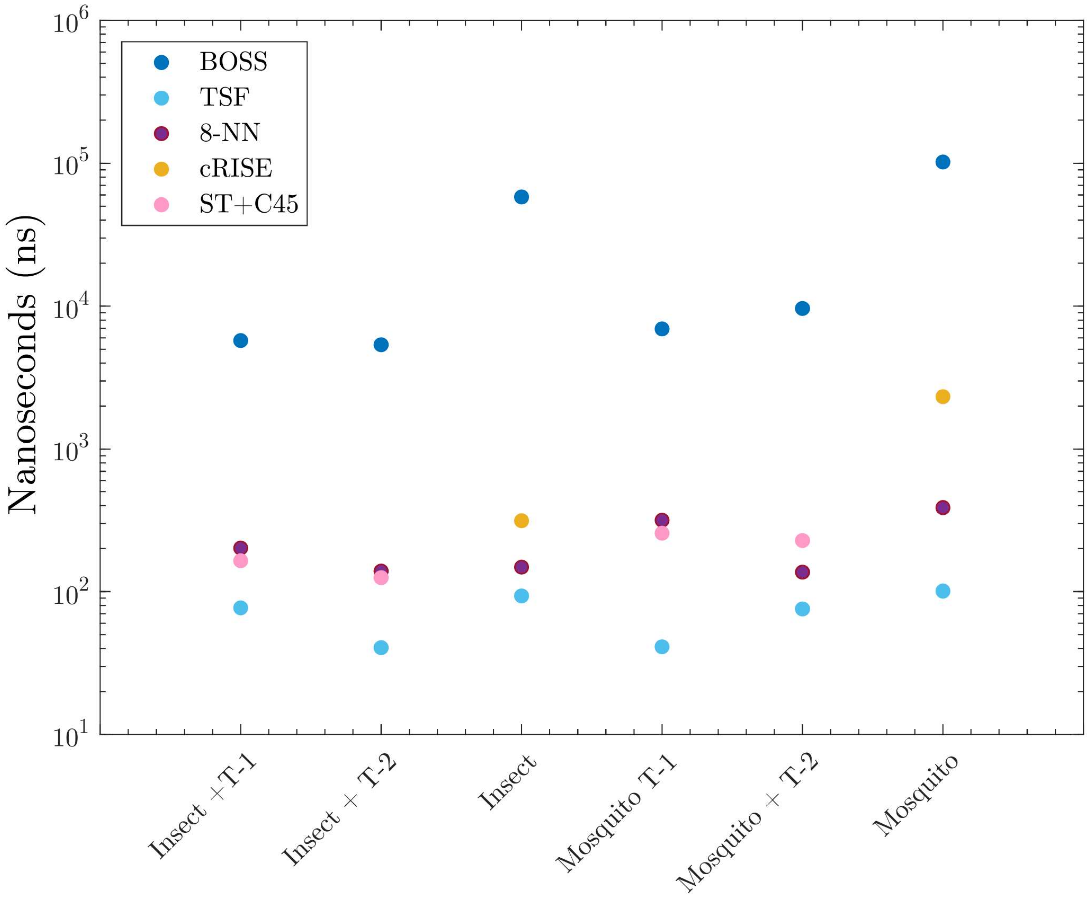


Fig. 5.3 Figure showing mean test time per instance for all combinations of STC, TSF, cRISE, 8-NN$_{\mathrm{ED}}$, BOSS, CAWPE ensembles, with no spectral transformation, T-1 and T-2 transformations.

clear when comparing InsectSound+T-1 and InsectSound+T-2 with respect to TSF. This is likely to be because of the difference in the number of attributes present for classification, where the T1 transform results in less.

5.5 Conclusion

In conclusion, the work presented in this chapter has shown that the combination of simple audio features and HIVE-COTEv1 performs best on the InsectSound dataset and second best on the MosquitoSound dataset, beaten only by cRISE. It is acknowledge that no deep learning approaches were included in this evaluation and refer the reader to Chapter 6, where the application of state-of-the-art deep learning classification approaches to the insect classification problem is addressed. HIVE-COTEv1 in conjunction with spectral features is shown to be 12.69% more accurate than than a previously published benchmark approach on the InsectSound dataset and the cRISE approach was found to be 19.61% more accurate than the same approach on the new MosquitoSound dataset, despite both omitting powerful time of flight information. In both cases the best performing approach does not produce test estimates in a timely manner. Futhermore, as the cBOSS constituent of HIVE-COTEv1 is very slow, even threaded HIVE-COTEv1 would not produce results in real time and therefore would need a considerably faster processor to meet the requirements of an application setting.

The InsectSound and MosquitoSound datasets provide an opportunity to comment on the feasibility of classifying insects, based on species and genus and the InsectSound dataset provides further information relating to the mosquito sex. The accuracy of classifying fly species using the relatively simple $1\mathrm{NN}_{\mathrm{ED}}$ and NB approaches in conjunction with fundamental frequency, produced a poor result. In all cases, the reported accuracy was below 56%. This is unsurprising and mirrored all published results from the literature. An examination of fundamental frequency distributions from Figure 2.5.3 provide a suitable explanation for the performance and shows that the level of intra-class variance and subsequent overlapping between classes is high. However, further examination of the associated confusion matrices revealed that when framed as binary 'fly vs mosquito' and 'female vs all' problems, the performance is surprising, with an accuracy of 0.8944. The application of sophisticated TSC approaches to the insect wingbeat datasets, provided a promising indication that automatic classification could be viable. The approach that performed best on the InsectSound dataset, with regards to accuracy, was HIVE-COTEv1 whith an accuracy of 0.7951. This translated to a FNR of 0.0279 with respect to the classification of female mosquitoes in the InsectSound dataset. On the MosquitoSound dataset the best performing approach was cRISE with an accuracy of 0.7532.

It can thus be concluded that intra-class variance in fundamental frequency prevents its use as a discriminant feature on a species level. However, analysis of confusion matrices from both the fundamental frequency and spectral series experiments has shown it to be a powerful feature in determining the sex of a mosquito, or potentially whether the target is a mosquito species. In a real world setting, this feature is likely to play a key role. It is argued here that an appropriate algorithm architecture would consist of layers designed to minimise power consumption, by preventing unnecessary computations. In this context, fundamental frequency could prove an adequate method of determining whether to apply more sophisticated methods to incoming intervals. In cases where the sophisticated methods are utilised they could be done so with high dimensional audio data. By virtue of the recording process this data bares hallmarks of the insects morphology, such as body size or wing shape. providing an opportunity for the extraction of more robust features.

Chapter 6

Insect Classification II

6.1 Introduction

In this chapter, raw and spectral representations of multiple datasets are used to benchmark approaches. Folling this an evaluation of whether features such as time of flight and fundamental frequency can contribute to accuracy in the context of insect classification is undertaken. This is achieved via an investigation into methods of combining simple classifiers trained on expert features, with state of the art approaches. The methods are split into 2 categories: hierarchical, designed to filter out cases before they are classified by the more complicated stat-of-the-art approach and cumulative, designed to maximise accuracy. The expectation is that powerful approaches, such as HIVE-COTEv2.0 and InceptionTime will benefit from spectral data representations. However, the results show that contrary to expectation, the best results are not achieved in conjunction with spectral data. Furthermore, results show that convolutional features learned in approaches such as Arsenal [62], RESNET[45] and InceptionTime [33] on the raw audio datasets are superior, with the best performance being achieved by the InceptionTime algorithm.

The remainder of this chapter is laid out as follows: In Section 6.2 the methods of transformation and structure of experiments is described. In Sections 6.3, 6.4 and 6.5, the results from experiments on expert features, the raw series and spectral series are discussed. In Section 6.6 an ablation study is undertaken on HIVE-COTEv2.0, where the cRISE$_{\text{All}}$ approach is also introduced. In Section 6.7 methods of combining test distributions are defined before their respective performances are discussed. Finally, in Section 6.8 some conclusions are presented.

6.2 Experimental Methodology

For the evaluation presented in this chapter four publicly available pseudo-acoustic insect datasets were used. These were described in detail in Section 2.5 and summarised in Table 2.3. In all four cases, perspex boxes were used to confine insects of differing classes for recording. The Aphids, FruitFlies and the MosquitoSound dataset were all recorded with hardware developed at TIEC, whereas, the InsectSound dataset was recorded with hardware developed at UCR. Both systems are described in Section 2.4. The information present in each recording is the result of partial or total occlusion of the infrared signal from the photodiode during flight. It can be interpreted as audio and the data captured is similar to that of conventional audio recording devices [74]. In order to maintain consistency and ensure a reasonable chance of obtaining results both the InsectSound and MosquitoSound datasets have undergone the T-1 preprocessing steps outlined in Section 5.3.

All results presented are the mean over 30 experiments undertaken on stratified random re-samples of each dataset. The dataset splits are created with a seed values 0 - 29 to ensure they are reproducible. Furthermore, each dataset exists in 5 forms: raw series; spectral series; time of flight (TOF); fundamental frequency (HSP) and, time of flight and fundamental frequency combined. The order of instances between all representations were maintained such that instance 1 of the raw dataset was used to derive instance 1 in all alternative representations.

For all approaches other than TDE both the train and test phases were completed within 7 days without intervention. In the case of TDE a 24 hour contract on the training phase was enforced on all datasets but AphidsSpec. This was after at least one fold exceeded the 7 day execution limit of the UEA HPC for 120, 80, 60, 40 and 30 hour train time contracts.

A link between time of day and insect activity has long been established [43]. However, until recently there has been very little opportunity to asses how this information might impact classification accuracy. In one study Chen et. al.[16] showed that the use of TOF information was attributed to an increase in classification accuracy of 10%. All datasets used in this study were captured over multiple, not necessarily contiguous, days. The time at which each recording was made is also included with each dataset alongside, the raw time series. Inclusion of this information allows us to investigate how TOF may impact classification accuracy across multiple large datasets. Figure 2.12 summarises the TOF information included with the FruitFlies dataset. It is clear from Figure 2.12 that the level of activity at any given time of day differs per class. The TOF information is agnostic of date. It is expressed in minutes, as a result a value of 1 indicates a recording captured at 00:01 and a value of 1439 indicates a recording captured at 23:59.

In conjunction with results from the TOF, the fundamental frequency feature is a logical, relatively accessible and computationally fast starting point for comparison, when discussing the performance of classification approaches. Typically, accuracy in relation to this feature is poor. However, this is only true when classes are morphologically similar, e.g. the same sex. Classes which do not significantly differ morphologically, exhibit similar wingbeat motions. This often leads to substantial overlap between wingbeat frequencies of different classes, a problem that is only made worse as the number of classes increases. Commonly, an audio interval is transformed into the spectral domain at one resolution in order to determine its fundamental frequency component. However, as discussed in Section 5.3 the HSP approach 10 is favoured. This approach takes advantage of the repetitive nature of harmonics and measures the extent to which harmonics align at different spectral frames. Technically, the HSP algorithm is designed to determine the pitch of audio, which may differ from the fundamental frequency in polyphonic examples. However, in monophonic examples, where there is a dominating source of data, the pitch and fundamental frequency align. As a result, the approach provides the ability to measure the fundamental frequency of wingbeats that is robust to low level noise.

Algorithm 10 The Harmonic Spectral Product algorithm

```
Require: A time series, $\boldsymbol{x}$, and it's corresponding samplerate, $fs$.
Ensure: The fundamental frequency, $f0$, of $\boldsymbol{x}$.
procedure HSP($\boldsymbol{x}$, $fs$)
    Let $\boldsymbol{F}$ be a highpass filter with a stop band frequency of 100hz and an attenuation of 60db.
    $\boldsymbol{x} \leftarrow \text{filter}(\boldsymbol{F}, \boldsymbol{x})$
    $\boldsymbol{x} \leftarrow |\ \text{fft}(\boldsymbol{x}, fs)\ |$
    $\boldsymbol{d^1} \leftarrow \boldsymbol{x_0} \ldots \boldsymbol{x_{(fs/2)}}$
    $\boldsymbol{d^2} \leftarrow \text{downSample}(\boldsymbol{d^1}, 2)$
    $\boldsymbol{d^3} \leftarrow \text{downSample}(\boldsymbol{d^1}, 3)$
    $\boldsymbol{d^4} \leftarrow \text{downSample}(\boldsymbol{d^1}, 4)$
    for $i \leftarrow 1$ to length($\boldsymbol{d^4}$) do
        $p \leftarrow \boldsymbol{d_i^1} \times \boldsymbol{d_i^2} \times \boldsymbol{d_i^3} \times \boldsymbol{d_i^4}$
        $\boldsymbol{y_i} \leftarrow p \div 4$
    $f0 \leftarrow \max(\boldsymbol{y})$
    $f0 \leftarrow f0 \times 2$
    return $f0$
```

In Chapter 5, it was shown that the spectral series produces superior performance with respect to accuracy. The spectral series of an audio excerpt provides information regarding a signals spectral composition, via a function such as Fourier transformation information regarding the power and phase of each frequency up to the Nyquist limit can be obtained. In this work, each instance was transformed into the spectral domain via the FFT function. Each datasets corresponding sample rate was used to produce a full resolution spectral series.

The remainder of this chapter is laid out as follows. In Sections 6.3, 6.4 and 6.5 experimental results are presented and discussed. In Section 6.7 multiple methods of combining test distributions are presented, before their merits are discussed with respect to accuracy and test time. Finally, in section 6.8, some conclusions are presented.

6.3 Expert Features

As discussed in Section 6.2, each dataset has been transformed into 3 additional representations. In this sub-section, the results of experiments undertaken on the HSP and TOF representations are presented in Table 6.1. These two attributes represent the most obvious and easily collected features in the context of insect classification and as such, provide a sensible benchmark. As discussed in Section 2.3, historically the accuracy obtained via experiments using the fundamental frequency is often used as a starting point, on which further experiments expand. However, it is well established that when used alone, it makes a poor discriminant feature [80] [81] [85]. This is particularly true in cases where the aim is to differentiate between insects of different species of the same genus and sex, where morphological differences are slight.

Information on TOF is easily collected and requires almost no processing before its incorporation into a model. As an attribute, its effectiveness as a discriminant feature is relatively unexplored. However, the relationship between Circadian rhythm and activity in some insect species is well documented. Figure 2.9 visualises TOF information for the Aphid dataset. It shows that level of activity differs as a function of time and that each classes profile is distinct.

Table 6.1 Table showing mean accuracy over 30 folds for expert features datasets.

	Default	C4.5	BayesNet	ED	SVML	NB
Aphids HSP+TOF		**0.7327** ±0.004	0.7260 ±0.006	0.6438 ±0.006	0.4650 ±0.017	0.4691 ±0.002
Aphids HSP	0.4203	**0.7150** ±0.004	0.7034 ±0.006	0.7071 ±0.004	0.4470 ±0.002	0.4669 ±0.002
Aphids TOF		**0.5071** ±0.006	0.5058 ±0.005	0.4486 ±0.004	0.4202 ±0.000	0.4082 ±0.003
FruitFlies HSP+TOF		**0.7233** ±0.002	0.7126 ±0.005	0.6444 ±0.003	0.6745 ±0.001	0.6501 ±0.004
FruitFlies HSP	0.5305	**0.6947** ±0.001	0.6942 ±0.001	0.6922 ±0.001	0.6733 ±0.002	0.6287 ±0.008
FruitFlies TOF		0.5515 ±0.003	0.5430 ±0.003	**0.5582** ±0.003	0.5305 ±0.000	0.5264 ±0.001
InsectSound HSP+TOF		0.6962 ±0.002	**0.7002** ±0.002	0.6221 ±0.002	0.5192 ±0.006	0.5447 ±0.007
InsectSound HSP	0.1000	0.5353 ±0.009	0.5472 ±0.006	0.5825 ±0.005	**0.6899** ±0.005	0.5535 ±0.006
InsectSound TOF		0.2816 ±0.003	**0.3070** ±0.002	0.2239 ±0.001	0.1961 ±0.001	0.2254 ±0.001
MosquitoSound HSP+TOF		**0.6289** ±0.001	0.6203 ±0.001	0.5153 ±0.001	0.5382 ±0.004	0.5413 ±0.000
MosquitoSound HSP	0.166	**0.5568** ±0.000	0.5563 ±0.000	0.5552 ±0.000	0.5415 ±0.002	0.5330 ±0.000
MosquitoSound TOF		**0.4242** ±0.000	0.4229 ±0.000	0.4196 ±0.000	0.3060 ±0.000	0.3277 ±0.000

The results in Table 6.1 are grouped by parent dataset. Each group is then ordered such that the top row indicates the configuration that produced the highest accuracy, it also shows the accuracy of an approach that selects only the most prevalent class in the training set for all test instances, providing an indication of true transform/classifier combination benefit. Information on class distribution can be found in Table 2.3. The results shows that the C4.5 tree is most accurate in 8/12 datasets overall and 3/4 datasets when only considering the TOF+HSP combination. In all groups, the best performance exceeds the default approach of picking the most prevalent class present in the training set.

The results show that the effectiveness of TOF and HSP features vary with respect to the dataset characteristics. For instance, relative to HSP the TOF feature fares poorly on the sex separated InsectSound compared to the MosquitoSound dataset where classes are not sex separated. This is likely to be due to the fact that circadium rythm is correlated to specie not sex, as shown in Figure 2.15. However, the HSP+TOF combination always outperforms its constituents. This indicates that despite the low performance of the TOF feature, it adds additional insight to class membership. This is likely to be impacting instances for which the HSP value falls into a region of overlap between neighbouring classes.

Figures B.1, B.2, B.3 and B.4 show the confusion matrices for the results presented in Table 6.1. These reveal that the accuracy achieved with respect to the Aphids dataset fails to accurately reflect the performance of the C4.5 approach. Despite being a minimum of 8% more accurate than selecting the majority class, the approach fails to predict the *M. persicae*, *P. testudinaceus*, *Ps. chrysocephala* or *R. padi* classes on all three representations of the data. These classes are small and this is certainly a factor in the performance. However, the performance of expert features on FruitFlies dataset, where all classes consist of a reasonable number of instances, is also pathological. The difference between the achieved accuracy and the hypothetical majority class approach is 19.28%. In this case, the accuracy is primarily the result of a strong performance on the zaprionus class. Whereas, predictions for the melanogaster class are at best random, when using the TOF+HSP representation. Furthermore, analysis of the MosquitoWingbeat confusion matrices reveals a similar pattern. The TOF feature results in no predictions for classes 2, 3 and 5 and the HSP feature results in no predictions for classes 2 and 3. Whereas, the TOF+HSP combination does produce predictions for all classes. In all 3 datasets results from expert features are better than the default approach. These performances are the result of a strong performance on either 1 or 2 classes and reveal that the predictive power of expert features is not balanced across all target classes. In all 3 datasets the combination of TOF and HSP does improve predictions. But, the improvement is marginal. However, this trend of pathological predictions does not extend to the InsectSound dataset. In this case the TOF and HSP representations produce distinct and patterns of confusion. The TOF feature produces a clear split between classes of the *Culex* genus and the remaining classes. Whereas, the HSP feature forms 3 groups: flies, male mosquitoes and female mosquitoes. With classes from each group more likely to be misclassified as classes from that group.

Table 6.2 TPR of 5 subgroups of the TOF, HSP and TOF+HSP InsectSound datasets.

	Mosquito	Female	Male	Fly	Aedes	Culex
TOF+HSP	**0.9640**	**0.9209**	**0.9237**	**0.8730**	**0.8513**	**0.9269**
HSP	0.8909	0.8944	0.8443	0.8224	0.6951	0.7502
TOF	0.8457	0.5699	0.5183	0.5696	0.2032	0.9175

Table 6.2 quantifies the effectiveness of the 3 representations on the InsectSounds dataset. The table shows the TPR for 6 class subgroups from the data. The values are derived from the confusion matrices, which in turn are derived from the test

predictions over all folds. The results suggest that the classes from the *Culex* genus exhibit a significantly different TOF profile from that of the 'Fly' and *Aedes* classes. Consultation of the TOF plot in Figure 2.15 shows that despite sharing periods of high activity during the twilight hours the *Culex* genus have nocturnal tendencies. The high accuracy achieved in this case is certainly because the other classes are predominately active during the day and the poor performance of differentiating between the Fly and *Aedes* groups highlights how fragile the TOF feature is. The TPR of the HSP feature, presented in Figure 2.16, shows the best performance when classifying sex - where randomly selecting classes would achieve 50% accuracy. This is unsurprising, as sex separated groups have a lower intra-group variance than genus groups and the Fly group has very little overlap with the mosquito groups. Unsurprisingly, the combination of features improves the TPR of all groupings. Furthermore, the TPR of the Mosquito and Fly groupings lend weight to the idea of using expert features to filter incoming data that is unlikely to be a mosquito recording.

6.4 Raw Series

Table 6.3 presents the mean accuracy for 8 approaches on the raw time series of each dataset. Traditionally, benchmark results are not achieved using raw time domain representations of audio data. This is demonstrated in Chapter 2. Typically, approaches which perform an explicit spectral transform internally produce significantly higher accuracies. However, deep learning approaches have been shown to produce competitive accuracies by extracting meaningful features via convolutional layers.

The results in Table 6.3 show that InceptionTime (IT) is most accurate on 3 out of 4 datasets. Whereas the spectral approach cRISE$_{\text{ALL}}$ presents an average performance which outperforms the TDE approach on all datasets and the STC approach on all but the InsectSound, but looses out to the resident HIVE-COTEv2 spectral approach, DrCIF, in all four cases. As expected, a comparison between these results and the expert feature experiments given in Table 6.1, confirms that InceptionTime in conjunction with raw data, also produces considerably higher accuracies than reported approaches used in conjunction with expert features. Furthermore, in a comparison between these tables, we see that in 29/32 of cases approaches in conjunction with the raw data outperform the corresponding highest performing expert feature combination, the exceptions being the TDE with InsectSound and MosquitoSound and STC with MosquitoSound.

Table 6.3 Table showing accuracy for approaches in combination with raw data.

Datasets	$\mathrm{cRISE}_{\mathrm{All}}$	TDE	Arsenal	DrCIF	STC	$\mathrm{HC}_{\mathrm{v2}}$	IT	RESNET
Aphids	0.9391	0.9075	0.9556	0.9521	0.9117	0.9570	**0.9744**	0.9502
FruitFlies	0.8700	0.7975	0.9633	0.9269	0.8074	0.9650	0.8721	**0.9794**
InsectSound	0.7589	0.5461	0.7923	0.7761	0.7594	0.7992	**0.8545**	0.8269
MosquitoSound	0.7835	0.4138	0.8152	0.7960	0.6153	0.8240	**0.9058**	0.7976

Table 6.4 TPR of the presented approaches on 5 subgroups of the InsectSound dataset.

	Mosquito	Female	Male	Fly	Aedes	Culex
InceptionTime	**0.9865**	**0.9714**	0.984	**0.9656**	**0.8838**	**0.9424**
$\mathrm{HC}_{\mathrm{v2}}$	0.9851	0.9582	**0.9867**	0.9406	0.8221	0.9190
RESNET	0.9811	0.9597	0.9803	0.9421	0.8552	0.9314
Arsenal	0.9858	0.9585	0.9842	0.9304	0.8212	0.9167
DrCIF	0.9773	0.9516	0.9846	0.9644	0.8026	0.9067
STC	0.9799	0.95	0.9794	0.936	0.787	0.9023
$\mathrm{cRISE}_{\mathrm{All}}$	0.9775	0.9467	0.9821	0.9482	0.8	0.9123
TDE	0.9521	0.8711	0.9023	0.8332	0.5878	0.8459

In all cases, the performance of approaches in conjunction with the raw data representations out-performs the approach of selecting the most prevalent class in the training set. Furthermore, a review of the confusion matrices from each approach/dataset combination showed that there was only pathological behaviour on the Aphids dataset, where all but the InceptionTime approach failed to make predictions for one or more classes.

Tables B.2, B.3, B.4 and B.5 show the results of pairwise paired t-tests for all approaches on the raw data. The critical t-values are computed from the accuracies achieved over all resamples. The results show that InceptionTime on the Aphids, InsectSounds and MosquitoWingbeat datasets is significantly more accurate than the other approaches, and that RESENT on the FruitFlies dataset is significantly more accurate than the other approaches.

Table 6.4 shows that for the same 6 subgroups of the InsectSounds dataset all but the TDE approach outperforms the results obtained from the TOF+HSP representation. InceptionTime performs best overall but loses out to HIVE-COTEv2 on the male subgroup. However, the differences between the TPR of the InceptionTime and best performing expert features approach were in some cases marginal. The Fly and Mosquito groups were found to differ by 9.26% and 2.25% respectively, where the difference in the Fly group was the largest and the difference in the Mosquito group was second smallest, behind a 1.55% difference in the *Culex* group.

6.5 Spectral Series

As discussed in Sub-Section 5.4.2, classification of audio problems are often aided by transformation from the time domain into the spectral domain. Typically, the discriminant features identified in the spectral structure are a more informative than those found in the time domain. The benefit of applying a spectral transform can be seen when comparing the performance of the ST approach on the MosquitoSound dataset across Tables 6.3 and 6.5.

In this case, the datasets were transformed in MATLAB and made use of the FFT function. The imaginary portion representing information on phase was discarded and

Table 6.5 Table showing accuracy for approaches in combination with spectral data.

Datasets	cRISE$_{\mathrm{All}}$	TDE	Arsenal	DrCIF	STC	HC$_{\mathrm{v2}}$	IT	RESNET
Aphids	0.9011	0.8840	0.9363	0.9456	0.9427	0.9465	**0.9525**	0.9090
FruitFlies	0.8364	0.7761	0.8870	0.8655	0.7924	0.8880	**0.9009**	0.8398
InsectSound	0.7395	0.6419	0.7532	**0.7620**	0.7259	0.7601	0.6499	0.3319
MosquitoSound	0.7701	0.6554	0.7677	**0.7927**	0.6887	0.7783	0.7337	0.5974

the attributes in the remaining series represents the power of corresponding frequencies. Data up to the Nyquist limit was retained and corrected with respect to amplitude. The sample rate of each dataset was used as the NFFT parameter, ensuring a high spectral resolution. The process is described in more detail in the Sub-Section 4.4.1. The resultant datasets are summarised in Table B.1.

Table 6.5 shows the accuracy of 9 benchmark approaches on the spectral representation of each dataset. The results show that, as expected, the accuracy is reduced in approaches which perform spectral transforms internally, such as cRISE$_{\mathrm{All}}$ and DrCIF. Unexpectedly, these results also show that RESNET, InceptionTime and Arsenal are also negatively effected by the spectral representations.

A comparison between Tables 6.3 and 6.5 show that the highest accuracies achieved by approaches in combination with spectral representations, do not exceed that of the corresponding non-spectral combination. This result highlights the effectiveness of feature creation in deep learning approaches. The RESNET and InceptionTime architectures differ significantly and yet these approaches are both negatively effected by the spectral data. Furthermore, these results show that even the simple features derived in Arsenal, which are also convolutional, are more effective when used with raw audio.

Table 6.6 Table showing HIVE-COTEv2 (HC2) variants for ablation study.

	cRISE$_{\mathrm{All}}$	TDE	Arsenal	DrCIF	STC
All	X	X	X	X	X
HC2		X	X	X	X
1	X		X	X	X
2	X	X		X	X
3	X	X	X		X
4	X	X	X	X	

6.6 Varying HIVE-COTEv2.0 constituents

As discussed in Section 6.4, HIVE-COTEv2 is significantly less accurate than IT. The results in Table 6.3, show that the TDE component of HIVE-COTEv2 does not perform well on any of the four datasets in question. One advantage of the HIVE-COTEv2 structure, is the ease in which additional components can be included, so long as the training set accuracy is available. In Chapter 4 we discus the motivation for including multiple transforms aimed at extracting descriptive features from data of an oscillatory nature and presented the expectation that this approach will excel on audio problems. In the experiments presented in Section 6.4 cRISE$_{\mathrm{All}}$ does not produce an exceptional performance. However, the results show that it does outperform two current constituents of the HIVE-COTEv2 ensemble, TDE and STC. In the majority of cases the approaches were shown to perform best in conjunction with the raw audio.

Table 6.6 shows the approaches included in Section 6.4 ensembled in six combinations, one of which is the standard HIVE-COTE2. The results of these variants are shown in Table 6.7. As expected, inclusion of cRISE$_{\mathrm{All}}$ has a positive effect on accuracy, indicated by the results of the 'All' variant. Furthermore, in variants 1, 3, and 4 where TDE, DrCIF and STC are removed in favour of cRISE$_{\mathrm{All}}$ there is a net positive effect. However, for all the approaches that produce accuracies greater that HIVE-COTE2, the distribution of accuracies over 30 folds overlaps with that of HIVE-COTE2. This is true when looking at performance on individual datasets as well as overall. On the other hand, the performance of variant 2, in which Arsenal is removed in favour of cRISE$_{\mathrm{All}}$, is negatively affected and does not overlap with HIVE-COTEv2. The approach shown in the final column in Table 6.6, HCBR, is an ensemble approach which makes use of 9 constituents. These are: the standard HIVE-COTEv2 approaches built on raw data and on spectral data, and the RISE$_{\mathrm{All}}$ approach. Paired t-tests undertaken per dataset showed that the HCBR approach was significantly more accurate than IT on the FruitFlies and InsectSound datasets and significantly less accurate on the Aphids and MosquitoSound datasets. Moreover, a t-test over all folds of all problems showed that there was no significant difference between the two approaches, although IT does produce a higher mean accuracy overall.

Table 6.7 Table showing mean accuracy and standard deviation for HIVE-COTEv2 (HC2) variants described in table 6.6.

	All	HCV2	1	2	3	4	HCBR
Aphids	0.9574±0.0023	0.9570±0.0023	0.9582±0.0023	0.9483±0.0028	0.9564±0.0022	**0.9587**±0.0022	0.9531±0.0025
FruitFlies	0.9641±0.0015	0.9649±0.0014	0.9652±0.0011	0.9117±0.0039	0.9634±0.0012	**0.9663**±0.0014	0.9466±0.0038
InsectSound	0.8007±0.0020	0.7992±0.0020	0.8004±0.0019	0.7813±0.0024	0.7983±0.0019	0.7996±0.0018	**0.8766**±0.0164
MosquitoSound	0.8270±0.0033	0.8237±0.0033	0.8270±0.0033	0.7957±0.0035	0.8216±0.0031	**0.8285**±0.0033	0.8130±0.0034
Mean	0.8873±0.0022	0.8862±0.0023	0.8877±0.0022	0.8592±0.0032	0.8849±0.0021	0.8883±0.0022	**0.8973**±0.0065

In conclusion, variants in which cRISE$_{\text{All}}$ is present do not produce a meaningful increase in accuracy with respect to the performance of HIVe-COTEv2. However, this difference was shown to be less than one standard deviation. The most accurate HIVE-COTEv2 approach HCBR outperformed the IT approach on two datasets and on one of those, InsectSound, was also shown to differ from HIVE-COTEv2 by more than one standard deviation. However, over all datasets the difference between HCBR and IT was not found to be significant, and the IT approach produced a marginally better accuracy overall. Furthermore, the configuration of HCBR makes the train and test time prohibitive. As a result, we still consider the best approach to be IT.

6.7 Combining features

In the conclusion of Chapter 5 it was argued that easy to capture expert features may be adequate in acting as a filter, preventing expensive processes being run on cases that are extremely unlikely to be mosquitoes. The feasibility of this hierarchical classification pipeline is bolstered by the results presented in Section 6.3, which showed that simple approaches, such as the C4.5 decision tree and Bayesian Networks are effective at discerning Mosquitoes from Flies in the InsectSound dataset.

The TPRs shown in Table 6.2 do not exceed those produced by InceptionTime. However, the potential of utilising expert features does present an opportunity to reduce both the processing power and therefore energy required, and as discussed in Section 5.1 this is desirable in an application setting.

The effect of combining expert features with the best performing series approach, InceptionTime, in conjunction with the raw series is therefore explored in this section. These experiments are undertaken on the InsectSound dataset. A feature of this dataset is sex separated classes. This affords us the opportunity to explore combination techniques that operate at different resolutions - for example, Flies vs Mosquitoes, male vs female and *Aedes* vs *Culex.* This is particularly useful as the real terms cost associated with the misclassification of each class is not equal. The performance of the combination techniques are assessed by the effect on accuracy, and speed up offered relative to the base InceptionTime approach, referred to as InceptionTime$_{\text{Raw}}$.

6.7.1 Combination Methods

Methods of combining the predictive power of the expert features and Inceptiontime$_{\text{Raw}}$ approach fall into two categories, Hierarchical (A) and Cumulative (B). Hierarchical methods aim to utilise the expert features in order to prevent unnecessary processing, whilst the Cumulative methods use the predicted distributions of the expert and series approaches combined in an attempt to increase accuracy. An informal description of the 7 methods are described below and formal descriptions can be found in appendix B.

Hierarchical Methods (A)

A.1. For each test instance retrieve the prediction of BayesNet$_{\text{HSP+TOF}}$ (making use of both the HSP and TOF expert features). If the class predicted is a member of the fly group, accept the expert models probability distribution and prediction. Else, defer the instance to the InceptionTime$_{\text{Raw}}$ approach for classification.

A.2. For each test instance, retrieve the predicted class probabilities of BayesNet$_{\text{HSP+TOF}}$, (making use of both the HSP and TOF expert features). Then compute the cumulative probabilities of the instance belonging to both the fly and mosquito groups. If the cumulative probabilities of the fly group exceed the $belief$ parameter, by default set to 0.8, check that the predicted class is also a fly class. If it is, return the expert models predicted class distribution. If it isn't, return a one hot array that indicates the fly class with the highest predicted probability. If the cumulative probabilities of the fly group doesn't exceed the *belief* parameter, defer the instance to the InceptionTime$_{\text{Raw}}$ approach for classification.

A.3. For each test instance, retrieve the predicted class probabilities of SVML$_{\text{HSP}}$ (making use of just the HSP expert feature). Then compute the cumulative probabilities of the instance belonging to both the fly and mosquito groups. If the cumulative probabilities of the fly group exceed the *belief* parameter, by default set to 0.8, check that the predicted class is also a fly class. If it is, return the expert models predicted class distribution. If it isn't, return a one hot array that indicates the fly class with the highest predicted probability. If the cumulative probabilities of the fly group doesn't exceed the *belief* parameter, defer the instance to the InceptionTime$_{\text{Raw}}$ approach for classification.

A.4. For each test instance retrieve the prediction of SVML$_{\text{HSP}}$ (making use of just the HSP expert feature). If the class predicted is a member of the fly group and the cumulative probabilities of the fly group exceed the *belief* parameter, by default set to 0.8, accept the expert models probability distribution and prediction. Else, take the per class mean of the BayesNet$_{\text{HSP+TOF}}$,(making use of both the HSP and TOF expert features) and InceptionTime$_{\text{Raw}}$ predictive distributions and use the new distribution to find the predicted class.

A.5. For each test instance retrieve the prediction of SVML$_{\text{HSP}}$ (making use of just the HSP expert feature). If the class predicted is a member of the fly group and the cumulative probabilities of the fly group exceed the *belief* parameter, by default set to 0.8, accept the expert models probability distribution and prediction. Else, take the per class mean of the SVML$_{\text{HSP}}$ and InceptionTime$_{\text{Raw}}$ predictive distributions and use the new distribution to find the predicted class.

Cumulative (B)

B.1. For each test instance, take the per class mean of the BayesNet$_{\text{HSP+TOF}}$ (making use of both the HSP and TOF expert feautres) and InceptionTime$_{\text{Raw}}$ predictive distributions and use the new distribution to find the predicted class.

B.2. A cumulative approach that incorporates a hierarchical framework, in this approach the combined expert and series test distributions are tested multiple times for consensus as fly, male or female. Otherwise the distributions are combined and the new distribution is used for classification.

For each test instance, retrieve the predicted class probabilities of BayesNet$_{\text{HSP+TOF}}$ (making use of both the HSP and TOF expert feautres) and InceptionTime$_{\text{Raw}}$ and compute the cumulative probabilities of the instance belonging to both the fly and mosquito groups, for both distributions. If in both cases the probability of the fly group is higher than the mosquito group return a one hot encoded distribution reflecting the fly class with the highest probability across both distributions. Otherwise, compute the cumulative probabilities of the instance belonging to both the male and female mosquito groups for both distributions. If in both cases the probability of the male group is higher than the female group, return a one hot encoded distribution reflecting the male class with the highest probability across both distributions. Repeat again for the female group. Finally, if there is no consensus, take the per class mean of the expert and series predictive distributions and use the new distribution to find the predicted class.

6.7.2 Results

Table 6.8 presents the overall accuracy for each of the 7 combination methods alongside InceptionTime$_{\text{Raw}}$. The table also presents the true positive and false positive rates for the same 6 sub groups defined in Section 6.3. Table 6.9 shows that 4 of the combination methods produce a higher accuracy then InceptionTime over 30 stratified resamples of the InsectSound dataset, 2 from the hierarchical A category and 2 from the cumulative B category. The most accurate method is shown to be B.1,. with an increase of 4.15% over InceptionTime$_{\text{Raw}}$. The method is simple and sees new test distributions formed from the per-class mean of the BayesNet$_{\text{HSP+TOF}}$ and IncpetionTime$_{\text{Raw}}$ approaches test distributions. A.4. is shown to be the most accurate hierarchical method, with a 4.11% increase in accuracy over InceptionTime$_{\text{Raw}}$.

Table 6.8 Table showing the performance of combination approaches and InceptionTime on the InsectSounds dataset.

	Accuracy	Mosquito		Female		Male		Fly		Aedes		Culex	
		TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
A.1.	0.8025	0.9515	**0.0054**	0.9415	0.0081	0.9447	**0.0048**	**0.9946**	0.0485	0.8498	0.0292	0.9104	0.0566
A.2.	0.8237	0.9750	0.0106	0.9606	0.0088	0.9724	0.0062	0.9894	0.0250	0.8589	0.0301	0.9374	0.0595
A.3.	0.8543	0.9865	0.0379	0.9704	0.0121	0.9853	0.0120	0.9621	0.0135	0.8806	0.0309	0.9443	0.0734
A.4.	0.8966	0.9927	0.0227	0.9849	0.0066	0.9890	0.0087	0.9773	0.0073	0.9430	0.0204	**0.9664**	0.0349
A.5.	0.8615	0.9825	0.0147	0.9762	**0.0056**	0.9772	0.0070	0.9853	0.0175	0.9227	**0.0201**	0.9602	**0.0305**
B.1.	**0.8969**	0.9932	0.0228	0.9853	0.0067	0.9895	0.0087	0.9772	0.0068	**0.9434**	0.0205	0.9667	0.0350
B.2.	0.8777	**0.9972**	0.0947	**0.9902**	0.0175	**0.9923**	0.0220	0.9053	**0.0028**	0.9423	0.0272	0.9663	0.0667
IT	0.8555	0.9865	0.0344	0.9714	0.0112	0.9840	0.0119	0.9656	0.0136	0.8838	0.0323	0.9424	0.0701

However, an analysis of the TPR and FPR of the presented subgroups provides further insight as to when each method excels, or does not. The discussion in section 2.2 highlights the difference in impact that male and female mosquitoes have in society. As a result, the consideration of performance on the subgroups presented is arguably more important in a real world setting, than that of overall accuracy. Across all subgroups, neither B.1 or InceptionTime produces the highest TPR or lowest FPR in any subgroup. Despite B.1 being the most accurate method, B.2. is undoubtedly the better method with regards to minimising the misclassification of mosquitoes and subsequently female mosquitoes. The FPR in the fly group is low, indicating that the occurrence of mosquitoes, male or female, being classified as flies is rare and as then expected, the TPR of the mosquito subgroup is high. Furthermore, the TPR of the female subgroup is also high and shows that 99.02% of female mosquitoes are labelled correctly. On average, over 30 stratified resamples, this equates to missing 98 out of 10,000 female mosquito cases in real terms, an improvement on the 286 missed by IncpetionTime.

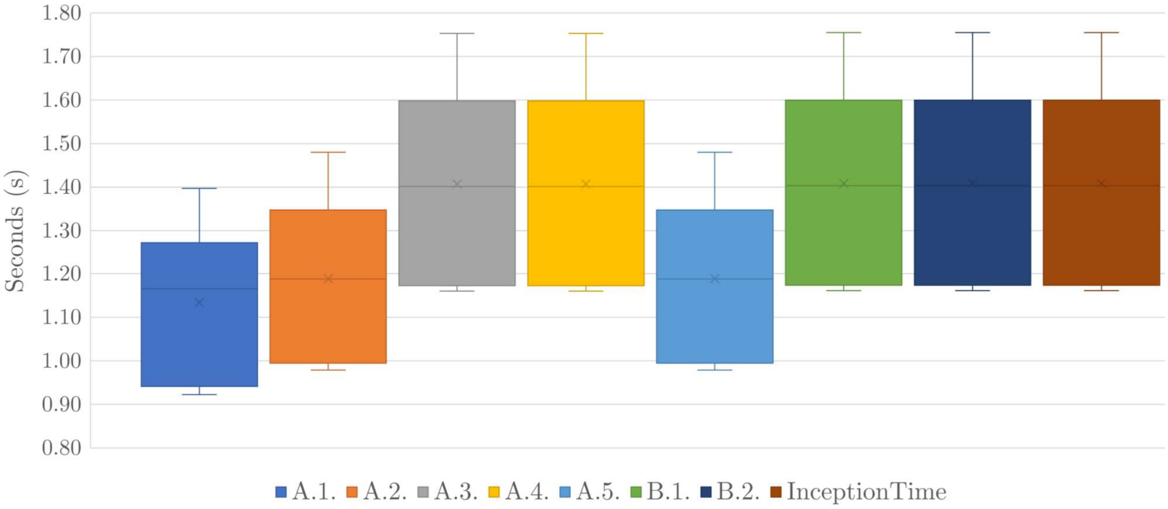
d.f.	28
Critical value (one-tail)	1.701131
Critical value (two-tail)	2.048407
α	0.05

Table 6.9 Table of pairwise t-values for 7 combination methods and the InceptionTime approach, computed from a paired t-test on 30 folds on the InsectSounds dataset.

	A.1	A.2	A.3	A.4	A.5	B.1	B.2
IT	148.6496	88.5943	10.6914	-49.0257	-7.679	-49.0399	-30.8764
A.1		-54.6986	-150.234	-104.057	-64.5047	-103.424	-91.2666
A.2			-86.5764	-91.6846	-52.289	-91.4992	-73.2062
A.3				-48.9685	-7.9410	-48.9356	-31.0481
A.4					124.9654	-11.0558	51.5151
A.5						-127.336	-32.1587
B.1							51.4304

The boxplots in Figure 6.1 show the distribution in the mean per-instance test time over 30 stratified resamples. B.1, B.2 are cumulative so are in fact processing more information than InceptionTime per instance. The plot suggests that at the default $belief$ value, there are two pairs of methods that behave similarly with respect to test time. The plots show the distribution of test time over a large number of cases. For all methods, both the median and mean test times are greater than the duration each case represents, 1 second. The plot suggests the benefit of using a hierarchical approach is minimal with the greatest difference in mean times between the hierarchical and cumulative methods being 0.29 seconds.

Fig. 6.1 Combination approaches test times.



However, the class distribution of the InsectSounds dataset is equal and the number of fly cases in the test set, 5,000, is significantly lower than the number of mosquito cases, 20,000. Table 6.10 shows the number of filtered cases, per-instance test time and per-instance test time of filtered cases. The table illustrates more clearly the difference in processing time for those cases that are filtered. The time required is between 50 and 30 microseconds, compared to the overall mean of 1.12 seconds and over a large period, could equate to a significant reduction in power consumption. The table also reveals that at the default $belief$ value, 2 pairs of methods produce the same number of filtered case and per instance test times. Interestingly, there is no repetition in the overall accuracies of these 4 methods and all 5 methods differ in the implementation of the filtering rules.

Table 6.10 Table showing the mean: number of filtered cases, total test time and test time of filtered cases for each of the hierarchical combination methods over 30 stratified ramdon resamples.

	Filtered cases	Per-instance test time (s)	Filtered cases per-instance test time (s)
A.1.	5087.2414	1.1220	2.7692E^{-6}
A.2.	3899.5862	1.1888	2.9045E^{-6}
A.3.	25.2069	1.4072	4.7917E^{-5}
A.4.	25.2069	1.4072	4.7645E^{-5}
A.5.	3899.5862	1.1888	2.8578E^{-6}

Figure 6.2 provides more detail on the effect that different $belief$ values have on each of the methods from Table 6.8. Sub-figures (b) and (c) present the test time per instance and number of cases filtered for each method. Unsurprisingly, the test time per-instance and number of filtered cases are shown to have a strong negative correlation. What is surprising, is the way in which the performance of the methods coalesce into 2 pairs with respect to cases filtered and test time, whilst remaining separate in terms of accuracy, female TPR and Fly FPR. This can be accounted for by the requirement for each cases predicted class being from the ‘fly’ group, as well as the cumulative probability of fly classes being greater that the $belief$ parameter in the conditional decision statement. This additional clause in the if statement prevents pathological behaviour, as the $belief$ parameter approaches 0. The separation of the paths into 2 streams is based on the expert approach used for filtering. Methods A.3. and A.4. use approaches in conjunction with the HSP representation. Whereas, methods A.2. and A.5. use approaches in conjunction with the TOF+HSP representation. Finally,

the emergence of 2 maximum accuracies is dependant on the whether the method is incorporating the prediction distribution of the expert approach in the final prediction, which is the case for the A.4. and A.5. methods, clearly showing the benefit of expert feature inclusion.

This additional information tempers the assumption that A.4. is the best hierarchical approach, because it is the most accurate. Figure 6.2 (a) shows that by a $belief$ value of 0.8, method A.4. has reached a plateau. Sub-figure (b) shows this is because the number of filtered cases is extremely low, indicating the more powerful InceptionTime$_{\text{Raw}}$ approach is being leveraged and as a result, the mean test time per-instance is high. Looking just at Sub-figures (b) and (c) shows this superior efficiency performance of A.5. only emerges for reasonable belief values, >0.5. and a reasonable implementation might switch method, based on the $belief$ value in order to maximise efficiency and accuracy. However, Sub-figures (d) and (e) reveal that the A.4. performs significantly worse than A.5. with respect to determining both fly and female mosquito classes, until the more powerful InceptionTime$_{\text{Raw}}$ approach begins to dominate classifications.

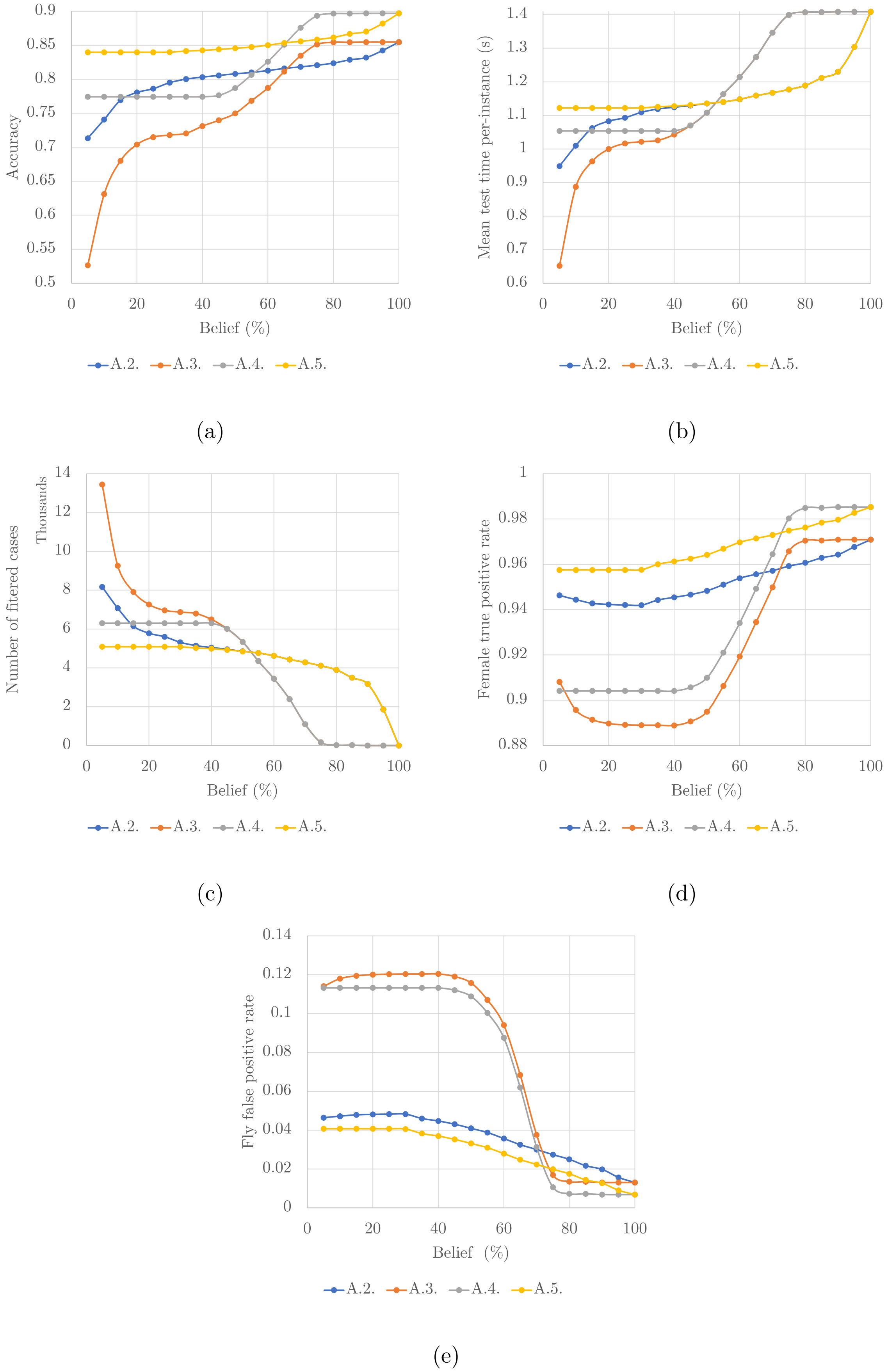
6.8 Conclusion

In conclusion, this chapter has presented experimental results that support the idea that expert features can be leveraged to improve classification accuracy in insect wingbeat classification, thus challenging the established paradigm that, in the context of insect classification, approaches perform best with spectral features. Instead, in the context of insect classification, deep learning approaches trained on raw audio produce superior accuracies.

The results in this chapter suggest that the use of fundamental frequency in conjunction with relatively simple approaches could be an adequate method of filtering, preventing unnecessary processing. An examination of both the overall accuracy, as well as the TPR and FPR have demonstrated that on the InsectSound dataset, These features are particularly effective at distinguishing between mosquitoes and flies. The reported experiments also reveal the limitations of the time-of-flight and fundamental frequency features. For example, the fundamental frequency is less informative when distinguishing between many classes of genus or species. This is because these groupings exhibit significant variance in wingbeat frequency, which is actually a function of size and therefore best suited for determining sex. The time-of-flight feature is less helpful, but was shown to make a difference. This is particularly evident in edge cases where: insect interception is either late or early in the day, or the wingbeat frequency is in an area overlapping distributions. The feature is an effective indication of prior-probability and is an intuitively good starting point for classification. However, in 3/4 of the datasets, the expert features produced models that were fundamentally flawed, and in multiple cases, classes were not predicted at all. In some cases, such as the Aphids dataset this is likely impacted by poor class representation, but fundamentally it appears, that the behaviour is caused by overlapping frequency and TOF distributions.

Experiments on both raw pseudo-acoustic data and the spectral counterpart provided unexpected results. Firstly, for all 3 convolutional based approaches, the performance was best on the raw dataset. This behaviour is contrary to that of so called 'traditional' approaches, where typically the spectral decomposition of audio data is most likely to yield useful features. Secondly, the performance of deep learning approaches, in combination with the raw audio, exceeded all non-convolutional approaches in combination with spectral data. Contrary to expectation the performance of the cRISE$_{\text{All}}$ approach was mediocre. However, spectral representations were unexpectedly shown to be less useful generally. There is some support for these in the literature discussed in Chapter 2.3 where the relevance of deep learning approaches in insect classification has been recently highlighted. It is thus arged that methods that make use of convolutional features present a promising avenue of research for flying insect classification.

Finally, an examination of 7 methods of combining the test distributions of approaches built using expert features and the InceptionTime approach trained on the InsectSound raw data, shows that the fundamental frequency feature is more effective than the combined fundamental frequency and time-of-flight features, despite the latter producing a superior classification accuracy. Whereas, the inclusion of the combined expert features increases the classification over the best single approach $\text{InceptionTime}_{\text{raw}}$. The results in Sub-Sfection 6.7.2 show that simply combining the output distributions from the TOF+HSP and InceptionTime, B.1., produces the highest accuracy. However, nuances in the methods presented, show that a hierarchical cumulative method, B.2., produces an approach which preforms best in key measures, such as maximising female mosquito TPR and minimising fly FPR. Additionally, a review of the hierarchical methods presented, shows that the most accurate A.4. method is ineffective as a filter. Whereas, the A.5. method exhibits far greater consistency in the examined performance measures, for varying levels of the $belief$ parameter, including the ability to filter cases. Where, on average over the 30 stratified resamples, the processing time of A.5. was 1.5 hours less than B.2. for a decrease in overall accuracy of 1.62%.



(a)

(b)

(c)

(d)

(e)

Fig. 6.2 Figure showing the changes in: accuracy (a), mean test time (b), number of filtered cases (c), the TPR of female mosquitoes and the FPR of flies for different values of the belief parameter in 4 hierarchical combination methods.

Chapter 7

Conclusions And Future Work

This thesis is primarily concerned with improvements in the automatic classification of flying insects via insect-centric psuedo-acoustic datasets. The prevailing method of screening in emerging smart trap technology has led to the creation of large multiclass problems. This technology is still in its infancy. However, some practitioners are already generating rich and diverse datasets, which by virtue of implementation are accompanied by metadata, such as the time of flight (TOF) feature. These datasets provide the opportunity for insect classification approaches to keep instep with the advancements in smart trap hardware. This thesis for the first time, presents experimental results of 'state-of-the-art' approaches on a collection of these datasets. Described in Chapter 2, this Thesis presents 4 datasets, generated by 2 distinct pieces of hardware. These datasets have allowed us to investigation the effectiveness of classification approaches from multiple domains, including powerful ensemble approaches such as HIVE-COTEv2 and InceptionTime. Whilst a unified methodology has enabled us to directly compare the performance of these approaches despite implementations in different languages.

Motivated by the experimental results in Chapter 5 we undertook a review of the HIVE-COTEv1.0 approach's audio domain expert, RISE. Changes in the RISE implementation led to: refinements in the internal transformations undertaken; a novel mechanism of controlling build time in tree-based approaches; changes to the interval selection policy and an investigation of multiple spectral transformations, as well as new methods of internal combination techniques. Presented in Chapter 4, these changes amounted to a significant improvement to both performance and usability of the RISE approach, named RISE-All.

Alongside the recently published HIVE-COTEv2.0 and InceptionTime approaches, results of multiple powerful approaches, including $\mathrm{cRISE}_{\mathrm{All}}$ have been presented. Through a series of experiments on multiple dataset representations, a framework that both maximises predictive performance, whilst also reducing the amount of unnecessary processing is outlined. Via an examination of multiple methods of test distribution combination, the number of female mosquito candidate cases passed to intensive processing approaches was reduced by leveraging the predictive power of expert features. The experiments in this Thesis have shown that the deep learning approaches RESNET and InceptionTime perform well on the pseudo-acoustic insect-centric datasets. Furthermore, the convolutional approach Arsenal also performed well. Interestingly, all 3 of these approaches were detrimentally affected by the transformation of data into the spectral domain. Our belief is that these approaches are confounded by the large intervals of barren attributes that represent high frequency information.

7.1 Discussion Of Contributions

The work in this thesis brings together, for the first time, multiple large insect-centric pseudo-acoustic datasets for experimentation and evaluation. Experimental results of powerful approaches such as HIVE-COTEv1.0, which at the time were considered state-of-the-art, are first presented in Chapter 5. The contributions of this chapter include a review of common assumptions in the literature, such as: the effectiveness of wingbeat frequency as a descriptive feature, the effectiveness of the spectral series as a representation from which to learn features and the apparent difficulty of producing useful features from the raw data representation. Across multiple large datasets, the claims that the approaches built on spectral representations perform best are challenged, adding weight to the intuition that data generation via the methods described in Chapter 2 is comparable to audio. Furthermore, the performance of an additional feature, Time of flight (TOF) is presented. Using this relatively unexplored feature with wingbeat frequency is shown to produce an accuracy greater than either feature individually. In an analysis of these results, it was shown, by the type I and type II errors of key class sub-groups, that these features can be effective in discriminating between classes of male mosquitoes, female mosquitoes and flies, adding weight to the idea of a hierarchical classification framework that would attempt to minimise the use of resources.

The work in Chapter 5 provided motivation for the attempts to improve RISE, the spectral component of the HIVE-COTEv1.0 approach. These contributions are detailed in Chapter 4. The contributions made, fall into 2 broad categories, improvements in usability and improvements in accuracy. Preliminary experiments on large datasets had previously highlighted how prohibitive the runtime complexity of RISE was for large problems. An ablative study on the transforms used revealed that 2 of these were in fact redundant. Although this did lead to a reduction in runtime, there was still a requirement to further control the train time specifically. Two training time control mechanisms were introduced and tested. The naive approach represented an obvious approach to controlling the train time of a forest archetype - checking and, if necessary, ending the training phase between each tree, whilst the adaptive approach used a regression model to change the maximum interval space, ensuring that the contract time was strictly adhered to. The ability of each mechanism to scale, was explored via a study of large datasets from the UCR archive, where the consistency of the adaptive mechanism was clearly shown. Having improved the usability of the RISE algorithm with an effective and reliable method of controlling training time, the focus shifted to attempting to improve the accuracy of the approach. As with most tree based approaches, the cRISE architecture comprised of 3 main parts: interval selection, transformation and base learner. The second algorithmic contribution was the result of a review of the interval selection policy. A total of 4 policies were tested, including the original policy. An analysis of selected attributes showed that the process of selecting length before start point, or start point before length, was fundamentally flawed and resulted in a skewed distribution. A new approach in which intervals were chosen based on a number of split points, had a profound effect on the distribution of selected attributes. Through experiments on the UCR archive, this approach was also found to be significantly more accurate, and as a result became the default procedure. Finally, a lengthy review of spectral transforms and transform combination approaches was undertaken in order to maximise the predictive accuracy of cRISE. In preparation for this review, additional transforms were implemented, including: 6 descriptive audio features quantifying characteristics such as brightness or the spectral 'centre of mass'; cepstral coefficients and spectrograms. Firstly the performance of each transform separately was assessed, contrasting performance with the default cRISE implementation. These experiments led to an investigation into intelligently selecting transforms. The work was centred around the idea of selecting a transform based on either a proxy, or performance on resamples of the train set and 3 approaches were implemented. The findings indicated that none of these approaches were able to outperform the default cRISE configuration and all attempts to intelligently select a single transform, were found to be significantly worse. This led to considerations of alternative ways of combining the expanded pool of transforms, including ablation studies, in order to identify redundancy in the included transforms. The experiments culminated in 2 transformation combination techniques, that both outperformed the default cRISE configuration. Firstly, there was combination via CAWPE method, introduced by Large et. al. [52], in which each transform is trained individually before the train distributions are combined. Secondly, there was the cumulative approach, in which individual transforms are concatenated. The difference in test accuracy between the best configurations of the two approaches was found to be insignificant, although the cumulative approach was marginally more accurate and exhibited a lower variance in accuracy over 30 stratified resamples of the UCR archive. Furthermore, it was shown that the runtime of this approach was lower on larger problems. As a result, it was favoured moving forward and this new cRISE configuration was denoted as $\text{cRISE}_{\text{All}}$.

Armed with a significantly improved spectral approach, the work in Chapter 6 focuses on two contributions: assessing the current state-of-the-art time series approaches on the application of insect classification, including the newly presented state-of-the-art InceptionTime and HIVE-COTEv2.0 ensembles, and using these findings to present and analyse methods of hierarchically combining both expert and series approaches to improve accuracy and reduce runtime on the datasets presented in Chapter 2. Starting with a review of expert feature performance, questions were raised regarding the reliability of these features. In 3/4 of the insect-wingbeat datasets, the predictions of approaches trained on both the wingbeat frequency and the TOF feature were shown to be pathological. Furthermore, questions regarding the fragile nature of the TOF feature were raised, when considering its performance over multiple datasets. However, the effectiveness of these features was shown to be considerable on the InsectSound dataset. In an analysis of the confusion matrices from approaches trained on expert features, it was clear that these features worked best when distinguishing between classes that were sex or species separated. In experiments assessing performance in conjunction with raw and spectral dataset representations, the results contradicted expectation.

Approaches that produced features from convolutions, such as: Arsenal, RESNET and InceptionTime, were better able to extract meaningful information from the raw pseudo-acoustic data representation, whereas spectral approaches such as cRISE$_{\text{All}}$ did not perform as well as expected. InceptionTime was found to perform best, on 3 out of 4 of the insect-centric datasets. In the final section, results demonstrate that these features are also effective in improving classification accuracy, whilst reducing run time. In experiments focused on the InsectSound dataset, 2 hierarchical and 2 cumulative methods of distribution combination were found to be more accurate than the single best approach, InceptionTime$_{\text{Raw}}$. Of these, the most efficient approach was shown to be 0.6% more accurate and capable of producing 48/10,000 more true positive female mosquito predictions, 83/15,000 fewer false positive female mosquito predictions and used 1.5 hours less processing time than InceptionTime$_{\text{Raw}}$. In comparison, the most accurate of these combination methods was shown to be 4.14% more accurate, capable of producing 139/10,000 more true positive female mosquito predictions and, 45/15,000 fewer false positive female mosquito predictions then InceptionTime$_{\text{Raw}}$.

When referring back to the questions posed in Sub-section 1.1 we can confidently conclude that it is possible to classify multiple species of mosquitoes into their respective sex using wingbeat features. The extent to which this is possible depends on a number of factors and in the general case where the number of species is very large the performance will be degraded. Experiments devised to asses whether it is possible to determine a mosquitoes genera produced a less convincing result. In general the intra-class variance caused by grouping mosquitoes of both sexes into one class is much larger than the inter-class variance you expect to see between species. For this reason, the ability to classify mosquitoes into the respective genus is poor unless there is significant morphological differences between the species. However, thorough the experimental results presented in Chapters 5 & 6 it is clear that the IncpetionTime approach represents a promising avenue for research in this application moving forward. This is further evidenced in Section 6.7 where a hierarchical method is shown to perform very well with respect to both accuracy and processing time, when combined with a simple approach trained using time-of-flight and wingbeat frequency, showing that it is possible to devise a hierarchical approach to classifying mosquitoes that reduces processing time whilst maintaining or improving accuracy.

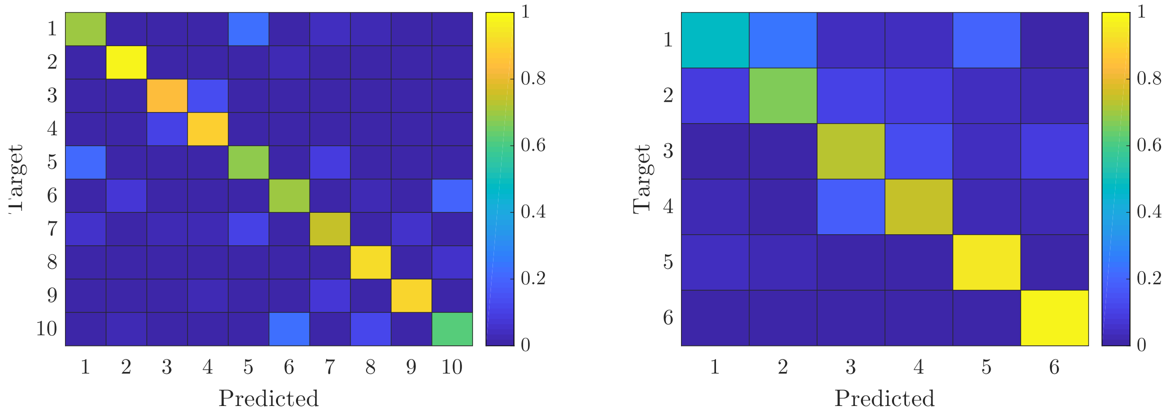
7.2 Future Work

In both Chapters 5 and 6 it was shown that expert features, particularly in combination have a potential to positively influence absolute accuracy. However, these features were also shown to be fragile and on only one of the four application focused datasets included, were they able to demonstrate practical use. It is clear to us that their effectiveness is coupled to the configuration of data. The overlap in distribution of wingbeat frequency across multiple species of mosquito is well documented, and is demonstrated again in Chapter 2, Sub-Section 2.5.4. However, until this point, little consideration has been given to the distribution of sex separated groups, the broader problem of discerning mosquitoes from other flying insects, or how the morphological differences might be leveraged in an application focused solution. That being said, it is clear that more work needs to be done, in order to fully explore the limitations of these features and further validate their role in a hierarchical classification framework.

The decision to frame the final approach entirely as a classification problem meant that the experimental methodology was consistent and the results throughout are directly comparable. However, it meant that the use of clustering approaches in the proposed hierarchical methods were not considered. The work in this thesis presents the idea that the correct classification of female mosquitoes is an important consideration, due to their role in the spread of disease. Collaboration with stakeholders directly involved with disease mitigation programmes would provide a valuable contribution and help inform the future direction of research with respect to the focus of performance and dataset curation. The work considered multiple dataset representations and looked closely at performance between groups of classes. Providing the maximisation of female mosquito accuracy is considered desirable the next step might be to reduce the number of class labels, to better reflect the problem, possibly incorporating 3 levels of hierarchy. Reconfiguration of data in this manner is likely to have a positive effect on accuracy, providing the opportunity for algorithms to learn features common to targeted characteristic agnostic of factors, such as specie. The expectation is that this will be preferential to the method of combining test distributions prior to classification, but could also be used alongside the methods presented here, to provide two classifications. In addition, future research would benefit from collaboration with hardware manufacturers, such as those mentioned in Chapter 2, Sub-Section 2.4.2, which would provide the opportunity to assess the impact of different algorithm and dataset configurations on power consumption in a real world setting, an important factor in proving the viability of the application in real terms.

In summary, this thesis has demonstrated the feasibility of a hierarchical approach to insect classification, via the use of expert features such as: fundamental wingbeat frequency and time-of-flight information. Experimental results have shown that useful and meaningful features can be extracted from the profiles of raw wingbeat data, produced via optical sensors. Deep learning approaches have been shown to be particularly effective on the datasets tested, outperforming competing non-deep-learning approaches in combination with spectral features. In the future, I expect the coalescence of these two streams of research to act as an accelerant on the road towards producing a viable ‘real-world’ solution.

Appendix A



(a) InsectSounds+T1, HIVE-COTEv1.0

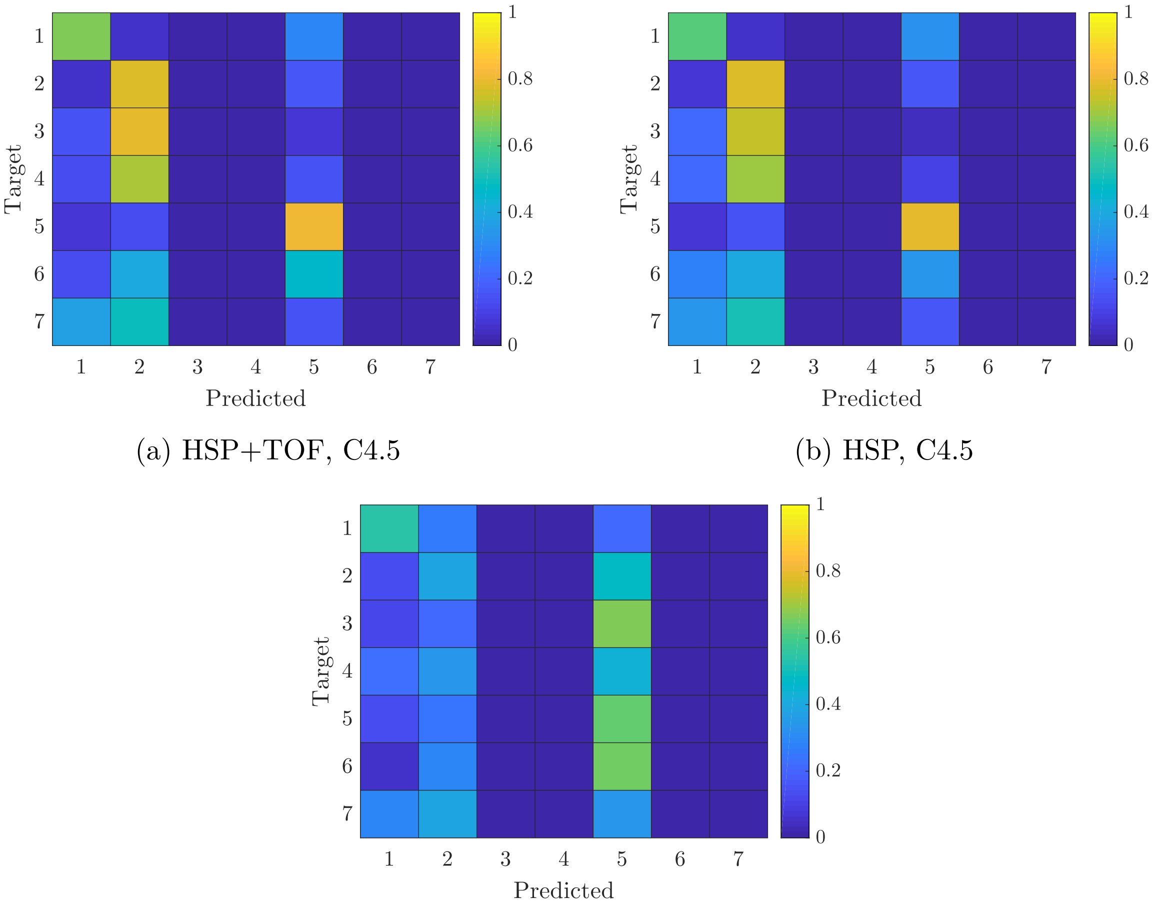
(b) MosquitoSounds, cRISE

Fig. A.1 Figures showing the confusion matrices for the HIVE-COTEv1.0 approach on the T1 transformed InsectSounds and cRISE on the MosquitoSounds datasets.

Appendix B

Table B.1 Dataset information for the insect wingbeat datasets in the frequency domain.

Dataset	Classes	Instances	Attributes	Majority class
AphidsSpec	*A. fabae*	2,036	4,000	0.4203
	D. platanoidis	3,192		
	M. persicae	19		
	P. testudinaceus	115		
	Pollen beetle	4,034		
	Ps. chrysocephala	194		
	R. padi	8		
FruitFliesSpec	melanogaster	6,064	4,000	0.5305
	suzukii	10,142		
	zaprionus	18,312		
InsectSoundSpec	*Ae. aegypti* ♀	5,000	3,000	0.1000
	Ae. aegypti ♂	5,000		
	Dr. simulans	5,000		
	Mu. domestica	5,000		
	Cx. quinquefasciatus ♀	5,000		
	Cx. quinquefasciatus ♂	5,000		
	Cx. stigmatosoma ♀	5,000		
	Cx. stigmatosoma ♂	5,000		
	Cx. tarsalis ♀	5,000		
	Cx. tarsalis ♂	5,000		
MosquitoSoundSpec	*Ae. aegypti*	5,000	3,000	0.166
	Ae. albopictus	5,000		
	An. arabiensis	5,000		
	An. gambiae	5,000		
	Cx. pipiens	5,000		
	Cx. quinquefasciatus	5,000		

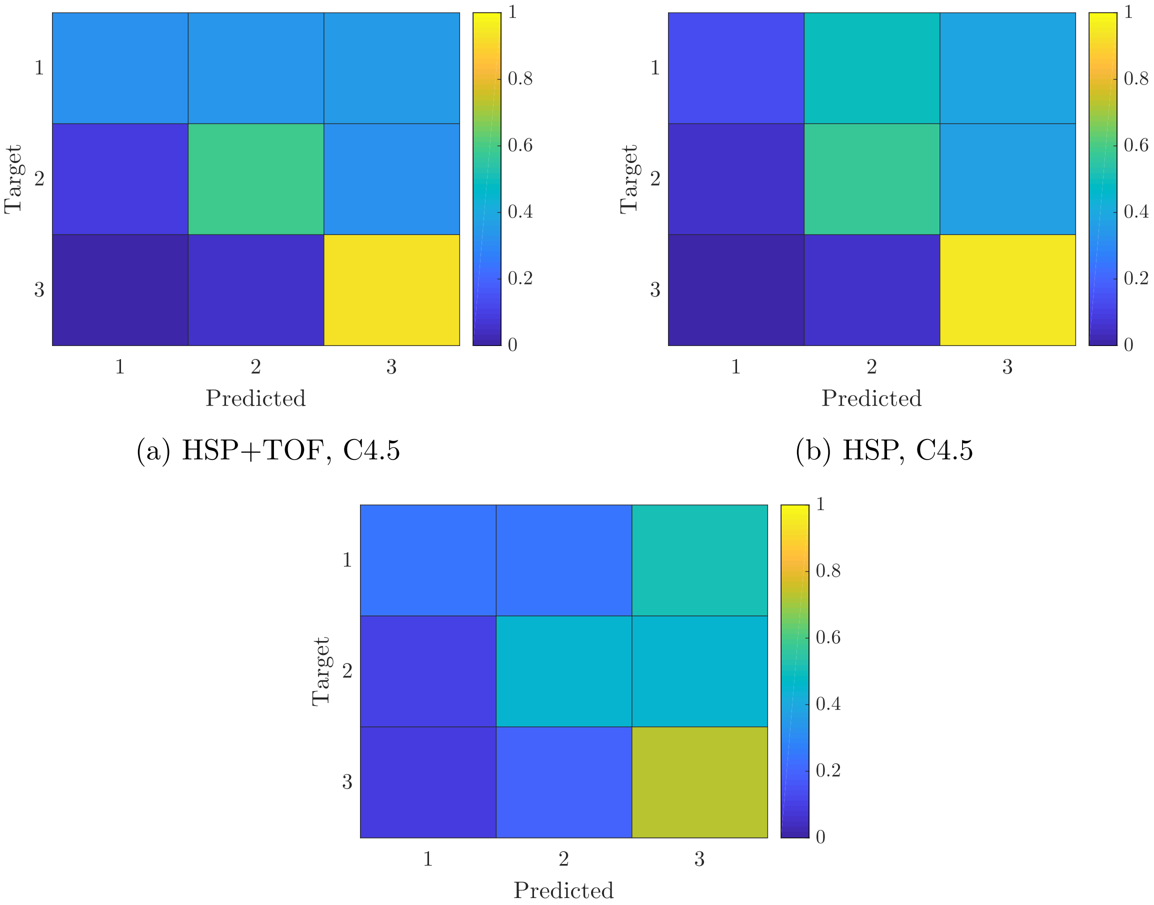


(a) HSP+TOF, C4.5

(b) HSP, C4.5

(c) TOF, C4.5

Fig. B.1 Figures showing the confusion matrices for: HSP+TOF, HSP and TOF transformed Aphids datasets.

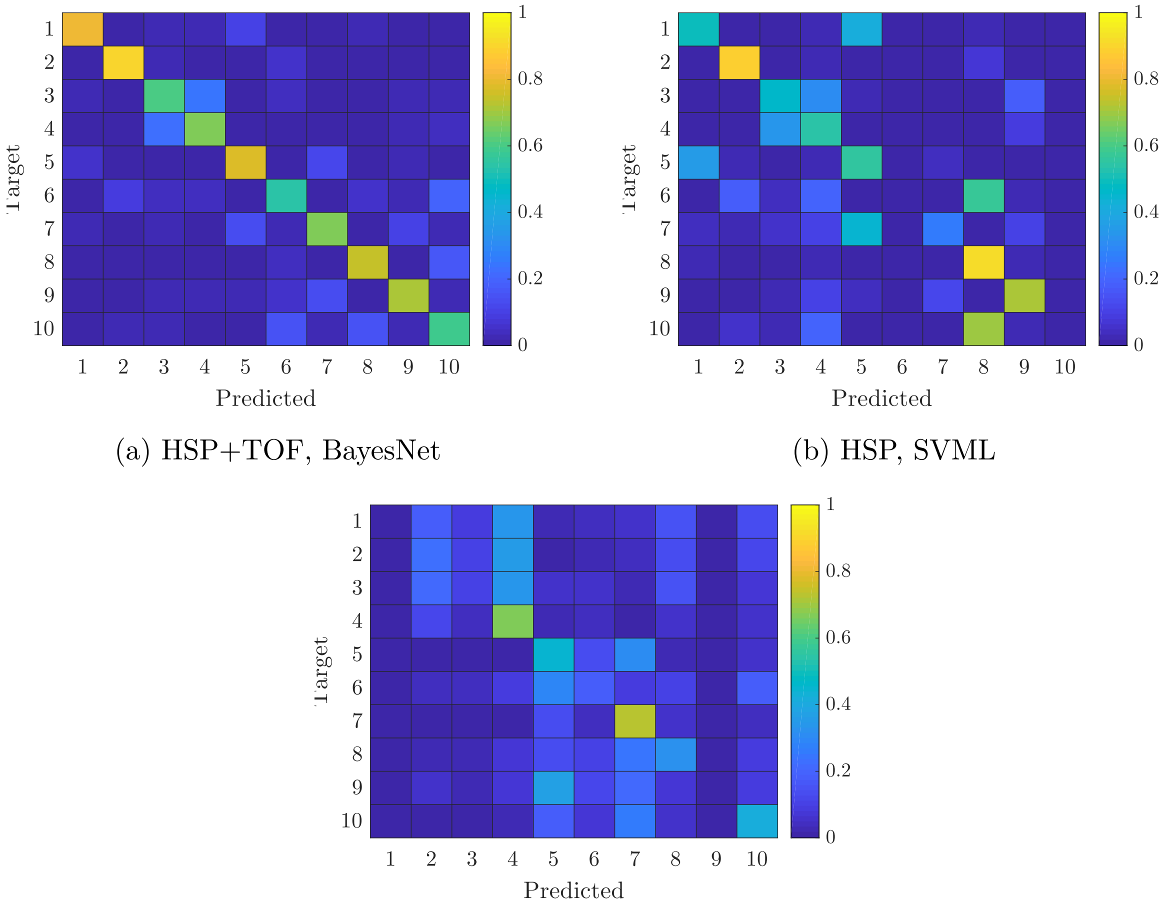


(a) HSP+TOF, C4.5

(b) HSP, C4.5

(c) TOF, ED

Fig. B.2 Figures showing the confusion matrices for: HSP+TOF, HSP and TOF transformed FruitFlies datasets.

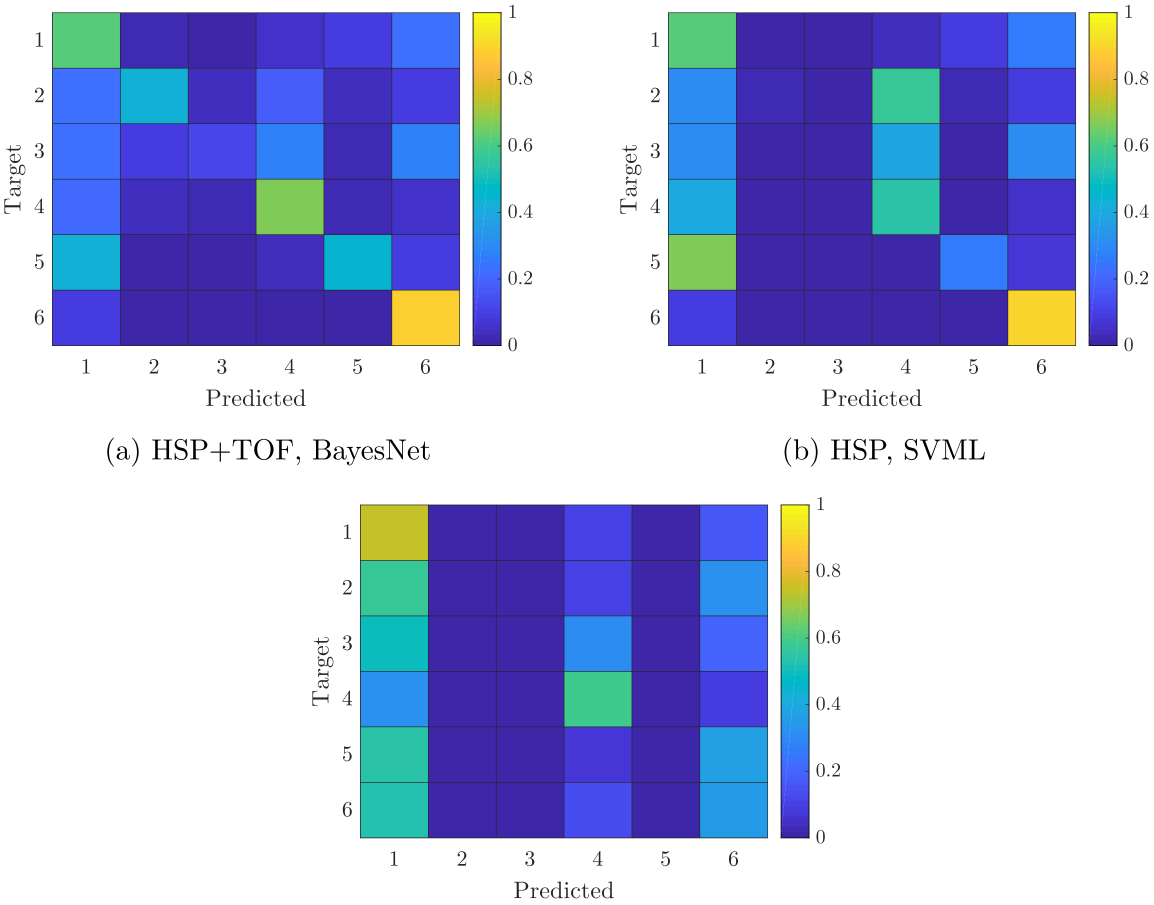


(a) HSP+TOF, BayesNet

(b) HSP, SVML

(c) TOF, BayesNet

Fig. B.3 Figures showing the confusion matrices for: HSP+TOF, HSP and TOF transformed InsectSound datasets.



(a) HSP+TOF, BayesNet

(b) HSP, SVML

(c) TOF, BayesNet

Fig. B.4 Figures showing the confusion matrices for: HSP+TOF, HSP and TOF transformed MosquitoSound datasets.

Table B.2 Table of pairwise t-values computed from a paired t-test on 30 folds on the Aphids dataset.

	HCv2.0	Arsenal	DrCIF	STC	TDE	RISE$_{\mathrm{All}}$	IT	RESNET
HCv2.0		6.1640	10.229	28.8332	33.1546	36.2304	-38.7348	7.6393
Arsenal			6.6372	27.4703	31.3505	32.2629	-36.6012	5.787
DrCIF				23.7358	28.0127	30.7277	-36.2440	**2.0312**
STC					**1.2087**	-17.6255	-40.8037	-20.8579
TDE						-19.0813	-39.6116	-25.8521
RISE$_{\mathrm{All}}$							-62.0329	-13.1532
IT								29.3565
RESNET								

df	29
Critical value (one-tail)	1.699127
Critical value (two-tail)	2.04523
α	0.05

Table B.3 Table of pairwise t-values computed from a paired t-test on 30 folds on the FruitFlies dataset.

	HCv2.0	Arsenal	DrCIF	STC	TDE	RISE$_{\text{All}}$	IT	RESNET
HCv2.0		8.3646	98.2049	80.1971	37.4912	201.7816	2.8515	-39.8425
Arsenal			109.9601	80.9508	36.4262	222.1941	2.8092	-49.4568
DrCIF				64.4605	27.5704	124.7528	**1.7486**	-158.3778
STC					**-1.0739**	-33.051	**-1.7491**	-88.5826
TDE						-13.7961	**-1.600**	-40.9348
RISE$_{\text{All}}$							**0.0783**	-251.64
IT								-3.2851
RESNET								

df	27
Critical value (one-tail)	1.703288446
Critical value (two-tail)	2.051830516
α	0.05

Table B.4 Table of pairwise t-values computed from a paired t-test on 30 folds on the InsectSound dataset.

	HCv2.0	Arsenal	DrCIF	STC	TDE	RISE_{All}	IT	RESNET
HCv2.0		43.7745	68.1316	96.1483	26.5258	109.2032	-83.444	-41.6243
Arsenal			48.7923	80.2915	25.3306	90.5023	-100.878	-59.2557
DrCIF				39.7805	22.8441	46.4195	-118.561	-78.663
STC					20.7009	**1.0624**	-129.973	-93.6003
TDE						-20.4886	-34.1422	-29.7568
RISE_{All}							-138.01	-101.254
IT								46.3631
RESNET								

df	29
Critical value (one-tail)	1.699127
Critical value (two-tail)	2.04523
α	0.05

Table B.5 Table of pairwise t-values computed from a paired t-test on 30 folds on the MosquitoSound dataset.

	HCv2.0	Arsenal	DrCIF	STC	TDE	RISE-ALL	IT	RESNET
HCv2.0		39.4179	37.2270	30.3396	22.1024	59.7902	-27.6146	**-0.2947**
Arsenal			26.2849	29.3109	21.6130	51.6935	-30.3167	**-0.8557**
DrCIF				26.9705	20.2567	18.7134	-36.9167	**-2.0648**
STC					7.7864	-24.7041	-35.7968	-13.0761
TDE						-19.6674	-27.0155	-17.8446
RISE-ALL							-39.8241	-2.9016
IT								5.5140
RESNET								

df	22
Critical value (one-tail)	1.717144
Critical value (two-tail)	2.073873
α	0.05

Algorithm 11 A.1.

```
function COMBINETESTDISTRIBUTIONS($\mathbf{t}$, $\mathbf{M}_1$, $\mathbf{M}_2$)
    Let $\mathbf{t} =< x_1 \ldots x_n, c >$ be a test case consisting of $n$ attributes and a class value.
    Let $c \in \mathbf{C}$.
    Let $\mathbf{M}_1$ be the trained BayesNet$_{\text{HSP+TOF}}$ model.
    Let $\mathbf{M}_2$ be the trained InceptionTime$_{\text{Raw}}$ model.
    $p_c \leftarrow \mathbf{M}_1$.getPrediction($\mathbf{t}$)
    if $isFlyClass(p_c)$ then
        return $\mathbf{M}_1$.getTestDistribution($\mathbf{t}$)
    else
        return $\mathbf{M}_2$.getTestDistribution($\mathbf{t}$)
```

Algorithm 12 A.2.

```
function COMBINETESTDISTRIBUTIONS($\mathbf{t}$, $\mathbf{M}_1$, $\mathbf{M}_2$, $belief$)
    Let $\mathbf{t} =< x_1 \ldots x_n, c >$ be a test case consisting of $n$ attributes and a class value.
    Let $c \in \mathbf{C}$.
    Let $\mathbf{M}_1$ be the trained BayesNet$_{\text{HSP+TOF}}$ model.
    Let $\mathbf{M}_2$ be the trained InceptionTime$_{\text{Raw}}$ model.
    $\mathbf{d}_1 \leftarrow \mathbf{M}_1$.getTestDistribution($\mathbf{t}$)
    if $getFlyProb(\mathbf{d}_1) > belief$ then
        $p_c \leftarrow \mathbf{M}_1$.getPrediction($\mathbf{t}$)
        if $isFlyClass(p_c)$ then
            return $\mathbf{M}_1$.getTestDistribution($\mathbf{t}$)
        else
            $\mathbf{d}_t \leftarrow< 0_1 \ldots 0_{\mathbf{C}} >$
            $\mathbf{d}_t[highestFlyProbIndex(\mathbf{d}_1)] = 1$
            return $\mathbf{d}_t$
    else
        return $\mathbf{M}_2$.getTestDistribution($\mathbf{t}$)
```

Algorithm 13 A.3.

```
function COMBINETESTDISTRIBUTIONS($\mathbf{t}$, $\mathbf{M}_1$, $\mathbf{M}_2$, $belief$)
    Let $\mathbf{t} =< x_1 \dots x_n, c >$ be a test case consisting of $n$ attributes and a class value.
    Let $c \in \mathbf{C}$.
    Let $\mathbf{M}_1$ be the trained SVML$_{\text{HSP}}$ model.
    Let $\mathbf{M}_2$ be the trained InceptionTime$_{\text{Raw}}$ model.
    $\mathbf{d}_1 \leftarrow \mathbf{M}_1$.getTestDistribution($\mathbf{t}$)
    if $getFlyProb(\mathbf{d}_1) > belief$ then
        $p_c \leftarrow \mathbf{M}_1$.getPrediction($\mathbf{t}$)
        if $isFlyClass(p_c)$ then
            return $\mathbf{M}_1$.getTestDistribution($\mathbf{t}$)
        else
            $\mathbf{d}_t \leftarrow< 0_1 \dots 0_{\mathbf{C}} >$
            $\mathbf{d}_t[highestFlyProbIndex(\mathbf{d}_1)] = 1$
            return $\mathbf{d}_t$
    else
        return $\mathbf{M}_2$.getTestDistribution($\mathbf{t}$)
```

Algorithm 14 A.4.

```
function COMBINETESTDISTRIBUTIONS($\mathbf{t}$, $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{M}_3$, $belief$)
    Let $\mathbf{t} =< x_1 \dots x_n, c >$ be a test case consisting of $n$ attributes and a class value.
    Let $c \in \mathbf{C}$.
    Let $\mathbf{M}_1$ be the trained SVML$_{\text{HSP}}$ model.
    Let $\mathbf{M}_2$ be the trained BayesNet$_{\text{HSP+TOF}}$ model.
    Let $\mathbf{M}_3$ be the trained InceptionTime$_{\text{Raw}}$ model.
    $\mathbf{d}_1 \leftarrow \mathbf{M}_1$.getTestDistribution($\mathbf{t}$)
    $p_c \leftarrow \mathbf{M}_1$.getPrediction($\mathbf{t}$)
    if $(getFlyProb(\mathbf{d}_1) > belief)\ \ isFlyClass(p_c)$ then
        return $\mathbf{M}_1$.getTestDistribution($\mathbf{t}$)
    else
        $\mathbf{d}_2 \leftarrow \mathbf{M}_2$.getTestDistribution($\mathbf{t}$)
        $\mathbf{d}_3 \leftarrow \mathbf{M}_3$.getTestDistribution($\mathbf{t}$)
        $\mathbf{d}_t \leftarrow meanDist(\mathbf{d}_2, \mathbf{d}_3)$
        return $\mathbf{d}_t$
```

Algorithm 15 A.5.

```
function COMBINETESTDISTRIBUTIONS($\mathbf{t}$, $\mathbf{M}_1$, $\mathbf{M}_2$, $belief$)
    Let $\mathbf{t} =< x_1 \ldots x_n, c >$ be a test case consisting of $n$ attributes and a class value.
    Let $c \in \mathbf{C}$.
    Let $\mathbf{M}_1$ be the trained BayesNet$_{\mathrm{HSP+TOF}}$ model.
    Let $\mathbf{M}_2$ be the trained InceptionTime$_{\mathrm{Raw}}$ model.
    $\mathbf{d}_1 \leftarrow \mathbf{M}_1$.getTestDistribution($\mathbf{t}$)
    $p_c \leftarrow \mathbf{M}_1$.getPrediction($\mathbf{t}$)
    if $(getFlyProb(\mathbf{d}_1) > belief)\ \ isFlyClass(p_c)$ then
        return $\mathbf{M}_1$.getTestDistribution($\mathbf{t}$)
    else
        $\mathbf{d}_1 \leftarrow \mathbf{M}_1$.getTestDistribution($\mathbf{t}$)
        $\mathbf{d}_2 \leftarrow \mathbf{M}_2$.getTestDistribution($\mathbf{t}$)
        $\mathbf{d}_t \leftarrow meanDist(\mathbf{d}_1, \mathbf{d}_2)$
        return $\mathbf{d}_t$
```

Algorithm 16 B.1.

```
function COMBINETESTDISTRIBUTIONS($\mathbf{t}$, $\mathbf{M}_1$, $\mathbf{M}_2$, $belief$)
    Let $\mathbf{t} =< x_1 \ldots x_n, c >$ be a test case consisting of $n$ attributes and a class value.
    Let $c \in \mathbf{C}$.
    Let $\mathbf{M}_1$ be the trained BayesNet$_{\mathrm{HSP+TOF}}$ model.
    Let $\mathbf{M}_2$ be the trained InceptionTime$_{\mathrm{Raw}}$ model.
    $\mathbf{d}_1 \leftarrow \mathbf{M}_1$.getTestDistribution($\mathbf{t}$)
    $\mathbf{d}_2 \leftarrow \mathbf{M}_2$.getTestDistribution($\mathbf{t}$)
    $\mathbf{d}_t \leftarrow meanDist(\mathbf{d}_1, \mathbf{d}_2)$
    return $\mathbf{d}_t$
```

Algorithm 17 B.2.

```
function COMBINETESTDISTRIBUTIONS($\mathbf{t}$, $\mathbf{M}_1$, $\mathbf{M}_2$)
    Let $\mathbf{t} =< x_1 \ldots x_n, c >$ be a test case consisting of $n$ attributes and a class value.
    Let $c \in \mathbf{C}$.
    Let $\mathbf{M}_1$ be the trained BayesNet$_{\text{HSP+TOF}}$ model.
    Let $\mathbf{M}_2$ be the trained InceptionTime$_{\text{Raw}}$ model.
    $\mathbf{d}_1 \leftarrow \mathbf{M}_1$.getTestDistribution($\mathbf{t}$)
    $\mathbf{d}_2 \leftarrow \mathbf{M}_2$.getTestDistribution($\mathbf{t}$)
    if $(getFlyProb(d_1) > getMosquitoProb(d_1))$ & $(getFlyProb(d_2) > getMosquitoProb(d_2))$ then
        $\mathbf{d}_t \leftarrow< 0_1 \ldots 0_{\mathbf{C}} >$
        $\mathbf{d}_t[highestFlyProbIndex(\mathbf{d}_1, \mathbf{d}_2)] = 1$
        return $\mathbf{d}_t$
    else
        if $(getMaleProb(d_1) > getMaleProb(d_1))$ & $(getMaleProb(d_2) > getMaleProb(d_2))$ then
            $\mathbf{d}_t \leftarrow< 0_1 \ldots 0_{\mathbf{C}} >$
            $\mathbf{d}_t[highestMaleProbIndex(\mathbf{d}_1, \mathbf{d}_2)] = 1$
            return $\mathbf{d}_t$
        if $(getFemaleProb(d_1) > getFemaleProb(d_1))$ & $(getFemaleProb(d_2) > getFemaleProb(d_2))$ then
            $\mathbf{d}_t \leftarrow< 0_1 \ldots 0_{\mathbf{C}} >$
            $\mathbf{d}_t[highestFemaleProbIndex(\mathbf{d}_1, \mathbf{d}_2)] = 1$
            return $\mathbf{d}_t$
    $\mathbf{d}_t \leftarrow meanDist(\mathbf{d}_1, \mathbf{d}_2)$
    return $\mathbf{d}_t$
```

References

[1] (2021). Confusion matrix.

[2] Ahmad, R., Ali, W. N., Nor, Z. M., Ismail, Z., Hadi, A. A., Ibrahim, M. N., and Lim, L. H. (2011). Mapping of mosquito breeding sites in malaria endemic areas in pos lenjang, kuala lipis, pahang, malaysia. *Malaria Journal*, 10(1):1–12.

[3] Bagnall, A., Lines, J., Bostrom, A., Large, J., and Keogh, E. (2017a). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3):606–660.

[4] Bagnall, A., Lines, J., Bostrom, A., Large, J., and Keogh, E. (2017b). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31:606–660.

[5] Bagnall, A., Lines, J., Hills, J., and Bostrom, A. (2015). Time-series classification with cote: the collective of transformation-based ensembles. *IEEE Transactions on Knowledge and Data Engineering*, 27(9):2522–2535.

[6] Bashar, K. and Tuno, N. (2014). Seasonal abundance of anopheles mosquitoes and their association with meteorological factors and malaria incidence in bangladesh. *Parasites & vectors*, 7(1):1–10.

[7] Battaglia, V., Gabrieli, P., Brandini, S., Capodiferro, M. R., Javier, P. A., Chen, X.-G., Achilli, A., Semino, O., Gomulski, L. M., Malacrida, A. R., et al. (2016). The worldwide spread of the tiger mosquito as revealed by mitogenome haplogroup diversity. *Frontiers in genetics*, 7:208.

[8] Baylis, M. (2017). Potential impact of climate change on emerging vector-borne and other infections in the uk. *Environmental Health*, 16(1):45–51.

[9] Boll, S. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on acoustics, speech, and signal processing*, 27(2):113–120.

[10] Bostrom, A. and Bagnall, A. (2015). Binary shapelet transform for multiclass time series classification. In *International conference on big data analytics and knowledge discovery*, pages 257–269. Springer.

[11] Braks, M., Anderson, R., and Knols, B. (1999). Infochemicals in mosquito host selection: human skin microflora and plasmodium parasites. *Parasitology today*, 15(10):409–413.

[12] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

[13] Cao, X., Wei, Z., Gao, Y., and Huo, Y. (2020). Recognition of common insect in field based on deep learning. In *Journal of Physics: Conference Series*, volume 1634, page 012034. IOP Publishing.

[14] Cardim Ferreira Lima, M., Damascena de Almeida Leandro, M. E., Valero, C., Pereira Coronel, L. C., and Gonçalves Bazzo, C. O. (2020). Automatic detection and monitoring of insect pests—a review. *Agriculture*, 10(5):161.

[15] Chadwick, L. E. (1939). A simple stroboscopic method for the study of insect flight. *Psyche*, 46(1):1–8.

[16] Chen, Y., Why, A., Batista, G., Mafra-Neto, A., and Keogh, E. (2014). Flying insect classification with inexpensive sensors. *Journal of insect behavior*, 27(5):657–677.

[17] Clements, A. N. (2013). *The Physiology of Mosquitoes: International Series of Monographs on Pure and Applied Biology: Zoology, Vol. 17*, volume 17. Elsevier.

[18] Conover, W. J. (1998). *Practical nonparametric statistics*, volume 350. John Wiley & Sons.

[19] Cooley, J. W. and Tukey, J. W. (1965). An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301.

[20] Cork, A. (1996). Olfactory basis of host location by mosquitoes and other haematophagous diptera. In *Ciba Foundation Symposium*, pages 71–84. Wiley Online Library.

[21] Crans, W. J. (2004). A classification system for mosquito life cycles: life cycle types for mosquitoes of the northeastern united states. *Journal of Vector Ecology*, 29:1–10.

[22] Cummins, B., Cortez, R., Foppa, I. M., Walbeck, J., and Hyman, J. M. (2012). A spatial model of mosquito host-seeking behavior. *PLoS Comput Biol*, 8(5):e1002500.

[23] Day, J. F. (2016). Mosquito oviposition behavior and vector control. *Insects*, 7(4):65.

[24] Deacon, T. W., Eisenkraft, N., Gilchrist, G. W., Reid, R., et al. (2001). Mosquito: A natural history of our most persistent and deadly foe. *American Scientist*, 89(5):456.

[25] Dempster, A., Petitjean, F., and Webb, G. I. (2020). Rocket: Exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495.

[26] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30.

[27] Deng, H., Runger, G., Tuv, E., and Vladimir, M. (2013). A time series forest for classification and feature extraction. *Information Sciences*, 239:142–153.

[28] Dickinson, M. H. (1999). Haltere–mediated equilibrium reflexes of the fruit fly, drosophila melanogaster. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 354(1385):903–916.

[29] Duncan, D. B. (1955). Multiple range and multiple f tests. *Biometrics*, 11(1):1–42.

[30] Ephraim, Y. and Malah, D. (1984). Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(6):1109–1121.

[31] Fanioudakis, E., Geismar, M., and Potamitis, I. (2018). Mosquito wingbeat analysis and classification using deep learning. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2410–2414. IEEE.

[32] Farmery, M. J. (1981). *Optical studies of insect flight at low altitude.* PhD thesis, University of York.

[33] Fawaz, H. I., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., Webb, G. I., Idoumghar, L., Muller, P.-A., and Petitjean, F. (2020). Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962.

[34] Flynn, M. and Bagnall, A. (2019). Classifying flies based on reconstructed audio signals. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 249–258. Springer.

[35] Flynn, M., Large, J., and Bagnall, T. (2019). The contract random interval spectral ensemble (c-rise): the effect of contracting a classifier on accuracy. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 381–392. Springer.

[36] FOSTER, W. A. and WALKER, E. D. (2002). 12 - mosquitoes (culicidae). In MULLEN, G. and DURDEN, L., editors, *Medical and Veterinary Entomology*, pages 203–262. Academic Press, San Diego.

[37] Georghiou, G. P. and Wirth, M. C. (1997). Influence of exposure to single versus multiple toxins of bacillus thuringiensis subsp. israelensis on development of resistance in the mosquito culex quinquefasciatus (diptera: Culicidae). *Applied and Environmental Microbiology*, 63(3):1095–1101.

[38] Gjullin, C. M. (1949). *The mosquitoes of Alaska.* Number 182. US Department of Agriculture.

[39] Goeldi, E. (1905). Os mosquitos no para reuniao de quarto trabalhos sobre os mosquitos indigenas, principahnente as espceies que molestam o homem. *Memorias do Museu Gceldi (Museu Paraense), de Historia Natural e Ethnographia, Para, Brazil*, 5.

[40] Gonzales, K. K. and Hansen, I. A. (2016). Artificial diets for mosquitoes. *International journal of environmental research and public health*, 13(12):1267.

[41] Gullan, P. J. and Cranston, P. S. (2014). *The insects: an outline of entomology*. John Wiley & Sons.

[42] Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12.

[43] Harker, J. E. (1973). Circadian rhythms in insects. *Biological Aspects of Circadian Rhythms*, pages 189–233.

[44] Hassall, K., Dye, A., and Bell, J. (2020). Opto-acoustic audio recordings of aphids and beetles.

[45] Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963.

[46] Kampen, H. and Werner, D. (2014). Out of the bush: the asian bush mosquito aedes japonicus japonicus (theobald, 1901)(diptera, culicidae) becomes invasive. *Parasites & Vectors*, 7(1):1–10.

[47] Kasinathan, T., Singaraju, D., and Uyyala, S. R. (2020). Insect classification and detection in field crops using modern machine learning techniques. *Information Processing in Agriculture*.

[48] Kawada, H., Tatsuta, H., Arikawa, K., and Takagi, M. (2006). Comparative study on the relationship between photoperiodic host-seeking behavioral patterns and the eye parameters of mosquitoes. *Journal of insect physiology*, 52(1):67–75.

[49] Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Stanford, CA.

[50] Kovats, R. S., Bouma, M. J., Hajat, S., Worrall, E., and Haines, A. (2003). El niño and health. *The Lancet*, 362(9394):1481–1489.

[51] Kupferschmidt, K. (2016). After 40 years, the most important weapon against mosquitoes may be failing. *Science*.

[52] Large, J., Lines, J., and Bagnall, A. (2019). A probabilistic classifier ensemble weighting scheme based on cross-validated accuracy estimates. *Data mining and knowledge discovery*, 33(6):1674–1709.

[53] Leisnham, P. (2010). Taking a bite out of mosquito research.

[54] Li, J., Zhou, H., Wang, Z., and Jia, Q. (2020). Multi-scale detection of stored-grain insects for intelligent monitoring. *Computers and Electronics in Agriculture*, 168:105114.

[55] Li, Z., Zhou, Z., Shen, Z., and Yao, Q. (2005). Automated identification of mosquito (diptera: Culicidae) wingbeat waveform by artificial neural network. *Artificial Intelligence Applications and Innovations*, pages 483–489.

[56] Lines, J., Taylor, S., and Bagnall, A. (2018). Time series classification with hive-cote: The hierarchical vote collective of transformation-based ensembles. *ACM Transactions on Knowledge Discovery from Data*, 12(5).

[57] Lubba, C. H., Sethi, S. S., Knaute, P., Schultz, S. R., Fulcher, B. D., and Jones, N. S. (2019). catch22: Canonical time-series characteristics. *Data Mining and Knowledge Discovery*, 33(6):1821–1852.

[58] Lunde, T. M., Bayoh, M. N., and Lindtjørn, B. (2013). How malaria models relate temperature to malaria transmission. *Parasites & vectors*, 6(1):1–10.

[59] Mayhew, P. J. (2007). Why are there so many insect species? perspectives from fossils and phylogenies. *Biological Reviews*, 82(3):425–454.

[60] Middlehurst, M., Large, J., and Bagnall, A. (2020a). The canonical interval forest (cif) classifier for time series classification. *arXiv preprint arXiv:2008.09172*.

[61] Middlehurst, M., Large, J., Cawley, G., and Bagnall, A. (2020b). The temporal dictionary ensemble (tde) classifier for time series classification. In *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*.

[62] Middlehurst, M., Large, J., Flynn, M., Lines, J., Bostrom, A., and Bagnall, A. (2021). Hive-cote 2.0: a new meta ensemble for time series classification. *arXiv preprint arXiv:2104.07551*.

[63] Middlehurst, M., Vickers, W., and Bagnall, A. (2019). Scalable dictionary classifiers for time series classification. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 11–19. Springer.

[64] Moore, A. (1991). Artificial neural network trained to identify mosquitoes in flight. *Journal of insect behavior*, 4(3):391–396.

[65] Moore, A. (1998). Development of a data acquisition system for long-term outdoor recording of insect flight activity using a photosensor. In *13th Conference on Aerobiology and Biometeorology, American Meteorological Society, Albuquerque, New Mexico*.

[66] Moore, A., Miller, J. R., Tabashnik, B. E., and Gage, S. H. (1986). Automated identification of flying insects by analysis of wingbeat frequencies. *Journal of economic entomology*, 79(6):1703–1706.

[67] Moore, A. and Miller, R. H. (2002). Automated identification of optically sensed aphid (homoptera: Aphidae) wingbeat waveforms. *Annals of the Entomological Society of America*, 95(1):1–8.

[68] Oastler, G. and Lines, J. (2019). A significantly faster elastic-ensemble for time-series classification. In *International Conference on Intelligent Data Engineering and Automated Learning*, Lecture Notes in Computer Science, pages 446–453.

[69] Organization, W. H. (2015). *Global technical strategy for malaria 2016-2030*. World Health Organization.

[70] Organization, W. H. (2016). *World malaria report 2015*. World Health Organization.

[71] Organization, W. H. et al. (2020). World malaria report 2020: 20 years of global progress and challenges.

[72] Peach, D. A. and Gries, G. (2020). Mosquito phytophagy–sources exploited, ecological function, and evolutionary transition to haematophagy. *Entomologia Experimentalis et Applicata*, 168(2):120–136.

[73] Peeters, G. (2004). A large set of audio features for sound description (similarity and classification) in the cuidado project. *CUIDADO Ist Project Report*, 54(0):1–25.

[74] Potamitis, I. and Rigakis, I. (2015). Novel noise-robust optoacoustic sensors to identify insects through wingbeats. *IEEE Sensors Journal*, 15(8):4621–4631.

[75] Potamitis, I. and Rigakis, I. (2016). Large aperture optoelectronic devices to record and time-stamp insects' wingbeats. *IEEE Sensors Journal*, 16(15):6053–6061.

[76] Price, R. D. (1958). Notes on the biology and laboratory colonization of wyeomyia smithii (coquillett)(diptera: Culicidae). *The Canadian Entomologist*, 90(8):473–478.

[77] Reed, S., Williams, C., and Chadwick, L. (1942). Frequency of wing-beat as a character for separating species races and geographic varieties of drosophila. *Genetics*, 27(3):349.

[78] Robert, V., Rocamora, G., Julienne, S., and Goodman, S. M. (2011). Why are anopheline mosquitoes not present in the seychelles? *Malaria Journal*, 10(1):1–11.

[79] Sarpola, M., Paasch, R., Mortensen, E., Dietterich, T., Lytle, D., Moldenke, A., and Shapiro, L. (2008). An aquatic insect imaging system to automate insect classification. *Transactions of the ASABE*, 51(6):2217–2225.

[80] Säwedal, L. and Hall, R. (1979). Flight tone as a taxonomic character in chironomidae (diptera). *Entomol. Scand. Suppl*, 10:139–143.

[81] Schaefer, G. and Bent, G. (1984). An infra-red remote sensing system for the active detection and automatic determination of insect flight trajectories (iradit). *Bulletin of entomological research*, 74(02):261–278.

[82] Sherman, J. (1949). Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix. *Annals of mathematical statistics*, 20(4):621.

[83] Shifaz, A., Pelletier, C., Petitjean, F., and Webb, G. I. (2020). TS-CHIEF: a scalable and accurate forest algorithm for time series classification. *Data Mining and Knowledge Discovery*, 34(3):742–775.

[84] Simard, F., Nchoutpouen, E., Toto, J. C., and Fontenille, D. (2005). Geographic distribution and breeding site preference of aedes albopictus and aedes aegypti (diptera: Culicidae) in cameroon, central africa. *Journal of medical entomology*, 42(5):726–731.

[85] Sotavalta, O. (1947). *The Flight-tone (wing-stroke Frequency) of Insects:(Contributions to the Problem of Insect Flight 1.).* PhD thesis, Suomen Hyönteistieteellinen Seura.

[86] Statistical Analysis System Institute, Inc., R. (1982). *SAS user's guide: statistics.* SAS institute.

[87] Sukumaran, D. et al. (2016). A review on use of attractants and traps for host seeking aedes aegypti mosquitoes. *Indian Journal of Natural Products and Resources (IJNPR)[Formerly Natural Product Radiance (NPR)]*, 7(3):207–214.

[88] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

[89] Unwin, D. and Corbet, S. A. (1984). Wingbeat frequency, temperature and body size in bees and flies. *Physiological Entomology*, 9(1):115–121.

[90] Unwin, D. and Ellington, C. (1979). An optical tachometer for measurement of the wing-beat frequency of free-flying insects. *Journal of Experimental Biology*, 82(1):377–378.

[91] Van Dam, A. and Walton, W. (2008). The effect of predatory fish exudates on the ovipostional behaviour of three mosquito species: Culex quinquefasciatus, aedes aegypti and culex tarsalis. *Medical and veterinary entomology*, 22(4):399–404.

[92] Wang, Z., Yan, W., and Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE.

[93] Weetman, D., Kamgang, B., Badolo, A., Moyes, C. L., Shearer, F. M., Coulibaly, M., Pinto, J., Lambrechts, L., and McCall, P. J. (2018). Aedes mosquitoes and aedes-borne arboviruses in africa: current and future threats. *International journal of environmental research and public health*, 15(2):220.

[94] Wen, C. and Guyer, D. (2012). Image-based orchard insect automated identification and classification method. *Computers and Electronics in Agriculture*, 89:110–115.

[95] Wigglesworth, V. (1933). The adaptation of mosquito larvae to salt water. *Journal of Experimental biology*, 10(1):27–36.

[96] Zhao, B., Lu, H., Chen, S., Liu, J., and Wu, D. (2017). Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28(1):162–169.