2012

CLASSIFYING EMOTION USING STREAMING OF PHYSIOLOGICAL CORRELATES OF EMOTION

Nathan J. Elmore
The University of Montana

CLASSIFYING EMOTION USING STREAMING

OF PHYSIOLOGICAL CORRELATES OF EMOTION

By

NATHAN JAMES ELMORE

Bachelor of Science, Rocky Mountain College, MT, 2010

Thesis

Presented in partial fulfillment of the requirements
for the degree of

Master of Science
in Computer Science

The University of Montana
Missoula, MT

December 2012

Approved by:

Sandy Ross, Associate Dean of The Graduate School
Graduate School

Dr. Douglas Raiford, Chair
Computer Science

Dr. Min Chen
Computer Science

Dr. Allen Szalda-Petree
Psychology

Contents

List of Tables

List of Figures

List of Equations

Elmore, Nathan, M.A., Summer 2012 *Computer Science*

Committee Chairperson: Dr. Raiford

1 Abstract

The ability for a computer to recognize emotions would have many uses. In the field of human-computer interaction, it would be useful if computers could sense if a user is frustrated and offer help (Lisetti & Nasoz, 2002), or it could be used in cars to predict stress or road rage (Nasoz, Lisetti, & Vasilakos, 2010). Also, it has uses in the medical field with emotional therapy or monitoring patients (Rebenitsch, Owen, Brohil, Biocca, & Ferydiansyah, 2010). Emotion recognition is a complex subject that combines psychology and computer science, but it is not a new problem. When the question was first posed, researchers examined at physiological signals that could help differentiate an emotion (Schachter & Singer, 1962). As the research progressed, researchers examined ways in which computers could recognize emotions, many of which were successful. Previous research has not yet looked at the emotional data as streaming data, or attempted to classify emotion in real time. This thesis extracts features from a window of simulated streaming data to attempt to classify emotions in real time. As a corollary, this method can also be used to attempt to identify the earliest point an emotion can be predicted. The results show that emotions can be classified in real time, and applying a window and feature extraction leads to better classification success. It shows that this method may be used to determine if an emotion could be predicted before it is cognitively experienced, but it could not predict the emotion transitional state. More research is required before that goal can be achieved.

2 Introduction

Emotion recognition is a popular field in both psychology and computer science. There are numerous applications and benefits that emotion recognition could provide, and it has recently become a popular subject in both Human-Computer Interaction (HCI) and behavioral sciences. There are also many applications that could benefit from reliable emotion classification. Emotion recognition could be applied to create more intelligent computer interfaces, educational systems, "driver and pilot's safety applications" (Lisetti & Nasoz, 2004), and telemedicine applications, such as monitoring dementia patients for episodes (Rebenitsch, Owen, Brohil, Biocca, & Ferydiansyah, 2010).

At the most basic level, emotions reflect how an individual "feels" inside. They have a powerful influence on actions and a key role in social life. Individuals often attribute their actions to emotion. They react in fear, or they are irrational because they are angry. Scientists recognize emotions as complex psychological processes that affect one's mind and body. Their purpose is likely to influence behavior. For example, if a negative event occurs to an individual who then experiences sadness, fear, etc., the individual is likely to adapt their behavior to handle or prevent a similar event. Emotions are also linked to communication and social behaviors. Some research suggests that when social connections are formed with another we not only experience their emotional states but their physiological states.

One of the first major advances in emotion quantification was the James-Lange theory in the late 19th century (James, 1884). They suggested that emotions are a perception of our body changes. Meaning, "we feel sad because we cry", and not "we cry because we feel sad" (James, 1884). This was challenged in 1927 by the Cannon-Bard Theory, which suggests that individuals experience both emotion and physiological signals simultaneously (Cannon, 1927). The Schachter-Singer theory of emotion (1962) suggests that emotion is both cognitive and physiological. Schachter and Singer came to this conclusion with an experiment in which they injected subjects with epinephrine to increase

physiological arousal. An actor performed certain actions that would either show anger or euphoria. They observed that subjects who were either misinformed or not informed about the side effects of the injection would have feelings of euphoria or anger. Conversely, the subjects who were correctly informed about the side effects would not have these feelings. They conclude, “people search the immediate environment for emotionally relevant cues to label and interpret unexplained physiological arousal” (Schachter & Singer, 1962).

Today, emotions are thought of as comprising three parts. The first are physiological changes: heart rate, temperature, breathing rate, etc. Second are behavior changes. Lastly, it is a conscience experience (Myers, 2004). This might suggest that if one measures the correct physiological responses, it might be possible to determine the emotion experienced by an individual. It may also suggest one could leverage emotion experience below cognitive awareness. Cognitive awareness of an emotion means the individual both cognitively recognize that they are experiencing an emotion, and identifies which emotion they are experiencing.

There are many studies that examine physiological correlates of emotion, and it is generally accepted that certain physiological signals suggest a specific emotion. Paul Ekman, Paul Levenson, and Wallace Friesen conducted a study on the autonomic nervous system response to emotions in 1983. Their data shows emotion-specific autonomic activity, particularly with heart rate, temperature, and galvanic skin response. A few years later, Lanzetta et al. (1986) found a correlation between fear and galvanic skin response. Galvanic skin response is the measure of electric conductance of the skin. Vrana et al (1993) studied correlations between heart rate and fear. Sinha et al (1996) showed that fear and anger are “accurately differentiable” (Sinha & Parsons, 1996). Lisetti et al (2004) showed that “emotions can be recognized from physiological signals via noninvasive wireless wearable computers” (Lisetti & Nasoz, 2004) using the minimum, maximum, mean and variance of three physiological correlates.

While many of the physiological correlates of emotion are recognizable, physiological responses currently provide very general information about emotions being experienced. It is reasonable that the integration of different complex physiological

streams would provide increased information about our biological response to emotions. These in turn can be leveraged to help identify emotions by physiological response. This thesis explores physiological data streams to attempt to classify emotions using a Random Forest classifier. It attempts to identify the earliest point an emotion can be classified and determine whether that point comes before cognitive awareness of the emotion.

3 Preliminaries

3.1 Contributions

3.1.1 Emotion

There are a few ways to model emotion. The two most common are a discrete, categorical model, and a multidimensional model. The first suggests a set of universal basic emotions (Ekman, 1992). The multidimensional model uses multiple dimensions (usually two), which "enable the description of different emotions and the distinction between them" (Oehme, Herbon, Kupschick, & Zentsch). The two dimensions most commonly used are valence and arousal. Valence is a measurement of positivity and negativity. Happy, and relaxed emotions are in the positive quadrants while sadness and anger in the negative quadrants. Arousal is a measure of the reactivity to stimuli. These two dimensions form a graph (Figure 1) that determines emotion based on the levels of arousal and valence. These two dimensions create four quadrants: High Arousal High Valence (HAHV), High Arousal Low Valence (HALV), Low Arousal Low Valence (LALV), and Low Arousal High Valence LAHV.

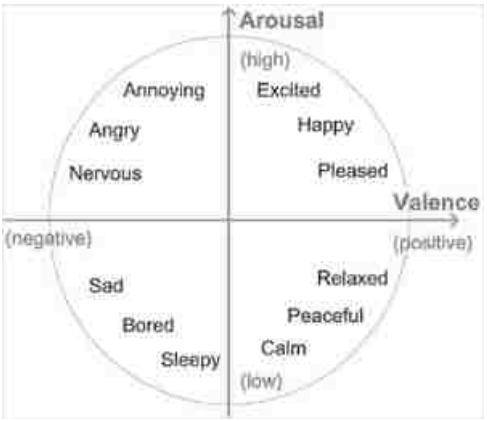


Figure 1: Valence-Arousal Model
Image adapted from **(Oehme, Herbon, Kupschick, & Zentsch)**

The dimensional model is commonly used for classifying and testing emotion. There have been studies on physiological processes that correlate to valence or arousal (Russel, 1980). Such processes are physiological correlates of emotion. "The most commonly used physiological parameters applied in studies based on the dimensional model are skin conductance level (SCL), facial electromyogram (EMG) and heart rate (HR)" (Oehme, Herbon, Kupschick, & Zentsch). Many others have been used. Oehme *et al* used skin conductance level, heart rate, breathing rate, and skin temperature, and show a correlations between breathing rate (BR) and arousal, SCL and fear-situations, and HR correlates with valence.

There have been many other studies in identifying emotions from physiological correlates. Most of them suggest that emotions can be classified using physiological correlates. Picard et al used EMG, blood volume pressure (BVP), skin conductance and respiration to try and classify eight different emotions (Picard, Vyzas, & Healey, 2001). This thesis used the dataset collected by Picard et al to explore feature selection, classification methods, and to construct a proof of concept. Picard et al show the difficulty of acquiring realistic emotional data. They identify five factors that influence the data collection:

1. Is the emotion subject or event elicited?
2. Is it in a lab setting or a real world setting?
3. Is the emotion expressed and felt internally? Or just expressed externally?
4. Is it an open recording or a hidden-recording?
5. Is it emotion-purpose or other-purpose? “Does the subject know that the experiment is about emotion?”

They used Sequential Floating Forward Search (SFFS), Fisher projection (FP) and a hybrid of the two, SFFS-FP, to classify the eight emotions. They found that they could predict emotions better with fewer emotional classes. This appears to be because some emotions are very similar in terms of the arousal and valence model. For example, classifying anger and grief is difficult because they both have high levels of arousal and negative valence, whereas distinguishing between anger vs. joy is easier because joy has a positive valence and a medium-high level of arousal. Picard et al are the first to show what they call “day-dependence”. They observed “...emotions from the same day often clustered more closely than did features for the same emotions on different days” (Picard, Vyzas, & Healey, 2001). This could be caused by several factors. The three factors proposed are:

1. Skin-sensor interface influences
    a. Changes in positioning, hand washing, etc.
2. Variations in physiology caused by caffeine, sugar, sleep, or hormones
3. Variations in physiology that are mood dependent
    a. It would be hard to experience joy if subject was in a sad mood that day.

Lisetti et al administered an experiment where they elicited emotions using film clips and measured physiological correlates using BodyMedia SenseWear Armbands (Lisetti & Nasoz, 2004). They measured galvanic skin response (GSR), temperature and heart rate (HR). Then they normalized the data and used the minimum, maximum, mean, and variance to predict emotions with three machine learning algorithms. The three algorithms were k-Nearest Neighbor, Discriminant Function Analysis, and Marquardt Backpropagation. The

most successful of these was the neural network using Marquardt Backpropagation. Their research shows that “emotion can be recognized from physiological signals via noninvasive wireless wearable computers” (Lisetti & Nasoz, Using Noninvasisve Wearable Computers to Recognize Human Emotions from Physiological Signals, 2004).

3.1.2 Data Stream Mining

A data stream is a sequence of data signals that transmit data about some process. There are many types of data streams. Some of the most common types of streaming are network traffic, phone conversations, video streaming and sensor data. This thesis focuses on sensors that measure physiological signals. Streaming data presents its own set of problems and requires unique considerations. The availability of these vast, real time data sets has led to many new studies in computing. It “has gained a high attraction due to the importance of its applications and the increasing generation of streaming information” (Gaber, Zaslavsky, & Krishnaswamy, 2005).

Data stream mining is the process of extracting meaning from a data stream. Data streams generally are high speed, high volume, and can fluctuate in transmit speed. This can prove to be a difficult set of challenges. Modern computers cannot often keep up with the volume (storage) or speed of the incoming data (processors). This implies that the application must have some intelligent way of determining which data to process or store.

Many techniques have already been employed for data steam mining. Two of the major types are Data-Based and Task-Based (Gaber, Zaslavsky, & Krishnaswamy, 2005). Data-based techniques are generally used for summarizing the data as a whole. Task-Based techniques are used to “address computational challenges of data stream processes” (Gaber, Zaslavsky, & Krishnaswamy, 2005). In this project, the problem is not summarizing the data as a whole, but summarizing data in parts, then classifying that summarization. This leads us to the Task-Based techniques. Of the Task-Based techniques that Gaber *et al.* describe, the sliding window appears to be the most useful for our purposes. This is because predicting the current class of emotion is more concerned with the most recent data.

Overall analysis of emotion over longer periods of time could be difficult because a person's physiology is changes day to day (Picard, Vyzas, & Healey, 2001).

For our purposes, the system will have to handle multiple streams simultaneously and store the most recent changes in a window. This presents a few challenges. The first is managing resources because data streams are often high volume, and speed. In addition, the system must have a central location to combine the data. This is because it's "impossible to offload classification decisions to individual data sources, each of which lacks full knowledge for the decision making" (Bai, Wang, & Zaniolo).

4 Proof of Concept

4.1 Data

The dataset used for the proof of concept comes from a study at MIT (Picard, Vyzas, & Healey, 2001). In this study, a single subject was used and data was collected over many days. They used a single subject because it has been argued that emotions can have "different interpretations across individuals within the same culture". This means that classifying emotions between multiple individuals greatly increases the difficulty of classification. To elicit emotions, a graduate student spent six weeks and "tried to experience eight affective states with the aid of computer controlled prompting system". They collected data from facial Electromyography (EMG), breathing rate (BR), galvanic skin response (GSR), and blood volume pressure (BVP). The emotions collected were *no emotion, anger, hate, grief, platonic love, romantic love, joy* and *reverence.* This data was used by Picard et al to classify the eight different emotions over 20 days with notable accuracy.

4.2 Principal Component Analysis

First we analyze the data to see how separable it is. Principal components analysis is a procedure that uses an orthogonal transformation to rotate data into linearly

uncorrelated data (Jolliffe, 1986). The new variables are called principal components. They are sorted in a way that the first component captures the largest variance. PCA can be used to determine the axis greatest of variance and to help visualize data with high dimensionality. Using the statistical package R (R: A language and environment for statistical computing., 2012), PCA was performed on the data set to visualize data correlations of the eight emotion classes.

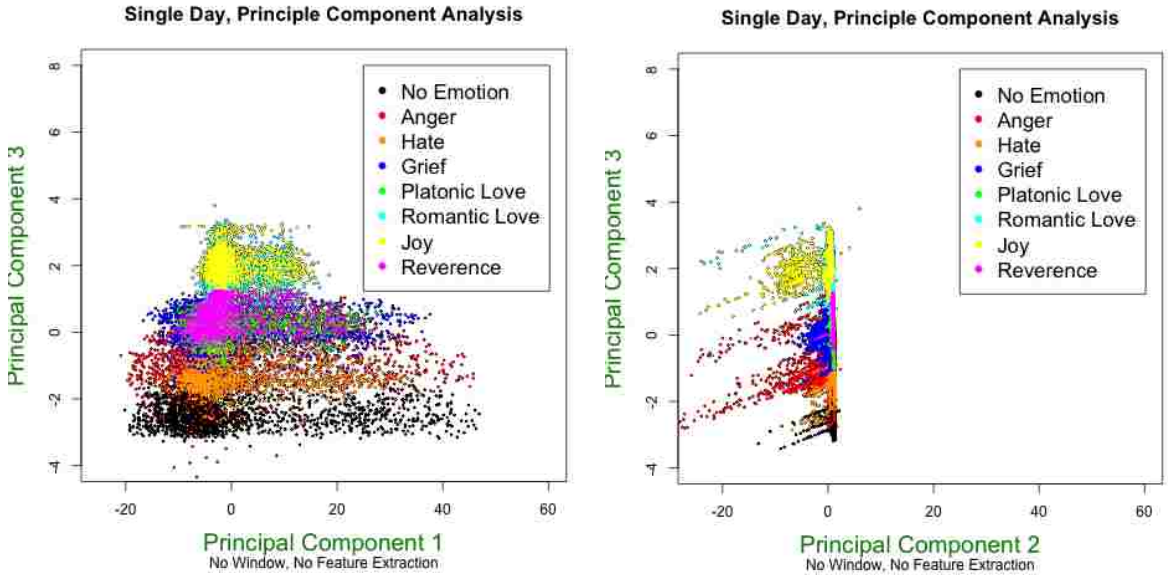


Figure 2: Single Day PCA

PCA performed on a single day emotion datasets show the divisions of the data and the emotions represent. Notice that many of the emotions that are overlapping are close proximity in the valence arousal model.

The graphs above compare the first three components. There are fairly defined clouds and bands when comparing the PC1 with PC3 and PC2 with PC3. They also support other results. The confusion matrix shown in Table 3 shows some support. For example, Joy and Romantic Love are likely to be misclassified with each other. Their data points in the PCA are in very close proximity. The PCA is much less discernable when plotting all days, likely because of the very large amount of data points and the dependence phenomenon (Picard, Vyzas, & Healey, 2001). Adding windowing appears to increase the variability as the bands/clouds of data points are appear more separable for each of the classes. The explanation for this is

that a windowing function reduces the impact of noise from the biofeedback sensors. A window length of twenty-five is shown below.

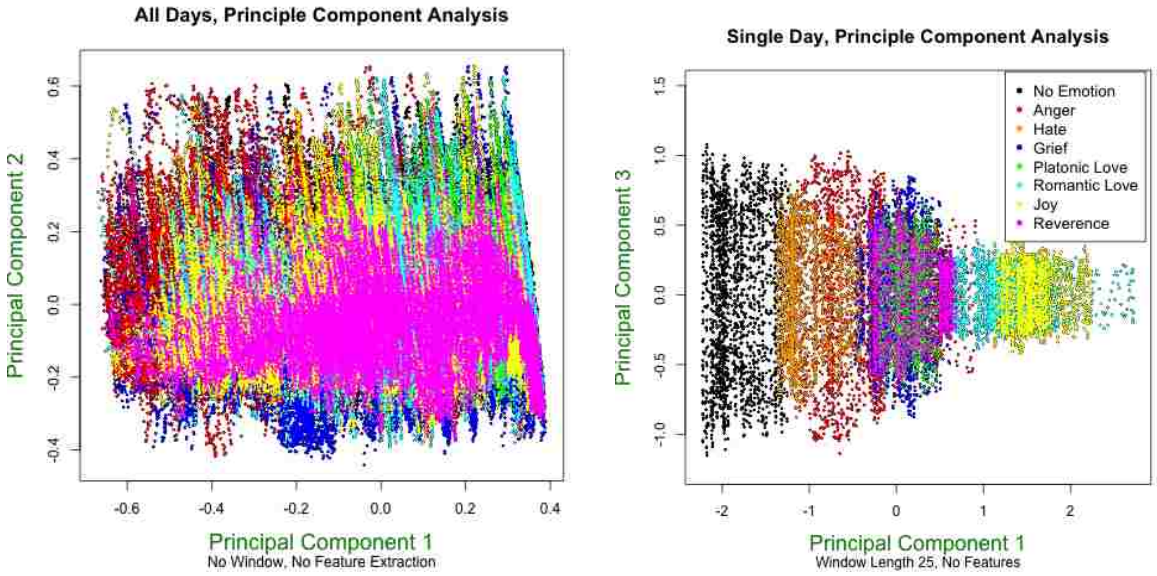


Figure 3: All Days, No Window PCA (Left), Single Day Window Size 25 (Right)

PCA performed on combined, "All Days", dataset and windowed dataset show the divisions of the data and the emotions represent. Notice that many of the emotions that are "stacked" are close proximity in the valence arousal model.

4.3 Preliminary Classification

Before classifying the entire dataset, I used the WEKA data mining software to explore the best performing classification methods for a single day. The table below shows some of the more accurate classifiers for classification of the single day, without windowing emotion data. The classifiers were tested using ten fold cross validation.

Table 1: Single Days Accuracies

Single day, No windowing, Accuracy Rates					
	Decision Table	Bayesian Network	Multilayer Perceptron	Random Forest	J48
Mean Accuracy	80.82	83.75	75.3	91.64	90.31
Max. Accuracy	92.29	93.43	88.23	96.50	95.99
Min. Accuracy	66.32	70.31	58.95	78.99	77.53

Table 1 shows the five classifiers' mean, maximum and minimum percent of correctly classified instances for all twenty raw data sets.

These classification runs are a small subset of the algorithms initially tested for performance. These were chosen on the following criteria:

1. Accuracy: The best performing classifiers are given preference
2. Type and Implementation: A variety of algorithms were necessary to test a broad spectrum of methods.

The first classifier shown is the decision table (Kohavi, 1995). This is a decision table that uses rule mapping to predict the class. It comprises a schema and a body of labeled instances. To classify, it compares the instance with table and returns the closest match. If no matches are found, it returns the majority class of the table is returned. J48 is a type of classification tree. A classification tree is a tree where each branch represents a decision, and terminal nodes are classifications (shown on next page). The J48 is specifically decision tree implemented with the C4.5 algorithm. It determines which features provide the most information, then creates decisions in the form of a tree. The leaf nodes of the tree contain the labels.

In **Figure 4**, a classification tree is represented by the nodes $t_1 \ldots t_5$. The round nodes represent decision nodes while the square nodes represent terminal, or classification leaves. There are two labels represented by x's and o's, and two attributes, x_1, x_2. The decisions were used to split the data space into three sections. Each section represents a space in which a specific label is returned.

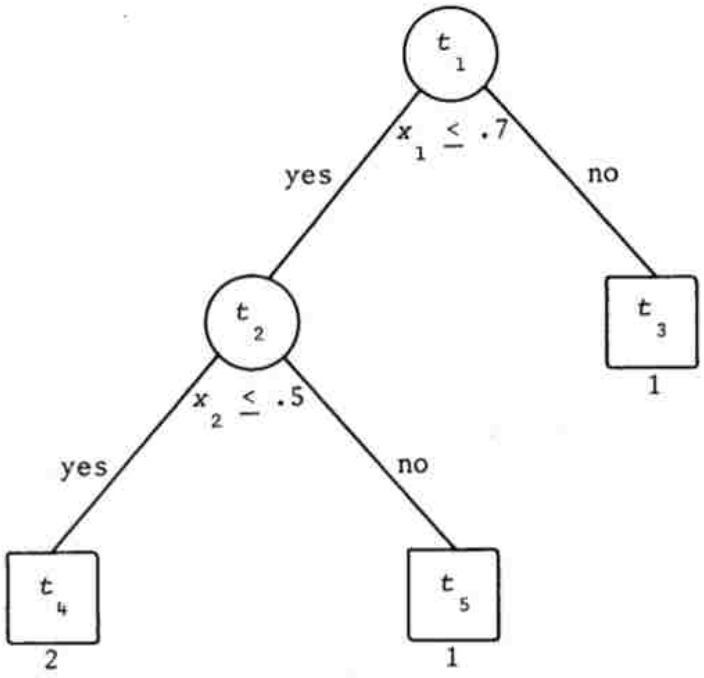
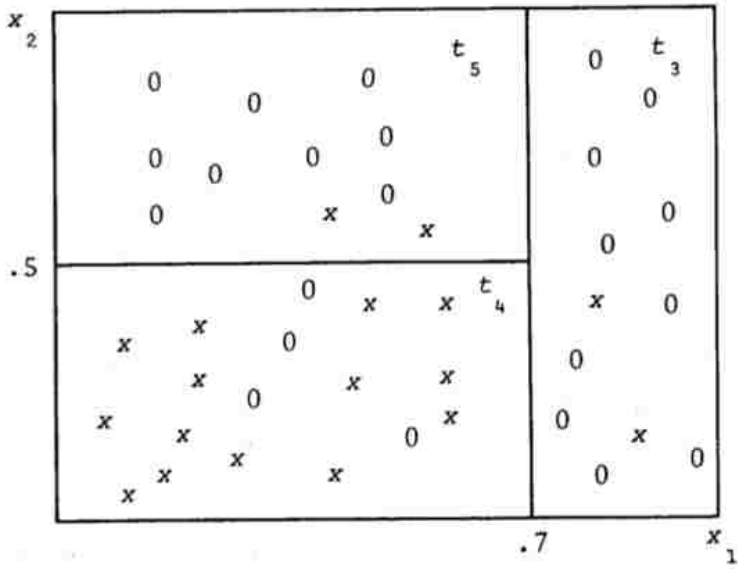


FIGURE 2.8



0 = class 1, x = class 2

Figure 4: Classification Tree. Reproduced from **(Breiman, Friedman, Olshen, & Stone, 1984, p. 31)**

This shows an example classification tree with two classes: $\{x, 0\}$, and two features: $\{x_1, x_2\}$. The circular nodes in the tree represent decisions, while the square boxes represent predictions. This splits the data space into sections that represent a certain class. Each line dividing the data space comes from a decision. An instance is classified based on the section it falls into.

A random forest is a set of classification trees where the trees in the forest contain a random subset of the features. Each tree gets a single vote on the class. Each tree in the forest is queried and the class with the most votes is returned. The preliminary experiments below show that the random forest is the best performing algorithm.

In initial testing, the best performing algorithms were classification trees. Classification trees in machine learning often suffer from a phenomenon called over fitting. It is also difficult to update a decision tree since all trees must be rebuilt from the entire data set to do so. So, I found it prudent to include Bayesian network and the multilayer perceptron (a type of artificial neural network). The advantages of this are that these methods have more resilience to over fitting, and with a neural network, the model could easily be updated with relevance feedback by the user/subject.

This data suggests that classification for a single day is most plausible with various classifiers and is most accurate with a Random Forest. However, this was not surprising based on previous studies. The real test is how the classifier performs over multiple days as Picard et al has already shown that physiological correlates of emotion vary from day to day. Next, all the days were combined into a single dataset and classified using ten fold cross fold validation.

Table 2: Combined Days Accuracies

All days, No windowing, Accuracy Rates					
	Decision Table	Bayesian Network	Multilayer Perceptron	Random Forest	J48
Accuracy	58.52	42.81	31.84	82.7	80.38

Table 2 shows the five classifiers' percent of correctly classified instances using all twenty days combined into one dataset.

The accuracies dropped fairly substantially, but the Random Forest and J48 still perform with reasonably high accuracies around 80% correct.

The sample confusion matrix for a random forest below shows how each of the eight classes is likely to be misclassified for a single day.

Table 3: Single Day Confusion Matrix

Confusion Matrix – Single day, No Windowing

```
   a    b    c    d    e    f    g    h   <-- classified as
1980    0   10    0    0    0    0    0 |    a = No Emotion
   0 1819   15  103   45    2    0    6 |    b = Anger
  13   22 1955    0    0    0    0    0 |    c = Hate
   0  101    0 1822   39    5    0   23 |    d = Grief
   0   42    0   27 1767    0    0  154 |    e = Platonic Love
   0    3    0    3    0 1839  139    6 |    f = Romantic Love
   0    0    0    0    0   98 1892    0 |    g = Joy
   0    6    0   25  305    6    0 1648 |    h = Reverence
```

This is a sample confusion matrix for a single day dataset using the random forest classifier.

According to our preliminary results, anger is most often misclassified as grief and visa versa. Anger and grief are both classified as high arousal, negative valence. This suggests that with these features, emotions close in the valence-arousal model are more likely to be misclassified as emotions in close proximity. As more data and complexity is added to the classifiers, the number of emotions may have to be generalized into the four quadrants of the model.

5 Methods

5.1 Data Examination and Features

5.1.1 Sliding Window

Picard *et al* (2001) extracted features for entire days for classification of other days. In this thesis I use raw data and extract features from a sliding window of varying lengths to explore the best methods and most discriminative features for accurate classifications for all twenty days. The features are analyzed using a variety of methods including Kullback-Leibler divergence (information gain), chi-squared, and principal components analysis (PCA) to help determine the most discriminative features.

Sliding windows of four different lengths are applied to the dataset to create distinct datasets. Imagine that zero represents the current point in time. Sampling points in time are represented by numbers less than zero; numbers in the future are greater than zero. Given a window size of n, the range the window examines is from $-n$ to 0. The dataset of instances, D, is now composed of $n * |F|$ features, where F is the set of original features. Now, the previous n data points of each feature f are contained in each element of D:

Equation 1: Sliding Window

$$f \in F, \{f_{-n+0}, f_{-n+1}, \dots, f_0\}$$
$$d \in D, \{F_0 \cup F_1 \cup \dots F_{|F|}\}$$

Given the current time as zero and a window size of n, the new feature set contains the values of the n previous features of the same feature type. The new instance $d \in D$ is comprised of the union of the new feature sets.

Next, classification techniques and feature analyses are used on the new dataset's raw points. Features are extracted from the windows and then another round of classification and feature analysis is performed.

I applied sliding windows with sizes *5, 10, 25 and 50* to the data sets. The data was collected with a rate of 20Hz. So, a window size of 25 is 1.2 seconds. Using the WEKA API in Java, I created 5 programs to run each of the classifiers. The datasets are very large, ranging from 320,000 data points to just shy of eighty million data points. Classifying a single set takes anywhere from an hour to a few days depending on the classifier and the size of the dataset. A bash script was used to run the five classifiers over the 85 datasets. When classifying raw windowed data, there is a significant increase in accuracy, especially for the Random Forest.

Table 4: Single Days, Windowed - Average Accuracies

Single day, Windowed, Average Accuracy over all 20 Days					
Window Size	Decision Table	Bayesian Network	Multilayer Perceptron*	Random Forest	J48
5	80.78	86.92	82.64	95.42	91.46
10	80.82	88.30	86.38	97.18	92.58
25	81.31	90.43	91.07	99.01	94.06
50	82.40	92.58	N/A	99.68	95.681

Table 4 shows the five classifiers' percent of correctly classified instances using single datasets with various window sizes. The data was collected at 20Hz, so a window size 10 is half a second.
* Multilayer Perceptron is the slowest to train some tests ran for days at a time. A window size of 50 ran for a two weeks before it crashed.

All algorithms except Decision Table see a substantial increase in accuracy, with Random Forest jumping to 99% accuracy. Classifying all twenty days results in a similar increase for the tree algorithms (Random Forest, J48), but does not show such a significant increase in the classification rate for the other three methods.

Table 5: All Days, Windowed - Accuracies

All days, Windowed, Average Accuracy over all 20 Days					
Window Size	Decision Table	Bayesian Network	Multilayer Perceptron*	Random Forest	J48
5	58.8600	44.5072	38.3823	90.0084	82.0233
10	58.8872	45.6415	42.0370	93.6073	84.1723
25	61.6733	47.6323	47.3480	97.5913	87.4408
50	64.5502	50.0735	N/A	99.2496	90.0176

Table 5 shows the five classifiers' percent of correctly classified instances using the combined dataset with various window sizes. The data was collected at 20Hz, so a window size 10 is half a second.
* Multilayer Perceptron is the slowest to train some tests ran for days at a time. A window size of 50 ran for 2 weeks before it crashed.

By looking at 2.5 seconds of data instead of 0.05 seconds, the accuracy for a random forest increased by 16.5%. Here are some explanations as to why we see this increase in accuracy:

1. Noise Resilience – many physiological sensors are susceptible to noise. EMG, for example, will often measure heartbeats. By looking at sequential data points, noise may only appear in a small percentage of the window.
2. Emotions are Fluid – emotions are fluid and happen over time and rates of time. Most models of emotion contain an event, physiological response, and cognitive response (naming the emotion). How and when they happen depends on the model. (Myers, 2004)

The random forest performs very well when training with all twenty days. It is already established that emotions appear differently from day to day (Picard, Vyzas, & Healey, 2001). The real test is how it would perform when classifying on a day it hasn't seen yet. Up to now, the classifiers were trained on 90% of the data points over all twenty days. So, I wrote a few new programs with the WEKA API. These programs implement the Leave-One-Out method on all twenty of the days so that the classifier is trained on 19 days, and tested on the one day that has not been seen.

5.1.2 Feature Selection and Analysis

Feature selection is important in this process because of the size and speed of these data sets, and the volume and rates streaming data. While its arguable that less discriminative features will have less effect on the classification, the extra computational load is not desired for a system that needs to handle large amounts of data quickly and accurately. In addition to speeding up the learning process and providing better generalization of the model, it can also reduce the curse of dimensionality (Bellman, 1957).

To gauge the impact of each feature on the classification model, information gain (Kullback-Leibler divergence) is calculated. Information gain is calculated by comparing entropy of an attribute to the entropy of the attribute when learning of a new variable. This can be useful for feature selection, and is often used in decision trees. The formulas for both are given:

Equation 2: Probability Mass Function

$$T = Training\ set$$

$$x = a\ class\ in\ training\ set$$

$$p(x, T) = \frac{|t \in T|\ x = t|}{|T|}$$

Equation 3: Entropy

$$T = Training\ set$$

$$Entropy(T) = E(T) = -\sum_{i=1}^{n} p(t_i, T) \log_2 p(t_{i,}, T)$$

Equation 4: Information Gain

$$A = \{a \in A \mid a \text{ is a unique attribute of } T\}$$

$$vals(a) = \{v \in V \mid v \text{ is unique value for attribute } a\}$$

$$InfoGain(T, a) \equiv E(T) - \sum_{v \in vals(a)}^{a} \frac{|\{x \in T | x_a = v\}|}{|T|} \cdot E(\{x \in T | x_a = v\})$$

In feature selection, information gain can be used as in feature ranking to help determine which features to include or trim from the model. Some other common techniques for feature ranking are chi squared, Pearson Correlation, and the One-attribute rule. This project uses Information gain because it is often used in decision trees, and the random forest is a set of decision trees. The tables below show the information gain in this data set. When testing with all twenty days, experiments are implemented using both raw and normalized data. In the normalized set, each day's data is normalized before being combined in the full "all days" data set. This is an attempt to establish a baseline for each day to try and mitigate the problem that emotions appear different on different days. As shown below, the normalized data set exhibits greater information gain in almost all attributes. This also positively affects the accuracy of some classifiers. This poses a problem when classifying in real time from streaming data sources. That is because normalizing data for that day requires data (such as max/min) that has not been seen yet.

Table 6: Information Gain, No Window

	Single Day	All Days	All Days, Normalized
Attribute	**Information Gain**		
EMG	0.80	0.27	0.72
BVP	0.33	0.05	0.05
GSR	1.89	0.44	1.67
RES	0.51	0.17	0.30

Table 6 shows the information gain for four biofeedback streams: Electromyography (EMG), Blood Volume Pressure (BVP), Galvanic Skin Response (GSR), and Respiratory Rate (RES). It shows it for three datasets, a single day, all days combined, and normalized all days combined.

This shows that GSR and EMG are the most important attributes in the data set, at least for a decision tree. This is not surprising; previous studies have shown skin conductance to be strong correlate of emotion (Lisetti & Nasoz, Using Noninvasisve Wearable Computers to Recognize Human Emotions from Physiological Signals, 2004). EMG is also considered a strong correlate. Facial EMG has been shown to accurately differentiate between anger and disgust (Vrana, 1993), which have fairly close proximity on the valence-arousal model. Next information gain was calculated for the windowed data sets. Below shows the mean information gain for the four streams with the respective window sizes.

Table 7: Single Day, Windowed Information Gain

	Single Day			
	Window size 5	Window Size 10	Window Size 25	Window Size 50
Attribute	**Mean Information Gain**			
EMG	0.80	0.80	0.79	0.79
BVP	0.34	0.34	0.34	0.34
GSR	1.89	1.89	1.90	1.91
RES	0.51	0.51	0.50	0.50

Table 7 shows the mean information gain of a single day for the four biofeedback streams: Electromyography (EMG), Blood Volume Pressure (BVP), Galvanic Skin Response (GSR), and Respiratory Rate (RES)

Adding windows for single day did not have much effect on the information gain, but it did increase the accuracy of most classifiers. The information gain with all days and windowed

is similar to the single day in that it does not greatly affect the information gain by adding windowing; however, the information gain does change when adding features.

I wanted to see how extracting some basic vector features would affect information gain and classification. In previous studies maximum, minimum, mean and variance were used with some success in emotions classification (Lisetti & Nasoz, Using Noninvasisve Wearable Computers to Recognize Human Emotions from Physiological Signals, 2004). Using a Perl script I created, these features and also standard deviation, and range were extracted for each of the windows. Then, using WEKA, the data was discretized and then the information gain was calculated in R.

Table 8: Single Day, Features Extracted - Information Gain

Single Day, Featured Information Gain with Window size of 5

	EMG	BVP	GSR	RES
Standard Dev.	0.0433	0.2574	0.0498	0.1214
Max	0.1023	0.3497	1.6129	0.3445
Min	0.0624	0.4539	1.6317	0.3522
Mean	0.0882	0.3612	1.6306	0.3417
Range	0.0531	0.2607	0.0509	0.1246
Variance	0.0114	0.2229	0.0180	0.0493

Single Day, Featured Information Gain with Window size of 25

	EMG	BVP	GSR	RES
Standard Dev.	0.1137	0.9938	0.0981	0.1684
Max	0.1640	1.0386	1.6057	0.3968
Min	0.2824	1.0396	1.6848	0.4954
Mean	0.1518	0.2652	1.6574	0.3974
Range	0.1289	1.0576	0.0990	0.1851
Variance	0.0508	0.8877	0.0265	0.1104

These tables show the information gain for single day datasets of various features of four biofeedback streams: Electromyography (EMG), Blood Volume Pressure (BVP), Galvanic Skin Response (GSR), and Respiratory Rate (RES). The first table shows a window of size 5 (0.25 seconds) and the second shows 25 (1.25 seconds).

For the single day, this suggests something interesting that has not been previously seen: BVP is more important than originally thought. Now, with windowing and features, BVP has the second highest information gain to GSR. Of these features, Max, Min, and Range provide the most information gain which suggests that BVP is much more useful in stream

analysis when it monitors the changes in BVP. Also, notice that while the change in information gain for GSR is small, the increase in information gain more than doubles by increasing the window size to twenty-five. We see a similar increase for both normal and normalized datasets when computing the information gain over all days (shown below).

Table 9: All Days, Features Extracted - Information Gain

All Days, Featured Information Gain with Window size of 5

	EMG	BVP	GSR	RES
Standard Dev.	0.0061	0.0401	0.0012	0.0162
Max	0.0142	0.0644	0.1739	0.0265
Min	0.0041	0.0460	0.1748	0.0274
Mean	0.0082	0.0479	0.1748	0.0271
Range	0.0085	0.0429	0.0011	0.0161
Variance	0.0026	0.0298	0.0005	0.0022
	Normalized			
Standard Dev.	0.0225	0.0394	0.0012	0.0408
Max	0.0695	0.0453	0.1931	0.0790
Min	0.0199	0.0554	0.1931	0.0787
Mean	0.0547	0.0346	0.1928	0.0790
Range	0.0289	0.0410	0.0010	0.0408
Variance	0.0058	0.0310	0.0005	0.0020

All Days, Featured Information Gain with Window size of 25

	EMG	BVP	GSR	RES
Standard Dev.	0.0141	0.1338	0.0174	0.0244
Max	0.0272	0.1646	0.1751	0.0306
Min	0.0972	0.1150	0.1761	0.0494
Mean	0.0105	0.0064	0.1764	0.0312
Range	0.0237	0.1491	0.0093	0.0251
Variance	0.0040	0.0832	0.0029	0.0113
	Normalized			
Standard Dev.	0.0434	0.1322	0.0208	0.0551
Max	0.1007	0.1621	0.1933	0.0883
Min	0.0331	0.1389	0.1945	0.0900
Mean	0.0723	0.0084	0.1935	0.0843
Range	0.0672	0.1454	0.0118	0.0580
Variance	0.0147	0.0899	0.0030	0.0240

These tables show the information gain for the all days combined dataset of various features of four biofeedback streams: Electromyography (EMG), Blood Volume Pressure (BVP), Galvanic Skin Response (GSR), and Respiratory Rate (RES). The first table shows a window of size 5 (0.25 seconds) and the second shows 25 (1.25 seconds).

5.2 Classification and Data Streams

5.2.1 Classifier Implementation

As demonstrated above, the best performing algorithm of those looked at is the random forest, followed by J48 trees. The random forest was chosen for this thesis. The random forest classifier is composed of a set of optimal split selection trees. An optimal split splits the dataset into two subsets maximizing the separation of the datasets classes. Each tree contains a random subset of features, and each tree is independent of each other. The data subset distribution is the same across all trees. In this work, each tree gets a single vote, and the majority class is predicted. Besides, being on of the best performing classifiers, another benefit of the Random Forest is that it is more resilient to over fitting than a decision tree (Breiman, Random Forests, 2001). This is largely a result of using many random subsets of features and instances for training. The authors claim, "…over fitting is not a problem". Breiman gives evidence using the Law of Large numbers to say, "…random forests do not over fit as more trees are added, but produce a limiting value of the generalization error" (Breiman, Random Forests, 2001). This is met with some skepticism, as random forests are still composed of optimal split trees.

The trees are constructed using the C4.5 algorithm (Quinlan, C4.5: Programs for Machine Learning, 1993). Another popular method is the CART method, which was introduced by Breiman *et al* in 1984. In short, this technique describes the steps to creating a classification or regression tree. The steps are as follows:

1. *The selection of the splits*
2. *The decisions when to declare a node terminal or to continue splitting*
3. *The assignment of each terminal node to a class*

This is similar to the C4.5, except for the methods of determining splits. There are a few methods for determining the splits. The CART method generally uses the Gini coefficient (Breiman, Friedman, Olshen, & Stone, 1984). Other methods include Information Gain like in the C4.5 algorithm (Quinlan, C4.5: Programs for Machine Learning, 1993), CHI-squared, and multivariate adaptive regression splines (MARS) (Friedman, 1991). The reason for using

the C4.5 algorithm is its performance, simple construction and the ability to handle continuous data, so the data does not need to be discretized before hand.

In the C4.5 algorithm, the splits are determined by maximizing the information gain. To handle continuous data, the information gain is calculated for a set of splits on the attribute. The maximum information gain is saved with the corresponding threshold. The graph below shows information gain for thresholds with steps of $1/25^{th}$ of the range of the attribute. The peaks of the graph show the maximum information gain on splitting on this attribute.

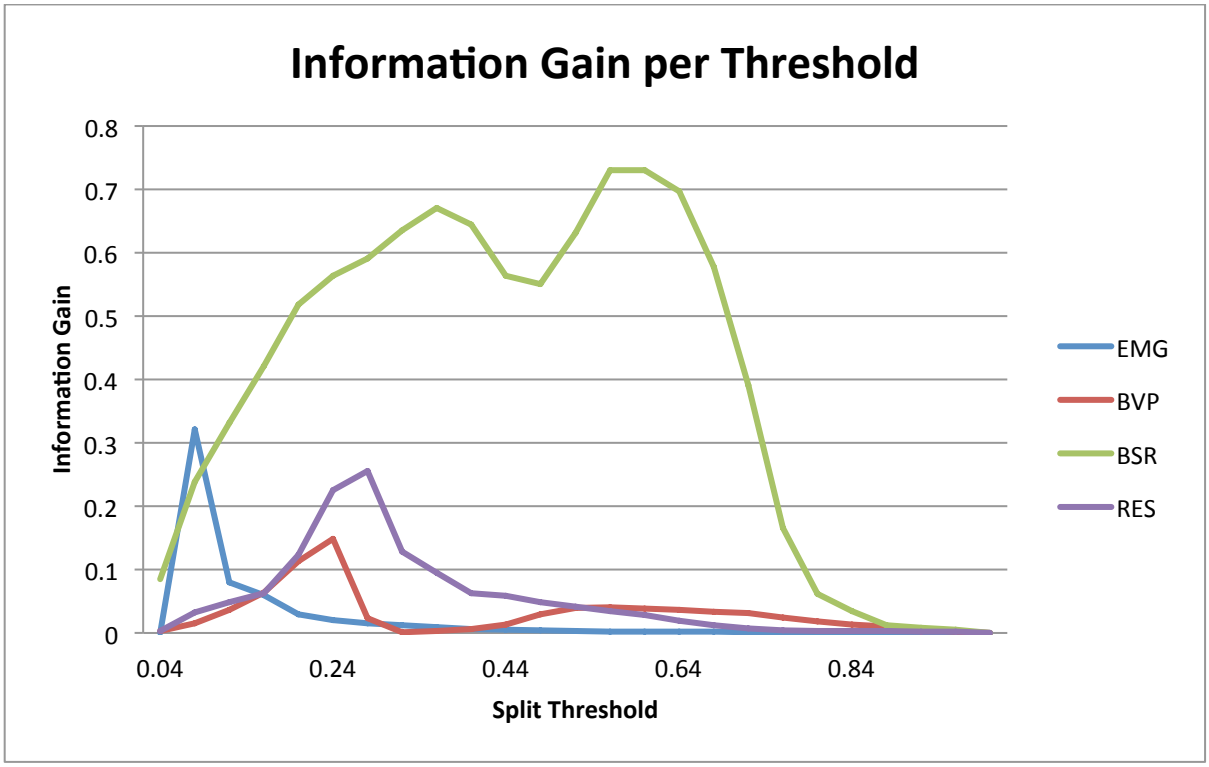


Figure 5: Information gain at each threshold

This figure shows the information gain as the split threshold changes for four biofeedback signals: Electromyography (EMG), Blood Volume Pressure (BVP), Galvanic Skin Response (GSR), and Respiratory Rate (RES).

An important aspect of random forests is the feature subset selection. That is because if there is no subset selection, all the trees in the forest will be equivalent. This defeats the purpose of having a set of trees. This random forest uses bagging, which is short for bootstrap aggregating. Given a dataset D, bagged set D_i is a set of n randomly selected

elements from D, with replacement (Breiman, Random Forests, 2001). When $n = |D|$, there are theoretically 63.2% unique elements, and the remainder are duplicates. In this application, each tree is generated using a random subset of features, and elements are selected with bagging.

5.2.2 Streaming with a Sliding Window

There are many methods for mining data from streams such as sampling, load shedding (Bai, Wang, & Zaniolo), sketching, aggregations, approximation algorithms, clustering algorithms and time series analysis (Gaber, Zaslavsky, & Krishnaswamy, 2005). Each method tackles different problems with streaming and work toward different goals. A sliding window is a method of capturing the data at time points along with a small amount of adjacent historic data. In this application, the most recent data is the most important, so the sliding window is a sensible choice for this model (Gaber, Zaslavsky, & Krishnaswamy, 2005) (Datar, Gionis, Indyk, & Rajeev, 2002). Using the archival test data, it will be streamed with inter-process communication and each new data point will become the most recent data.

In 2002, Datar et al proposed a sliding window construction in which the problem they tackled was simply to count the number of 1's in a byte stream. Their work on the sliding window and definitions influenced the design of this project (Datar, Gionis, Indyk, & Rajeev, 2002). In the implementation for this thesis, a stream manager contains a set of streams. The streams contain information about themselves including window size, stream type, and the data type that the stream is expected to emit. The streams allow multiple data attributes for the same stream. This means that multiple windows can be used for the same stream. The random forest in a separate thread reads the windows.

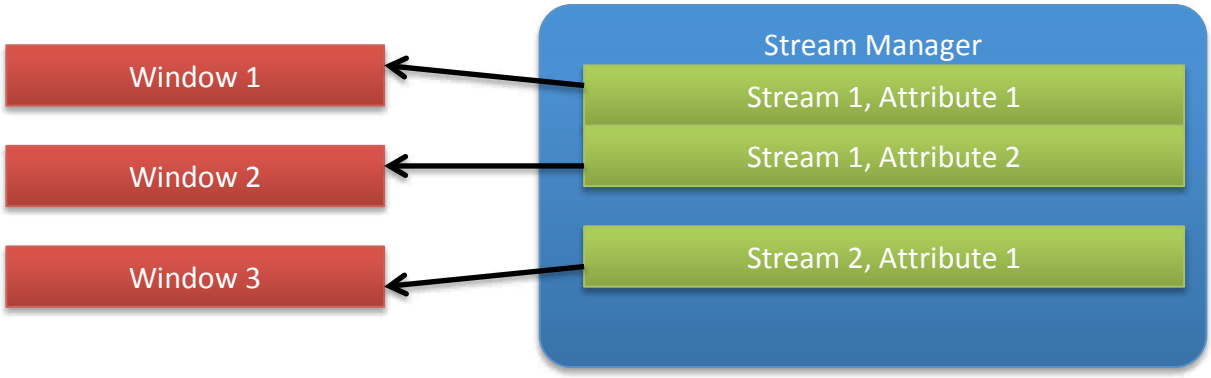


Figure 6: Stream Manager

The stream manager manages a set of streams. Some streams may contain more than one attribute. The stream manager creates a separate window for each attribute.

There are a few methods in place to track the validity of the read data. First, each window in a stream tries to verify the data type in the new data packets as they are read. This helps prevent dirty or invalid reads from a stream. Secondly, all data points are given a time stamp at the moment they are read from the stream. This enables one to calculate a few things that help make decisions on what to do with the data:

1. *The age of the data*. The age is an important factor in determining whether or not to run expensive calculations in a time sensitive situation.
2. *The real-time range.* The real time range can help the program decide whether data is being lost. If a stream is expected to output at 20Hz and the time range shows a smaller rate, then one may conclude that there is either a slowed rate on the stream, or the reader cannot keep up with the rate that stream is updating.
3. *Stream Synchronization.* When looking at sliding windows over multiple streams on can determine if they sliding windows are in the same time frame. Otherwise, data points from different times could be falsely combined

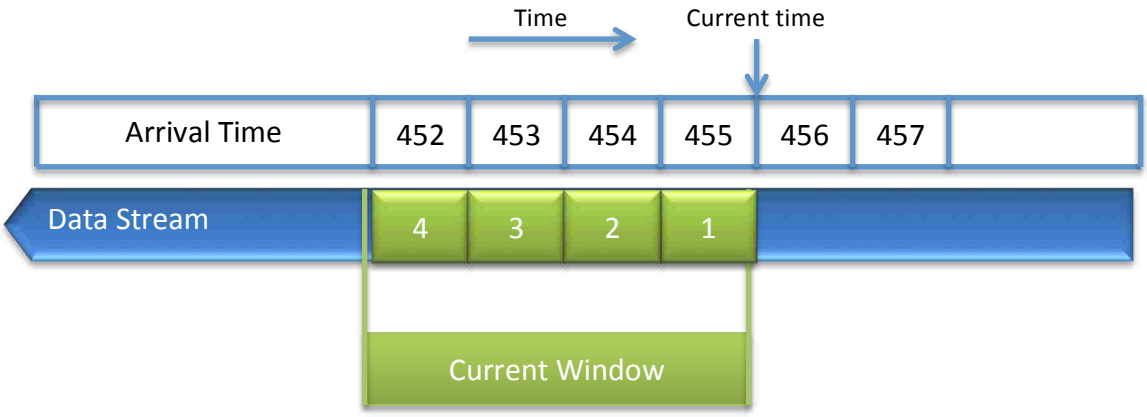


Figure 7: Sliding window

Figure 7 shows a sliding window with a size 4 as it slides across a data stream over time.

5.2.3 Stream Simulations

Since biofeedback-streaming hardware is not available at this time, sending messages at specified rates is used to simulate a stream. The software developed for this research uses a named pipe inter-process communication. To send messages at specific rates, the program uses a partial leaky token bucket. It is partial, because reading data from a file will never give too much data. A stream will produce overflow if the stream frequency is greater than the read frequency. For example, if a data stream outputs at a frequency of 1MHz and the classifier is limited to 0.5MHz, then it will have to leak data. Otherwise, the classifier will fall behind and be using old data. This does have a few disadvantages that are not easily implemented without a real stream:

1. Streaming hardware can transmit at very high speeds, often with dedicated hardware. This simulation with this program could not achieve those rates. However, for this application it works okay because the data rates are relatively low. The data used in the initial testing was collected at 20Hz
2. This simulation can match the amount of data that a stream provides. Streams can have an enormous amount of data. Theoretically, they do not have a beginning or end, and the amount of data grows linearly when increasing the rate.
3. Some other issues with streams are not simulated, such as varying rates.
4. Named pipes are queues by nature, meaning that data is not lost if a program is not listening to the stream. In a real stream situation, data is only seen if someone is listening. This may be remedied using a virtual serial connection.

5.3 Transition Synthesis

The archival data collected shows time periods of various emotional states. It does not show transitions from one emotion to another. The transitions are important in trying to identify the earliest point in which an emotion can be predicted. Since transition data is not available at this time, the data is synthesized to create a simulation of streaming data. While this data is synthetic, it could be beneficial in determining the plausibility of a system like this.

The software creates the transitions by overlapping a portion of the data points at the two edges of an emotion. The weighted average is calculated depending on the position of the data point. Given a transition window of *n*, the weights are determined increasing linearly from $1/n$ to 1 for the first emotion and decreasing linearly from 1 to $1/n$ for the second emotion. This gives a larger weight to the first emotion, and decreases as it progresses into the second emotion. A new data point in the synthetic transition is calculated in the formula that follows:

Equation 5: Weighted Averages

$n =$ Window or frame size of the transition

$i =$ The position or index of the data point in the transition

$$Weight\ of\ Stream\ a\ at\ point\ i = W_{ai} = 1 - \frac{i}{n}$$

$$Weight\ of\ Stream\ b\ at\ point\ i = W_{bi} = \frac{i}{n}$$

$$t_i = A_i W_{ai} + B_i W_{bi}$$

The frame size allows for an adjustable transition timeline when the frequency of the biofeedback signals is known. In this case, the frequency is 20Hz so a transition time of five seconds is a frame size of 100. The charts below show some synthesized transitions.

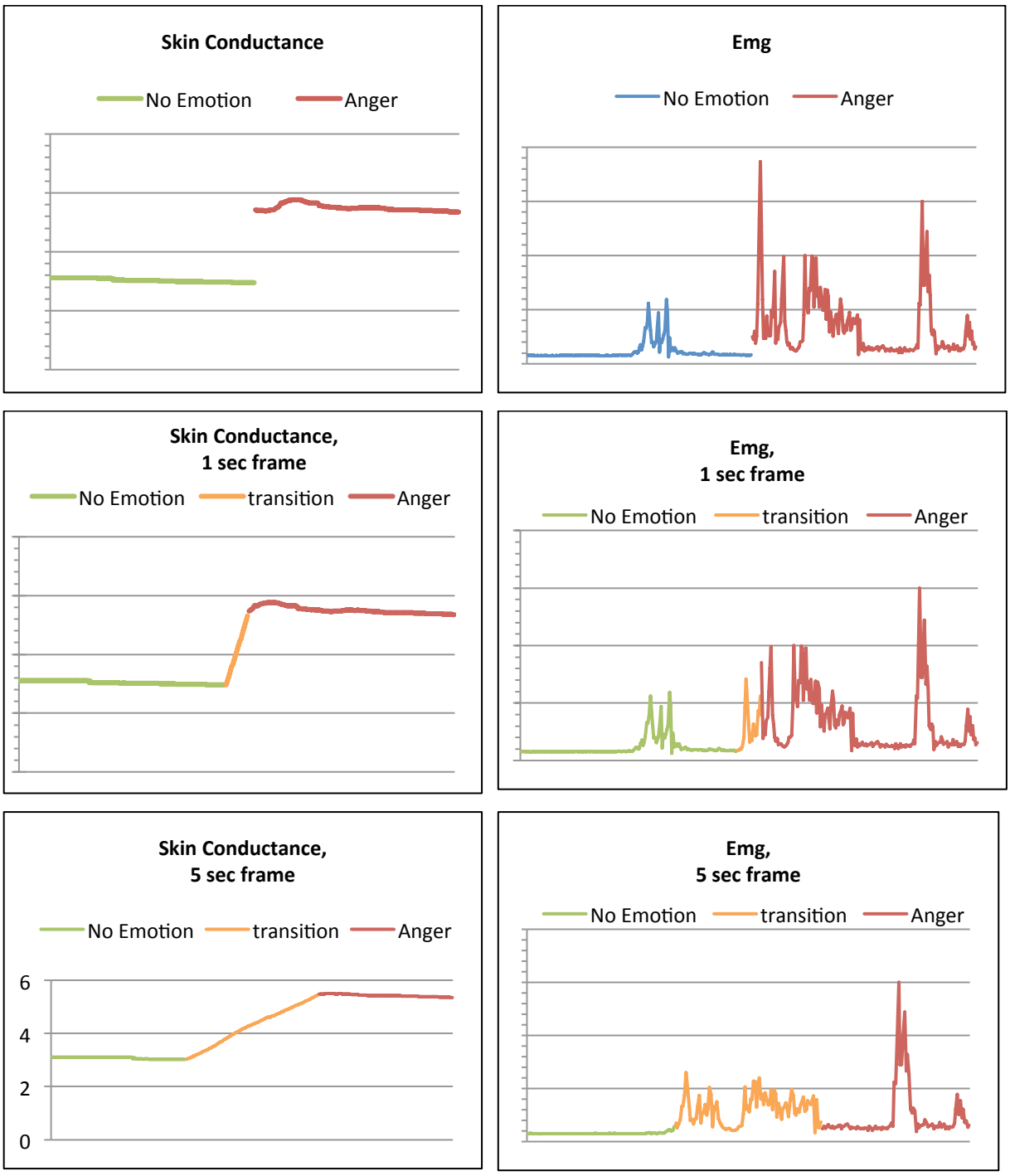


Figure 8: Linearly Increasing Weighted Average

This figure shows the transtion from no emotion to anger for two biofeedback signals: Skin Conductance (GSR), and Electromyography (EMG). The first row shows the raw data with no transition. The second shows the same signals with a one second transition using a weighted average. The third shows the signals with a 5 second transition using a weighted average.

6 Results

6.1 Classification

As shown in the methods section above, the random forest is a reliable classifier for emotions. Initial tests on unprocessed biofeedback data show an average of 91.6% accuracy over twenty days. This accuracy drops by 10% for the random forest, and as much as 50% for other classifiers when the classifier tackles the problem of day-dependence. That is, experiments classifying all twenty days combined drops the success rates substantially.

This thesis applies the windowing function and feature extraction to the biofeedback signal data. This is important to the streaming aspect of the project for a couple reasons. The first is that biofeedback data is time series data. The correlates of emotion are measured at specific times and have order. Many classification techniques randomize order and do not consider data as having a temporal component. By windowing the data the system can also consider the physiological events that occurred in the most recent past. The second is that windowing is an established methodology in modern data stream mining research (Gaber, Zaslavsky, & Krishnaswamy, 2005).

Using the windowing and extraction techniques, the average accuracy over all twenty days is increased substantially. For this experiment, features were extracted for the various window sizes over the twenty days. This gives six features per data stream, for a total of twenty-four features. Combine twenty-four features with 320,000 instances, and you get 7.68 million data points. This is way too much data for many computers to handle, so we can trim the data by removing the features that do not provide as much information gain. Using **Table 9: All Days, Features Extracted - Information Gain**, features with the lowest information gain score are eliminated to make the dataset more manageable. The table below shows the results of windowing and feature extraction over various size windows.

Table 10: All Days, with Feature Extraction

All Days, Windows and Features Extracted with the Random Forest

	Window 0.25 s	Window 0.5 s	Window 1.25 s	Window 2.5 s
Accuracy (%)	96.17	98.79	99.91	99.94

Table 10 shows the percent correctly classified when combining all days into a single dataset and applying windowing and feature extraction techniques.

6.2 Transitions

The transitions software is used on a non-windowed data set, and a data set where features where extracted with a window size of 2.5 seconds. It creates transitions over 1 second, 2.5 seconds, and 5 seconds. The sections are labeled separately so that the training set for the classifier does not use the transition set when building the classifier. Then, the transition sections are classified as a streaming source. The figures below show the predictions over time from left to right. First series (in red) shows the emotion, while the second (in green) shows the predicted emotion.

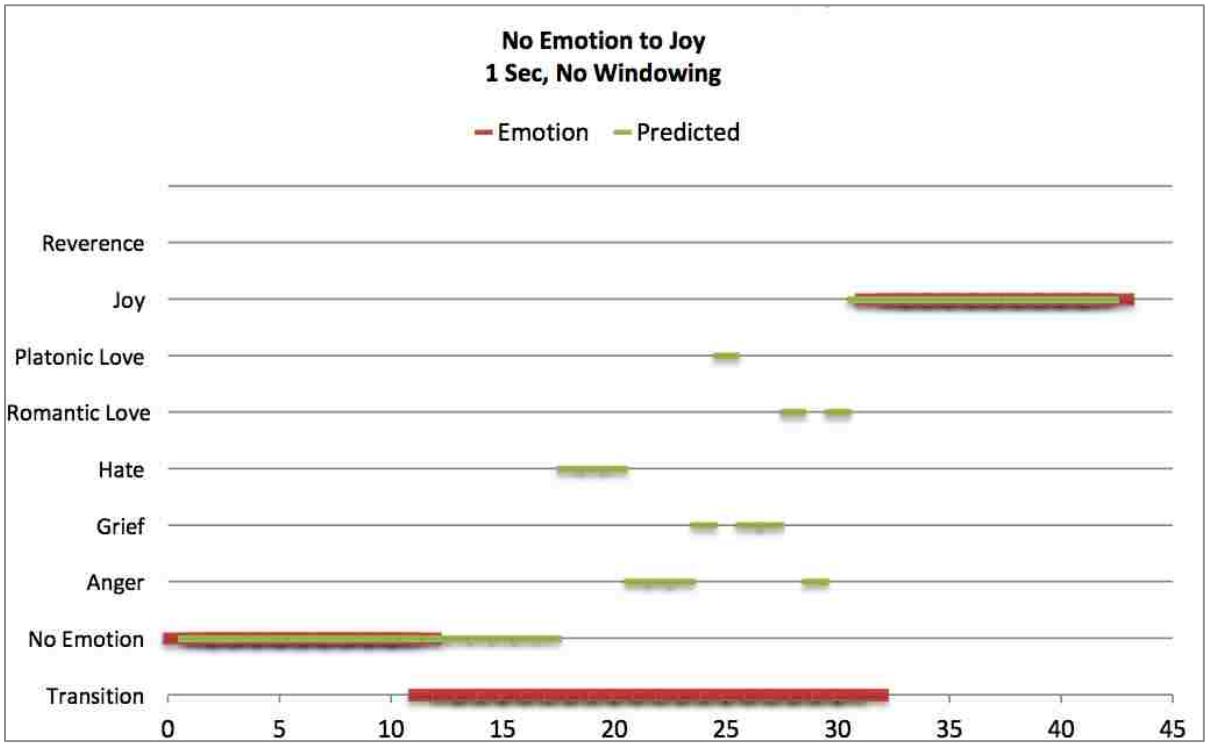


Figure 9a: This shows the transition from no emotion to joy with a one second transition and no windowing.

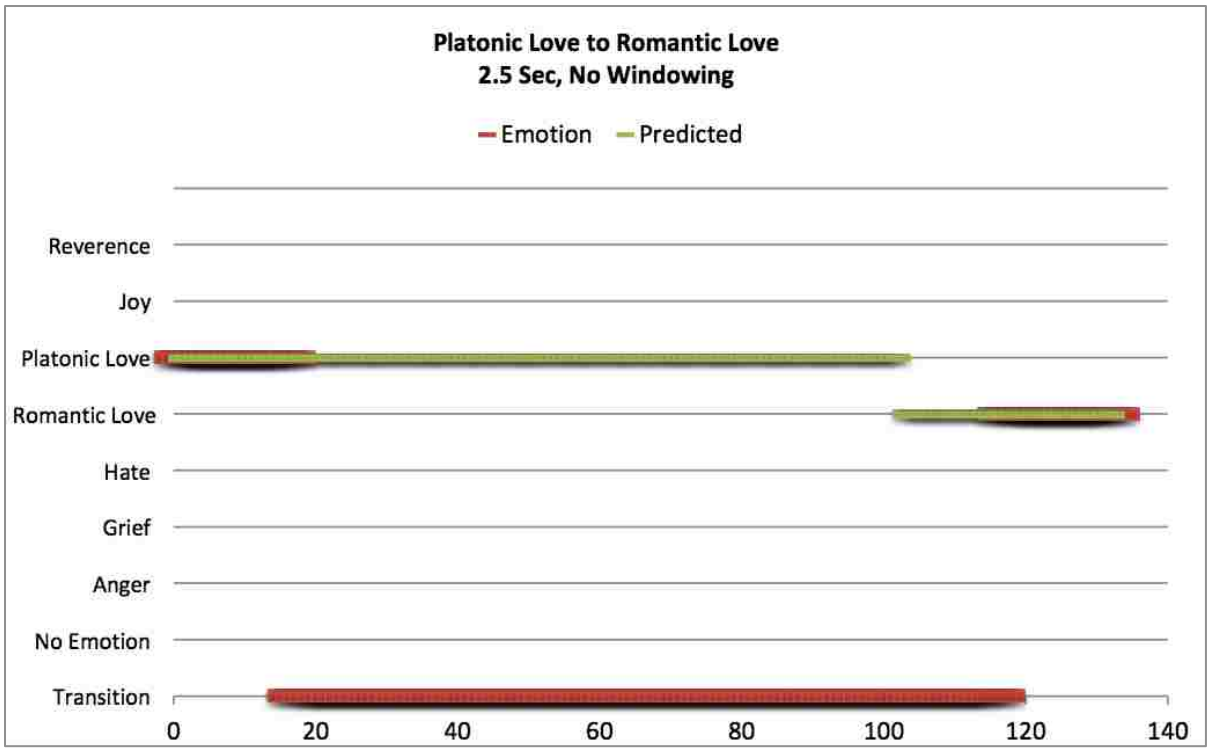


Figure 9b: Shows a transition of platonic love to romantic love with a 2.5 second transition and no windowing.

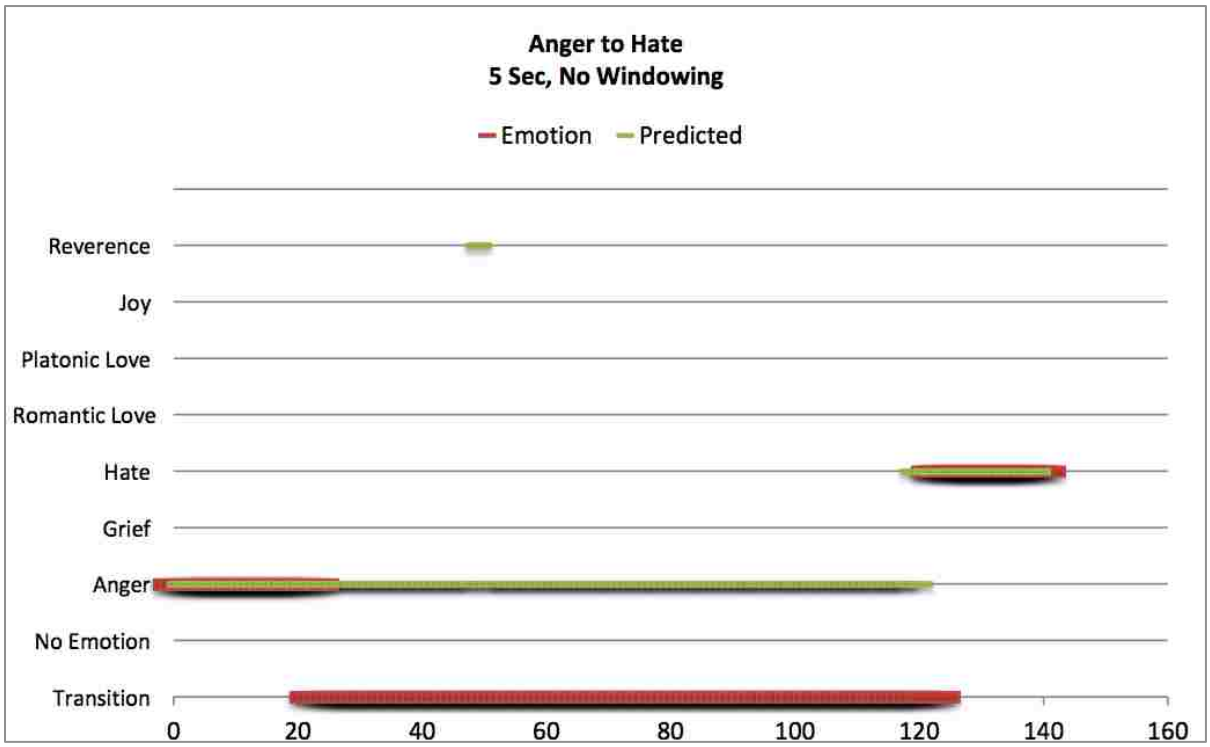


Figure 9c: Shows a transition from anger to hate over 5 seconds and now windowing.

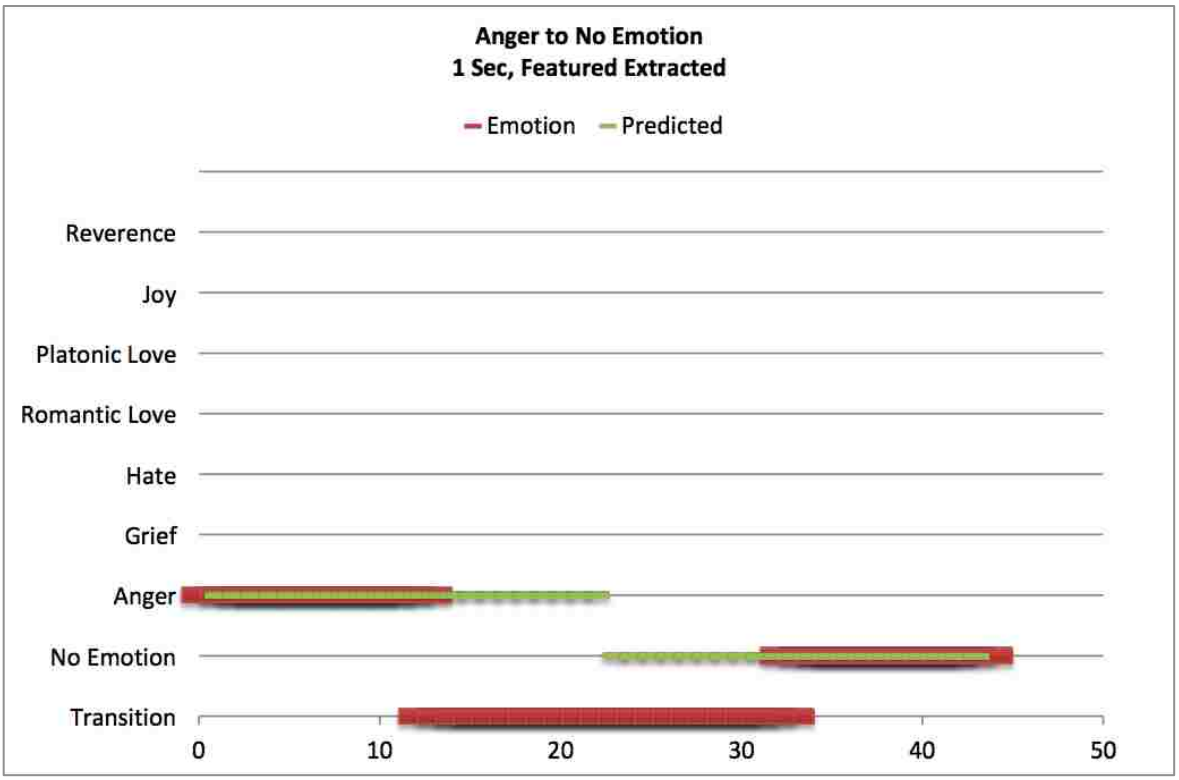


Figure 9d: Transition from anger to no emotion over 1 sec. Uses a 1.25 second window and feature extraction.

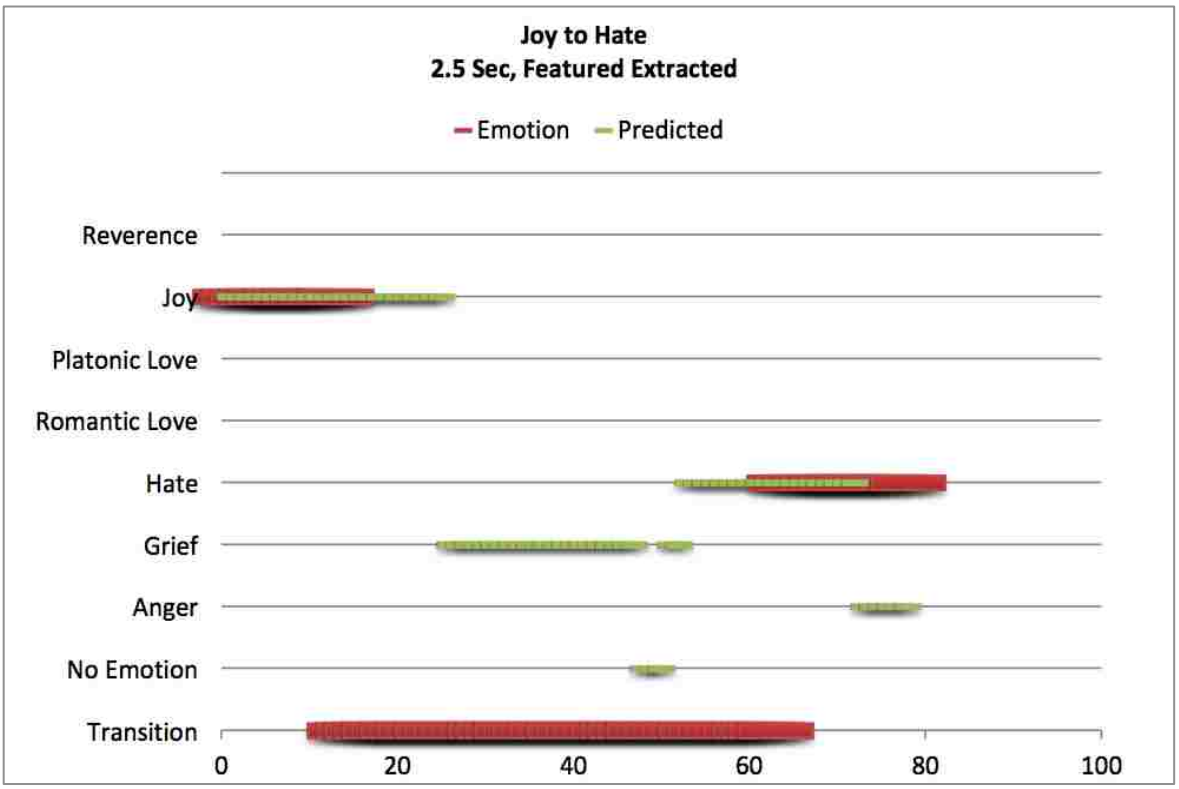


Figure 9e: Transition from anger to no emotion over 2.5 sec. Uses a 1.25 second window and feature extraction.

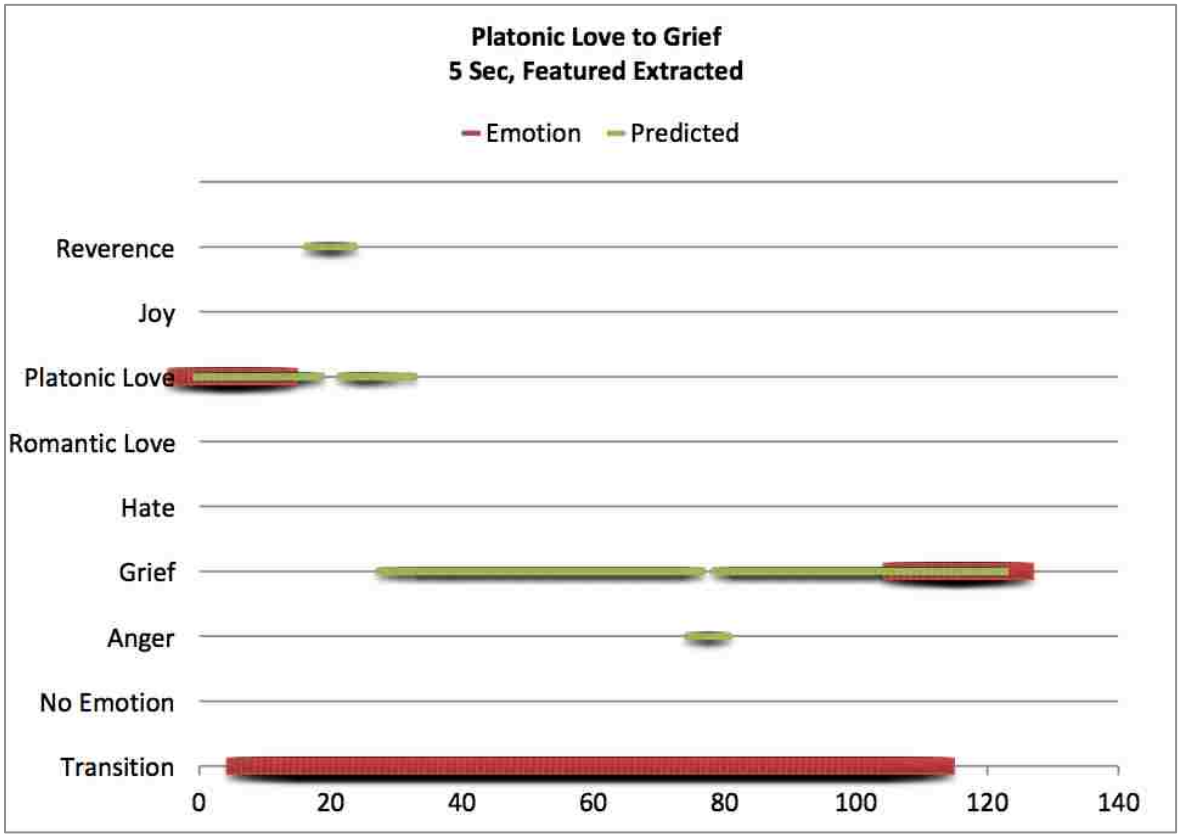


Figure 9f: Shows the transition from platonic love to grief with a five second transition. It uses a 1.25 second windowing and feature extraction.

Figure 9: Transition Classification

Figure 9 shows examples of the transition analyses using three transition lengths with no windowing in 9a, 9b 9c. 9d, 9e and 9f show results when a 1.25 second window with feature extraction is applied. The emotions can be predicted noticeably earlier when the windowing function is applied. The red bar represents the actual classification, and the green shows the predicted emotions over time.

Figure 9: Transition Classification shows an example of the varying results of transitions classification. As with previous classification attempts, the results are more successful when a windowing system is used. For the first three charts, the classifier predicts the emotions late. Meaning there is little to no warning of an impending transition. It also jumps around to many emotions before concluding to the second emotion (Figure 9a). When an emotion translates into a similar emotion (e.g. Figure 9b), the classifier does not "jump around" a lot. This suggests that, physiologically, two similar emotions are not that distant, which supports the valence arousal model of emotion. Also, it suggests that it

may be easier to classify physiological data as a quadrant of this model, than as a discrete emotion.

The last three charts show the transitions when a 2.5 second sliding window and feature extraction is applied. Overall the point in which the transition is predicted is earlier than that of the windowless transitions. This is because the window holds the data of the each emotion longer while it is in the transition phase. For a one second transition period, the window is actually longer than the transition and that leads to very accurate classifications. It would be remarkable to see if a similar result occurs when the sliding window is longer than an organic transition.

While examining transitions like this is interesting, it is not exactly predicting transitions. It is predicting that a person is in an emotional state during a transition period. The real goal is to be able to predict that a subject is in transition from emotion *a* to emotion *b.* It would be remarkable if the algorithm could predict that a person is transitioning to anger, and the earlier, the more remarkable. The experiments are modified to attempt this.

Using the same 1, 2.5, and 5-second transition lengths, the training set was modified so that the transition periods were classified as a transition from the first emotion to the second (e.g. no emotion_anger is a transition from no emotion to an anger state). With 8 emotions, this bumped the number of classes to more than 50 classes. This multiplicatively increases the computation time, and the machine used could not handle this much computation. So, the set of original classes is reduced to: No emotion, Anger, Grief, and Joy. The transitions and feature extraction are created for all twenty data sets, and for window sizes of 0.25, 0.5, 1.25, 2.5, 15, and 30 seconds. For each emotion and a transition length of *n*, the training and test sets contain *n* points that transition into to concrete emotion state, *n* points of the emotion state, and *n* points transitioning out of the state.

For each of the window lengths, and each of the transition lengths, the experiment is run using the leave-one-out method. That is, the classifier trains on 19 datasets, and tries to predict the dataset that was “left out”. Each of the 20 datasets will be left out, which

gives 20 runs, and the average accuracy is recorded. This means that there are 300 experiments total, and the results are shown below.

Table 11: Transition Prediction Accuracy (% Correct)

Transition Length (Seconds)	Window Size (seconds)					
	0.25s	0.5s	1.25s	2.5s	15s	30s
1s	28.8%	27.4%	22.7%	22.5%	13.1%	17.5%
2.5s	15.5%	16.6%	16.1%	13.8%	13.8%	13.4%
5s	15.8%	17.7%	16.6%	15.0%	13.1%	17.5%

Table 11 shows the average percentage of correctly classified instances when classifying transitions to attempt to predict an emotion before it occurs. Each column represents window length, in seconds. The rows represent the transition length, in seconds. The results show that the method attempted does not accurately predict and emotion transition.

Table 11 shows the results of this experiment. It shows that the method attempted cannot be used to accurately predict an emotion transition. Adding the transition classes also reduced the accuracy when predicting concrete emotional states. One explanation for this is that the transition data covers a wider range of data points that have close proximity to the two emotions in its transition. This transition may also pass through a range of another emotion. Another explanation is that the experiment does not contain enough concrete emotion data points in the training.

To explore if this is true, the new experiment multiplies the number of data points in the concrete emotion state. This gives the random forest more information about the concrete emotional state to hopefully increase the accuracy when predicting the concrete emotional states. If it stabilizes the predictability of the concrete emotional state, the experiment could provide better conclusions about predicting transitional states. This did increase the accuracy, but not by enough to be able to analyze the transitions. The classifier still incorrectly predicts many of the concrete emotion states so we cannot examine the transitions with confidence. **Table 12** shows the percent of correctly classified instances.

Table 12: Extended Concrete Emotion, Transition Accuracies

Transition Length (Seconds)	Window Size (seconds)					
	0.25s	0.5s	1.25s	2.5s	15s	30s
1s	22.0%	23.7%	21.1%	23.6%	16.0%	19.0%
2.5s	24.4%	26.8%	26.2%	27.2%	20.9%	21.2%
5s	28.8%	31.7%	32.6%	30.2%	27.7%	27.8%

Table 12 shows the average percentage of correctly classified instances when classifying transitions to attempt to predict an emotion before it occurs. This experiment uses the extended concrete emotion training sets, in which the number of data points in the concrete states is much larger than the transition states. Each column represents window length, in seconds. The rows represent the transition length, in seconds.

6.3 Streaming Data

The streaming data implementation consists of three parts: the simulator, the stream manager, and the sliding window. This is illustrated in **Figure 10**. The simulator contains a user specified rate that it uses to write data with equal intervals between messages. Next the stream manager reads data from the simulated steam via inter-process communication. It then writes to a sliding window of the most recent data. A trained random forest then reads from the sliding window and classifies that instance. The results show that windowing and feature extraction are valuable assets in emotion classification. This thesis shows that a properly trained random forest would have no trouble classifying emotion in real time and, therefore provides a method of transferring these concepts into real world applications.

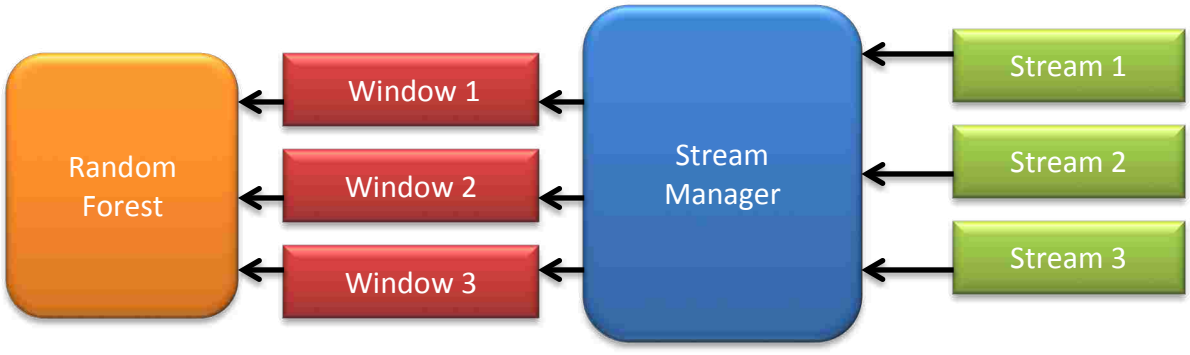


Figure 10: Stream Simulation Process

This figure shows the stream simulation. The stream manager contains a set of streams in which it reads from and sends data to the windows. The random forest can then read from the windows to make a prediction.

7 Conclusions

The proof of concept concluded that random forests perform particularly well in this application. So, this thesis builds on that with implementations of a stream simulator, a stream reader, and a random forest. The combination of these implementations has shown that the classifier can predict an emotion in real time. The major benefit of the windowed stream reader is that the window allows a classifier to see the data as a time series. After careful analysis of the results from the preliminaries, feature extraction, random forest implementation, stream simulation, and Weka, the following conclusions are proposed.

While the main focus of this thesis is the predictive ability of windowing streaming biofeedback signals, an interesting side effect is the increased accuracy. By comparing the success rates of the windowless classifier and the features extracted, it is easy to see that using windowing greatly increases the success rate. The success can increase by almost 20% by extracting features from the most recent data. The explanation for this is that emotions are not instantaneous experiences. They are experienced over time as reactions to external stimuli. An emotion doesn't look the same each day, but the results suggest that the changes in the biofeedback signals are more important than their specific values.

Given proper data, random forests can be reliably trained to classify a single person's emotional state, and it can predict transitions into a new state. Even in the preliminaries with the unprocessed data, the classifiers give accurate results, especially the random forest and the J48 tree. When the data sets were combined, the accuracies dropped substantially. The random forest and J48 still give respectable results, hovering around 80%. This is increased when the windowing function and feature extraction are applied **(Table 10)**. When this is applied to transitions, there are similar results.

The thesis examines emotion transition in two ways. The first classified the states without trying to predict a transition. That is, it simply predicted one of the eight emotions, even in a transition state. When using this method without any windowing, the new emotion is not predicted until almost the exact moment of the change in state (Figure 8a, Figure 8b, Figure 8c). When a windowing function is applied this is increased with variable

rates. In Figure 8d, the transition is predicted half way through the transition, so 0.5 seconds before the emotion. This is with a one second transition, and a 1.25 second window. In Figure 8e, the emotion is not predicted as early, but it is still earlier than when using no window. In the last example, Figure 8c, the emotion is predicted around four seconds earlier than the new emotion, given a 5 second transition from platonic love to grief.

The problem with this method is that it is not actually predicting that the subject is transitioning to a new state. It is predicting that the subject is in the new state. Next, transitions are introduced as classes. The system tries to predict that the subject is transitioning from *a* to *b*, but the results are disappointing. That is not to say it cannot be done. There are many more facets to explore. Emotions are complicated states and are not necessarily discrete.

The designed system could help determine the earliest time point an emotion is predictable. Testing with organic emotion transitions is required for stronger conclusions. In the transitions section, the data shows that earliest points could be found. The big disadvantage is twofold. The first is that the data is synthetic. This is obvious, but the synthetic data does have its use in showing this possibility. The second is that there is no clear transition point. The data simply picks two edges in the archival data and splices them together. The second is that a subject-specified instant is needed at the point of cognitive recognition. This is to determine whether the classifier can predict the emotion before the subject acknowledges the emotion.

Emotion classification in real time is plausible with the methodology introduced in this thesis, but more questions still exist. Can a classifier be trained to accurately predict the emotions of multiple people? Can we use different features or biofeedback signals to accurately predict transitions? Could a classifier be resilient to changes in a person's physiology over time? This thesis does show that windowed steaming method performs particularly well with this data, and it demonstrates that the data should be examined with temporal element. For a single person, real time emotion classification can be successful with proper classifier training. This method can potentially be integrated with real world

systems such as vehicles, mobile phones, hospital rooms or any other application one could imagine.

8 Future Work and Open Problems

8.1 Classifier Improvement

As mentioned above, there are many ways to implement a random forest. While Weka's classifier predicted very high accuracy (90-99%), the classifier implemented for the thesis predicted between 10-20% less accurate. The implementation section above established that the difficult part is making the decision on where to split. In this case, it is affected by two major design components:

1. Information Gain
2. Handling Continuous data.

One thing that might increase the accuracy into the range of Weka's random forest is experimenting with the different splitting methods. Such as the Gini coefficient, chi squared method, etc. The other problem is how to handle continuous data. It might be that finding the best split does not perform as well as discretizing the data first by binning it. This introduces a new problem. Since all incoming data is continuous, and the system would need a way to normalize data that it has not seen before.

8.2 Synthetic Data Transitions

One of the major problems in studying the transition of different emotions is that the data does not have transitions between emotions. So, they are synthesized. The ability to predict the earliest point in an emotional transition is remarkable, however, without real data, we don't know how the transitions happen. The major issues with this are:

1. Transitions are not likely to be linear
2. Transitions for different signals likely have different rates of change (e.g. fear might happen very quickly, while another may take more time to build up)

It does show that a windowing system possibly could be used to predict the earliest point the emotion is classifiable.

8.3 Multi Person, Multi Day Data

Another project that could be explored in the future is the ability of these classifiers to predict emotions reliably for different people. The archival data focuses on one person over twenty days, and previous research suggests that emotions look different for different people. Predicting emotions over multiple people adds another layer of complexity, but it would certainly be remarkable to accurately classify one person's emotional state while training the classifier on another person's biofeedback.

It is also known that correlates of emotion look different from day to day (Picard, Vyzas, & Healey, 2001). This suggests that a person's emotional model could also change over time too. One example of this is that a person may work hard at getting into shape. This affects many things physiologically and could change the way the emotional correlates appear over time. If this were the case, perhaps there would be a way to collect baseline of the correlates of emotion, and therefore, calibrate the classifier from time to time.

9 References

1. Bai, Y., Wang, H., & Zaniolo, C. *Load Shedding in Classifying Multi-Source Streaming Data: A Bayes Risk Approach.*
2. Bamidis, P. D., Frantzidis, C. A., Konstantinidis, E. I., Luneski, A., Lithari, C., Klados, M. A., et al. (2009). An Integrated Approach to Emotion Recognition for Advanced Emotional Intelligence. *Human-Computer Interaction, Part III* , 565-574.
3. Bellman, R. (1957). *Dynamic Programming.* Courier Dover Publications.
4. Breiman, L. (2001). Random Forests. *Machine Learning , 45*, 5-32.
5. Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees.* Belmont, CA: Wadsworth International Group.
6. Cannon, W. (1927). The James-Lange theory of emotion: A critical examination and alternative theory. *American Journal of Psychology , 39*, 10-124.
7. Datar, M., Gionis, A., Indyk, P., & Rajeev, M. (2002). Maintaining Stream Statistics over Sliding Windows. *In Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete Algorithms (SODA '02)* , 635-644.
8. Ekman, P. (1992). An Argument for Basic Emotions. *Cognition and Emotion* (6), 169-200.
9. Ekman, P., Levenson, R. W., & Friesen, W. V. (1983). Autonomic Nervous System Activity Distinguishes among Emotions. *Science , 221*, 1208-1210.
10. Friedman, J. H. (1991). Multivariate Adaptive Regression Splines. *Annals of Statistics* , 1-67.
11. Gaber, M. M., Zaslavsky, A., & Krishnaswamy, S. (2005). Mining Data Streams: A Review. *SIGMOD , 34*, 18-36.
12. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutmann, P., & Witten, I. H. The WEKA Data Mining Software: An Update. *SIGKDD Explorations , 11* (1).
13. James, W. (1884). What Is an Emotion. *Mind* (34), 188-205.
14. Jolliffe, I. T. (1986). *Principal Component Analysis.* New York: Springer-Verlag.
15. Kohavi, R. (1995). The Power of Decision Tables. *Proceedings of the European Conference on Machine Learning* , 174-189.
16. Lisetti, C. L., & Nasoz, F. (2002). MAUI: a Multimodal Affective User Interface. *Proceedings of the tentch ACM international conference on Multimedia (MULTIMEDIA '02)* , 161-170.
17. Lisetti, C. L., & Nasoz, F. (2004). Using Noninvasisve Wearable Computers to Recognize Human Emotions from Physiological Signals. *EURASIP Journal on Apllied Signal Processing* , 1672-1687.
18. Myers, D. G. (2004). Theories of Emotion. In *Psychology* (7th ed., p. 500). New Yortk, NY: Worth Publishers.
19. Nasoz, F., Lisetti, C. L., & Vasilakos, A. V. (2010). Affectively Intelligent and Adaptive Car Interfaces. *Information Sciences , 180*, 3817-3836.
20. Oehme, A., Herbon, A., Kupschick, S., & Zentsch, E. (n.d.). Physiological Correlates of Emotion.

21. Picard, R. W., Vyzas, E., & Healey, J. (2001). Toward Machine Emotional Intelligence: Analysis of Affective Physiological State. *IEEE Transactions on: Pattern Analysis and Machine Intelligence , 23* (10), 1175-1191.
22. Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers.
23. Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning* , 81-106.
24. Rebenitsch, L., Owen, C. B., Brohil, C., Biocca, H., & Ferydiansyah, R. (2010). An Exploration of Real-Time Environmental Interventions for Care of Dementia Patients in Assistive Living. *PETRA'10* (p. 8 Pages). Samos, Greece: ACM.
25. Russel, J. A. (1980). A Circumplex Model of Affect. *Journal of Personality and Social Psychology , 39* (6), 1161-1178.
26. Schachter, S., & Singer, J. (1962). Cognitive, Social, and Physiological Determinants of Emotional State. *Psychological Review , 69*, 379-399.
27. Sinha, R., & Parsons, O. (1996). Multivariate response patterning of fear and anger. *Cognition and Emotion , 10* (2), 173-198.
28. Team, R Development Core. (2012). R: A language and environment for statistical computing. Vienna, Austria.
29. Villon, O., & Lisetti, C. (2007). A User Model of Psycho-physiological Measure of Emotion. *Proceedings of the 11th international conference on User Modeling (UM '07)* , 319-323.
30. Vrana, S. (1993). The psychophysiology of disgust: differentiating negative emotional contexts with facial EMG. *Psychophysiology* , 279-286.