COMP 4121 Major Project

Emotion Recognition Using Physiological Signals in Real World Unconstrained Contexts

Contents

1. Introduction

1.1. Background

Emotion recognition systems make use of machine learning algorithms to look for patterns in data in order to try and predict when a user is happy or sad, or when they are calm or stressed. Physiological bio-signal data, such as heart rate or skin temperature, contains potentially useful information which we can use to predict emotion. However, in order for machine learning algorithms to carry out this prediction successfully we require a large amount of training data which is correctly labelled so that we know what kind of emotion each segment of the collected data corresponds to.

Practical applications of emotion recognition such as emotion based recommender systems, emotion tracking for mental health patients, or adaptive services like mood based music players generally require ongoing recognition of emotions in a real world and unconstrained context. This is very different to a lab environment where many variables can be controlled or observed and data is simple to collect. In the past this has been an issue for physiological emotion recognition because the devices required to collect many of the physiological signals have been impractical for use in ambulatory settings. Now days, however, many users are beginning to widely adopt wearable devices such as fitness tracking wristbands or smart watches which are capable of recording physiological signals. As a result physiological information can be collected from users without the need for any additional hardware and in a manner that is almost completely unobtrusive. However, while data might be becoming easier to collect (and thus attractive as means of recognising emotion) this data is of little use for training a machine learning algorithm unless accurately labelled, a task which is particularly difficult in these sorts of unconstrained contexts.

Finding an accurate way to label data with its ‘true’ value is often referred to as ‘establishing a ground truth’[1] and is a recurring issue in studies around emotion recognition. The approaches typically used depend heavily on the context in which emotion is being studied. If a study is performed in a lab, analysing emotions that are induced by a known stimulus, then the label applied to the data could simply be derived from the label given to the stimulus. However if the study is being performed in an everyday natural setting, where the stimuli are uncontrolled, then labelling becomes a much less trivial task.

1.2. Motivations

The majority of studies done to date using physiological signals for emotion recognition have done so in a lab context. Nowadays the technology to collect these signals can be just as easily used in real world as in settings, however, the same cannot be said for the collection of ground truth data. For the previous studies involving emotion recognition in the real world generally one of two approaches to data annotation is adopted, both of which are a form of self assessment. The first is the use of intermittent 'on the spot' ratings where a user is prompted at multiple points throughout the day to rate how he/she is currently feeling. This has been implemented in some phone based applications such as Moodscope, an iOS application for creating mood calendars [2]. The second approach is to ask the user to complete retrospective surveys at the end of their day to describe how they were feeling at various times. This is the approach that has been taken by various studies completed by teams at the MIT Affective Computing Lab to investigate both physiological and phone based recognition [3-6]. Both of these commonly used annotation approaches each have their own pros and cons, but there has been little work done to assess or compare their effectiveness experimentally. What's more the time scale granularity of these labelling schemes is quite low, generally labelling data in day long blocks with the smallest time windows being around 4 hours [2]. Predicting emotions for long time periods like this is not very helpful in applications where an instantaneous prediction of emotion is required, such as recommender systems which take emotions into account, so there is clearly benefit in exploring ways to increase the timescale granularity of labelling and thus prediction.

1.3. Aims

When building any emotion recognition system it is of course desired to achieve the best performance possible as measured by a variety of metrics. As such one of our aims will be to investigate what levels of performance can be achieved for a physiologically based classifier, in a real world setting, making predictions on a smaller time scale than the current minimum scale of 4 hours. The study will also investigate which settings or classifier configurations lead to the best performance. Because machine learning approaches like this are so dependent on the collection of solid ground truth data this study will also investigate the effects of the multiple ground truth collection methods which are applied to label the physiological data.

1.4. Outline

This study starts off by conducting a review of the relevant literature in our field. This review will look at collection methods for both physiological signals and their labels, feature extraction techniques and of course classification approaches. The report then goes on to outline the overall system design that has been chosen to allow for the most effective investigation of the real world physiological classification and the associated various annotation methods, before delving in to the design of each stage of the system in more depth. A description of the methods employed to implement the designed system is then provided. Finally the results collected from the system indicating the performance of various classifiers and their settings are outlined before being analysed in a discussion section.

2. Literature Review

In this section we will provide an overview of the important concepts related to our problem and a review of the relevant work completed to date. We begin by exploring emotional models and the way in which emotions can be defined before looking at the various aspects of data collection. Following this we look at ways of processing this data to prepare it for classification with a variety of feature extraction methods. Finally we explore the methods commonly used in the literature for the classification itself.

2.1. Emotion Models

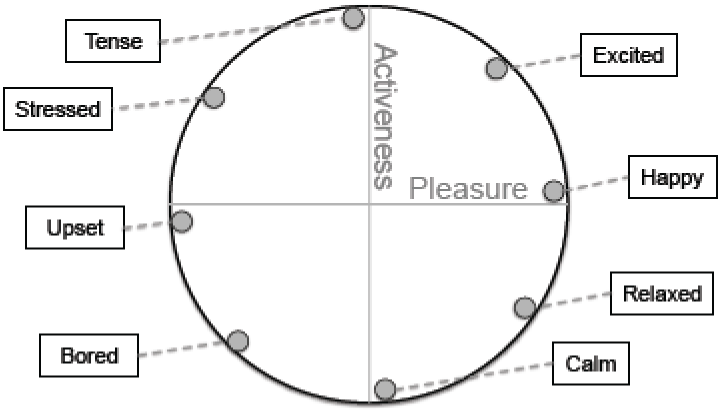


Figure 1 - The circumplex model [2]

Before exploring potential methods for collecting emotion labels it is important to define the way in which emotions can be represented, as this will influence the way in which these labelling systems are designed. There are two common representations (or models) used for emotions [1, 7] . The first is a set of discrete classes, such as Clynes' set of eight basic emotions. In this case the model classifies every sample into one of the categories in the set e.g. anger, fear, joy, or sadness. This model was used more in early research into emotion recognition such as [8]. The second representation is a continuous dimensional model often referred to as the circumplex model. This model consists of two continuous axes, valence (positive or negative) and arousal (level of excitement or “activeness”) [9]. Each sample is classified by assigning it a value for each axis independently. This places all samples in a 2 dimensional plane. Regions of this plane can be chosen to represent specific emotions as can be seen in figure 1 taken from [2]. Note that valence has been replaced with pleasure, and arousal with activeness. A third axis also exists in some studies called “dominance”, however, it is seldom used.

2.2. Collection

2.2.1. Annotation Methods

The labelling of data with an accurate ground truth is a very important step in the machine learning process and one that we will focus much attention on in this study. There are a number of methods used, and the choice depends on the setting in which the study is taking place.

Studies carried out in the lab or constrained environments can quite easily make use of pre classified stimuli for the purpose of labelling. In such cases an emotion is induced by providing the subject with a stimulus, such as a film clip, and then the data collected is simply given the same label as the stimulus which has been pre-defined. Examples include images such as those in the International Affective Picture System [10], audio such as that in the International Affective Digital Sounds [1], video clips or short movies [11, 12], or music [13]. It is also possible to use pre-classified stimuli in natural contexts by orchestrating specific emotion inducing events for a subject, for example arranging for a close friend to call them with good news [1]. This approach, however, could not be repeated frequently without arousing suspicion in the subject and also requires significant planning and organisation.

The most common approach for labelling data in natural settings, however, is through self assessments performed by the test subject. These self assessments generally come in two forms. The first is ‘on the spot’ ratings which ask the user at various points throughout the day how they are feeling at the current time. These could also come in the form of a push-notification on the subject’s phone asking them to rate their mood on a simple scale [2]. The second approach is to ask the user to complete retrospective surveys at the end of their day to describe how they were feeling at a range of times [3-6].

Most of the related studies undertaken to date have used one of these two methods, however they have limitations. Firstly in the case of ‘on the spot’ ratings it is possible that by interrupting the user to ask them to respond to questions on their current mood state we might be changing or affecting this state through the mere action of the interruption [14]. Similarly users may forget to respond when they are busy and there is a limit to how regularly we can reasonably expect a user to answer such questions before our system loses the property of being unobtrusive. For these reasons using such a system for labelling data limits the temporal resolution that we can achieve (the minimum time interval that we can break a day up into). On the other hand for end of day reviews, while the user is not at risk of being interrupted since they can choose to submit reviews at a time convenient for them, there is a chance that the subtleties of the emotions they experienced might be forgotten. A further option for self-assessment is for the model to ask for clarification only when it is unsure or predicts a change in emotion state [2].

Another method sometimes used for obtaining ground truth is the use of trained judges to decide on the emotion being displayed [1]. Clearly in a natural setting it becomes difficult to capture enough observational data for such a judge to review which rules out the feasibility of this in our application.

2.2.2. Modalities

Modalities are the actual indicators upon which a machine learning algorithm can base its predictions. They represent the raw data that a system needs to collect. Modalities of various forms have been successfully used for the task of emotion recognition. The most commonly used are physiological signals, speech and audio based signals, image and video signals and more recently smartphone sensor signals.

While speech and image processing are both well established techniques for emotion recognition this study has chosen not to focus much attention on them. Neither approach is particularly well suited to an ambulatory environment [8] due to the more obtrusive nature of sensors required such as microphones (with cords and added hardware) and cameras (which are difficult to place). Seen as investigating annotation methods in continuous real world unconstrained environments is our primary aim it was decided these options were less appropriate. Of course with more complex systems and specialised hardware these difficulties could be overcome.

2.2.2.1. Physiological Modalities

Studies that make use of physiological modalities [4, 5, 8, 11, 13, 15-17] tend to collect similar sets of signals. The most common signals appears to be heart rate (often derived from blood volume pressure), skin conductance (or galvanic skin response), accelerometer data, and skin temperature. Other signals which are often used are respiration measures such as breathing rate, electromyography signals (EMG), and electroencephalography signals (EEG), however the sensors required to collect these signals are not well suited to an ambulatory environment.

2.2.3. Collection Methods

Relevant studies have used a range of different devices for collecting both physiological signals and smartphone data. We present a summary of previously used and potential devices for collecting physiological signals in table 1.

Device	Notes
Empatica E4 [18]	Complete set of most common sensors Easy to interface with good APIs - accessible raw data Bluetooth connectivity
Affectiva Q Sensor [4, 5]	Discontinued
Jawbone	Successor to Affectiva Q Sensor Consumer targeted, no raw data access
BodyMedia Sensewear [11]	Discontinued
Biopac System [17]	Requires electrodes and patches
Actigraph [18]	Accelerometer only Can add external bluetooth HR monitor
Generic HR Monitors	Potential limited access to raw data

Table 1 - Collection devices for physiological signals

Another aspect of the collection methodology that is worth considering is the choice of participants for the study. Some studies focus only on one participant and build a personalised model, while others use multiple participants. In the latter case further consideration must be given to the age, gender, and cultural background of participants [1]. The length of time over which data is collected is also important, and whether or not collection is continuous or segmented.

2.3. Feature Extraction

Features are the meaningful values that we are able to extract from analysing raw data and their nature will vary depending on the form of the data. Feature extraction is simply the process of computing these values, but depending on the quality of the data or signal this can include further processes such as noise reduction [5].

Data that comes in the form of continuous signals such as skin conductance can be analysed by statistical means to extract features. Common statistics include mean, minimum, maximum, variance, range, standard deviation, area under the curve, multi-bin histograms and other forms of frequency analysis [8, 11, 19]. These statistics can also be extracted from normalised, differentiated, and first or second difference versions of the raw signals [4].

In a study done my MIT's Affective Computing lab on real world emotion recognition [4] features used were evaluated based on maximising information gain, which is equivalent to minimizing entropy, which can be seen as the reduction in uncertainty about one variable after observing another [20].

Entropy is expressing in the following equation, where P(x_i) is the probability of x_i being the value of random variable X:

$$H(X) = -\sum_{i} P(x_i)\log P(x_i)$$

The information gain can then be expressed as the change in entropy for X resulting from the observation of Y:

$$I(X,Y) = H(x) - H(X|Y)$$

By using this method the study was able to identify which features provided the highest information gain. For physiological features it was shown that various skin conductance statistics (such as median, and standard deviation) as well as accelerometer statistics (standard deviation) provided the highest information gains. Similarly they found that features related to sleep were also highly effective.

In another study a similar process was followed again using entropy to choose the best features for classification [5]. Both skin conductance and accelerometer data were again shown to be effective features.

2.5. Classification

Classification approaches are widely varied in relevant studies, but it is not particularly meaningful to compare their performance across studies due to the varied nature of data collection methods that these approaches are applied to and then the various combinations of feature selection and machine learning algorithms. None the less it is useful to summarise the approaches out there and to look at those which are frequently shown to be successful.

2.5.1. Feature Selection

Once features have been created it is sometimes found to be beneficial to reduce or simplify the feature set by using methods of feature selection or transformation before providing the feature vectors to a machine learning algorithm. A summary of the feature selection and transformation approaches used in the relevant literature is presented in table 2. Of these the most commonly used types of feature selection seem to be sequential forward selection (SFS) and sequential forward floating selection (SFFS), both very similar algorithms with SFFS just adding an extra "floating" step at each iteration (where it is checked if one of the currently selected features can be removed to increase performance).

Feature Selection	**Notes**
Sequential Forward Selection [2, 5, 13]	Most frequently used, very simple
Sequential Backward Selection	Very similar to SFS
ANOVA [15]	Analysis of variance method
Fisher Projection [8, 13]	Feature transformation rather than selection
Sequential Floating Forward Search [8]	Similar to SFS but at each step a step of SBS is also run
Principal Component Analysis [6, 13]	Doesn't consider class information

Table 2 - Feature selection approaches

2.5.2. Machine Learning Algorithms

Once features have been created (and optionally selected) the final step of classification (or prediction) can be carried out. A portion of the collected data is first used to train a classification model (using machine learning algorithms), and then a second portion of data is used to test performance. A summary of frequently used machine learning algorithms for physiologically based emotion recognition systems is given in table 3. Among the most frequently used, which can be a useful indicator of an algorithm's strength, is K Nearest Neighbour (KNN). Some studies have also compared different algorithms on the same data sets, and SVMs appear to have performed well against other classifiers in these cases.

Algorithm	**Notes**
K Nearest Neighbour [4, 8, 11, 13]	Most common, and very simple
Logistic Regression [2, 21]	Output can be continuous, weights meaningful
Neural Networks [4, 11, 13, 17]	Common, little information about features importance
Support Vector Machines [4, 5]	Shown to perform well against other methods
Random Forests [4]	Shown to perform well against other methods
Naïve Bayes [4, 8]	Simple, easy to add further evidence later

Table 3 - Machine learning algorithms

2.5.3. Validation Methods

While training classifier models it is important to have methods of validating the models' effectiveness. This is useful firstly for deciding which parameters and settings (provided to the machine learning algorithms) are optimal but also to output final performance results. The three prominent validation methods that are used are K fold cross validation, leave one out cross validation and the use of held out validation sets. These methods are often used in combination [11].

The leave one out cross validation (LOOCV) [2, 8, 11, 13] approach is implemented by removing one training sample at a time from the training set, and training the model with the remaining sample. The model is then tested with the single training sample that was left out, and its predicted value compared to the known value. This is repeated for each training sample in the set until each has had a turn being 'left out'. Because labelled emotion based training data is often costly to collect this approach is very popular because it optimises the size of the training set [1]. This approach does have issues however as the models trained at each iteration are highly overlapping (as they differ by only one sample).

To ensure that there is no overlap between training and validation sets we can use a held out validation set [4, 11]. In this scenario a subsection of the data will be put aside (the validation set) for final testing and the remaining data (the training set) will be used to optimise parameters and train the model. Once parameters are decided and the model is trained, the held out validation set is used to evaluate the classifier's performance. While this ensures that the training and validation sets have no overlap it reduces the size of the training set.

K fold cross validation [5, 21] works similarly to LOOCV but instead of removing individual samples, removes partitions of the data set. The first step is to split the data into K folds, and then a model is trained on K-1 of the folds and tested on the left out fold. This approach is almost a trade-off between the previous two approaches causing less overlap between training sets than LOOCV but decreasing test set size by less than the held out method.

2.5.4. Evaluation Methods

Many papers use accuracy as a scoring method for optimising and evaluation the performance of classifiers. When the problem involves normal multi-class classification then this is quite appropriate, however when the scale that one is classifying on has a natural ordering (i.e. is ordinal) then this measure does not tell the whole story [22]. Accuracy only tells us how many predictions were made exactly correct, it doesn't take into account how far from the correct prediction and incorrect predictions were. For example if the true value of a sample on a 5 point scale is 4, then a classifier which predicts 5 is much closer than another classifier which predicts 1. To account for this two common scoring methods that can be used are mean absolute error (MAE, the mean of the absolute difference between true and predicted values) and mean squared error (MSE, the mean of the square of the difference between true and predicted values) [22, 23]. Of these two measures MSE places the greatest performance penalty predictions that are very wrong.

Whether accuracy, MAE, or MSE is used as a scoring method, the score alone does not provide enough information about the performance of a classifier. A comparison should also be made with a trivial classifier as a benchmark [23]. Various trivial classifiers exist such as the majority class classifier, average class classifier, or classifiers that use ratings assigned to adjacent samples [2].

3. System Design

In this section we revisit the initial aims outlined in the introduction and use them to inform our generation of a design goal for our system. Once the overall goals are established we will provide a high level overview of the system's construction and show how the various components fit together. Following this high level overview we will then explore the design of each sub component of the system in terms of requirements, inputs and outputs. We leave details of implementation to the following section.

3.1. Design Goal

The main aim of this study is to investigate what levels of performance can be achieved for a physiologically based classifier, in a real world setting. At the same time we also want these predictions to be made on a smaller time scale than is currently used. So the goal will be design a system which maximises prediction performance based on a range of metrics (not just accuracy). We also want this system to be applicable to real world contexts and as such we want to create a system which can collect data in this type of context. We would also like the system to be as unobtrusive as possible to the user but at that same time we want to try and maximise the granularity for which labels are collected.

3.2. System Overview

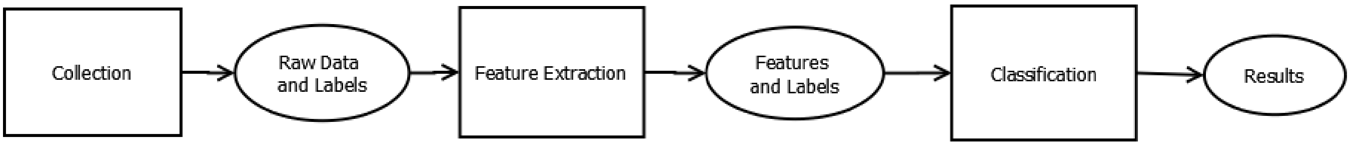


Figure 2 – System Block Diagram

The emotion recognition system that has been designed comprises of 3 stages or modules, each with a relatively unique purpose. This is demonstrated graphically in figure 2 below. The first stage in the system is collection. The collection module is responsible for recording both physiological data as well as asking the user to provide emotion labels in variety of forms to be attributed to this data. These data and labels are then downloaded from their respective devices and loaded into the second stage which we call the feature extraction module. After reading in all the labels provided to it (whether they be ratings or reviews) the feature extraction module associates all the physiological data with its correct label. Once this association has been made the data is used to compute various features which are then passed to the classification module. The final classification module takes the features and labels and creates a classifier whose settings and hyper parameters are optimised before evaluating the performance of the classifier and outputting the evaluation results for analysis.

3.3. Collection Module Design

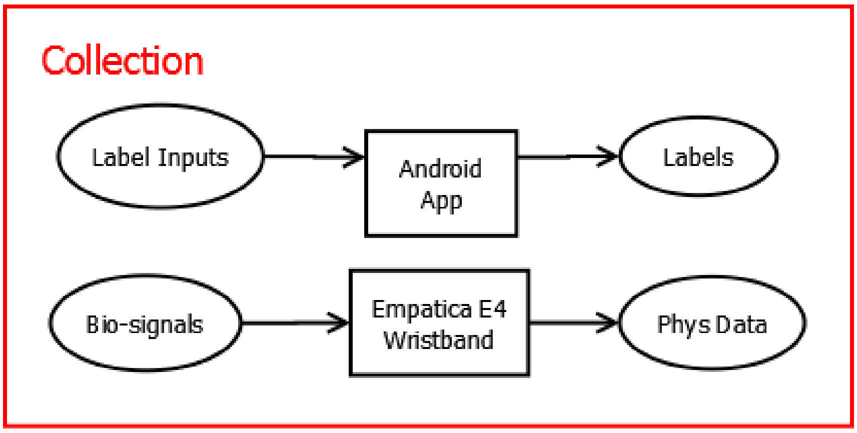


Figure 3 – Collection Module Block Diagram

The collection module is made up of two components, one for collection of the actual physiological raw data and another for the collection of the associated labels. The first component which collects the labels is a smartphone application. The function of this sub-component is the collection of annotation data, denoted in the diagram as label inputs. The second component of this module is a wearable device which is able to collect a range of physiological bio-signals, the Empatica E4 wristband.

3.3.1. Requirements

The two primary goals of the collection system are first to collect data, and second to collect labels to associate with this data. Each of these goals can be broken down further.

Data should be collected continuously and reliably in a way that user error is unlikely to result in the erasure, corruption or non-recording of data. This data collection process should occur silently in the background such that it is as unobtrusive to the user as possible. A large enough volume of data of data should be collected to allow for effective prediction, and to avoid issues such as overfitting.

Emotion labels similarly need to be collected continuously and reliably. Labels of both ‘on the spot’ rating and retrospective review style should be collected for the same data. Both reviews and ratings should be convenient for the user and create the smallest interruption possible so as to still be deemed unobtrusive. At the same time it is desirable to have annotations as frequently as is practical, in order to maximise the time scale granularity. The system should also have methods in place to ensure that annotations are made regularly and not forgotten.

Finally, both data and labels must be easily accessible for transfer to PC for analysis. The label collection process should be a drain on system resources such as battery life, CPU time, or mobile data allowances, again for the sake of being as unobtrusive as possible.

3.3.2. Inputs

As seen in figure 3 there are 2 inputs (or groups of inputs) to this module.

- Label inputs – This is the annotation data provided by the user. This is collected through the app in the form of on the spot ratings and also end of day reviews. The user is prompted to submit these by the app.
- Physiological Bio-signals – A range of bodily signals to try and capture the physiological response to emotions that are being experienced. These signals include indicators such as how much the user is sweating, their skin temperature, their movement and heart activity.

3.3.3. Outputs

As seen in figure 3 there are 2 outputs (or groups of outputs) for this module.

- Labels – The annotation data provided by the user is recorded along with a timestamp of exactly when the annotation was provided. Ratings and reviews are saved in separate files.
- Physiological (Phys) data – Physiological signals are sampled at varying rates and the samples are saved into a separate file for each physiological signal type e.g. a skin temperature file, and a skin conductance file. Each file is labelled with a timestamp signifying the beginning of collection, and a sampling rate.

3.4. Feature Extraction Module Design

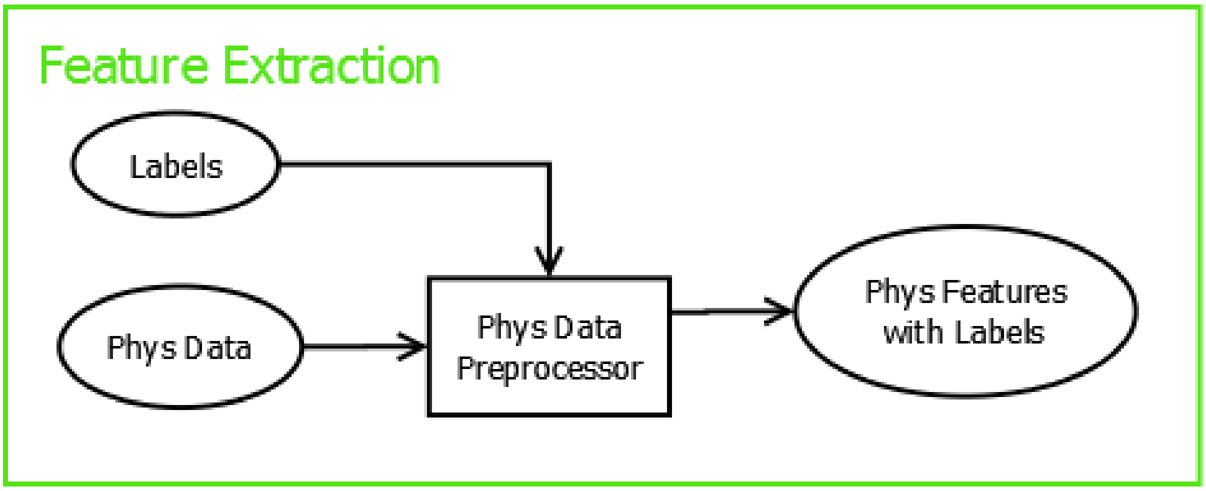


Figure 4 – Feature Extraction Module Block Diagram

The labels and physiological data produced by the collection module are taken in by the physiological data pre-processor which first reads in each label and creates a data point to associate with it. After reading in the labels and their time stamps the physiological pre-processor then reads in the physiological data provided by the collection module and associates the correct segments of this data with each data point. Features are then calculated and output in a set of feature vectors.

3.4.1. Requirements

The main goal of the feature extraction stage is to take in the collected data and labels and compute features to associate with each label. After taking in all labels this module will need to correctly associate the segments of data in time to each label, and be able to do this for a variety of annotation schemes. For each of these segments a good range of features should be calculated. These features must then be output in a format that can be read by the classifier stage. The module will also need to gracefully handle corrupted, erroneous or missing data. All this should be done in a way that is easily extensible to allow for the addition of new data types if desired.

3.4.2. Inputs

As seen in figure 4 there are 2 inputs (or groups of inputs) to this module. Each of these inputs is received from the collection module.

- Labels – The annotation data provided by the user. Ratings and reviews are saved in separate files, and each is labelled with a timestamp.
- Physiological (Phys) data – Physiological signals are sampled at varying rates and the samples are saved into a separate file for each physiological signal type. Each file contains a header with a timestamp signifying the beginning of collection, and a sampling rate.

3.4.3. Outputs

As seen in figure 4 there is 1 output from this module which is passed to the classification module.

- Physiological features and labels – Each data point (corresponding to a label) has all its features output on one line. Thus clearly the number of lines should equal the number of labels.

3.5. Classification Module Design

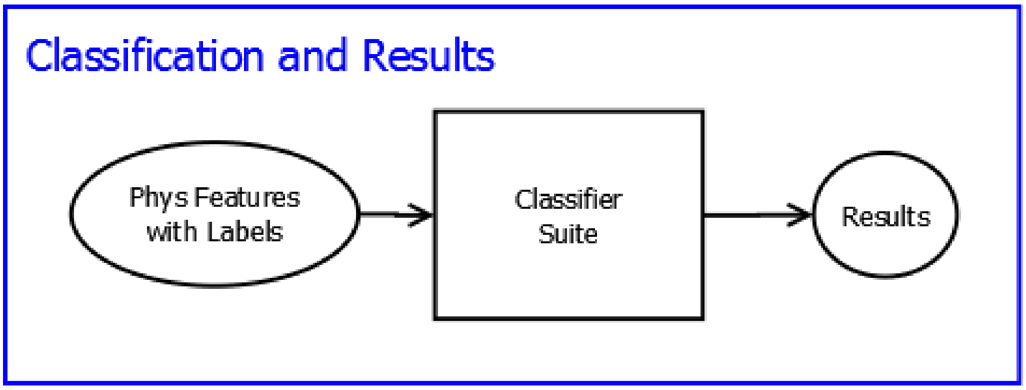


Figure 5 – Classification Module Block Diagram

The Classification Module consists of one core component, the classifier suite. This component receives inputs from the feature extraction module. As a first step the classifier suite takes in the multiple different types of labels and the physiological features and generates a range of different classifier scenarios characterised by the different combinations of these features and with each label type along with other settings. Once this range of classifier scenarios has been generated, a classifier can be created for each scenario which can be then trained on its associated data and labels. Each classifier's performance is then evaluated by comparing with a trivial classifier on the same data set over a range of metrics. The results are printed out in both a readable form and saved in a numerical format for ease of analysis.

3.5.1. Requirements

Once the system has created features and applied various annotation schemes to them the goal is to use these combinations to create a range of different classification scenarios. These classification scenarios should also apply more than one type of machine learning algorithm to create greater breadth in the types of scenarios created. For each scenario that is created a classifier should be built, and its hyper-parameters and settings (such as feature selection/transformation) tuned, to maximise its effectiveness at predicting the associated data. There will be many different classification scenarios so it should also be possible to generate these scenarios, train optimal classifiers and evaluate the performance of each in a way that is automatic. The module should also provide the performance of a trivial classifier for each scenario against which performance of the trained classifier can be compared. Evaluation should provide a range of metrics and scores for each classifier which are saved in both human readable and numerical formats.

3.5.2. Inputs

As can be seen in figure 5 there is only 1 type of input to this module.

- Physiological features with labels – The physiological features computed by the feature extraction module and their associated labels of various types.

3.5.3. Outputs

As can be seen in figure 5 there is only one output from this module.

- Results – Various performance metrics for the classifier in each scenario are output alongside metrics computed for a trivial classifier on the same data.

4. System Implementation

Now that the overall design and requirements of the system have been established we will now present the details of the implementation. To achieve this we will group the descriptions of implementation into the modules that were introduced in the section above. First we will look at the collection module where physiological data was collected using a multi-sensor wristband along with emotion labels using a smartphone application. Then we shift our attention to the feature extraction module to look at how the raw data was processed and which features were created. Finally we explore the classification module to look at how different classifier scenarios were generated, how classifiers were trained, and then of course how they were evaluated. While presenting the various design choices, the reasons behind these decisions and how they help to achieve our aims of physiologically based classification in a real world unconstrained context will also be explained.

4.1. Collection Module Implementation

As explained in the design section the collection module is made up of two important components, a wearable device for collecting physiological signals and an android app for collecting annotation data. However, before giving a summary of the work done on these two components we will first explain the emotional scale upon which ratings will be made and make a note on the participants used in the study.

4.1.1. Emotional Model

As explained in section 2.1 when it comes to emotional models there are generally two options. The first is the discrete model where emotions are classified into one class in a set. The seconds is the circumplex model which is a 2 dimensional continuous model where emotions are classified according to both a valence and an arousal score. In this study we have chosen to use a simplified version of the circumplex model by investigating valence. Essentially the valence only model projects the circumplex model into one dimension. Thus when users are asked to rate how they are feeling they provided a valence rating on a scale of 1 to 5. The primary reason for choosing only one axis was to make ratings as easy as possible for a user. Less scales to report on means that ratings are quicker and easier to submit and create less of a distraction. The reason that valence was chosen as the axis (as oppose to arousal) was firstly because it is a less ambiguous concept for users but secondly because valence has also been found to correlate better with other useful mental health indicators.

4.1.2. Participants

It is worth noting that data collected in this study comes from a single participant. While data from more participants is desired it was not feasible within the scope of the project and with the time and resources available. It could also be important to note that the participant was also the researcher, an Australian male university student in his early 20s. To ensure that a large enough volume of data was collected, data collection was carried out over a continuous period of a month (with some days omitted when collection devices were out of power, or when the devices could not be used).

4.1.3. Collection of Physiological Signals

The device chosen for collection of physiological data was the Empatica E4 wristband. This device was chosen primarily because of its complete and varied set of sensors that all come bundled together in the one device. The 4 sensors which record skin conductance, skin temperature, blood volume pulse and 3 axis accelerometer data cover all the most commonly used modalities in the relevant literature. The device is also coupled with a very user friendly software package. Downloading data off the device is as simple as connecting the device to a PC via USB and once downloaded all the data is all stored in an online account which allows the researcher to view sessions in graphical forms and also to download raw data in readily accessible CSV formats for analysis. Furthermore the device has good battery life of up to 72 hours and a small form, both factors which make the device less obtrusive to the user who has to wear it. An example file header can be seen below for a file containing the skin conductance data.

```
1475911978.000000
4.000000
0.000000
0.052519
0.081980
```

The first line of the header is the time after the epoch in milliseconds when the session was started, the second line contains the sample rate and then subsequent lines contain data values.

4.1.4. Annotation Collection

The implementation of annotation collection was by far the most substantial volume of work for the collection module. This was done using a smartphone application as this was seen to be the simplest and least obtrusive method for collecting this information. Almost all users have smartphones so it means that further hardware is not required. Most users also take their smartphone with them everywhere so this solution also makes it unlikely that a user will forget or be unable to make annotations so long as they are prompted. The android operating system was chosen as the platform to develop for, firstly because it is supported by more devices but secondly because the provided system APIs provide a much wider range of functionalities and access than other common smartphone OSes like iOS.

To explain the implementation of the app we will first provide a structural overview and then explain some of the decisions relating to the appearance of the application, before going on to explain the details involved in the code design for both the rating and review activities.

4.1.4.1. App Overview

The app for collecting annotation data is comprised of 6 components as can be seen in the block diagram in figure 6. Three of these components are android activities which essentially represent the different UI screens (and associated background implementation) of the tasks performed by the app. The central activity is the MainActivity whose UI component is the app's main menu screen. This activity is also responsible for scheduling all the background services that need to be run. The 3 other components (which are not activities) are all services which means that they are designed to be run in the background at a scheduled time, and as such are all scheduled by the MainActivity. The RateNotificationService and ReviewNotificationService, as the names suggest, are responsible for creating notifications which are sent to the user's phone to remind them to submit either a rating or a review. The AlarmStopper is a service which exists to stop the RateNotificationService from being scheduled after the user's chosen bed time. The main activity can also open the two other activities (the RateActivity and the ReviewActivity) which are the components which display the UI screens with which a user can interact to submit emotion annotations.

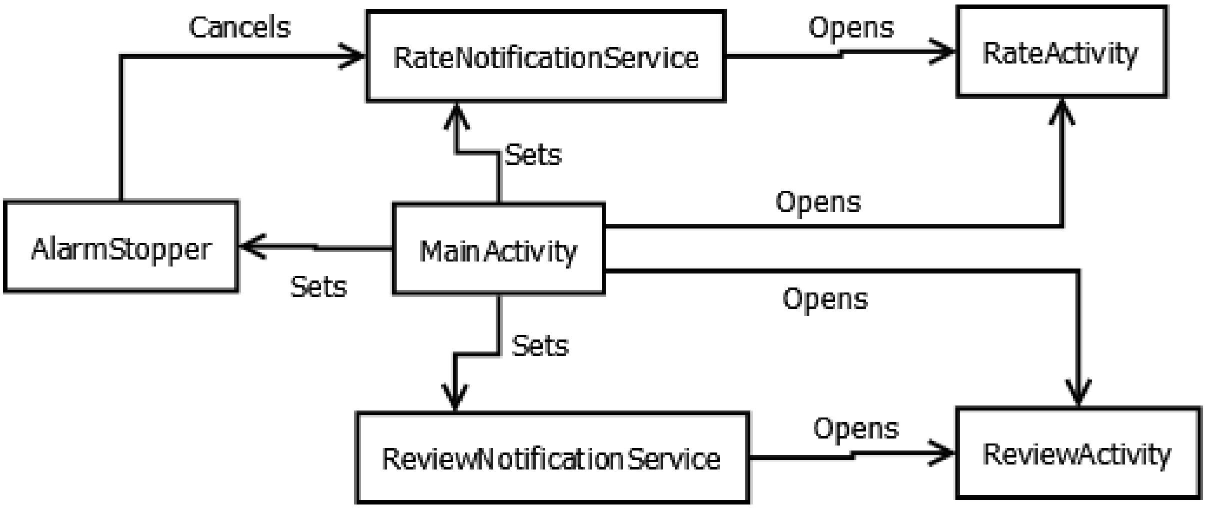
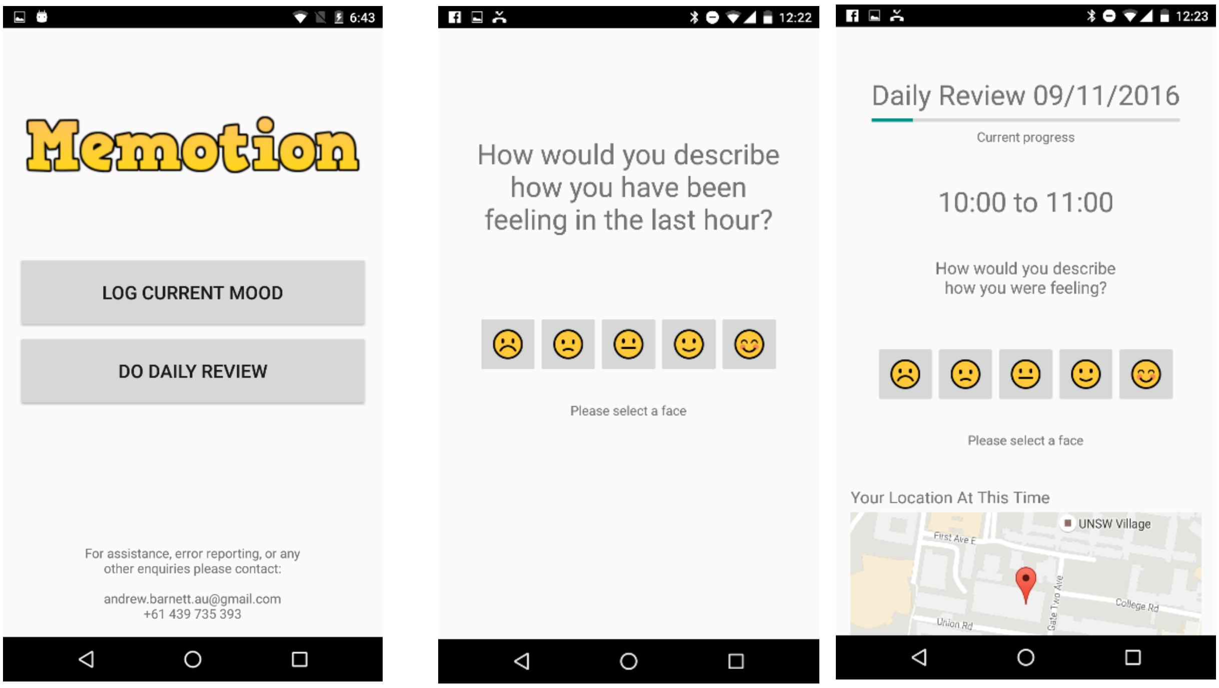


Figure 6 – Annotation Application Components

4.1.4.2. UI Screens

The UI screens associated with the MainActivity, the RateActivity and the ReviewActivity can be seen in the figure below. The general approach with the appearance of the app was to keep everything as plain and simple as possible. Obviously this makes the app intuitive and easier to use but more importantly it means that the app is less likely to have an effect on the user's emotions subconsciously.



For assistance, error reporting, or any other enquiries please contact:

andrew.barnett.au@gmail.com
+61 439 735 393

How would you describe how you have been feeling in the last hour?

Please select a face

Daily Review 09/11/2016

Current progress

10:00 to 11:00

How would you describe how you were feeling?

Please select a face

Your Location At This Time

Figure 7 – MainActivity (left), RateActivity (middle) and ReviewActivity (right) UI Components

4.2. Feature Extraction Module Implementation

The Feature Extraction Module is made up of one component, the Physiological Data Pre-processor, which takes the form of a Matlab file. We will first explain the overall function of this module by walking through pseudocode in an overview. Following this we will look at the most important steps in the pseudocode in more detail, such as initial processing, matching of data with labels, and calculation of features, providing small code snippets where appropriate.

4.2.1. Overview

The general functionality and structure of the Physiological Data Pre-processor is demonstrated in the following pseudocode:

```
Read in rating timestamps from chosen rating file

For each recorded session (i.e. set of data files):
      For each data file type:

            Read in the data
            Run a median filter on the data

            For each timestamp:

                  If the current data file overlaps with this
                  timestamp :
                        Take all the data relating to this
                        timestamp
                        and compute the associated features.
                        Store features in results matrix

Print results matrix out as CSV file
```

Running the code represented by the above pseudocode will result in a feature file being output relating to one type of annotation data, and as such must be run multiple times to generate the features associated with each annotation type.

4.2.2. Processing

Once the raw data has been read in a small amount of processing must be carried out. If the data type is blood volume pulse then the absolute value of the signal is taken. Similarly if the data comes from the accelerometer then each reading is made up of an x, y, and z component and we choose to combine these in quadrature to give an acceleration magnitude i.e.

```
data(i) = sqrt(accData(i,1)^2 + accData(i,2)^2 + accData(i,3)^2);
```

Then regardless of data type all raw data signals are filtered using a median filter with order 9 in order to remove any noisy spikes in data.

4.2.3. Attributing Samples

In order to attribute samples to the correct emotion labels, a sample index is maintained as we loop through a data file. By knowing the start time of the file and the sampling rate we are able to use the sample index to easily compute the time that a sample was collected. We attribute all data in the 3 hours proceeding a label to that label. The process of updating this sample index and determining the start and finish indexes for the current segment of data can be seen in the code snippet below.

```
%Keep track of which sample we are up to
sampleCount = 1;
%Go through each labelled segment
for i = 1:1:size(ratingTimes,1)

    %Only include data that is up to 3 hours before the
        ratingTime
    MAX_DATA_AGE = 3*60*60;
    while(startTime + sampleCount/sampleRate + MAX_DATA_AGE
        < ratingTimes(i))
        sampleCount = sampleCount + 1;
    end

    %The current segment starts at and includes this index
    startIndex = sampleCount;

    while(startTime + sampleCount/sampleRate <
        ratingTimes(i))
        sampleCount = sampleCount + 1;
    end

    %The current segment finishes at and includes this index
    finishIndex = sampleCount - 1;

    %Make sure that we haven't run out of samples
    if(finishIndex < startIndex || startIndex >
        size(filtData,1))

        %We have no samples in this segment

    else

        %Compute features
        %startIndex and finishIndex can now be used
        %to access the correct segment of the data array
        ...

    end
```

4.2.4. Features

A range of features are computed for each signal so as to maximise the information that can be provided to the final classification module. The signals that are provided to the feature extraction module are all of those made available by the Empatica E4 wristband:

- Skin Temperature
- Skin Conductance
- Blood Volume Pulse
- Heart Rate
- Accelerometer Data

For each of these signals not only are features calculated for the raw data signal but we also transform the raw signal to create a normalised signal and a derivative signal. The normalised signal is calculated as:

```
    normData(x) = (rawData(x) - rawMean)/(rawMax - rawMin);
```

And the derivative signal (which is really an approximation) is calculated by taking the difference of adjacent elements.

For these 3 signals we then calculate the same range of features. These being:

- Mean
- Min
- Max
- Variance
- 5 Histogram Features

Where the 5 histogram features are quite simple calculated by creating a histogram with 5 equally spaced bins and then calculating the percentage of values which fall into each bin. With this combination of features we are able to capture much of the information about the distribution of values in the signal.

Finally there are two extra features which are calculated to try to identify if the wristband was removed for any period of time during the given segment of data. These features are calculated purely by looking at skin temperature when it falls below an experimentally determined threshold (to a temperature which is highly unlikely if the device were actually being worn). It was initially thought that segments identified as unworn would be removed however because the removal of the wristband was generally associated with certain activities that may associate with happiness (such as exercise or showering) this information was left in as a potentially useful feature.

4.2.5. Output

As features are calculated (in order of signal type) they are stored in a results matrix. The reason they are not immediately printed is that we would like to print all features associated with one label on the same line. Seen as we process the data in the order of signal type (and not in the order of labels) if we were to print out features as they are calculated then the order would be wrong, hence we save them in a matrix first. Once all features are calculated the results matrix is written to a CSV.

4.3. Classification Module Implementation

The classification module is where we get the chance to investigate all the data that has been collected and processed to see whether or not meaningful predications are able to be made. This module makes use of the machine learning toolkit called scikit learn which is based in python and as such this final module takes the form of a python script. To explore the implementation of this module we first look at the pseudocode in a general overview. Following this the finer details of the most important sections in the pseudocode are discussed.

4.3.1. Overview

The following pseudocode represents the procedure used by the classifier suite to set up the variety of classifier scenarios, create and optimise a classifier for each of these scenarios, and then calculate performance metrics for evaluation (also for each scenario).

```
Choose settings to be used this run.

Initialise lists of data types, scale types,
and classifier types over which to test.

Initialise lists of hyperparameters and classifier
specific settings from which to choose.

For each scale type:
     For each label data type:
          Load the data associated with that label type.
          Prepare the related features.
          Create training and validation sets.
          Compute the data distributions in these sets.
          Compute trivial classifier scores.
          For each score method:
               For each ML algorithm:
                    Tune hyperparameters with
                    cross validation.
                    Apply feature selection and/or PCA.
                    Calculate performance measures.
                    Output performance results.
```

When the above code is run two files are generated. One is a human readable output file while the other is a CSV which can then be opened in an analytical software package such as Microsoft Excel to extract useful results.

4.3.2. Setting Selection and List Initialisations

A variety of settings are able to be selected when running the classifier suite which include options such as whether or not feature selection is enabled, which results are printed, and what debug information is displayed. Similarly the range of data types, scale types, and classifier types which are used to generate the different classifier scenarios are also initialised at the top of the file. These initialisation lists are composed of:

- Data types: Reviews, Ratings, Combination (of reviews and ratings)
- Scale types: 5 point scale, 3 point scale
- Classifier types: Support Vector Machines (SVM), K Nearest Neighbours (KNN)

The range of data types are simply the two different styles that were collected with the app as well as a third style that is created by combining the two (taking ratings where possible and filling in the gaps with reviews). The scale types refer to the scale that is used for labelling. The 5 point scale is the standard scale that emotion data was collected upon (with 5 points each corresponding to a different face which the user can select in the app as can be seen in section 4.1.4.). The 3 point scale is a simplified version of this, grouping emotions as simply negative, neutral or positive. This is done by grouping labels of 1 or 2 out of 5 together to represent negative, and 4 or 5 out of 5 together to represent positive. Finally the two classifier types used are the SVM and KNN classifiers provided by the scikit learn libraries.

As mentioned previously combinations of these different types are used to generate the variety of classifier scenarios which are tested. All possible combinations of the 3 type groups above (data, scale, classifier) are created and as such this leads to 12 (3x2x2) different classifier scenarios.

4.3.3. Training and Validation Set Separation

In order to select appropriate hyper parameters and train each classifier produced we separate our data into a training set and a held out validation set. The held out validation set is only used to obtain final performance measures after the hyper parameters have been optimised and the classifier trained. This ensures that there is no overlap between training and test data. This implementation used 3 quarters of the data for training and 1 quarter for validation, with the division made by a random selection of indices which is different each time a classifier scenario is evaluated. This can be seen in the code snippet below:

```
#Create a training set and validation set
indices = np.random.permutation(len(currentData))
numValSamples = math.ceil(len(currentData)/4)

data_train = currentData[indices[:-numValSamples]]
target_train = currentTarget[indices[:-numValSamples]]
data_val = currentData[indices[-numValSamples:]]
target_val = currentTarget[indices[-numValSamples:]]
```

Once these two sets are generated their label distributions are then calculated and shown i.e. for each label value the percentage of the total labels with that value is calculated. The value which is most frequent (has the highest percentage of occurences) in the training set is then identified and from this a “trivial classifier” which always predicts that value is evaluated on the validation set.

4.3.4. Machine Learning Algorithms and Hyper-parameters

As previously mentioned the two machine learning algorithms that are used in this study are SVMs and KNN. Both algorithms were chosen based on their popularity in previous studies on emotion recognition and reported success relative to other algorithms in some of these studies. Support vector machines work by separating data in multiple dimensions using a set of hyperplanes. This can still be used to achieve essentially non linear separation of data by using what is referred to as the “kernel trick”. To achieve this the inputs are mapped into even higher dimensional vector spaces. One example of such a kernel is the radial basis functions, and this is the kernel that is used in the support vector classifier in this study. K Nearest Neighbours is a simpler algorithm where a model is trained by essentially creating a database of all the training examples it receives. When a new sample is given, for which a prediction is to be made, the algorithm simply calculates which are the K nearest samples in its training database (nearest can be defined in a range of ways) and the associated true values for each of these K “neighbours” are counted and the dominant class is returned as the prediction.

Both of these classifier algorithms also require hyper parameters to be provided. For the support vector classifier, using a radial basis function (RBF) kernel, these parameters are:

- C – A prediction penalty weighting which essentially allows a trade off to be made between training set classification accuracy and simplicity of the separating surface.
- Gamma – Determines the distance of influence of individual samples. Low gamma means that samples had a wide ranging effect, where as high gamma means their region of influence is only small.

- Balanced – Decides whether or not the accuracy achieved over classes will be weighted naturally or “balanced” to account for datasets where the number of training samples in each class is not roughly equal.
- Linear – Decides whether the SVM will use a linear kernel (in which case the Gamma parameter is not applicable) or an RBF kernel.

On the other hand the KNN classifier has only one hyper parameter that needs to be trained:

- K – The number of nearest neighbours to be considered when determining a new sample’s value

In order to determine the values of these hyper parameters a grid search is carried out. To do this a set of possible values to be considered for each hyper parameter are provided e.g. C = [0.1, 1, 10, 100], and every possible combination of these hyper parameters is used to create a classifier which is evaluated using cross validation on only the training set. Then whichever set of parameters achieves the highest cross validation accuracy is chosen to train the final classifier which is then evaluation using the validation set.

4.3.5. Feature Selection and Transformation

Feature selection can optionally be carried out using sequential forward floating search. The library which provides this is known as mlxtend. Sequential forward floating search essentially selects features by beginning with the empty set and iteratively adding features whichever un-included feature increases the accuracy of the classifier the most. At each iteration, however, it is also checked that no currently included feature can be removed to increase the accuracy. This can also be applied during the grid search however it is prohibitively slow so instead in this study we opted to tune hyper parameters without feature selection and then use it at the end when evaluating performance. Principle Component Analysis was also used in a similar way.

4.3.6. Performance Metrics

A range of performance metrics are calculated for each classifier scenario. For a given scenario 4 different classifier variations are reported:

- Plain validation set
- Validation set with PCA
- Validation set with SFFS
- Trivial classifier performance

For each of these variations we also report using 3 different scoring methods:

- Accuracy

- Mean absolute error
- Mean squared error

Then finally for each of these 12 (4x3) combinations the hyper parameter tuning is carried out using these 3 different scoring methods.

This results in 36 (4x3x3) different performance metrics being reported for each of the 12 classifier scenarios. These are saved in 2 separate files. One which is human readable, and one which is a CSV. In the next section we will present these results.

5. Results

Classifier performance metrics for each of the 12 classifier scenarios were output by the final classification module in a CSV format. This CSV was opened in Microsoft Excel for analysis and presentation in a useful format.

As explained in section 4.3.6. there are 4 performance metrics which we print. Three of these metrics are the performance of the trained classifier on the validation set (with no feature changes, PCA, and SFFS) and the fourth is the performance of the trivial classifier on that dataset. These metrics were calculated for each of the 12 scenarios.

We investigate 3 main results in this section. The first is the number of classifier scenarios in which the trained classifiers were able to beat the trivial classifier. The second is the mean margin by which the trained classifiers were able to beat the trivial classifier. The third is the maximum margin by which the trained classifiers were able to beat the trivial classifier. These averages help to give us a good idea of how the system has performed overall in the real world unconstrained setting rather than looking at simply its best case performance under a restricted scenario. These 3 results are presented in tables in the subsequent sections. It is also worth noting that because each scenario was evaluated with no feature changes, PCA, and SFFS we present all these results, and further more for each of these we present the results when hyper parameter tuning was carried out with accuracy (ACC), mean squared error (MSE) and mean absolute error (MAE).

Once we have presented these three sets of results, a quick inspection reveals that the strongest results are found when feature selection is used, and that accuracy is the scoring method which is best for training. Using this information we simplify our subsequent investigations where we explore which classifier scenarios were most well suited to the use of physiological signals for emotion recognition in a real world unconstrained context. Here we again calculate the number of scenarios (in each category) which beat the trivial classifier and the mean of the associated margins (note that we leave out maximum margin as it doesn't prove to be very useful in showing patterns). The scenarios are grouped, and thus compared, in 3 different ways – by annotation type, by scale type, and by machine learning algorithm.

5.1. Accuracy Scoring Results

The results in the following table 4 show those calculated based on accuracy as the validation scoring method. As can be seen feature selection, with accuracy based hyper parameter tuning, is able to achieve the greatest number of classifier beating the trivial with 10/12. However feature selection, with MAE based hyper parameter tuning is able to achieve the best mean margin of increase over the trivial classifier, as well as the largest maximum increase.

Training Method	ACC	ACC	ACC	MAE	MAE	MAE	MSE	MSE	MSE
Validation Method	SEL ACC	PCA ACC	VAL ACC	SEL ACC	PCA ACC	VAL ACC	SEL ACC	PCA ACC	VAL ACC
Margin Mean	0.0171	-0.0318	0.0004	0.0194	-0.0407	0.0002	-0.0176	-0.0362	-0.0108
Margin Max	0.0822	0.1000	0.0822	0.1233	0.0274	0.0822	0.1000	0.0274	0.0822
Num Beat	10.000	5.0000	7.0000	8.0000	5.0000	7.0000	5.0000	5.0000	6.0000

Table 4 - Accuracy Scoring Results

5.2. Mean Squared Error Scoring Results

The results in the following table show those calculated based on mean squared error as the validation scoring method. As can be seen feature selection, with either accuracy or MAE based hyper parameter tuning, is able to achieve the greatest number of classifier beating the trivial with 10/12. However feature selection, with MAE based hyper parameter tuning is able to achieve the best mean margin of increase (note that since we are not looking at error, the most negative value is desired) over the trivial classifier, while largest maximum is achieved with MSE tuning.

Training Method	ACC	ACC	ACC	MAE	MAE	MAE	MSE	MSE	MSE
Val Method	SEL	PCA	VAL	SEL	PCA	VAL	SEL	PCA	VAL
Margin Mean	-0.1042	-0.0496	-0.0289	-0.1423	-0.0362	-0.0242	-0.0878	-0.0578	-0.0291
Margin Max	-0.2329	-0.3562	-0.4110	-0.4250	-0.3562	-0.4110	-0.4750	-0.3562	-0.4110
Num Beat	10.0000	6.0000	6.0000	10.0000	6.0000	6.0000	8.0000	7.0000	5.0000

Table 5 - Mean Squared Error Scoring Results

5.3. Mean Absolute Error Scoring Results

The results in the following table show those calculated based on accuracy as the validation scoring method. As can be seen feature selection, with accuracy based hyper parameter tuning, is able to achieve the greatest number of classifier beating the trivial with 11/12. However feature selection, with MAE based hyper parameter tuning is able to achieve the best mean margin of increase over the trivial classifier.

Training Method	ACC	ACC	ACC	MAE	MAE	MAE	MSE	MSE	MSE
Validation Method	SEL	PCA	VAL	SEL	PCA	VAL	SEL	PCA	VAL
Margin Mean	-0.0461	0.0047	-0.0099	-0.0604	0.0151	-0.0082	-0.0176	0.0049	-0.0025
Margin Max	-0.1096	-0.1500	-0.1918	-0.1918	-0.1370	-0.1918	-0.2250	-0.1370	-0.1918
Num Beat	11.0000	5.0000	5.0000	9.0000	6.0000	5.0000	7.0000	8.0000	4.0000

Table 6 - Mean Absolute Error Scoring Results

5.4. Annotation Methods

The first way in which we compare scenario performance is by annotation methods. Here we look at how many classifier scenarios were able to beat the associated trivial classifier for each method of ratings, reviews, or a combination of both. There were 4 scenarios in each of these categories and the number of these which beat the trivial classifier for each category is shown in the graph of figure 8. This information is also displayed in table 7 below along with the average margin by which each category of scenarios was able to beat the trivial classifier.

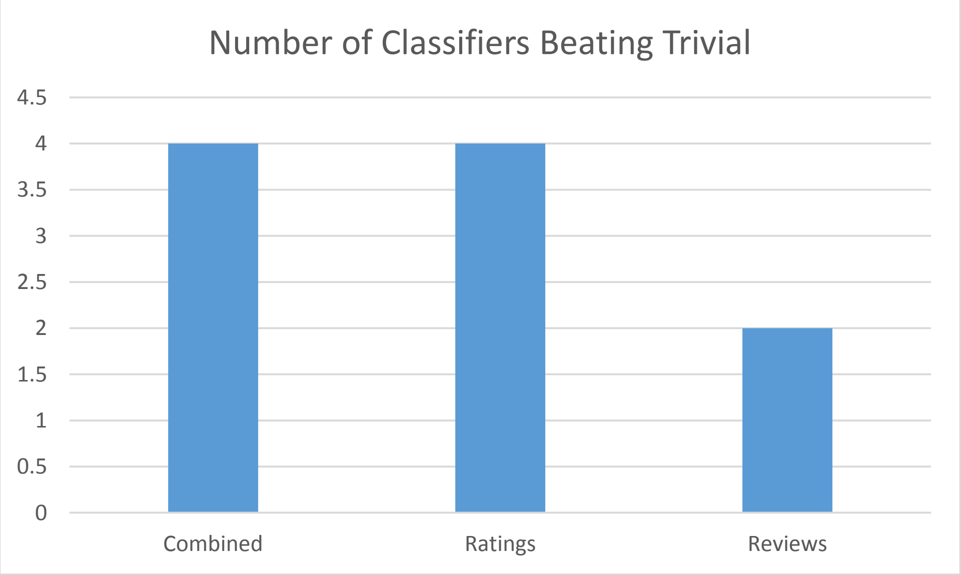


Figure 8 – Graph of Annotation Method Performance

	Combined	Ratings	Reviews
Num Beat	4	4	2
Margin Mean	0.047945	0.0125	-0.00909

Table 7 – Annotation Method Performance

5.5. Scale Types

The second way in which we compare scenario performance is by scale types. Here we look at how many classifier scenarios were able to beat the associated trivial classifier for both a 5 point scale and a 3 point scale for emotion labelling. There were 6 scenarios in each of these categories and the number of these which beat the trivial classifier for each category along with the average margin by which each category of scenarios was able to beat the trivial classifier is show in table 8 below.

	5 Point	3 Point
Num Beat	5	5
Margin Mean	0.009174	0.025062

Table 8 – Scale Type Performance

5.6. Machine Learning Algorithms

The third and final way in which we compare scenario performance is by machine learning algorithms used. Here we look at how many classifier scenarios were able to beat the associated trivial classifier for both a KNN and an SVM classifier. There were 6 scenarios in each of these categories and the number of these which beat the trivial classifier for each category along with the average margin by which each category of scenarios was able to beat the trivial classifier is shown in table 9 below.

	KNN	SVM
Num Beat	5	5
Margin Mean	0.01936	0.014877

Table 9 – Machine Learning Algorithm Performance

6. Discussion

In this section we explore the implications of the results presented in the previous section. In order to do this we will frequently refer back to the aims presented at the beginning of this paper and explore how the system has met these aims. The first results to be discussed will be the overall classification results presented in sections 5.1-5.3 which allow us to make inferences about which settings are most appropriate for emotion recognition classification in this context. Following this we explore the effect that annotation methods have on data collected for this type of problem, and similarly how the type of scale impacts the performance of classification. Finally we discuss the effects of the two different machine learning algorithms used. After discussing all these results we make a series of comparisons between our findings and the relevant literature. We then evaluate the importance of the findings, before noting some of the limitations of the research and allowing these to motivate potential future improvements.

6.1. Overall Classification Results

The results discussed in this section relate to the 3 tables found in sections 5.1 to 5.3 inclusive. These 3 sections explores the effect of using accuracy, mean squared error and mean absolute error as the scoring method for validation. The first thing to note is that the results, indicating which classifier settings are optimal, are very similar for each of these 3 methods and as such we can simplify our discussion of these results somewhat.

In terms of the number of classifier scenarios which were able to achieve better predication performance than the trivial classifier it is clear that the best option was feature selection when trained using accuracy as the scoring method for hyper parameter tuning. This was able to beat the trivial classifier in 10/12 scenarios in terms of accuracy and MSE, and 11/12 scenarios for MAE.

In terms of mean margin (i.e. on average by how much the trained classifiers beat their associated trivial classifiers) feature selection is still the best, however, it appears that using MAE for the hyper parameter tuning stage results in the best performance. These measures help us to achieve our aim of determining how effective we can make a classifier to work in the real world unconstrained context with mean improvements over trivia classifiers being 0.0194 for accuracy, 0.1423 for MSE, and 0.0604 for MAE.

While the maximum margin achieved for each validation type is not very informative about general trends it can help us achieve our aim of investigating the maximum effectiveness that can be achieved. Again this was achieved using feature selection and the maximum increase in accuracy over the trivial classifier was 12.33%.

From this collection of observations we can see that feature selection is consistently achieving the best results and thus is clearly a useful option to adopt when trying to predict emotions in a real world unconstrained context.

In terms of choosing the scoring method to use for hyper parameter tuning it seems that MAE achieves the most accurate classifiers as shown by its high mean margin, but in general it is perhaps not as consistently good across a range of classes as accuracy which results in a higher number of classifiers beating the trivial counterpart.

It can also be seen that for hyper parameter training using MAE or accuracy as the scoring method that PCA consistently gives poorer results than if it were not included at all, and for MSE it only provides a very marginal improvement. Thus we do not have any strong evidence that PCA is a useful technique for this problem.

6.2. Annotation Method Discussion

One of our main aims was to investigate what effects different types of ground truth collection methods would have on the effectiveness of classifiers in our chosen context. In light of the results explained in the previous section we simplify our investigation of our subsequent results by considering only the results for the most successful setting of using feature selection. When we do this we get the results displayed in section 5.4. From these results it is quite clear that in terms of number of classifiers that beat the trivial, for each type of annotation, the ratings method and combined method are more effective (both with 4/4) than the review method. When we look at the mean margin however we can see that the combined annotation method is even better than ratings alone with a 4.8% margin compared to only 1.25% for ratings alone. This tells us that in this context ratings are the best option if only one method can be used or if we want to minimise the amount of annotations a user needs to make, but that if it is possible then a combined method is desirable.

6.3. Scale Type Discussion

Both scales are showing themselves to be effective at improving classification accuracy over a trivial classifier in this context and so we can see that despite the fact that we have decreased the minimum time scale on which we label (i.e. the time gap between subsequent labels) we are still able to achieve meaningful prediction, as was one of our aims. The three point scale appears to be even more effective than the 5 point in terms of margin but we should also recognise that as there are a lower number of classes to choose from in this case an absolute increase in accuracy over the trivial classifier might be easier to achieve.

6.4. Machine Learning Choice

Both the KNN and SVM algorithms are effective according to the number of classifiers that beat the trivial and the corresponding mean margins. KNN has a slightly higher margin mean but the difference is not significant enough to provide grounds for saying that one classifier is more appropriate than the other for our problem.

6.5. Comparison with Previous Works

Many previous studies have shown that Sequential Floating Forward Search as a feature selection method was able to improve classifier performance [5, 13] as these results have also shown, and as such this is not a surprising observation.

The result that was found with PCA providing no significant benefit is also similar to what is described in certain literature [13]. The reason given for this is that the PCA approach does not take into account any class information when transforming the features.

The majority of other studies only report their results in terms of accuracy and measures such as MAE and MSE are not used extensively. However, those studies which have explored the use of such measures for classifying ordinal scales have found it to be a useful scale [22, 23] so it is unsurprising that it frequently reinforced the trends we saw in accuracy results but also that in the case of hyper parameter training was able to produce a classifier with the highest mean margin.

No previous work, that this study is aware of, has ever compared annotation methods in an experimental fashion, however, the result that was achieved is what was expected. Firstly the fact that a combined method is more successful than its two constituent methods individually, makes sense in that a larger amount of useful information is being provided to the classifier. It also makes sense that ratings submitted at the time an emotion is being felt might produce more predictable results than reviews which are submitted at the end of the day when the user has potentially forgotten exactly how they were feeling.

The scales used in other studies are most frequently a 2 or 3 point scale so it is not all that surprising that the 3 point scale was most effective, also due to the fact that simply by pure chance the classifier is more likely to predict correctly. However, as 5 point scales are not themselves uncommon it is also unsurprising that they perform almost as well.

As was previously discussed, the result of both KNN and SVM providing similar levels of effectiveness is not unsurprising either as both classifiers were chosen due to their popularity in this domain.

6.6. Limitations and Future Work

Due to various factors such as resources, available time and level of knowledge of the researcher there were limitations to what was possible in this study. Most of these limitations are issues that could be overcome given more time and resources or slightly modified approaches and as such suggest potential areas for future work.

One of the most important limitations was that the system was only able to be tested on one participant, and what's more that participant was the researcher. Only one physiological data collection device was available and so data could not be collected from users in parallel. What's more even if there had been more devices much more work would have been required to recruit participants ethically, ensure that other ethical concerns were appropriately considered, add privacy preserving aspects to the data, and provide tech support for issues with different devices using the android app. As such using multiple participants was not feasible. Clearly this is an issue, however, because emotion recognition systems are highly user dependent and results could differ wildly between people. This is particularly the case for physiologically based systems as different users have different physiological responses to emotion. It is also not ideal for the researcher to be used as a participant in the study as they are aware of how data is being collected, and how it will be analysed, and it is possible that even subconsciously this could affect the way they report data.

Another limitation of this system is that it has only made use of a 1 dimensional model, that is essentially the valence axis from the circumplex model (even though this is never explicitly stated to the user, and they simply report how they are “feeling” by choosing the most relevant face). Most other studies also collect arousal data. This allows emotions to be separated more easily and often also provides more insight into the subtleties of the user's emotional state.

The set of physiological signals which were collected and analysed were also limited by those which were able to be collected by the Empatica E4 wristband. While this set does cover the most frequently used and proven successful signals there could be added benefit in the use of further signals.

The final limitation that we will discuss (and of course there are many more) is the fact that the periods which are used for prediction in this system never overlap. It was not investigated whether it could be useful to use data over longer periods of time before a label is given in the calculation of features. Some other studies do this and it could prove to be beneficial. Similarly if this system were to be adapted to be able to make predictions real time then it might be useful to look at predictions from recent points in history. This adaptation to make this system work in real time is yet another improvement that could be explored in future works.

7. Conclusion

Our primary aim of investigating the settings that lead to the most effective emotion classifier in a real world unconstrained context led us to find that this was achieved with a classifier using feature selection, a combined rating system and a 3 point scale. Our aim of decreasing the time scale while still making meaningful predictions was also achieved with times between labels of only 1 hour being used for the combined method of annotation. Similarly we were able to show that the combined method of annotation was the most effective in this real world unconstrained context as well.

Significant contributions that can be taken away from this study include the experimental comparison of labelling methods which has never been done before. The demonstration of the possibility of using shorter time periods was also shown. In the process a useful annotation collection tool for real world contexts, in the form of an android app, was also created and could be used in future work.

As discussed in the previous section there are many possibilities for future work including the use of more participants in a similar system to compare how the system performs on multiple users, the collection of more sophisticated labels such as ones which include arousal information, or the general creation of a wider variety of features perhaps from new signals.

8. Bibliography

1. Constantine, L. and H. Hajj. *A survey of ground-truth in emotion data annotation*. in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*. 2012. IEEE.
2. LiKamWa, R., et al. *Moodscope: Building a mood sensor from smartphone usage patterns*. in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. 2013. ACM.
3. Rachuri, K.K., et al. *EmotionSense: a mobile phones based adaptive platform for experimental social psychology research*. in *Proceedings of the 12th ACM international conference on Ubiquitous computing*. 2010. ACM.
4. Jaques, N., et al. *Predicting students' happiness from physiology, phone, mobility, and behavioral data*. in *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*. 2015. IEEE.
5. Sano, A., et al. *Recognizing academic performance, sleep quality, stress level, and mental health using personality traits, wearable sensors and mobile phones*. in *Wearable and Implantable Body Sensor Networks (BSN), 2015 IEEE 12th International Conference on*. 2015. IEEE.
6. Sano, A. and R.W. Picard. *Stress recognition using wearable sensors and mobile phones*. in *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*. 2013. IEEE.
7. Bianchi-Berthouze, N. and A. Kleinsmith, *11 Automatic Recognition of Affective Body Expressions.* The Oxford Handbook of Affective Computing, 2014: p. 151.
8. Picard, R.W., E. Vyzas, and J. Healey, *Toward machine emotional intelligence: Analysis of affective physiological state.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2001. **23**(10): p. 1175-1191.
9. Jerritta, S., et al. *Physiological signals based human emotion recognition: a review*. in *Signal Processing and its Applications (CSPA), 2011 IEEE 7th International Colloquium on*. 2011. IEEE.
10. Lang, P.J., M.M. Bradley, and B.N. Cuthbert, *International affective picture system (IAPS): Technical manual and affective ratings.* NIMH Center for the Study of Emotion and Attention, 1997: p. 39-58.
11. Lisetti, C.L. and F. Nasoz, *Using noninvasive wearable computers to recognize human emotions from physiological signals.* EURASIP Journal on Advances in Signal Processing, 2004. **2004**(11): p. 1-16.
12. Carvalho, S., et al., *The emotional movie database (EMDB): a self-report and psychophysiological study.* Applied psychophysiology and biofeedback, 2012. **37**(4): p. 279-294.
13. Wagner, J., J. Kim, and E. André. *From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification*. in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*. 2005. IEEE.
14. Charland, L.C., *Emotion experience and the indeterminacy of valence.* Emotion and consciousness, 2005: p. 231-54.
15. Bamidis, P.D., et al., *An integrated approach to emotion recognition for advanced emotional intelligence*, in *Human-Computer Interaction. Ambient, Ubiquitous and Intelligent Interaction*. 2009, Springer. p. 565-574.
16. Ekman, P., R.W. Levenson, and W.V. Friesen, *Autonomic nervous system activity distinguishes among emotions.* Science, 1983. **221**(4616): p. 1208-1210.
17. Lee, C.K., et al. *Using neural network to recognize human emotions from heart rate variability and skin resistance*. in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*. 2006. IEEE.

18. Garbarino, M., et al. *Empatica E3—A wearable wireless multi-sensor device for real-time computerized biofeedback and data acquisition*. in *Wireless Mobile Communication and Healthcare (Mobihealth), 2014 EAI 4th International Conference on*. 2014. IEEE.

19. Haag, A., et al. *Emotion recognition using bio-sensors: First steps towards an automatic system*. in *Tutorial and research workshop on affective dialogue systems*. 2004. Springer.

20. Robert, C., *Machine Learning, a Probabilistic Perspective.* CHANCE, 2014. **27**(2): p. 62-63.

21. Yang, N. and A. Samuel, *Context-rich Detection of User's Emotions using A Smartphone.*

22. Gaudette, L. and N. Japkowicz. *Evaluation methods for ordinal classification*. in *Canadian Conference on Artificial Intelligence*. 2009. Springer.

23. Baccianella, S., A. Esuli, and F. Sebastiani. *Evaluation measures for ordinal regression*. in *2009 Ninth international conference on intelligent systems design and applications*. 2009. IEEE.