IDENTIFYING EMOTIONAL STATES THROUGH KEYSTROKE DYNAMICS

A Thesis Submitted to the College of

Graduate Studies and Research

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In the Department of Computer Science

University of Saskatchewan

Saskatoon, CANADA

By

Clayton Epp

Keywords: Affective computing, keystroke dynamics

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

ABSTRACT

The ability to recognize emotions is an important part of building intelligent computers. Extracting the emotional aspects of a situation could provide computers with a rich context to make appropriate decisions about how to interact with the user or adapt the system response. The problem that we address in this thesis is that the current methods of determining user emotion have two issues: the equipment that is required is expensive, and the majority of these sensors are invasive to the user. These problems limit the real-world applicability of existing emotion-sensing methods because the equipment costs limit the availability of the technology, and the obtrusive nature of the sensors are not realistic in typical home or office settings. Our solution is to determine user emotions by analyzing the rhythm of an individual's typing patterns on a standard keyboard. Our keystroke dynamics approach would allow for the uninfluenced determination of emotion using technology that is in widespread use today. We conducted a field study where participants' keystrokes were collected in situ and their emotional states were recorded via self reports. Using various data mining techniques, we created models based on 15 different emotional states. With the results from our cross-validation, we identify our best-performing emotional state models as well as other emotional states that can be explored in future studies. We also provide a set of recommendations for future analysis on the existing data set as well as suggestions for future data collection and experimentation.

ACKNOWLEDGMENTS

I would like to convey my appreciation to my supervisor Regan Mandryk for her support and guidance throughout my graduate career at the University of Saskatchewan. I would also like to thank the faculty and staff in the Department of Computer Science and the members of the Interaction Lab both past and present. In particular, I would like to express my gratitude to Mike Lippold, Andre Doucette, and Craig Yellowlees for their assistance during this process. Finally, I would like to thank Carrie Demmans Epp for all the support that she provided throughout my time as a graduate student.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ARFF	Attribute-Relation File Format
AV	Arousal/Valence
CMC	Computer mediated communication
CSV	Comma Separated Values
EKG	Electrocardiography
EMG	Electromyography
ESM	Experience Sampling Methodology
FAR	False Alarm Rate
FP	False positive
FRR	False Reject Rate
GSR	Galvanic Skin Response
GUID	Globally unique identifier
IM	Instant Messaging
IP	Internet Protocol
IPR	Imposter Pass Rate
LIWC	Linguistic Inquiry and Word Count
MIP	Mood Induction Procedure
PCA	Principle component analysis
PHP	PHP: Hypertext Preprocessor
TP	True positive
WEKA	Waikato Environment for Knowledge Analysis

CHAPTER 1

INTRODUCTION

1.1 PROBLEM

If computer systems were capable of recognizing users' emotions they would be able to make more intelligent decisions; however, today's computer systems typically do not incorporate the emotional context of a situation in the decision making process. A form of emotional intelligence would provide a richer context from which computers could make better decisions.

In some situations, computer systems with emotional intelligence could attempt to infer the possible causes of these emotions through the situation variables, and then respond appropriately. For example, in tutoring programs the subject material could be altered depending on the student's emotional state. If the student is frustrated, the program could provide assistance in some form (e.g., an alternate explanation/example). Conversely, if the computer detects that a student is bored and yet performing well, the computer system could then provide the student with more challenging activities or speed up the pace of the material presented.

In other situations, it may not matter what the specific cause of the emotion is, just that the computer should take some course of action. For example, detecting when users are in a stressed, distracted, or fatigued state would be beneficial in high-stress occupations, such as the monitoring of mission-critical systems, because mistakes have the potential for catastrophic outcomes. Perhaps the user is fatigued from a poor night's sleep due to a noisy neighbour or is stressed about a recent performance review in which he did poorly. In these types of situations,

the cause of the user's state is not pertinent to the immediate situation. Even if it were, it is unlikely the computer would be able to assist with the root causes of these matters. The more important issue is that the user may not have his mind fully on the task at hand which may lead to mistakes. If computer systems could detect when the user was in one of these states, the system could provide either the user or a supervisor with feedback identifying a potentially dangerous situation.

An emotionally intelligent computer could also be used to facilitate computer-mediated communication (CMC). Current systems (email and instant messaging applications) rely on explicit cues such as emoticons to convey the tone of a message. With an emotional instant messaging client, people could communicate more naturally, integrating both the content and the tone of the message through subtle cues. This could lead to fewer misunderstandings between users in cases where the message may be ambiguous, or where a specific tone (e.g., sarcasm) may be missed.

For any of these emotional state applications to be realized, there first needs to be some method of detection and recognition of particular affective states in users. The term *affect* can be understood as the physical experience of feeling, which we interpret and experience as emotions. The area of Affective Computing [45] is mainly concerned with providing computers with an emotional capacity. This includes the ability to recognize emotions as in the mission-critical monitoring example as well as the ability to express emotion like in the emotional chat client example. We focus on the computer's ability to recognize emotions in this thesis.

In recent years, there has been an increasing amount of research looking into different methods of detecting user affect [28,31,38,37,34,46,52]; however, current solutions are limited in a number of ways. First, many of the current methods of measuring affect require specialized sensors that are directly affixed to the user's skin or body [38,37]. The intrusiveness of these methodologies causes difficulties for two reasons: the fact that the user knows that they are being measured could alter their affective state undesirably; and it is unlikely that attaching sensors directly to the skin will happen in a real-world office or home context.

Second, the current methods of measuring affect can be very expensive because they utilize specialized equipment that is uncommon in home or office environments. This limits the real-world applicability of any affective solution as this equipment is not as widely used as standard computer equipment. For example, studies based on determination of different user states through thermal imaging avoid the problem of invasive sensors because the user's image can be captured without the user realizing it [46]. However, these techniques still require the use of specialized equipment that is both expensive and non-standard in the home or office.

1.2 SOLUTION

Our solution is to identify particular affect states by analyzing the differences in the user's typing rhythms, an area of research known as keystroke dynamics.

This research was encouraged by keystroke dynamics research in authentication systems where users gain access to computer systems by providing the password as well as the correct typing rhythm of the original user [43]. Monrose and Rubin observed that the user's affective state actually interfered with identifying participants due to changes in their keystroke rhythms. In our research, we attempt to exploit these changes in keystroke rhythms to see whether they correspond to particular affective states of the user.

The advantage to identifying affective states using keystroke dynamics is that it avoids some of the previously-described issues found in affective state determination research such as the expense, intrusion and use of specialized hardware. Keystroke logging is very unobtrusive to the user and is undetectable by the average user without the aid of special computer programs. This is advantageous when measuring emotional state as it should reduce the interference effect of our data collection on the user's true affective state. Keystroke dynamics also has the advantage of using the common keyboard which is inexpensive and ubiquitous on most computer systems. Identifying affective states through keystroke dynamics could allow us to implement affective computing solutions using standard equipment that is currently available on a large scale.

1.3 STEPS IN THE SOLUTION

The development of an emotional recognition system using keystroke dynamics requires a number of steps. We first need to gather a large amount of emotionally-labeled typing data. Then, we need to extract the relevant keystroke features, and build and validate models of emotional state. In our research, we used a field study to gather emotionally-labeled data as the user performs their daily activities. From this data set we extract keystroke features and use answers from an affective questionnaire for the ground truth in supervised machine learning classification.

1.3.1 Experience Sampling Field Study

Our study differs from that of other studies [26,21,28,31,34,38,37,46,57] in that we do not try to induce our participants into particular emotional states. Our emphasis in this research is ecological validity so we decided to use a field study to collect our data using an experience-sampling methodology, which asks participants to record their experiences or feelings in real time in their real daily activities [22]. The purpose of this methodology is to gather temporal feelings 'in the moment' rather than retrospectively, and to gather real world data instead of inducing emotional states in a lab. This type of study introduced some unique benefits and disadvantages, which will be discussed in Chapter 6. The data collection software was installed on participants' computers for a period of 3 weeks. The participants were free to use their computer as they normally would, but were asked to fill out an affective state questionnaire when the software prompted them. The affective state questionnaire contained questions on 15 different affective states: anger, boredom, confidence, distraction, excitement, focus, frustration, happiness, hesitance, nervousness, overwhelmed, relaxation, sadness, stress, and tiredness.

1.3.2 Data Collection Software

In order to collect the necessary keystrokes, we developed software that ran as a background process and recorded keystrokes as they were entered regardless of which program was currently being used. This allowed the participants to carry out their daily computer activities without requiring them to type into a specific program. Due to the sensitivity of the data collected, a

number of keystroke-specific features had to be considered in our software to ensure participants' privacy and the validity of the data entered. These features are described further in Chapter 4. Periodically, based on the activity of the user, the participants were asked by the software to fill out a short self-report on their current emotional state. The keystroke data as well as the answers to the self-report were then collected and stored for further processing.

1.3.3 Post Processing and Feature Extraction

Upon completion of the field study, we collected all the data from the server, performed data cleaning, and identified the features that we wanted to extract from the raw keystroke data. Features were then extracted using extensive processing of the raw data. Due to the large number of features that were extracted, we then needed to perform attribute selection to reduce the number of attributes in order to facilitate the machine learning classification process.

1.3.4 Model Building

Decision trees were used to create our classifiers with the features that were extracted from the previous step as input, and the answers from the affective state questionnaires as target classes. Each of the 15 emotional states that we collected was trained individually. We also identified a number of different variations that we trained on each emotional state, which are discussed in greater detail in Chapter 5. In total, 376 distinct classifiers were trained to account for the different variations identified. To evaluate the predictive performance of these models, we used 10-fold cross-validation on our dataset.

1.4 CONTRIBUTIONS

There are four main contributions presented in this thesis. First, we present a methodology that can be used for creating affective user models based on keystroke dynamics. Second, we describe an experience sampling field study that focuses on ecological validity when measuring affect. This study is unique in that we measure affect *in situ* and without artificially inducing emotions. Third, we created classifiers for 2 levels of 2 affective states (relaxed and tired) with

classification rates of 79.5% and 84.2% respectively. Fourth, we have identified other affective states that show potential given a larger sample size.

1.5 THESIS OUTLINE

In the remainder of this thesis, we will provide a discussion of related work and describe our experiment, data analyses, and results in detail.

In Chapter 2 we present a survey of the related literature that formed the basis of the research presented in this thesis. We first focus on the related research in assessing affect, as well as the technology used in affective state measurement. We then present research on keystroke dynamics as well as similar solutions that combine affect recognition and keystroke dynamics.

Chapter 3 describes the first half of our methodology – the data collection process. We present the details of our experience sampling study as well as the software that was developed for data collection. We present the particular features of the software that were implemented to ensure the privacy and anonymity of the participants of our study, given that there were a number of special considerations due to the sensitive nature of the data being collected.

Chapter 4 presents the second half of our methodology, the data processing on the raw data that was collected during the study. This included feature and class extraction as well as various other data points that could be used during the analysis and model building process.

Chapter 5 describes the numerous combinations and variations of processing that were used to analyze the data. We describe how we reduced our large feature set and performed data cleaning. We also describe the different machine learning techniques that we used in creating our classifiers. We finish the chapter by presenting the results of our analysis, identifying the best classifiers as well as the emotional states that show potential for further investigation.

In Chapter 6, we discuss the outcomes of the results presented in Chapter 5 focusing on generalizing the results of the different variations. We discuss some of the lessons learned from

performing this research as well as the advantages and disadvantages of using an experience-sampling approach to data collection.

Chapter 7 summarizes our research, identifies the contributions and discusses possible future directions that this research has revealed.

CHAPTER 2

RELATED WORK

In this chapter, we present the related research that informs our work. We start by reviewing the terminology commonly used in the research on affect and the methods that have been used to measure affect. We present the previous research in keystroke dynamics that inspired our work, as well as some of the initial research that has been performed in Affective Computing, using keystroke dynamics.

2.1 AFFECT, MOOD, AND EMOTION

In this section we introduce some of the common terms used in the literature as well as some of the different ways that these terms have been described.

2.1.1 Terminology

The terms *affect*, *mood*, and *emotion* can be confusing and are often used interchangeably; however, it is important to understand the distinction between these terms. In this thesis, we use *affect* in a more general sense that encompasses both mood and emotions [14]. *Moods* are subtle, long in duration, and are usually spoken of in general terms. The subtle nature of moods can mean that they go unnoticed to the person experiencing them until their mood is brought to their attention. In contrast, *emotions* are usually reactionary; they are often triggered by some particular cause either physical or cognitive and are short in duration. Also, the individual is usually aware of the presence of an emotion [45].

There are two different approaches that have traditionally been taken in describing emotion: those that emphasize the cognitive (mental) aspects and those that emphasize the bodily (physical) aspects. The cognitive approach has been attributed to Walter Cannon who suggested that emotion is experienced within the brain, independently of the sensations of the body [7]. The physical approach focuses on the physiological response (e.g. elevated heart rate) that occurs just prior or during an emotional episode; this approach has largely been attributed to William James [45].

Recent approaches see emotion as the combination of these two aspects (cognitive and physiological) simultaneously contributing to emotion. Thoughts as well as changes in body chemistry alone can cause emotions to arise in individuals [45]. For example, Schachter suggests that emotion is the result of our interpretation of our bodily responses and our situation, which we attach a label to (e.g. fear) [48]. When we refer to emotional *state* in this thesis, we mean the internal dynamics (both cognitive and physiological) that are present during an emotional episode and we describe the emotional *experience* as what an individual perceives of their emotional state [45].

2.1.2 Describing Emotion

There are two approaches in which the related research describes emotions: by using discrete categories of emotion or by using a continuous dimensional approach.

2.1.2.1 Discrete Categories

The categorical approach is based on how we describe emotions through language; we typically give specific labels to different emotional episodes. Examples of such labels (or categories) include happiness, anger, indignation, contempt, nostalgia, satisfaction, and sadness. In fact, there have been a number of different categories that have been suggested; the variability and disagreement in the literature suggests that clear definitions or boundaries are lacking for these states, which has caused difficulties when comparing different research approaches. The definition of these categories vary not only within a language but also across languages. Specific

categories may or may not exist in other languages also making research using this approach difficult [61].

2.1.2.2 Continuous Dimensions

The dimensional approach described by Russell in [47] describes the idea of *core affect* which is central to emotion and mood. Core affect can account for overall lasting feelings (e.g. moods) as well as immediate feelings triggered by specific events, and is described as being composed of two independent dimensions: *arousal* and *valence*. Figure 2.11 illustrates the concept of core affect using some of the common emotion categories for better understanding of core affect.

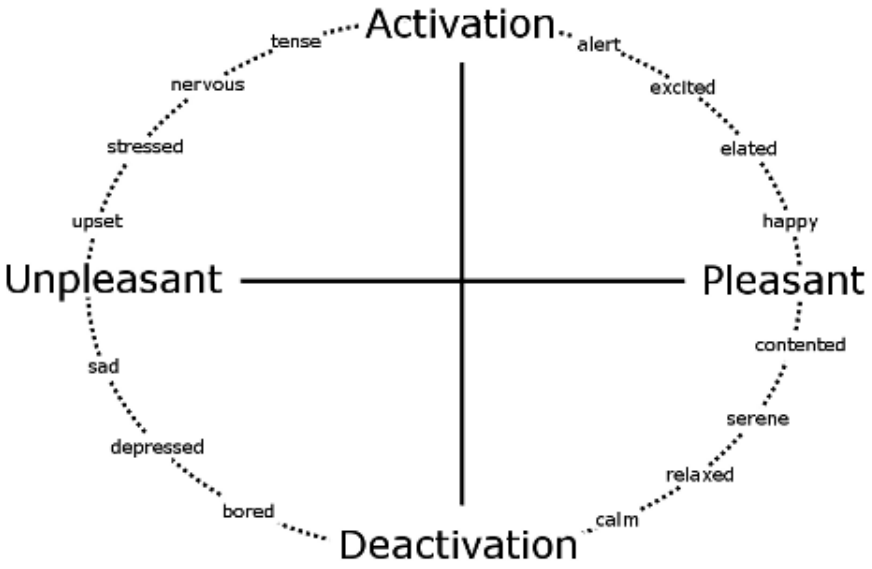

Figure 2.11 The two-dimensional core affect with arousal along the vertical axis (activation – deactivation) and valence on the horizontal axis (unpleasant - pleasant) [47].

Arousal refers to the sense of mobilization or energy and is sometimes referred to as the degree of activation of an individual. This concept originated from Cannon's theory of the "fight or flight" response [50] and focuses on the physiological changes that occur in the body during these situations. Typically, arousal is described in terms of low arousal (e.g. sleepiness) to high

arousal (e.g. excitement). *Valence* summarizes how an individual is feeling based on pleasure (positive valence) or displeasure (negative valence).

2.1.2.3 Using Discrete Categories and Continuous Dimensions

In our research, we used both the categorical and dimensional approaches to create our models. Initially, we used a categorical approach to gather the subjective emotional experiences of our participants and created models for each one of the emotional state responses. We gathered the information using a categorical approach because we needed a way to gather information about the user's emotional state in a language that they could understand (emotional categories rather than the degree of arousal or valence). We did not want to use a data collection process that would require the participant to relearn the terminology every time they had to fill it out. We also created arousal and valence models by mapping the original emotional categories to a dimensional model as in Figure 2.1 which we discuss further in Chapter 4.

2.2 RECOGNIZING EMOTIONS

There are many different emotional indicators that have been studied to determine affect including facial expressions, gestures, postures, vocal intonation, language, pressure, and pupil dilation [45]. These are all visible features that can be observed by others through day-to-day interactions. For example, in human-to-human interactions, facial expression can help us to determine whether someone is distracted, frustrated, or happy just through facial expression. Some researchers have used sophisticated face-tracking software to analyze facial expressions to infer the emotional state of the user [11,44]. Work by Khan et al. [31] extended this idea but used thermal imaging to identify changes of blood flow patterns in the face that correspond to different facial expressions.

There are also a number of other indicators that are less visible to another person, such as physiological changes in the body that occur during emotional episodes. In [38], Mandryk et al. used physiological metrics such as galvanic skin response (GSR), respiration,

electrocardiography (EKG), and electromyography of the jaw (EMG) as indicators of participants' affective states while playing video games. These indicators are measured through electronic sensors placed directly on the skin, face, chest, and hands of the participant.

To get an idea of how these types of sensors work, we briefly present two of these measures (EKG and respiration) in more detail.

EKG measures the electrical activity of the heart which is measured on the surface of the skin using electrodes. These electrodes are commonly placed on the chest, forearm, or legs and are applied with conductive gels on the bare skin of the person being examined. The area where the electrodes are placed must be free of hair so shaving these regions may be necessary to prevent interference with the sensors [50].

Respiration measures the rate or volume of air exchange in the lungs. Although accurate results can be obtained by measuring gas exchange, the apparatus that is required (a face mask) prevents the user from speaking and requires them to remain stationary during the measurement process. Alternatively, measuring chest cavity expansion can also be used for these metrics using less obtrusive sensors (e.g. stretch sensor around the chest of the participant) [50].

The major problem with the physiological approaches to measuring affect is the intrusive nature of the technology. Affixing sensors to users' skin would not be realistic in a real-world context (e.g. casual interactions with mobile phones). Sensors take time to attach to the user, conductive gels might be used, shaving may be necessary, and the sensors can be sensitive to movement and could fall off with activity. Furthermore, the presence and constant reminder of the sensors may alter the emotional state that the user would have been in, if the sensors were not present.

Some physiological approaches to detecting emotional states are less obtrusive because they do not require physical contact with the participant. For example, thermal cameras have been leveraged to identify increased blood flow in particular regions of the face when the user is experiencing emotional states such as stress [46]. Although this type of technology is not as obtrusive as some of the other physiological approaches (such as GSR), the main drawback is that the approach requires the use of expensive technology (a thermal camera) that is not widely

used in typical computer settings such as the home or office. This is also typical of the other physiological sensors previously mentioned (e.g. GSR) because they require specialized equipment that can be expensive, which limits the applicability of widespread adoption of this approach.

2.3 EMOTIONAL EXPERIMENTATION

There are different ways of collecting data on emotion; each have their own advantages and disadvantages. Here we discuss the most common approach in laboratory settings where moods are induced into the participant being studied in order to observe and collect data. We also discuss collecting data in a more naturalistic setting, capturing emotions as they occur naturally in participants, in a less influenced manner.

2.3.1 Laboratory Settings

One commonly used technique in studies on emotion is the use of mood induction in laboratory studies. A mood induction procedure (MIP) is an experimental technique devised to establish a particular mood in a subject. Westermann et al. lists nine different categories of MIPs from the literature: imagination, Veltren, film/story, music, feedback, social interaction, gift, facial expression, and combined MIPs [59]. We present two of these (Veltren and film/story) in more detail to get an idea of what mood induction entails.

The Velten MIP is the most widely used technique for mood induction and it uses a self-reference-statement technique. Subjects are presented with statements that are positive, negative, or neutral depending on the target mood desired. Subjects are instructed to feel the moods described by the statements. Examples of the positive, negative, and neutral statements that Westermann et al. used are presented in Table 2.1 below.

Table 2.1 Examples of positive, negative, and neutral statements used in Velten [55] to induce elation, depression, and to serve as a control respectively.

Positive	*"If your attitude is good, then things are good, and my attitude is good."* *"This is great-I really do feel good-I am elated about things."*
Negative	*"Every now and then I feel so tired and gloomy that I'd rather just sit than do anything."* *"I have too many bad things in my life."*
Neutral	*"This book or any part thereof must not be reproduced in any form."* *"Utah is the Beehive State."*

Another MIP, although less widely used, is the use of film or story to induce subjects into particular moods. In this technique a descriptive narrative is presented to subjects. This narrative can be either a short clip from a movie or a detailed story that helps the subject identify with the protagonist [59]. Each clip or story is selected according to the desired target mood. In [21], Hancock et al. induced a negative affective state through the use of a short clip from the film *Sophie's Choice* where a mother is forced to give up her child to the Nazis.

In the film/story MIP, subjects are either provided with or without instruction during the induction procedure. When instruction is used, the participant is explicitly asked to become engrossed and imagine how it would feel in that situation. Film and story MIPs have been seen to be to be the most effective technique for inducing both positive and negative mood in subjects [59].

A few concerns have arisen with the validity of MIPs. As previously implied with film and story MIPs, different induction techniques have varying success rates. For example, facial expression MIP was found to have a success rate of 50%. In this case, it would be necessary to use twice as many participants in a study than would otherwise be required so that the targeted number of participants is achieved. The administration time that is required also depends on the MIPs that are used, and can range between 7 minutes for an individual to 55 minutes; it varies depending on the technique used. Group administration could alleviate this time expense; however, individual testing has been used extensively whereas group administration has not. In a group setting there could be issues with particular subjects being inhibited from entering particular moods when being observed by other participants. Individuals can also react differently to mood

induction techniques and the demand characteristics of the experimental situation could influence the subject. In other words, the subjects may guess the type of mood that was desired by the experiment and artificially adjust their reactions towards that mood in an attempt to please the experimenter [39].

2.3.2 Naturalistic Settings

There are a number of different approaches that can be taken to observe subjects in their natural setting including self-report recall surveys, time diaries, direct field observation, and experience sampling. Self-report recall surveys and time diaries require the subject to record their experiences after they have occurred. Drawbacks to self-report recall are that participants can suffer from recall issues (they might not remember how they felt or what they did) and reporting bias due to the subjective nature of data collection. Direct field observation can provide a more objective viewpoint; however, administration can be time-consuming, costly, and may interfere with the subject's performance [24]. In addition, people are good at masking their true emotional state and thus their actual mood may not be observable.

The experience-sampling methodology (ESM) is a technique that is used to collect individuals' experiences (thoughts, feelings, sensations) as they occur in situ as well as the overall context of these experiences. Subjects provide responses (qualitative and quantitative) to questionnaires at random times throughout each day of observation. Typically, a signaling device is used (a beeper or handheld computer) that notifies the subject to record either their current experiences or their experiences since the last data collection period. The questions asked can vary and are customized to the particular research goals being studied, but they often contain questions about the subject's physical context (where they are), social context (who else is around), activities, thoughts and feelings [22].

Although the ESM shares characteristics of other methods, what distinguishes it from other methodologies is that it captures daily life as it occurs from moment to moment. This allows for examination of the changes that occur at different moments and facilitates identification of factors that may have influenced these changes. ESM minimizes the recall problem in

retrospective techniques because the subject is able to describe things as they happen or soon after they happen. ESM emphasizes the ecological validity of naturalistic observation and is less unobtrusive than diaries [22].

As with any approach, the ESM has disadvantages too. Depending on the frequency of the sample period, the interruption to subjects' daily activities can be burdensome and could lead to selective non-compliance [22]. Another limitation is the reliance on subjective self-reports to gather information [10]. As with any subjective measure, individuals may be biased, forgetful, repress certain information or change their response to fit with the social norms of the participant's culture [22].

Despite these limitations, ESM provides the opportunity to collect detailed accounts of participants' daily lives that would otherwise be difficult to obtain. Over the past 30 years, ESM has been used in a variety of studies; however, we are only interested here in the studies on affective measurement using ESM. In one such study [41], Moneta and Csikszentmihalyi used the ESM to test the relationship between an individual's skill-level and challenges to study the experience of *flow* or the experience of enjoyment one realizes when the appropriate balance of challenge and skill are met when performing an activity. Some of the variables that were measured were the participant's concentration, involvement, happiness, and desire to perform the activity.

We used an experience sampling methodology to collect keystroke and affective data from users as they performed their daily tasks. Mood induction was not used in this study because we wanted to test a wide array of emotions over a relatively long period of time. Experience sampling allowed us to collect our data with minimal interference during the collection process and across multiple naturally-occurring emotional states. This approach was similar to the approach used by Kapoor and Horvitz where predictive user models were created out of data collected using an ESM [29].

In the next section, we move away from the research on affect and focus on the related research on keystroke dynamics.

2.4 KEYSTROKE DYNAMICS

Keystroke dynamics is the study of the unique characteristics that are present in an individual's typing rhythm when using a keyboard or keypad. Keystroke dynamics research typically involves inspecting timing characteristics of individuals' typing in order to identify patterns in their keystroke data. This typically includes the analysis of characteristics such as duration of a key press or group of keys and the latency between consecutive keys (i.e. time elapsed from one key to a subsequent key). Timing features are the cornerstone of keystroke dynamics; however, there are other features that are often used in conjunction with these such as the content of the keystrokes (what the user is typing) and the application context (what program the user is typing in).

We illustrate how keystroke dynamics works through a simple example of identifying expert typists in a group of both non-expert and expert typists. In a group of typists, it would be reasonable to assume that expert typists are quicker at typing and have fewer errors than non-experts. We could identify those users that have the shortest timing features in the text. These features generally relate to either key duration (the time elapsed for a single key press) or key latency (the time from the release of one key to the next key press). From this set of users (with short timing features) we would then identify those users that made the fewest mistakes when copying a piece of text by comparing their keystroke characters with the original data. This would produce a range of typing abilities for each user, which could then be assigned into different proficiency groups resulting in the identification of non-expert and expert typists.

The previous example focused on identifying groups of people, but just as in handwritten letters and signatures, the way that individuals type can be distinctive too [3,2,5,6,9,12,16,18,20,27,42,43,49]. For instance, an individual could become adept at typing a certain small piece of text, such as their own name, resulting in a quick succession of keystrokes, but that same person may struggle to enter other keystrokes such as numbers or punctuation. Other factors such as finger dexterity [16] and the type of keyboard [56] could also affect these timings. The end result would be a rhythmic cadence of keystrokes for an individual, something

that would be very difficult for another user to replicate. To identify particular users, you would need to recognize their unique keystroke pattern amongst other keystroke patterns.

We try to identify emotional states from keystroke data by looking for patterns of keyboard usage that correspond to the particular emotional state of the user on various occasions during normal computer use.

2.4.1 Pattern Recognition

Our basic problem is that of pattern recognition. There are three main aspects in pattern recognition problems: the *representation* of the data to be analyzed, the *feature extraction* process, and the *classification* of the data into different categories [3].

In keystroke dynamics, the typical input data representation is the raw keystroke events that occur when the user types on the keyboard. When the user presses a key, a key down event is created in the computer's operating system and when the user releases the key, a key up event is created. These events are then captured by a software program, such as a key-logger, and each collection period is referred to as a *sample*.

Feature attributes are then extracted from the representation data set. These features vary greatly between studies; however, in keystroke dynamics typical features include keystroke duration and latency. We included these features along with a number of types of non-keystroke features in our pattern recognition approach which we describe in detail in Chapter 4.

In this thesis, we refer to *features* as the selected attributes that were used as input to the model building process. We also refer to *data points*, which are attributes in the data set but differ from features in that they are only used to describe the data set and to filter or separate the data into different sub-sections for analysis. For example, the duration of a specific key would be considered a feature whereas the sex of the participant (data point) could be used to separate the data into male or female sets if needed.

There are many different approaches to classification in pattern recognition problems. In keystroke dynamics, there have been a number of different approaches taken to build models

including neural nets [6,9], distance measures of feature vectors [27,42,43], decision trees [49], and various statistical approaches [2,12,16,18,5,43]. We decided to use a decision tree algorithm (C4.5 version 8) which is a *supervised* machine learning approach that uses known classes (emotional states in our research) to create a *model* or *classifier* from the keystroke features [60]. The process of creating models from the feature set is called *training*; once complete, the model can be used to describe the data set or to predict outcomes with new data sets (keystroke data).

2.4.2 Keystroke Dynamics Background

The idea of using an individual's typing rhythm as a form of identification was originally noticed upon the wide-spread adoption of the telegraph for communicating across long distances. Upon widespread adoption of the telegraph, experienced telegraph operators were noticed to have unique signatures that they used when sending messages. There are reports that by World War II, United States Military Intelligence identified and exploited this unique quality in messages sent via Morse Code using what they called "The Fist of the Sender". They used this idea to identify the original operators based on the unique rhythms that they supplied; this aided in tracking German telegraph operators [40,54].

From these beginnings, similar approaches have been applied to computer keyboards and keypads in computer security for user authentication and intrusion detection.

2.4.2.1 Authentication & Intrusion Detection Systems

Interest in keystroke dynamics was revived after a 1980 study by Gaines et al. [16] illustrated that individuals were seen to have a unique signature when they typed on a computer terminal. Gaines et al. suggested the use of keystroke dynamics as a method of user authentication when logging on to a computer terminal. In their approach [16], users were identified by their username and passwords as well as their typing rhythm by entering in a fixed piece of text. The keystroke rhythms were then analyzed, comparing the supplied pattern to a previously constructed template (also referred to as a profile or model in our case). Computer users had to supply the correct user-name and password along with the correct rhythms for that user's template.

In the 30 years since Gaines et al.'s initial research, there have been many different approaches that further refine this authentication process. The studies vary greatly on the classification algorithm, selected features, sample collection, and other experiment design factors. We will present each of these different approaches in the following sections starting with section 2.4.3 where we present some of the different terminologies that were seen in the related literature.

In authentication systems, keystroke dynamics are considered a biometric; a physical or behavioral characteristic used to identify an individual. Physiological biometrics are normally considered stronger than behavioral ones because they are fairly consistent over time and are unique across large populations. Examples of physiological biometrics include patterns found in the iris, face, finger print, hand geometry, vascular layout in the hand, wrist, or face. The difficulty of using these types of biometrics is that they are expensive (e.g. iris scanners) or require specialized equipment [25].

In contrast, behavioral biometrics are considered weaker than their physiological counterparts because they are less stable and can vary over time. For example, an individual's signature (a behavioral biometric) could adapt and change throughout the course of that person's life. Any system that uses behavioral biometrics would also have to adapt to these changes. In keystroke dynamics, there have been many reports of variability in authentication results due to both physiological (finger dexterity) and psychological factors such as stress and fatigue [27,43]. These indicators from early authentication research influenced our work in trying to identify such states when they occur.

Although behavioral biometrics have unstable qualities, they are still successfully used because it is still difficult to control or imitate others' behavior such as the intonation of one's voice, the style of handwriting, and typing rhythm. Behavioral biometrics such as handwritten signatures, have a long history of identifying individuals as well as imposters [42].

2.4.2.2 Commercial Products

The interest in using keystroke dynamics for authentication purposes resulted in the development of a number of commercial products such as those offered by Admit One Security [1] and Type

Sense [53]. These companies are currently offering keystroke dynamics in conjunction with traditional authentication techniques as a form of multi-factor authentication for workstations and web interfaces. As of this writing, there were at least 11 companies selling products using keystroke dynamics for authentication and 3 United States patents according to [30].

2.4.3 Terminology

In this section, we present some of the different terminology used in the related literature in keystroke dynamics. In particular, we look at the differences between static and dynamic approaches and the differences between models based on different types of user keystrokes (fixed and free text data collection). We define these terms and what they mean in the context of the authentication research as well as our research in emotional state recognition.

2.4.3.1 Static and Dynamic Text

In the related research, a distinction is made between *static* and *dynamic* approaches to collection and classification of keystroke data. *Static* approaches [2,3,5,9,16,18,19,27,42,44,49,56] asked all users to enter the same fixed piece of text (usually multiple times) during the data collection process. Authentication would then be attempted by the user entering the same fixed text that the model was built on. If the provided keystroke timings were similar enough, the user would be authenticated. In contrast, *dynamic* approaches [12,17,18,42,43,56] do not use the same text for collecting training data as used in testing.

This distinction of static and dynamic approaches is important due to the implications that it has on the end use of these systems. Authentication systems based on static text models can only be used at the time when the user logs in because there would be no way of continuously checking the user's credentials (continuous monitoring) during a session without prompting the user to re-enter the static text used to authenticate. Continuous monitoring is required by intrusion detection systems (IDS), security programs that continuously monitor computer usage by users and software. To be able to perform any type of continuous monitoring, the classifier would have to work on text that it was not initially trained on.

Continuous monitoring would be beneficial for detecting emotional states of users. One of the main benefits of using keystroke dynamics for emotional state recognition is that the input (set of keystrokes) is continuously available during a computer session [2]. This presents opportunities to determine the user's emotional state at any time during the computer session. The mission-critical monitoring example presented in Chapter 1 would require a system that could classify emotional states using keystrokes that were not available during training (dynamic text).

Although continuous monitoring would be the best application of our research, good static text results would help us identify potential emotional states that could provide strong indicators of affect. As with authentication systems, good static text results could lead to good results for models that could handle dynamic text and therefore make continuous solutions possible.

There has been much disagreement over the terms static and dynamic in the literature. Previous systems have been developed that claim to handle dynamic text but they either use a small set of predefined fixed text that the user can choose from or they use early authentication (attempting to authenticate the user before the entire fixed string was entered) [17]. Both of these methods are still just variations on the static approach and would not work in a continuous monitoring scenario. For continuous monitoring to be fully realized, the model must be able to handle any text that the user enters (dynamic text) [2]. Due to the confusion over terms, like Gunetti and Picardi [17], we use the terms *fixed* text for static, predetermined, text and *free* text for dynamic, unrestricted text throughout the remainder of this thesis.

2.4.3.2 Fixed and Free Text

Fixed text refers to any model that is built with a piece of text that is later evaluated using newly collected keystrokes of the exact same text entry. Free text is completely uninfluenced text; it is defined as any text that the user can enter during their typical keystroke interactions. Most of the literature on keystroke dynamics so far has been using fixed text (static) approaches [3,2,5,9,16,18,19,27,42,43,49,56] with only a few using free text (dynamic) approaches [12,17,18,42,43,56].

Fixed text studies typically involve the participant entering keystrokes into a text-box during the authentication phase. There are many different approaches to using fixed text; some studies used the participant's full name as the training text [5,27], and other variations gave participants a choice of a few different phrases [42]. The main aspect of the static approaches was that models were trained on the same text that they were later tested on.

It should be noted that free text does not imply that the keystrokes were obtained unobtrusively. In some free text studies, keystroke data was gathered by providing an open-ended text-box in which the user could enter any text they would like (with the exception of repetitive phrases or 'junk' input) as in [17]. However, depending on the activity of the user and the amount of text entry required, this could be taxing because the user would have to first think of something to type and then enter it.

Alternatively, Dowland and Furnell monitored all of the user's keystroke activity as a background process while the participant used their computer on their own daily tasks regardless of the application that was currently running [12]. This method has three benefits, data could be obtained unobtrusively, the user would be less influenced by the collection method, and it could reduce the cognitive load on the participant by removing the requirement of thinking of the text that they will enter. We also used this method of data collection because it was well-suited to our focus on ecological validity because the participants would be under less influence for our free text data collection.

Free text approaches have had low classification rates when compared to fixed text approaches [43]; however, recent studies in free text show promise in being able to provide good classification rates provided the sample is of sufficient length [18]. User authentication using free text has been shown to identify individuals even if they are typing in a different language than samples that the model was created with [18]. In [18], Gunetti et al. built classifiers that could identify users when they were typing in either English or Italian. They concluded that as long as the different languages have enough similar valid digraphs for each language, that it would be possible to identify individuals by their keystrokes.

2.4.4 Keystroke Features

The most common features that were used in the related research were timing features. These features included calculations based on individual keys as well as multiple keys.

One common single key feature was *key duration*, the time that the key was depressed by the user. This was calculated by finding the duration of the keystroke from when the user presses a particular key (the key down event) until the release of that same key (the key up event). Note that it could be possible for many keys to be depressed at the same time; this must be taken into consideration as a particular key's up event may not directly follow that key's down event. The key duration has also been extended to groups of consecutive key characters or *graphs* [3].

Digraphs contain two consecutive keystrokes, whereas trigraphs contain three; this continues for any number of combinations, which creates n-graphs. Using this terminology, the word 'emotion' would have six digraphs ('em', 'mo', 'ot', 'ti', 'io', 'on') and five trigraphs ('emo', 'mot', 'oti', 'tio', 'ion').

A common multiple key feature is *digraph latency*, or the time elapsed from one key being released to the next depressed key. For the digraph 'em', the latency would be from the time that 'e' was released to the time that 'm' was depressed. Note that digraph latency can be negative if the first key is not released until after the second key is released [6].

Features based on digraphs have been used [2,12,16-18,27,56] in authentication research, with only a few studies using trigraphs or larger graphs [3,2,17]. Gunetti and Picardi found that they achieved better classification rates with larger graphs; however, as graphs become larger, it is less likely that they will appear in the training samples or the text that is being tested [17]. This would be more of an issue in free text data where the text that users enter is not controlled in any manner.

One of the advantages of only using timing features is that privacy can be maintained as there is no need to process the characters that the user types, only timing values are viewed. This helps maintain the user's privacy; however, the drawback of this approach is that we are essentially

throwing away data that could otherwise help us identify particular emotional states. In [57], Viser uses a number of content features based on the textual information produced by the user's keystrokes including specific types of words, keystroke frequencies, and timing characteristics. In Chapter 6 we discuss how this data could be used in conjunction with keystroke dynamics to help identify different emotional states.

In addition to timing features, some studies [12,5] captured the active application for each keystroke. This would allow the data set to be dissected into different categories of user activity, which could help identify problems of low classification in specific activities. In [12], the entire window title was recorded to gather this contextual information; however, we realized that the application title may introduce privacy concerns because it can contain sensitive information (e.g. email subject lines can appear in the window title). Instead, we opted to include only the name of the process with each keystroke since this provides sufficient contextual information while still protecting the user's privacy.

2.4.5 Classification

There have been many different approaches to classification taken in the keystroke dynamics literature. In this section, we start by describing the different approaches to the training sample size and explain the metrics that were used for validating the different models. This section is concluded with an overview of the methods that others have used for classification.

2.4.5.1 Training Sample Size

The size of the keystroke samples that were collected from participants varied greatly in past keystroke dynamics studies. Sample text ranged anywhere from a few short words such as a participant's full name [5], to a few phrases [3], to full pages of text [16].

In [3], Bergadano et al. suggested that the longer sample texts create better performing classifiers; however, they also suggested that a carefully selected sample text could be used to create models that work just as well as long sample passages. They suggested that the issue would be how many different digraphs are present in the sample text. As long as the sample text

had many different digraphs, similar results to long sample passages could be obtained using this shorter, carefully-selected text. Ultimately, we did not use this approach due to the length of time it would have taken to create the text for each of our 64 different fixed text samples. A large number of these samples were desired because we wanted to avoid learning effects that may occur.

The implications that the sample text length has for real-world authentication systems would be that a very strong authentication system could be built using long sample text; however, usability would suffer due to the extended time needed to train a model as well as the added time to authenticate. There would have to be a trade-off between usability of the system and the strength of the security.

In continuous monitoring applications (e.g. intrusion detection systems), the more keystrokes that an imposter could enter without getting caught, the greater the risk to the system's security. Imposters could quickly inflict damage with only a few keystrokes; the command 'rm –rf /' is only 8 characters long but could delete the entire directory system in a Unix based computer if run as a user with sufficient privileges. Furthermore, since free text can be entered by the user, systems using only common English digraphs may not provide very much security at all. In the previous example, the command 'rm –rf /' uses only two English digraphs because only 4 of the characters are alphabetic. The user could quickly execute this command and do considerable damage to the system.

Similar examples could be imagined for our research in affective computing. Using the mission-critical monitoring example that was presented in Chapter 1, the longer it would take for the system to recognize that a user was fatigued, the greater the risk of dire consequences occurring due to fatigue. Another possible side effect could be that the longer it takes to recognize a particular emotional state, the increased chance that the user's emotional state may have changed.

2.4.5.2 Model Validation

Before we discuss the different classifiers that were used in the keystroke dynamics research, it is important to understand how these models were validated.

The authentication research borrows two security metrics to validate their models, the False Alarm Rate (FAR) and the Imposter Pass Rate (IPR). The FAR is the percentage of instances that a legal user was misclassified as an imposter. The IPR is the percentage of cases in which an imposter is able to pass as a legal user. Small percentages for both the FAR and IPR is desirable; however, both variables are dependent on the other so decreasing one will increase the other [2].

This type of validation using the FAR and IPR does not lend itself well to our research. These metrics imply that there were only two states (target classes), the user was either valid or an imposter. In our research, there were five possible target classes per emotional state model that were trained (i.e. each state had five responses ranging from strongly disagree to strongly agree). In the *Target Classes* section of Chapter 5 we discuss how the number of classes affects the meaning of classification rates such as the FAR and IPR.

Furthermore, the FAR and IPR do not apply to our research because the problem domain is different; there are no imposters in our applications. Instead we choose to validate our models based on the correctly classified rate and the Kappa statistic from our ten-fold cross-validation. We explain the details of our validation procedure and what these metrics mean in the context of our research when we discuss our analysis in Chapter 5.

2.4.5.3 Classifiers

There have been a number of different classification methods that have been used to create keystroke models for authentication.

Neural networks have been used for creating keystroke models. In [6], Brown and Rogers used three different classifiers (two of which were different types of neural nets) using keystroke duration and digraph latencies and they achieved a 0% IPR and a FAR of 4.2% for one set of users and an 11.5% FAR for a different set; this was accomplished by adjusting the neural net's

parameters specifically to the sample set in order to obtain the 0% IPR. The overall FAR was a result of taking the best performing classification methods for each participant based on the sample data. However, as pointed out by Bergadano et al. [3], by adjusting the parameters ahead of time specifically to the sample and by taking the best case from a set of classifiers, an artificial scenario was created based on the specific examples used. In a real-world application, it would be unlikely that the model would perform that well given that the model was tailored to the sample data.

One of the disadvantages of using neural networks is that they have long training periods and they require re-training every time new data should be integrated [43]. Initially we tested a neural net approach and found that this was not a desirable option for us due to the opaque nature of the trained model, the retraining requirement, and the additional time spent training the model.

In [27,42,43,56], researchers used distance measures between the trained vector (model) and the model that was being tested. If the measures were within some predetermined threshold value, the user would be permitted to login. Three different distance measures were used including the Euclidean distance, a weighted probability measure and a non-weighted probability measure. Using these measures, Joyce and Gupta obtained a FAR of 13.3% and an IPR of 0.17%. Other studies reported classification rates (FAR and IPR not reported) of 83%-92% [43] and 93.3%-97.9% [56] depending on different subsets of participants used in training or different conditions in which the samples were collected. For example, [56] found that the accuracy of their models significantly decreases when participants used different keyboards (i.e. laptops versus desktop keyboards). In our study, we asked participants to identify whether the computer that they were using was on a desktop or laptop. This allowed for the possibility of separating the data set to improve the accuracy of the models.

In [2], Bergadano and Gunetti designed their distance classifier using both absolute and relative digraph features. Trigraph durations were inserted into an array and then sorted by length. They then calculated the degree to which the new sample was out of order from the trained model (the degree of disorder). The relative features were added to mitigate the effects of the user's current emotional state such as fatigue. The authors suggested that an individual should have the same

relative cadence in their keystroke rhythms when they are tired (only slower) compared to when they are not. They had a successful classification rate of 97%, but admitted that they required a long sample length in order to achieve these results. In [3], they extended their research, by using short text samples, and achieved accurate classifications higher than 90% when using less than a full line of text.

Although these results are impressive for user authentication and identification purposes, we believe that this relative positioning method would be inappropriate for our purposes in identifying affect. This is due to the use of relative positions of digraphs, which was introduced to remove the effects of physiological or psychological changes in the users; however, this is the exact situation that we want to identify in our research.

Decision Trees [49] have also been used in keystroke dynamics research. In [49], multiple parallel trees were used to obtain higher classification rates based on a majority decision from the group of trees. They were able to achieve a FAR of 9.62% and an IPR of 0.88% using a Monte Carlo approach to attain sufficient training data. This study defined FAR and the false reject rate (FRR) to have the exact opposite definitions from the keystroke literatures FAR and IPR. The FAR and IPR expressed here are according to our definitions in Section 2.4.5.2 on validation.

From the literature alone, it was difficult to determine which classification methods produced the best model due to the variability in experimental conditions. Studies used varying amounts of training data, training sample length, different data collection methods, participant skill levels, and number of participants. A common data set would help research in this area to more easily compare the performance of different classifiers. Such a data set has been offered by Bergadano et al. only recently and it is yet to be seen if this data set will be adopted in future studies [2].

2.4.6 Typing Errors

Regardless of whether the samples are collected using fixed or free text, typing errors can occur when collecting keystroke data. This would be a problem especially for keystroke models that can only handle fixed text. For example, the user could misspell the fixed text causing different

keystroke patterns that falsely misidentify one individual's keystrokes for another's. Many studies discard these errors or they prevent the users from correcting mistakes [6,9,16].

However, mistakes can occur throughout typical computer usage; any application that handles free text (continuous monitoring) would need to account for possible mistakes that the user makes. In [17,18], the authors kept mistakes data as it did not affect their results because their classification used relative positions of common digraphs in the text; if the digraph was not present, it would essentially be ignored.

For our research, typing errors could be important because they may help us identify the user's different emotional states. For example, a relaxed user may have fewer mistakes than a fatigued or stressed user who may have faster than normal keystrokes or may not be focused on the task at hand.

2.4.7 Novice & Experienced Keyboard Users

In the initial research, Gaines et al. focused mainly on professional typists [16], that is, secretaries with formal training. Authentication studies since then have included participants of variable typing skills [3,5,6,17,18,43].

Research suggests that even with the very fast typing speeds of expert typists (very small timing values); participants were still able to be successfully identified by their keystroke dynamics [16]. Others have found that the most inconsistent classifications come from models that were built for novice typists [3,5,17]. Novice users, such as those that use only the two fore-fingers while typing seemed to have very inconsistent typing patterns with long pauses in between keystrokes.

Some studies included an outlier removal process to account for differences in participants' typing abilities as well as other naturally-occurring behaviors such as long pauses or breaks in typing. Early studies such as [35] used a single low-pass filter to remove these outliers from the data. Unfortunately, one single filter does not perform very well across all participants [43] due to differences in typing abilities. For instance, when looking at the key duration of an expert

typist, the timing values (duration and latency) are likely to be shorter than those of a novice user. Outliers in the expert typist's timing values could resemble typical novice user values. Depending on the threshold used to remove outliers, the threshold will be either too specific (removing the valid novice users' data) or too general (with no outliers removed for the expert user).

To avoid this problem, some studies used separate threshold values for each participant to accommodate for each user's typing proficiency. One common technique that was used was to calculate the mean and standard deviation from an individual's typing samples, and then any data that was three standard deviations away was removed from the data set and the mean and standard deviations were then recalculated on the remaining data [9,27,49,56].

In [56], Villani et al. illustrated the importance of threshold removal by plotting different threshold values against the classification rates to determine the optimum threshold to use during their outlier removal process. Other studies provided different results by segmenting portions of their populations to handle these differences; for example, in [56] segmentations were performed based on whether the participant was using a laptop or desktop. However, as [2] suggests, these types of scenarios may be unrealistic in real-world authentication systems. They claim that by fine-tuning the model to the data, the authors over-fit the model to the training data set.

In order to prevent over-fitting, we took the approach of using ten-fold cross-validation in conjunction with decision trees, which allowed for the capability of reducing the complexity of the tree through pruning and adjustments to the minimum number of nodes per level. By not testing on the training data and reducing the complexity of the tree, we could increase the predictive performance of the model on new data sets [60].

2.5 AFFECTIVE COMPUTING AND KEYSTROKE DYNAMICS

The keystroke dynamics research presented in the previous section consisted of authentication and verification systems as this was the area where the majority of the research originated from. In this section, we look at the studies that combine affective computing and keystroke dynamics.

In [62], Zimmermann described a methodology that could be used to find correlations between user interactions (keyboard and mouse) and their affective state. In this paper, he described an empirical study where he used film clips to induce participants into various states along the arousal and valence dimensions. To determine the affective state of the participants, physiological sensors were used to measure respiration, pulse, skin conductance level, and corrugator (the small muscle that controls part of the eyebrow) activity. Participants were also asked, at different times, to self-assess their current emotional state using the Self-Assessment-Manikin (SAM), devised by Lang [32]. The authors found significant differences between the neutral state when compared to the other emotional states; however, they were unable to distinguish between the other four states that were induced.

A study [21] performed by Hancock et al., looked at emotional contagion between participants in computer-mediated communication. In this empirical study, three induction techniques were used to induce a negative (sadness and frustration) affect state: a video clip, music, and an additional task that was either easy or hard depending on the condition being tested. The emotional state of the participants was assessed using a questionnaire with 7-point Likert scale statements that the users were asked to complete. The authors created profiles based on linguistic patterns produced by the Linguistic Inquiry and Word Count (LIWC) software package which looks at word frequency along different psychological dimensions. They found that individuals could identify the negative affective state in their partners through text-based cues and that emotional contagion also took place between the participants in the negative affect condition.

Recent work in affective computing using keystroke dynamics has been performed by Visor et al. in [57,58] where keystroke and linguistic features were used to identify both cognitive and physical stress. In this empirical study, free text was collected and five different data mining

techniques were built using features that focused on the words used in the sample, timing and keystroke features. They achieved classification rates of 62.5% for physical stress and 75% for cognitive stress, which they state was comparable to other proposed affective computing solutions. The authors mentioned that although their results indicate that they can detect stress through keystroke dynamics using their empirical methodology, the effectiveness should also be tested across a wider range of typing abilities, cognitive/physical abilities, and keyboard types, as well as in real-world stressful situations.

Our research attempts to address the real-world (ecological validity) aspect that was missing in the related literature. We did this by collecting keystrokes and a variety of emotional states in situ on the participants' computers during their daily activity. In the next chapter we describe the experience-sampling field study that was conducted as well as the data collection software that was used to create our data set.

CHAPTER 3

DATA COLLECTION

This chapter is divided into two sections: a description of the field study that we conducted and an overview of the data collection software that we developed for this study.

We conducted a field study using custom built software to gather participants' keystroke data as well as subjective ratings of emotion. This approach gathers input in a natural setting while participants perform their daily tasks. We chose to gather data in a real-world context using an experience-sampling approach in order to increase the likelihood of capturing uninfluenced emotional states. This approach described by Hektner et al. [22], asks participants to periodically take notes on temporal feelings as they occur in the moment. The software that the participants installed on their computer cued the data collection process.

There are tradeoffs to using this type of field study compared to a more controlled approach (e.g. laboratory study) for emotional state determination. Modeling emotion is difficult because a controlled approach is needed for clean and labeled data, while the process of eliciting emotional responses is hard to control. In laboratory studies, carefully selected pictures, video, or audio stimuli that are known to elicit specific emotions are presented to participants [38,37,34,28,31,26,21,46,52,59]. This approach can generate a significant amount of clean data for a single emotional state; however, because these emotions are induced, they may differ from the emotions that develop in real-world situations.

Another difficulty in modeling emotions is the amount of data that is required to create a model with sufficient predictive power. Controlled lab studies can be time-consuming and typically

only cover a small fixed set of variables or target emotions. This can be limiting when conducting exploratory research where it may be difficult to determine which variables will be of interest. Furthermore, the amount of data that you can realistically collect from a lab study (with respect to keystrokes) is very limited without having participants attend multiple sessions. These additional sessions could be costly because they may require additional compensation and administration.

We chose to use an experience-sampling approach over a laboratory approach after weighing the relative advantages and disadvantages of each approach.

In this research, we did not initially know which emotions could be detected via keystrokes since there has been little research performed in this area. Our approach allowed us to gather data on 15 different affective states over a relatively long period of time. Furthermore, it allowed us to do this without increasing the cost of administering the study and without requiring multiple long sessions in the lab.

Through the course of our field study, participants were asked to rate their feelings on 15 5-point Likert scale statements. The participants were asked to rate how much they agreed, (strongly disagree, disagree, neither disagree nor agree, agree, strongly agree) to each statement. Using the stress statement as an example, the user would indicate his agreement with the statement "I feel stressed". These subjective measures of ground truth were used because the nature of field studies does not lend itself well to the types of methods where sensors are placed in the participant's environment or on their body [38,37].

We then processed the field study data, extracting keystroke features as well as other attributes that we could use when constructing the model. This processing took over 10 hours of computing time on a standard quad-core workstation and resulted in a large feature set (over 100 000 attributes). To reduce the size of the feature set, we used a combination of attribute selection and reduction to focus on what we believe to be the more salient features.

This reduced feature set was then used to train separate models for each of the 15 emotional states. The responses provided by the participants for each emotional state question were used as

the ground truth in supervised machine learning. Using the previous stress example, there were 5 different options to answer from strongly disagree to strongly agree. If the participant answered they 'disagree' with this statement, the disagree state would be considered the target class during model training. The models that were built used the C4.5 decision tree algorithm as implemented in the WEKA machine learning toolkit [60].

This field study was performed in conjunction with a similar study on detecting affective states through mouse movement. Due to the simultaneous collection of data for two studies, certain considerations had to be included for each study. This meant that some aspects of the field study and the software were modified to include restrictions and features for the different studies' needs. While we will mention all of the restrictions of the keystroke study, we will only discuss the details of the features and restrictions of the mouse movement study that had an effect on the keystroke study.

3.1 FIELD STUDY

Our field study was conducted from July 9th, 2009 to October 17th, 2009 with participants contributing data for, on average, four weeks. Participants were recruited using an online university bulletin system, email, posters, and through word of mouth. Two incentives were used: an initial incentive to encourage recruits in taking part and a bonus incentive (draws for gift cards) based on participant activity in the study.

3.1.1 Getting Started

Participants were required to complete two consent forms for this study, one for permission to use the keystroke data in the study and another one that asked for permission to use text excerpts from the keystroke data in anticipation of future textual analysis of the data. These forms were provided electronically via a website; this facilitated the remote administration of the study and ensured that we received the forms quickly so that participants could get started immediately. Recruits were required to accept the main consent form to be allowed to participate in the study;

however the textual excerpts consent form was optional. This was enforced by the security settings implemented on the website. The user could only access the software if they had accepted the consent form for the study. Appendix A has details on both of these consent forms.

The consent forms were programmed in PHP[1] and the first consent form provided text inputs for the participant's first name, last name, email address, and a checkbox that the user could use if they wanted to be notified about the results of the study. We collected the participant's name and email address so that we could contact them for troubleshooting any issues and incentive distribution. When the user accepted the first consent form we collected the IP address of the participant's computer as well as the timestamp when the user accepted the consent form. The IP address and timestamp combination were collected as a way to identify system errors and track them to the appropriate user if such a situation presented itself during the study. This information was kept securely in a write only file on the server that hosted the consent form. Similar information was also recorded for the textual excerpts form: a timestamp when the consent form was submitted and a flag that indicated if the user accepted or rejected the form.

An email list was created for the study; participants were encouraged to submit any questions or concerns to this list. Multiple administrators were assigned to this list making study administration easy and providing quick turn-around for questions.

Participants were asked to install our logging software on the computer that they used most frequently. Only one installation of the software was allowed per computer to prevent data corruption and performance issues that would have arisen with more than one logging application running simultaneously. The study was administered remotely on the participants' personal or work computers, as was data submission, which provided the opportunity to recruit more participants than we would have been able to otherwise.

[1] PHP: Hypertext Preprocessor `http://php.net/index.php`

3.1.2 Restrictions

There were no restrictions on the participants' activities during the study; they had the freedom to work unimpeded. However, participants were screened for particular requirements before they were able to take part in the study. The participants were required to type in English only because the use of multiple languages would have complicated the model building process. Common English character sequences were used in the feature extraction process; the addition of other languages would have extended our already large feature set. Also, it is unlikely that enough participants would have a common second language, leading to sparsity in the aggregated data for that language.

We also restricted the operating system that participants could use, because they were required to install platform-dependent data collection software on their own computers. The software was built using the .NET 3.5 Framework[2] and was tested on Windows XP[3] and Windows Vista[4] operating systems (the two most prevalent Windows operating systems at the time), which restricted the study to only those participants that use these operating systems. In some cases, the participants were also asked to install .NET 3.5 if they did not already have it installed.

Laptop users were initially prevented from participating in the study. This was due to the common usage of second keyboards amongst laptop users. Some laptop users have docking stations or separate keyboards that they use in different locations (e.g. office vs. home). These keyboards are usually very different than the smaller constrained keyboards that are built into laptops. A participant's keystroke timing could be different given the different layout and spacing between the two keyboards. However, we later removed this restriction due to initial low participation in the study and we added this factor to the list of the limitations of this approach in Chapter 6.

---

[2] .NET 3.5 Framework `http://msdn.microsoft.com/en-us/library/w0x726c2(v=vs.90).aspx`

[3] Windows XP `http://www.microsoft.com/windows/windows-xp/`

[4] Windows Vista `http://www.microsoft.com/windows/windows-vista/`

Participants were also restricted to using standard computer mice-no trackballs, track pads, or other pointing devices were allowed. This restriction was added for the mouse study. All restrictions were clearly communicated to participants before they could install the software.

3.1.3 Maintaining Privacy

Due to the sensitive nature of the data that we were collecting, we needed to ensure that participants felt comfortable with the study. Because our software falls into the category of key loggers-which most people think of being synonymous with malicious software (malware) and invasions of privacy - we had to make some accommodations to address potential participant concerns. 'Successful' key logging applications usually go unnoticed by users (and sometimes antivirus programs), which is why they are effective as malware. This subversive aspect of key loggers is also one of the biggest advantages of using keystroke dynamics to determine the affective state of the user because he will not be continuously reminded that he is being recorded. The participant's emotional state should therefore be minimally affected by the knowledge that they are being recorded.

To help alleviate participants' trepidation over the software, they were provided with a detailed description of the data collection process before beginning the study. This included annotated illustrations of the software's interfaces, an explanation of the data collection process, and detailed instructions on how to opt out of a sample period. At the beginning of a sampling period, the user is presented with the text that was collected from their keystrokes. They could then decide whether or not to include this data in the study or to discard it. If the text contained sensitive information (e.g. passwords), they could select an 'opt out' option on the interface. This would prevent the data from being sent to the data collection server, thereby maintaining the user's privacy. This also ensured that only data that the participant approved would be included in the study.

Participants were also given the option of manually clearing the recently captured data through the use of a right-click option on the system icon. This affected only the keystrokes that were in the software's temporary memory and not previously submitted keystrokes.

3.1.4 Study Completion

After completion of the field study, participants were debriefed and were sent instructions on how to remove the software. The removal process also removed any data collected during the study, including any files that had not been uploaded. This could have been a result of transient network connectivity issues or server downtime. Non-participation based incentives were then distributed to the participants, but participation-based incentives had to wait until the entire field study ended. This was due to the staggered start times of participants, meaning that participants were in different stages of the study. The difficulty in recruiting participants meant that each participant may have started the study at a different time.

3.2 PARTICIPANT DEMOGRAPHICS

To obtain some general information on the demographics of our participant population, a one-time questionnaire was presented to the participants (see Table 3.1 for a list of these questions). This questionnaire was included to get a general understanding of our population and to provide possible data points that could be used to further subdivide the dataset for analysis.

Table 3.1 Demographic questions presented to the participant.

1	Sex
2	Age
3	What is your occupation?
4	Where did you install this software?
5	Are you running this software on a virtual machine?
6	Did you install this software on a laptop or desktop?
7	What is your first language?
8	What language do you usually type in on this computer?
9	Which is your dominant hand?
10	Which hand do you normally use to control your mouse?
11	What type of mouse pointing device do you normally use?
12	How many buttons does your mouse have?
13	How would you rate your typing abilities?
14	On average, how much time do you spend on computers a day?
15	How much time do you spend playing computer, video, or console games?
16	How much time do you spend using a word processor, email, or instant messaging?
17	What percentage of your time that you spend on computers is spent on this particular machine?

Note that due to the fact that the mouse study was performed in conjunction with this study, some of the questions in Table 3.1 are specific to the mouse study and were not applicable for our purposes. This pertains mainly to questions 10, 11, and 12 so we will not be presenting the summary statistics for these questions.

It should be noted that the participants' responses presented in this chapter represent only the participants that were the most active. Originally, we had 26 participants in the study; however, we removed the less active participants (with fewer than 50 responses) which left us with 12 participants. The results that we present in this section are for these remaining 12 participants only, as it is their data that was used for building our models. We will revisit this in Chapter 5 where we discuss instance reduction.

Of the 12 active participants that took part in our field study, 10 were male and 2 were female. Their ages ranged from 24 to 34 with an average age of 28.5. This age range was expected as most of the recruiting was done on a university campus. Occupations consisted of 9 university

students, 2 administration personnel and 1 technician. All participants indicated that they were at least of average typing abilities (5 average, 2 good, and 5 with expert typing abilities). Each participant indicated that they usually typed in English and 6 people indicated that their first language was English. Other first languages included Persian (2), Vietnamese (2), Chinese (1), and Yoruba (1). The participant's first language was asked to possibly identify the dataset rows that may not be in English if we had found this was a problem.

Looking at the participants' computer usage in Table 3.2, we see that we have representation from each one of our video game usage categories with 3 participants indicating that they are more regular video game players. The time spent using word processing, email, and instant messaging (IM) was weighted more towards high usage categories.

Table 3.2 Computer usage indicated by the participants.

Time	**Number of Participants**	
	Video games	**Word processing, email, IM**
None	3	0
Less than 3 hours a week	4	0
3-7 hours a week	2	3
1-2 hours a day	2	2
More than 2 hours a day	1	7

The data collection software was installed on 10 desktops and 2 laptops. Of the 12 installations, there was only one installation on a virtual machine. Work computers accounted for 8 of the installations with the remaining 4 being home installations. Table 3.3 reports the percentage of time that the participants indicated that they spent on the computer where the data collection software was installed. From this, we see that 10 of the 12 participants spent at least half of their time on the computer that was collecting data.

Table 3.3 Percentage of time the participants spent on the computer where the software was installed.

Time	Number of participants
Almost none	0
About a quarter	2
About half	2
About three quarters	4
Almost all	4

3.3 FIELD STUDY SOFTWARE

The data collection software was a custom built Windows desktop application that was developed in the C# language using the .NET 3.5 framework. The software was tested on Windows XP and Windows Vista operating systems; however, it should work with any Windows-based computer that supports .NET 3.5 (e.g. Windows 7[5]).

3.3.1 Installation & Operation

Upon installation, the software created a globally unique identifier (GUID) for the participant. All data that was sent to the server was collected under this GUID. This helped maintain the anonymity of the data as the participant's first and last names were not collected by the software itself but through the consent form only. The software then displayed a one-time demographic questionnaire where the questions from Table 3.1 were asked. See Figure 3.1 for a screen shot of this questionnaire.

After the installation process completed, the application automatically started and ran as a background process that collects keystroke activity regardless of which application currently was in focus. The software was also added to the startup programs in the operating system to ensure

[5] Windows 7 http://www.microsoft.com/windows/windows-7/

that it would run even after a reboot of the computer. When the key logger was running, it was visible by a system tray icon in the lower right-hand corner of the screen.

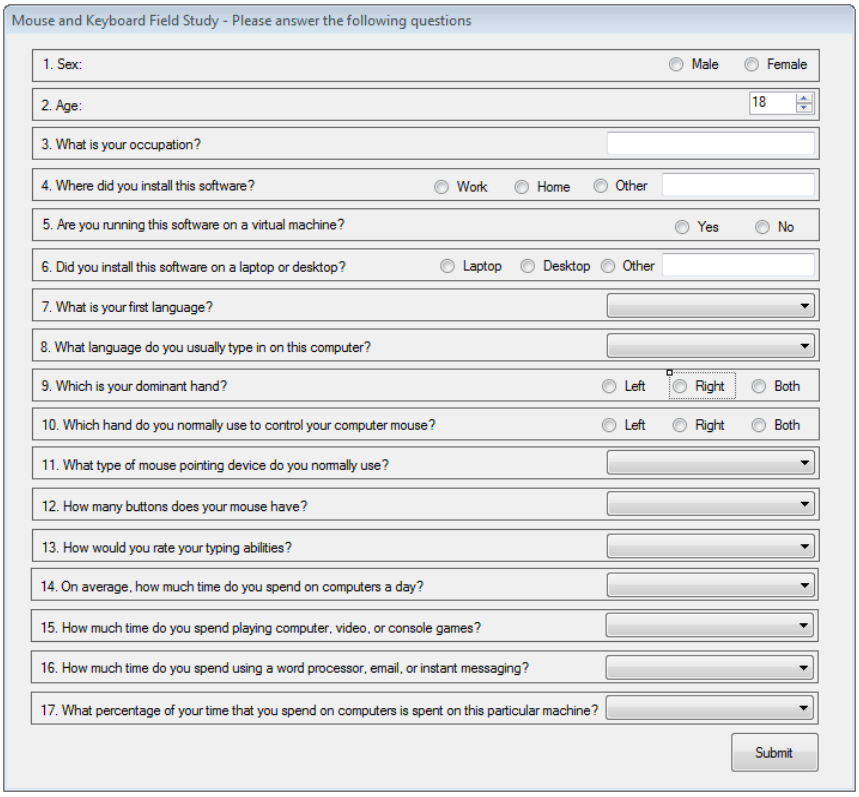

Figure 3.1 Demographic survey.

During pilot studies it was noticed that some participants would consistently enter passwords as they logged into applications after starting up their computer. These participants would most likely opt out of the first collection period due to the sensitive information captured and an hour would pass before it was possible to prompt the user again. To mitigate this situation, we implemented a delay of 10 minutes after initial startup of the computer before any data was logged.

Upon startup, the application also sent any pending logs to the data collection server. This was done to ensure that all of the logs that were meant to be included in the study were successfully collected. This feature ensured that we had minimal data loss due to transient network connectivity issues or temporary downtime in the data collection server.

3.3.2 Keystroke Capture

The application used a low-level Windows function accessed by unmanaged code in C#. This allowed us to screen each keystroke before passing it to the intended application, ensuring that the data was recorded regardless of which window had focus. On each keystroke event, a copy of the event was made and put into an internal processing queue, returning the original event to the intended application as soon as possible. This was done to minimize the latency that can occur when there were multiple events firing in rapid succession (e.g. an expert typist). The software used this internal event queue for its processing in a separate thread of execution.

The thread managing the event queue is used to ensure that only the previous 10 minutes of events (e.g. keystrokes) remain in memory and not entire hours worth of data. This was done for 3 reasons. First, this drastically reduces the amount of memory the application used as well as the speed of the application because it was very easy to gather a large amount of events in a short time period. Second, this makes the approval process easier for the participant since they did not have to page through a lot of keystroke data during the approval process. Third, we wanted to focus on the keystrokes that were collected near the time the participant completed the questionnaire as this data would more likely be influenced by the emotional states reported and not some previously experienced emotional state.

At any time the user could remove all collected events that were in the queue by right-clicking the system tray icon and selecting ‘clear keystrokes’ from the context menu. This was done to provide users with a way to quickly clear their data if they realized that they had just entered sensitive information that they did not want included in the study. For example, if the user had just signed into their online banking, they would most likely want this information cleared immediately.

Every 10 seconds, the software also determined the number of applications that the user had running and their active process names were also recorded. This was done to gather information on the participant’s context and as a possible indicator of how busy they were. However, it could

be that the number of open applications may just be an indicator of how a particular person works and not how busy they were.

As the user carried out their computer tasks, a notification would appear requesting them to complete a short questionnaire (Figure 3.2). The user could have chosen to ignore this notification and the program would delete the current keystrokes and prompt the user again in 30 minutes if they were still considered active at that point.

Figure 3.2 User notification to prompt a sample period.

A simple activity monitor was implemented to ensure that the user was not disturbed more than once every hour (except in the situation where the user ignored the notification without explicitly opting out). This monitor also ensured that the user was only prompted when their activity level was sufficiently high. With each new keyboard (and mouse) event that the software received, a timestamp was added to the activity monitor. A separate timer thread would check the activity monitor if the user was active enough to display the questionnaire. The activity monitor would check the most recent timestamp and would indicate the user as active if there was activity within the last five minutes (configurable) or if there was at least 2000 timestamps recorded (also configurable). Unfortunately, due to a bug in the system found after the field study completed, the minimum number of timestamps was not checked. This led to variability in the length of the free text data that we discuss in Chapter 5's results.

Alternatively, the user could have initiated the questionnaire at any time by double-clicking the icon in the system tray. This situation had the potential of creating variability in the free text as well because there was no limitation set in place if the user initiated the questionnaire.

3.3.3 Questionnaire Interface

When the participant initiated the questionnaire, a wizard-type interface appeared (Figure 3.3). At each step in this wizard, the participant had the opportunity to opt out of the collection period using the “Opt out this time” button that was visible at the lower left hand corner of each screen. If the participant chose to opt out of this collection period, the internal event queue was cleared from memory and the data not submitted. This opt out procedure was important because it allowed participants to easily skip the questionnaire if they were busy or if the captured keystrokes contained sensitive information. Note that this opt out procedure was only for a single collection period; the participants were given another chance to fill out another questionnaire one hour later. This was different than completely opting out of the study, for which there was no automatic method. The user would have to send an email to the study’s administrators and their data would have to be manually removed. However, this did not occur over the course of the study.

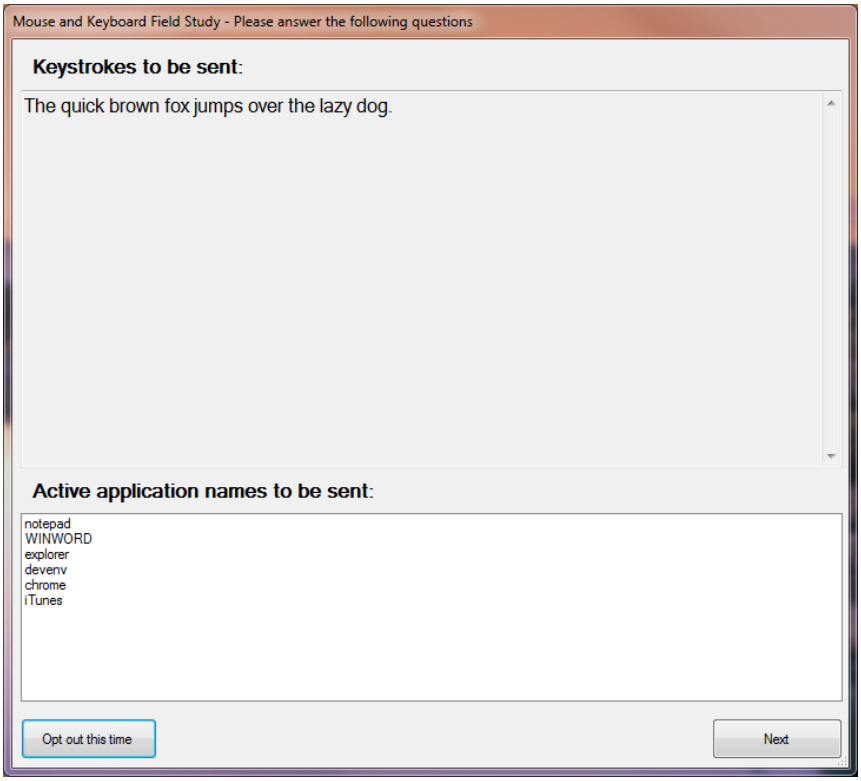

Figure 3.3 First screen of the data collection wizard.

The initial screen (Figure 3.3) presented all captured keystroke text as well as a list of active process titles that would be included in the sample period. The keystroke text here was referred to in this thesis as *free* text because there were no restrictions or influences on what the user typed. It was important to realize that this text contains all keystrokes, even keystrokes that the participant may have corrected. For example, deleted text using the backspace or delete keys were still recorded and were included in the text that was displayed to the user. It was important to include all of this text to ensure that it contain no sensitive information. Since this approval process was part of the first screen presented to the user, it provided them with an early chance to opt out of that collection period.

The next screen (Figure 3.4) presented consisted of a short questionnaire asking the user if they agreed or disagreed to the 15 statements, using a 5 point Likert scale. Some of the items in the questionnaire had opposing statements as a means of ensuring that the user is answering

consistently. For example, it would have been unlikely for a participant to be both distracted and focused at the same time; their answers should reflect this duality.

Figure 3.4 Emotional state self-report screen.

We included many different emotional state statements because of the exploratory nature of this research. Although we had some previous idea of what might work, we wanted to cover many different emotional aspects and the sample-experiencing methodology provided an opportunity for this. As there were many questions on this screen, rather than randomizing the order of the presentation of the questions, we kept them in a static order to reduce the cognitive processing requirements for the participant filling out the questionnaire repeatedly. To further reduce the cognitive load on the participant, question labels appeared bold if the question was unanswered. As soon as the question was answered, the text returned to a normal weighted font. Alternating row colors were used to assist the participant in easily associating their answers with the appropriate label.

The next screen presented was the *fixed* text input screen (Figure 3.5). On this screen, the participant was presented with a sample paragraph of text and was asked to type the paragraph into the text entry box below the sample. The system chose between 64 unique paragraphs that were extracted from the children's novel, Alice's Adventures in Wonderland [8]. These text excerpts were chosen due to the relatively simple sentence structures and absence of large uncommon words. Each piece of text was roughly the same length (190 - 210 characters); see Appendix B for a complete listing of these paragraphs. The paragraphs were rotated to reduce the chance that a participant could memorize any particular paragraph which could change their keystroke timings.

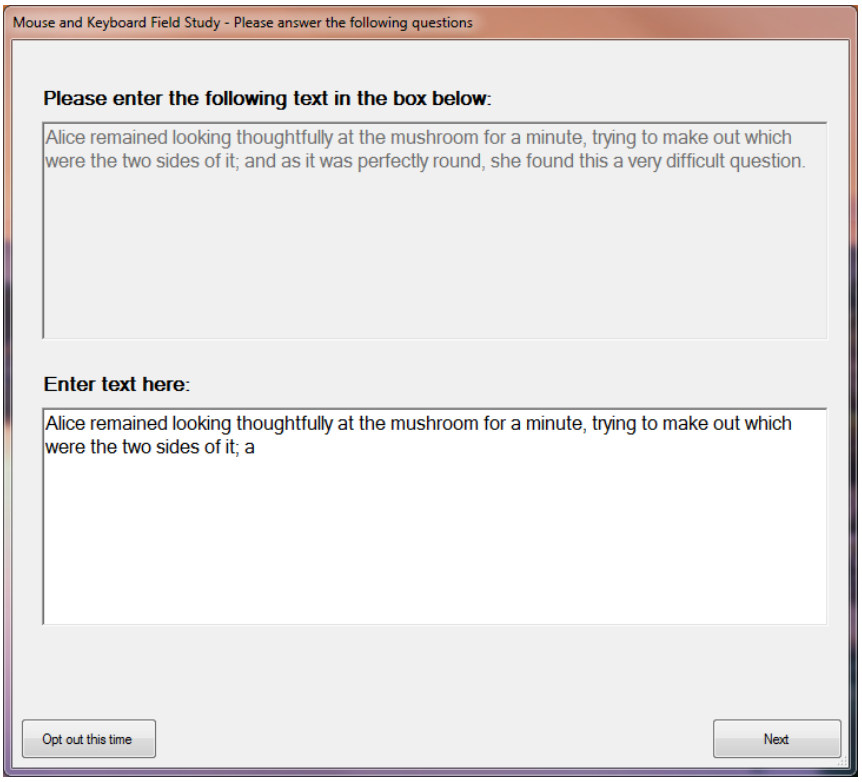

Figure 3.5 The fixed text entry interface.

Fixed text was included in the study for a number of reasons. The activity monitor was designed to be triggered by mouse events as well as keystroke events, so it was possible that the free text could have very few keystrokes recorded. The fixed text entry ensures that a minimum number of keystrokes were entered in each sample. The separation between the two types of text also

allows us the opportunity to identify which one (free or fixed text) had a stronger indicator of the participant's emotional state. In addition, the fixed text was included to ensure that English-based text was collected as the free text may not contain these in some cases (e.g. when the participant was playing a game).

There were, however, two drawbacks that the fixed text introduced. First, it increased the amount of work that the participants spent during a collection period, causing more disruption. Second, the fixed text presentation could have also influenced the participants' emotional state more compared to the more covert free text data collection. Due to these differences, it was important that we analyzed the fixed text and free text separately.

Due to the uncontrolled nature of the study, participants were prevented from selecting the fixed text paragraph to ensure that they did not copy and paste the text, but rather manually typed it in. Participants were also required to input a similar amount of text, within 15 characters; no additional checks were performed on the users' input. The keystrokes that were captured during the fixed text entry were then displayed to the user for review on the next screen (Figure 3.6). This was to provide an opportunity for the user to opt out again if any sensitive information was captured. For example, during the fixed text entry screen, the user could have got an instant message, replying to the message before completing the fixed entry screen. It was important to provide participants with this option in order to maintain their privacy.

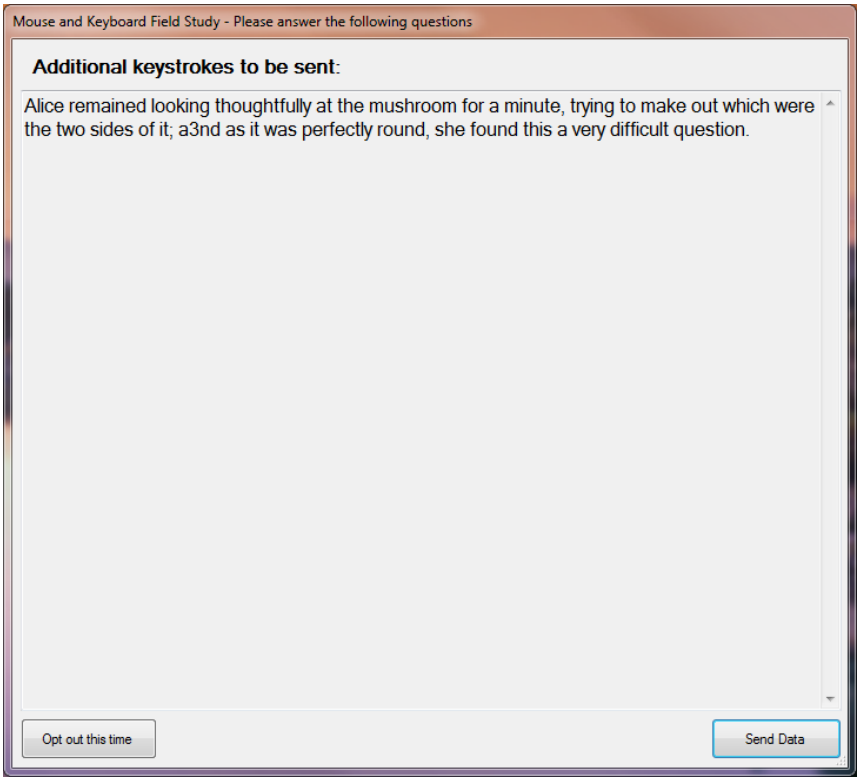

Figure 3.6 Final presentation of fixed text keystrokes entered.

Participants then clicked on the submit button to include all the data in the study. All events that were currently in the queue for this sampling period were saved as separate log files to the hard drive. These files were named based on the current date, type of log, and the number of the questionnaire (i.e. sample identifier). The files were sent to a data collection server via a web service. If an error occurred during the file transfer, the files that did not successfully upload would be sent again during next sample period or upon application start-up if the participant restarted their computer.

3.3.4 Event Logs

The data collection software produced 15 different types of files, 9 of which were used for our analysis (see Table 3.4) with the remaining 6 used for the mouse study that ran simultaneously. During each sample period, 8 of these logs were produced along with a unique identifier for that sample period. The log containing the participants' responses to the demographic questionnaire

(Demographics.log) was produced only once as the questionnaire was only displayed during installation.

Table 3.4 Logs produced for keystroke analysis.

Log Name	**Description**
Fixed-KeyboardEvents.log	The fixed text keyboard events captured.
Free-KeyboardEvents.log	The free text keyboard events captured.
Fixed-WindowEvents.log	The process names for running applications during the fixed text capture.
Free-WindowEvents.log	The process names for running applications during the free text capture.
QuestionniareEvents.log	Responses to the emotional state questionnaire.
SystemInformation.log	Operating system information (e.g. Keyboard speed)
application.log	Lists output statements used for debugging purposes.
error.log	Any errors that occur in the system are listed in this file.
Demographics.log	The participant's responses to the demographic questionnaire.

Keystroke event logs consisted of a line for each key pressed or released as well as other associated keystroke event data that we discuss in Chapter 4. The free and fixed text was maintained in separate log files. Each time the list of currently-running applications was retrieved, a row was recorded in the windows event log. The questionnaire log contained the answers to the emotional state questions and the demographic log contained the answers to the demographic questions. The general event and error logs contained debugging information to assist in debugging issues in the system. Examples of each of these logs can be seen in Appendix E.

3.3.5 Data Collection Server

We used a generic data collection server that was previously developed by another graduate student, Mike Lippold, for a class in distributed systems. For increased security, this application resided on a university server that was password protected and used the HTTPS protocol for data transfer. The data on this server was backed up daily to ensure minimal data loss during the study. The data was segregated on the server; projects had separate directories and each

participant had their own sub-directory in the project. To facilitate processing, we maintained this structure during data processing.

The remote data collection was implemented for four reasons. First, it reduced the work that the participant was required to perform since the software automatically submits the data. Second, if the participants encountered computer failures (e.g. hard drive failure) the data loss for our study would have been minimal as all data was uploaded to our backed up servers. Third, this maintained the anonymity of participants. If we had used a form of manual submission or collection, there may have been an opportunity to inadvertently identify the participant and we may not have received the data. Again this was especially important for this study due to sensitive information that may have been found in the dataset. Fourth, this form of data collection allowed for early preprocessing, while the study was still fielding. This was useful because it identified problem areas in our preprocessing early on. We will be discussing the special considerations that we had to handle due to these areas in Section 4.1.1 in the next chapter.

CHAPTER 4

FEATURE EXTRACTION

In this chapter, we describe the data processing and feature extraction that was performed on the keystroke data in preparation for our analysis. We begin with a general overview of the data processing and the special considerations made during this processing. We then describe the particular keystroke features and target classes (emotional states) that we extracted as well as additional data points that could be used for further analysis.

4.1 DATA PROCESSING

The raw log files generated by our software needed to be put into a form that could facilitate analysis of the data. This included the extraction of a number of features that we would later use to train our models. These features are discussed in detail in the Feature Extraction section in this chapter. A collection of 77 Matlab[6] scripts containing over 4000 lines of code were developed; they used the entire project directory from the data collection server as input during this process.

Processing was performed on each participant directory and summary features were extracted after each directory had been fully processed. Each log was read in and features were extracted based on the type of log (see Table 3.4 for a complete listing). For some log files this was trivial (e.g. questionnaire log); for others, such as the keystroke files, the process was more involved. For example, down and up keystroke events had to be matched in order to extract some of the features. Individual statistics and population statistics were calculated for these features as well.

[6] Matlab `http://www.mathworks.com/`

Once all of the logs for a questionnaire were processed, the data was then merged into one single row for that sample period. Features were extracted in this manner for all questionnaires for each participant. Once all of the files for a participant were read, a file with only that participant's data was saved. This was done to facilitate future studies of the data on an individual level. The individual participant data was then combined into one file and aggregate statistics were calculated.

4.1.1 Special Considerations

As previously mentioned, there were a few special cases that needed to be considered during the preprocessing phase for the keystroke features in particular. In this section we will describe these issues and how we handled each case.

When matching key-down events with key-up events for the same key, it was observed that occasionally key down events were found that did not have corresponding up events causing problems when extracting some of the keystroke features. This was the result of participants holding down keys for an extended period of time (greater than one second). There are a number of legitimate scenarios where this would be considered normal behavior. For example, this is seen when the user holds down the shift key while selecting multiple items in a file manager. From the software's perspective, this results in multiple down events of the same key firing repeatedly with only one matching up event. When this type of key pattern was found, the extra key down data was removed so that it would not skew the features. For example, key duration would be greatly affected in these situations. We modified the scripts to remove these down events from the dataset. The impact of this decision resulted in the removal of 5.9% of the data. Normalization of the features would exacerbate this situation as other features would be compressed due to such strong outliers.

The threshold that was used during this outlier removal process was configurable in the processing scripts; however, no further reductions were performed to handle outliers in the data. In Chapter 6, we discuss other options that could be taken to reduce the affects of outliers in the data.

Another problem occurred as a result of the event queue only storing 10 minutes of data. When older events in the queue are purged, no consideration is taken of which key down events match which key up events. It was found that there existed some key-up events that did not match to a prior key down event due to the key down event being on the other side of the 10-minute window. The preprocessing scripts were modified to remove these events and again the data lost was minimal as it would have only affected at most a few keystrokes at the beginning of each sample period. For example, it would be unlikely for the participant to have more than one key depressed at the start of the 10 minute window; this single key would be removed from the data set during this processing.

Modifier keys (shift, alt, control, system/Windows key) also created some unique challenges. Each key on the keyboard has a unique code, called the *vkcode*, associated with it. However, the character representation of this code on the screen varies depending on the modifier keys that are depressed. As this information was not recorded in the individual keystrokes, additional processing had to be performed to extract exactly when the modifiers were depressed for each keystroke event. Any keystrokes that were affected by these modifiers were updated with the modifier states during preprocessing so that they could later be easily used.

Toggle keys (caps lock, number lock) had similar issues. Depending on the state of the toggle key, the character representation of the vkcode is different. To further complicate matters, since these keys act as toggles they could be either on or off when the computer starts. To determine their initial states, the keyboard state was used to identify when a particular toggle key was depressed at the beginning of each log file. Similar to the previous modifiers, any keystrokes that could be affected by the toggle keys were then updated with the correct toggle state.

In order to properly analyze the data across all participants, some of the features described in this Chapter needed to be normalized. This was the last step in the feature extraction process as all of a participant's data had to be preprocessed before normalization could be performed. The

features were normalized using the mapminmax function in the Neural Nets toolbox in Matlab[7]. To normalize a participant's data for a particular feature, all samples for that feature (and for the participant) were first collected and then normalized from 0 to 1 using the mapminmax function. Although we normalized the data, we still kept the original values in the feature set, in case they were needed for future processing. This essentially provided a number of features that could be later picked from depending on the particular analysis that was desired.

4.2 FEATURE, CLASS AND DATA POINT EXTRACTION

We extracted 3 categories of information from the log files: keystroke features, emotional state classes, and various other data points. For each feature extracted, a number of statistics were also calculated and the timestamp of the event was recorded. This extraction process resulted in a great number of features (over 100 000) that we could use in modeling as well as a number of data points that we could use to further analyze the data. Although all of these features were programmed for this study, only a fraction (68) were used in this analysis which we discuss in Chapter 5.

4.2.1 Keystroke Features

Keystroke events were split between key down and key up events. The character representation of the key was extracted at run-time as this proved to be a difficult task to perform offline. The current state of all of the keys (key state) is also recorded for each new key event which is represented by an array of integers for each key on the keyboard. Capturing the key state was necessary to extract the character representation of the key that was pressed while the program was running. Since we already had this information, we decided to record this information in order to provide the opportunity to extract more features in future studies if desired.

[7] Matlab `http://www.mathworks.com/`

Each key event also included the current active window to facilitate segmentation of the data based on application type if this was found to be necessary during analysis. People may type differently in word processing programs than they do in integrated development environments (IDEs) used for computer programming. This additional data provides the opportunity to analyze the data based on the user's context.

The keystroke features are divided into two main types: single key features and compound key features. The compound key features are further separated into digraphs and trigraphs. Each feature was given a unique coded name in which the description of the feature could be identified. See Appendix C for a complete description of the naming conventions used in our data processing.

4.2.1.1 Single Key Features

There are two types of single key features - those features that are summary features of the complete sample text and those features that are created for each individual key on the keyboard. The features are briefly described in Table 4.1 and we will discuss the more complicated features in detail later in this chapter.

Table 4.1 Single key features: S = summary features and I = individual key features.

Name	Type	Description
[KEY]_Count	I	The count of unique keys found in the sample.
D2D_AllKeys	S	The duration from key down to the next key down across all keys.
KeyDur_[KEY]	I	The duration from key down to key up for a particular KEY.
KeyLat_AllKeys	S	The duration from key up to the next key down across all keys.
NumChars	S	The number of characters in the sample.
NumMistakes	S	The number of mistakes found (e.g. backspaces + deletes).
NumNums	S	The number of digits found in the sample.
NumSpecChars	S	The summation of NumNums, NumUpChars, and PuncMarks.
NumUpChars	S	The number of uppercase characters in the sample.
PercSpecChars	S	NumSpecChars as a percentage of the sample.
PuncMarks	S	The number of punctuation marks entered during the sample.

It is important to note that the 'I' type features listed in Table 4.1 are actually multiple distinct features, a separate feature for each key on the keyboard. The [KEY] notation is used to indicate a key name. For example, the [KEY]_Count feature for the 'a' key would end up being called 'a_Count' and would contain the summation of all of the 'a' key down events in the text. We will see a similar notation when we discuss the composite keystroke features.

Another important distinction is that some of the features (D2D_AllKeys, KeyDur_[KEY], KeyLat_AllKeys) contain multiple values. For these features the minimum, maximum, mean, mode, median, standard deviation, and variance were extracted instead of only extracting a single number. For example, KeyDur_A calculates the duration of an 'a' key event from key down to key up. However, it is very likely that a piece of text will contain multiple 'a' key presses. Instead of just returning one value (KeyDur_A), we calculate all the 'a' key durations, which results in the following features: KeyDur_A_Min, KeyDur_A_Max, KeyDur_A_Mean, KeyDur_A_Median, KeyDur_A_Mode, KeyDur_A_Std, and KeyDur_A_Var.

The NumMistakes feature described in Table 4.1 should also be explained further. This feature is the summation of the entire backspace and delete key down events that were found in the sample text. This was an attempt to determine the overall number of mistakes in the text. We thought

that this could be important as typos may indicate the presence of a particular emotional state (e.g. mistakes due to fatigue or stress). However, it should be noted that this does not include all forms of correction. For example, corrections could be made in other ways such as by moving the cursor in a text program or by making a selection and overwriting the text. These different methods of correction are very difficult to track in our data collection methodology where the participants are not forced to use a particular text editing program specifically designed to capture corrections. However, we do believe that our use of the backspace and delete keys will give us some indication of mistakes even if it may not cover all possible cases.

4.2.1.2 Compound Key Features

Many studies of keystroke dynamics have used features based on consecutive keystrokes called digraphs and trigraphs [3,43]. These are key event groupings of 2 or 3 key event pairs (a particular key's up and matching down event) from the first key down to the last key up and all the key events in between. For example, the first digraph in the word *'computer'* would consist of the following key events: *'c'* key down, *'c'* key up, *'o'* key down, *'o'* key up. Trigraphs are similar but include 3 key pairs instead of 2.

In our processing, these graphs (digraphs and trigraphs) include keys that do not have a visible character representation such as spaces and modifier keys (shift, control, system key). So the first diagraph in the word *'Computer'* (note the capitalization) would consist of the key events: *'shift'* key down, *'c'* key down, *'shift'* key up, *'c'* key up. For each graph found, a number of features are extracted (see Table 4.2). Similar to some of the previous single key features, all of the features listed in this table are created for each unique graph that is found in the text. Again, this leads to a large number of features for even a small sample of text.

Table 4.2 Digraph and trigraph specific features.

Name	Graphs	Description
2G_1D2D_[GRAPH]	2	The duration between the 1st and 2nd down keys.
2G_1Dur_[GRAPH]	2	The duration of the 1st key of the graph.
2G_1KeyLat_[GRAPH]	2	The duration between the 1st key up and next key down.
2G_2Dur_[GRAPH]	2	The duration of the 2nd key of the graph.
2G_Dur_[GRAPH]	2	The duration of the graph from 1st key down to last key up.
2G_NumEvents_[GRAPH]	2	The number of events that contributed or were part of the graph.
3G_1D2D_[GRAPH]	3	The duration between the 1st and 2nd down keys.
3G_1Dur_[GRAPH]	3	The duration of the 1st key of the graph.
3G_1KeyLat_[GRAPH]	3	The duration between the 1st key up and next key down.
3G_2D2D_[GRAPH]	3	The duration between the 2nd and third down keys.
3G_2Dur_[GRAPH]	3	The duration of the 2nd key of the graph.
3G_2KeyLat_[GRAPH]	3	The duration between the 2nd key up and next key down.
3G_3Dur_[GRAPH]	3	The duration of the third key of the graph.
3G_Dur_[GRAPH]	3	The duration of the graph from 1st key down to last key up.
3G_NumEvents_[GRAPH]	3	The number of events that contributed or were part of the graph.

Note that the number of key events in a graph is variable. When typing quickly, many keys may be depressed before others are lifted. Using digraphs as an example, it is possible that more than four key events can be included in a particular digraph. In this research, we consider the first two key down events as *contributing* to the digraph. All additional keystrokes found between the start and end of the digraphs are considered to be a *part* of the digraph. However, they do not contribute to the digraph. This is an important distinction to make as we refer to key events as being a part of a graph or contributing to it when we describe the compound features.

Graph duration, keystroke latency, and key down to key down have been used extensively in previous keystroke dynamics research [3,43] and were also used in this study. *Graph duration* is the time elapsed from the first key down event to the last key up event of the digraph or trigraph. *Keystroke latency* was defined as the time between the first key up event and the next key down event that contributes to the graph. Key down to key down was defined as the time elapsed from the first contributing key down event to the second contributing key down event.

The order of key events is not always sequential as the first key down may not be released until after the second key is released. For example, in the situation where the following key events are recorded: 1st key down, 2nd key down, 2nd key up, 1st key up. This could be the result of consecutive keystrokes using different hands (e.g. typing 'th'). Calculating the key latency in this scenario will result in negative values. The negative values were preserved in our data set.

Note that the features for digraphs and trigraphs are based on the same principles; however, there are more features for trigraphs than digraphs. This is due to trigraphs consisting of more key pairs than a digraph does. For example, key latency is a calculation between two events, in a digraph there would be only one calculation as there are only two keys (e.g. one space between the two keys). However, in a trigraph there are three keys, two sets of consecutive keys in which key latency can be determined. We decided to track these separately within the feature name (e.g. 3G_2KeyLat_[GRAPH] is the second key latency calculation of the particular trigraph).

In addition to the graph-specific features listed in Table 4.2, we included aggregate features across all of the digraphs (see Table 4.3). These are generally the same features; however, they summarize all of the digraphs and trigraphs found. Similar to the single key features, in the cases where there were multiple items (different digraphs or trigraphs), each of these features is further split up into the minimum, maximum, mean, mode, median, standard deviation, and variance across all of the digraphs or trigraphs. The majority of the features that we used in training our emotional state models in Chapter 5 are from Table 4.3.

Table 4.3 Aggregate digraph and trigraph features.

Name	Graphs	Description
2G_1D2D	2	The duration between the 1st and 2nd down keys of the digraphs.
2G_1Dur	2	The duration of the 1st key of the digraphs.
2G_1KeyLat	2	Duration between the 1st key up and next key down of the digraphs.
2G_2Dur	2	The duration of the 2nd key of the digraphs.
2G_Dur	2	The duration of the digraphs from 1st key down to last key up.
2G_NumEvents	2	Number of events (contributing/part of) found in the graph.
3G_1D2D	3	The duration between the 1st and 2nd down keys of the trigraphs.
3G_1Dur	3	The duration of the 1st key of the trigraphs.
3G_1KeyLat	3	Duration between the 1st key up and next key down of the trigraphs.
3G_2D2D	3	The duration between the 2nd and third down keys of the trigraphs.
3G_2Dur	3	The duration of the 2nd key of the trigraphs.
3G_2KeyLat	3	Duration between the 2nd key up and next key down of the trigraphs.
3G_3Dur	3	The duration of the third key of the trigraphs.
3G_Dur	3	The duration of the trigraphs from 1st key down to last key up.
3G_NumEvents	3	Number of events (contributing/part of) found in the graph.

4.2.1.3 Fixed and Free Text

As explained previously, there are two types of situations in which keystrokes can be gathered: fixed and free text. It was important to keep these keystrokes separate during analysis due to the different conditions under which the data was collected. Each feature explained in the previous subsection includes a fixed and free-text version, essentially doubling the number of keystroke features previously listed. These features were kept distinct by adding identifiers to the feature names.

4.2.1.4 Keystroke Feature Overload

The keystroke feature definitions that we used produced over 100 000 distinct features in our dataset. This was a problem for three reasons. First, the size of the output files was too large to handle. During initial feature extraction we needed to modify our scripts to load and save data to

the hard disk in order to free enough memory to continue processing; however, the data files that were produced were still too large to open in most programs. Second, the full feature set contained many missing values due to the fact that we were generating features for every possible digraph and trigraph in the data. For example, if only one participant enters the ‘quo’ trigraph, all other participants would have missing values for that column. Although there are data mining algorithms that handle missing values, the sheer number of missing values would have been problematic. Third, it is generally more difficult to find the signal in a data set with high dimensionality (a large number of features). With over 100 000 distinct features and only 1129 instances, we needed a way to significantly reduce the number of features that were included in the training set.

We took the approach of using only the most common English digraphs and trigraphs (see Table 4.4). This was possible due to the English language requirement of the participant screener. This restriction was added to the feature extraction early in the pre-processing. It significantly reduced the processing time and it reduced the number of features in the dataset to from over 100 000 to 10 076. The scripts were changed in such a way that graphs could be easily removed or added if future analysis required different graphs or the full data set. However, there were still too many features for our purposes so, during the analysis phase, we reduced these further. We will discuss how we did this in Chapter 5.

Table 4.4 Common English digraph and trigraphs [15].

Digraphs				**Trigraphs**		
AL	IT	EM	RO	AND	MEN	TIO
AN	ND	EN	SA	EDT	NCE	TIS
AR	NT	ER	SE	ENT	NDE	
AT	ON	ES	TE	FOR	OFT	
CO	OR	ET	TH	HAS	STH	
DE	RA	HE	TI	ING	THA	
ED	RE	IN	TO	ION	THE	

4.2.2 Emotional Class Extraction

Extracting the responses from the emotional state questionnaires was fairly straightforward when compared to the keystroke features. As presented in Section 3.3.3, each questionnaire asked the participant to rate how they were feeling through a series of 15 5-point Likert scale statements. For example, one of the statements asked the user whether they agreed or disagreed that they were feeling stressed. Similar statements were asked for frustration, anger, nervousness, happiness, excitement, confidence, sadness, boredom, sleepiness, relaxation, and feelings of being overwhelmed. Each statement was presented to cover a wide range of emotional states; similar to studies that use subjective self-reports. Some statements such as hesitation, confidence, and distraction were also considered because these feelings were present at times during the pilot study.

Each of the 15 responses contains 1 of 5 possible classes that we later consider as ground truth during supervised machine learning (see Chapter 5 for details). The 5 classes correspond to the different options (strongly disagree, disagree, neither agree nor disagree, agree, strongly agree) for each of the emotional state questions. This produced the class categories indicated in Table 4.5.

Table 4.5 Class categories extracted from questionnaires.

Affective Classes	
AngerRating	HesitanceRating
BoredRating	NervousRating
ConfidenceRating	OverwhelmedRating
DistractedRating	RelaxedRating
ExcitedRating	SadRating
FocusedRating	StressRating
FrustrationRating	TiredRating
HappinessRating	

Two additional features were also extracted from this data: the arousal and valence ratings. These features were based on the idea presented by Lang [33], in which he suggested that emotions can be classified in a two-dimensional space (AV space) defined by arousal (activation) and valence (pleasure) see Chapter 2 for a discussion of arousal, valence, and emotion. Each statement was classified according to where it fit in the AV space. Figure 4.1 displays how these states were classified. If we consider the stress state, it is seen to be high in arousal but low in valence.

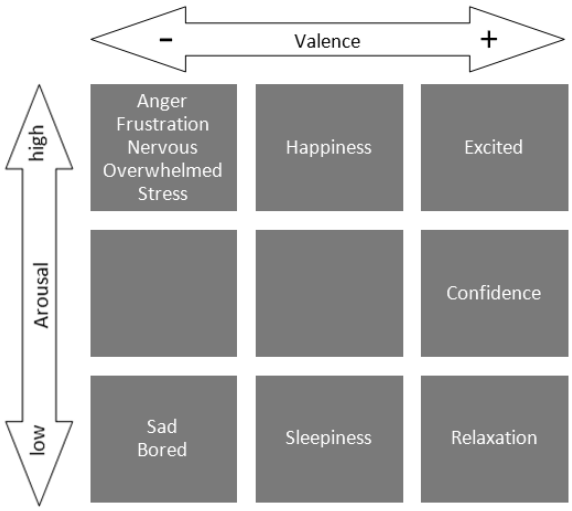

Figure 4.1 Emotional states in arousal/valence space

It is important to distinguish that Figure 4.1 illustrates the positions of each of the statements if the participant strongly agreed with the statement. However, if the participant disagreed with a statement, this would result in the opposite position across one of the axes. For example, if the participant disagreed to the excited statement, this meant that they agreed with the unexcited state, which is lower in arousal. This duality ensured that there was complete coverage of the cells in Figure 4.1.

The arousal and valence scores were determined by both the position of a particular item in Figure 4.1 and how strongly the participant agreed or disagreed with the statement. For example, if a participant's answers agreed with the stressed statement, the arousal score was incremented by 1 and the valence was decremented by 1. Higher weights were associated with the strongly agree and strongly disagree answers; they incremented or decremented the scores by 2. After the arousal and valence scores were calculated, they were assigned categories of low, neutral, and high based on the calculated scores.

4.2.3 Context Data Points

Our data collection methodology of capturing all of the users' keystrokes (regardless of the application) allowed us to capture contextual data such as the application name that was active for each keystroke. In [12], Dowland et al. included the active window title in the sample along with the keystroke data. Although we initially included the window text in our data collection software, this was later modified to include only the name of the active applications (e.g. winword.exe). This was done as we felt that privacy concerns could arise from recording the full window title in certain applications. For example, when using particular email programs, the window text could contain the full subject line of open email documents.

For each sample, we collected the process names of all the running user applications (polling done every 10 seconds) as well as the active application for each keystroke. These attributes allow for further dissection of the dataset into subsets of activity if needed. If we wanted to look at keystrokes that were just in chat applications, this would help to identify those rows of data where chat applications were used.

An attribute was created for each distinct application process that was running on the participant's computer. This attribute represents a sum of every new instance of an application as it appeared in the activity window log. This was added because it could indicate the type of work that the participant was doing for that sample period. The number of windows that the participant had open was also extracted as it may indicate how busy the participant was; however, this could just be an indicator of an individual's task management style.

Additional summations were calculated based on 4 application categories: text applications, internet browsers, communications programs (email, instant messaging), and integrated development environments (IDEs) used for computer programming (Table 4.6). These categories were chosen as they represented a range of interactions with the computer, from a keystrokes perspective. Text and communication applications, where participants are more likely to be entering full English sentences, may give very different keystroke timings than IDEs would, where proper sentences are atypically used. Internet browser usage could also be very different from the other categories. There could be very little text entered in the browsers for some users (e.g. internet searches); however, it is also possible for a portion of the users to enter full text (e.g., blogging) as well.

Table 4.6 User context features.

Name	Description
BrowserPrograms	The count of browser applications active during the sample period.
EmailPrograms	The count of communication applications active during the sample period.
IDEPrograms	The count of IDE applications active during the sample period.
NumWinOpen	The number of windows active during the sample period.
TextPrograms	The count of word processing applications active during the sample period.

This categorization of programs happened in two parts. It started with extracting a list of unique programs from the active window log. In our data set this list included 218 distinct processes. These programs where then manually categorized into one of 5 categories (‘other’ had to be added for applications that did not fit in the mentioned categories). This mapping was then used during the feature extraction process on the windows logs to count the appropriate number of instances per category.

4.2.4 Other Data Points Collected

A number of additional data points were also extracted, providing the opportunity to further divide the dataset, supporting multiple approaches of analysis. These data points are divided into 3 different types: the responses from the demographic survey, questionnaire-specific data, and other data that was obtained from the user’s operating system. Table 4.7 lists these additional data points and provides a short description of each.

Table 4.7 Demographic (D), questionnaire (Q), and system (S) attributes.

Name	Type	Description
Age	D	Age of the participant in years.
ComputerTime	D	The amount of time the participant spends on computers daily (less than 30 minutes. 30-60 minutes, 1-2 hours, 2-4 hours, 4-8 hours, more than 8 hours).
DominateHand	D	The dominate hand of the participant (left, right, or both).
FirstLanguage	D	The first language of the participant.
FirstLanguageLCID	D	.NET framework locale ID for the FirstLanguage attribute.
IPAddress	D	The IP address of the computer.
LaptopDesktopInstall	D	Type of computer the participant installed the software on (laptop, desktop, or other.
LaptopDesktopInstallOther	D	Open text answer to the LaptopDesktopInstall attribute.
Occupation	D	The occupation of the participant (open textbox).
PercentageTimeonThisMachine	D	Percentage of time the participant spends on this computer (almost none, about a quarter, about half, about three quarters, almost all).
Sex	D	The sex of the participant.
TypedLanguage	D	The language that the participant typically types in.
TypedLanguageLCID	D	The .NET locale ID of the TypedLanguage attribute.
TypingAbilities	D	The typing abilities as reported by the participant (novice, poor, average, good, and expert).
TypingSoftwareTime	D	Time the participant spends in word processing applications.[8]
VideoGameTime	D	Time the participant spends playing video games.[8]
VirtualMachine	D	If the software was installed on a virtual machine (yes or no).
WhereInstalled	D	Where the participant installed the software (home, work, or other).
WhereInstalledOther	D	Open text answer for the WhereInstalled attribute above.
QsFilled	Q	Number of questionnaires filled out.
QuestionnaireId	Q	Questionnaire ID can also be considered the sample id.
SampleTextDisplayed	Q	This is the identifier for the fixed text that was displayed.
CultureEnglishName	S	The current culture set in the operating system.
CultureKeyboardLayoutId	S	The .NET Keyboard layout ID set in the operating system.
CultureLCID	S	.NET culture locale ID for the CultureEnglishName attribute.
InstallTimestamp	S	Time and date when the software was installed.
KeyboardDelay	S	The keyboard delay as it was set in the operating system.
KeyboardSpeed	S	The keyboard speed as it was set in the operating system.
ParticipantId	S	The GUID assigned to the user.

[8] None, less than 3 hours a week, 3-7 hours a week, 1-2 hours a day, more than 2 hours a day.

4.2.5 Summary Data Points

Additional statistics were calculated across all of the data for each participant. For most numeric features in the dataset the maximum, minimum, median, mean, mode, standard deviation, and variance were calculated for each participant. For example, statistics for each of the emotional state statements were calculated to see the general trend in responses for an individual participant. This provided a quick overview of the range of participant responses. Similar statistics were calculated for each of the single keystroke features such as number of characters, punctuation, uppercase, and special characters. Other features were included to possibly identify how busy the person was and how much they work on certain types of applications. These include the overall statistics for program-type counts, number of mistakes, and the number of active windows.

CHAPTER 5

ANALYSIS & RESULTS

In this chapter, we present the analysis that we performed on the features extracted from our data set. We begin by explaining the data selection and reduction techniques that were used on the data set. Next we discuss how we trained our models and we discuss the overall classification process. We finish the chapter by presenting our results including the best classifiers that were created from our data set.

5.1 ANALYSIS

After the feature extraction process that was described in Chapter 4, we were left with too many features, which could result in over-fit models [4]. It was necessary reduce the number of features before proceeding with training the models. The next two sections discuss how we did this using a combination of feature selection and reduction.

5.1.1 Feature Selection

In Chapter 4, we described how during the feature extraction phase we had problems due to the large number of features that we were attempting to extract. Our solution was to keep only the most common English digraphs and trigraphs because the other graphs were more likely to contain sparse data between and within participants. Although the feature selection described in Section 4.2.1.4 alleviated these problems, we were still left with 10 076 features in the resulting data set.

Even with this initial reduction, we still needed to further reduce the remaining 10 076 features to build our models. We decided to only include aggregate features for this analysis and to remove specific key, digraph, or trigraph features. The features that we decided to use can be seen in Table 5.1.

Table 5.1 Features used in the analysis. *Mean and standard deviation were included for these features. Xs indicate the features that were included for fixed and free text.

Feature name	Fixed	Free	Description
NumChars		X	The number of characters in the sample.
NumNums		X	The number of digits found in the sample.
NumSpecChars		X	The summation of NumNums, NumUpChars, and PuncMarks.
NumUpChars		X	The number of uppercase characters in the sample.
PercSpecChars		X	NumSpecChars as a percentage of the sample.
PuncMarks		X	The number of punctuation marks entered during the sample.
NumMistakes	X	X	The number of mistakes (backspace + delete) in the text.
2G_1D2D*	X	X	The duration between 1st and 2nd down keys of the digraphs.
2G_1Dur*	X	X	The duration of the 1st key of the digraphs.
2G_1KeyLat*	X	X	Duration between 1st key up and next key down of the digraphs.
2G_2Dur*	X	X	The duration of the 2nd key of the digraphs.
2G_Dur*	X	X	The duration of the digraphs from 1st key down to last key up.
2G_NumEvents*	X	X	The number of events that contributed or were part of the graph.
3G_1D2D*	X	X	The duration between 1st and 2nd down keys of the trigraphs.
3G_1Dur*	X	X	The duration of the 1st key of the trigraphs.
3G_1KeyLat*	X	X	Duration between 1st key up and next key down of trigraphs.
3G_2D2D*	X	X	The duration between 2nd and 3rd down keys of the trigraphs.
3G_2Dur*	X	X	The duration of the 2nd key of the trigraphs.
3G_2KeyLat*	X	X	Duration between 2nd key up and next key down of trigraphs.
3G_3Dur*	X	X	The duration of the third key of the trigraphs.
3G_Dur*	X	X	The duration of the trigraphs from 1st key down to last key up.
3G_NumEvents*	X	X	The number of events that contributed or were part of the graph.

As you can see from Table 5.1, we chose a mixture of single and composite keystroke features. This included both fixed and free text versions; however, we separated these sets when training the models because of the different methods in which they were collected. After this division,

there were 31 features for the fixed text analysis, and 37 features for the free text analysis. Free text included more features due to the nature in which it was collected. The influenced nature of the fixed text keystrokes means that some of the features did not apply. For example, the number of characters does not make sense to include in the fixed text analysis, as it would be heavily influenced by the fixed text that was presented to them. For similar reasons, NumNums, NumSpecChars, NumUpChars, PercSpecChars, and PuncMarks did not apply to the fixed text analysis.

5.1.2 Feature Reduction

We were initially unsure if we would still have too many features at 31 and 37 so we decided to reduce these features using principle components analysis (PCA). PCA is a feature reduction technique that can be used to reduce the dimensionality or the number of independent variables of a data set. This is done by specifying the amount of variance that you would like to keep in the data set. We decided to keep 95% of the variance in the data, but because of a possible loss in accuracy (due to the loss of 5% of the variance), we decided to separately train both the PCA and non-PCA model variations.

Note that because different variations modified either the number of instances (we explain why in sections 5.1.4.2 and 5.1.5) or the number of attributes (e.g. fixed vs. free text), PCA was performed on the models for each emotional state separately from each other. This resulted in reducing the number of attributes to between 13 and 15 depending on the variation. A side effect of using PCA is that the resulting attributes can be difficult to read because they combine multiple original attribute names into one making the resulting decision trees difficult to read.

5.1.3 Instance Selection

As mentioned briefly in Chapter 3, we removed the data of participants who were less active during the field study. We removed all participants with fewer than 50 questionnaires submitted, which resulted in the removal 18.5% of our overall data. We wanted to ensure that everyone had a consistent activity level because it would have been difficult to identify inconsistencies (due to different emotional states) in users' typing rhythms if we did not have enough baseline samples.

5.1.4 Classification Method

We used supervised machine learning to build our models using the J48 decision tree algorithm found in the WEKA machine learning toolkit [60] written in the Java[9] programming language. This algorithm is based on the C4.5 revision 8 algorithm and although there was a successor to this algorithm (C5.0) available, the version provided in WEKA was chosen as it was freely available whereas the C5.0 was not. In addition, the WEKA project is open source, which ended up being more important during our analysis than first realized as we ran into two memory issues due to the size of our dataset. We were able to solve these memory issues by increasing the Java heap memory allowed and by fixing a bug in WEKA's source code that occurs when running large data sets during model training. These modifications would not have been possible in a proprietary data mining application where the source code was not accessible.

5.1.4.1 Decision Trees

There were a number of different supervised machine learning approaches that we could have used. We decided to use decision trees, and in particular the J48 algorithm, for a variety of reasons. This algorithm had the ability to handle numeric values as well as missing values. This was very important to keystroke features because the majority of our features were numeric timing values of keystrokes. The J48 algorithm could also handle missing values, which was required for the model building process because the data set could contain many missing values.

Decision trees were also used due to their simplicity of use compared to some of the other forms of supervised machine learning. For instance, neural nets were initially considered; they were not chosen because particular parameters, such as the number of hidden layers and nodes, were difficult to choose and refine. We also found that although models built with neural nets provided similar classification rates to decision trees, they took considerably longer to train compared to decision trees for each variation. This would have been a problem during our analysis as we anticipated wanting to run a large number of variations.

---

[9] Java `http://java.sun.com`

The J48 algorithm also provided an easy method of fine tuning the model (pruning) to make it more robust and to prevent over-fitting the model to our specific data set. The *confidence value* and the *number of objects per leaf node* are the parameters that were used to control the amount of pruning applied to the decision tree. After some initial trial and error with different values, we found that a confidence value of 0.10 and the number of objects per leaf node set to 5 gave us reasonable classification rates with relatively shallow trees. Shallow trees were desirable because unlike large trees they were not associated with an over-fit model.

Another desirable feature of decision trees was the ease of reducing them into a set of rules which would ease the task of future integration with real-world applications. Rule changes could be made easily with little change to the structure. In contrast, modifying classifiers based on neural nets would not be possible without re-training the entire classifier when new data is collected.

5.1.4.2 Target Classes

The results from the sampling questionnaires were used as target classes during our model training. In order to train our models, each emotional state was separated and trained individually. This resulted in five class-levels of 15 emotional models along with two additional models of three class-levels for the arousal and valence features described in Chapter 4.

During piloting, we noticed that some participants were not using the full range of the 5-point scale that was given to them; they seemed hesitant to use the extremes of the scale (i.e. strongly disagree and strongly agree). This could cause difficulties in creating classifiers for the full range of five levels. We decided to add additional models based on the original 15 emotional states. We did this by adding two additional sets of emotional models using two and three class-levels based on the participants' answers.

In the three class-level models, we combined the extreme responses with their corresponding responses leaving only disagree, neutral, and agree. Strongly disagree answers were combined with the disagree answers, and strongly agree answers were added to the agree answers. Two class-level models were created in a similar fashion; however, in this case we dropped the neutral

instances leaving only disagree and agree levels. This resulted in the 47 separate models listed in Table 5.2.

Unfortunately, by dropping the neutral category, the overall number of training instances were reduced. We revisit the implications of this reduction in the results section later in this chapter.

Table 5.2 Class-level breakdown for each emotional model. Xs indicate the class names that contained the different class levels.

Class name	5 class-levels	3 class-levels	2 class-levels
Anger	X	X	X
Bored	X	X	X
Confidence	X	X	X
Distracted	X	X	X
Excited	X	X	X
Focused	X	X	X
Frustration	X	X	X
Happiness	X	X	X
Hesitance	X	X	X
Nervous	X	X	X
Overwhelmed	X	X	X
Relaxed	X	X	X
Sad	X	X	X
Stress	X	X	X
Tired	X	X	X
Arousal		X	
Valence		X	

5.1.5 Adjustments for Class Skew

When looking at our pilot data, we noticed that some of our class distributions were skewed with over-representations in some classes while others were under-represented. This was due to the

uninfluenced nature of the data collection process because we could not artificially control the emotional state of the participant (as in mood induction). It would be unlikely that we would experience an even distribution of emotions when collecting data in natural settings. For example, the anger distribution would likely be skewed to the disagree category with only a few positive (agreements) sample periods because it would be unlikely that a majority of people would be angry most of the time.

This class skew resulted in high classification rates for some of our initial models; however, these overall classification rates were misleading when taking the true positive and false negative rates for each individual class into account. The strong classifications of the category with a majority of the instances were overriding the poor classification rates of the minority classes in our initial pilot evaluation.

We anticipated having similar class skew problems with the full study because of the uninfluenced nature of the data collection. In anticipation of this, we decided to try a method of adjusting the class distributions using an under-sampling [13] technique which we refer to as balancing the class distributions. To under-sample, we found the lowest number of instances across all classes in an emotional state, essentially the minority class. Next, we randomly removed instances from the other classes until all of the classes had an equal number of instances in them. This essentially levels the class distribution and removes excess instances. This process of random instance removal was repeated ten times and results were averaged over all instances.

Depending on the distribution of the classes, this method could remove a significant amount of data. We were unsure how this would affect our model training so we opted to include in this thesis both balanced and unbalanced variations that we trained separately.

5.1.6 Variations

Throughout the course of our analysis description, we have mentioned a number of cases where we constructed many different combinations in order to identify good models in our data set. This resulted in 376 different variations once all of the target classes, text types, balanced types, attribute reduction, and class-levels were taken into account. Figure 5.1 summarizes these

different combinations; for a detailed description on how we processed these variations, please refer to Appendix C.

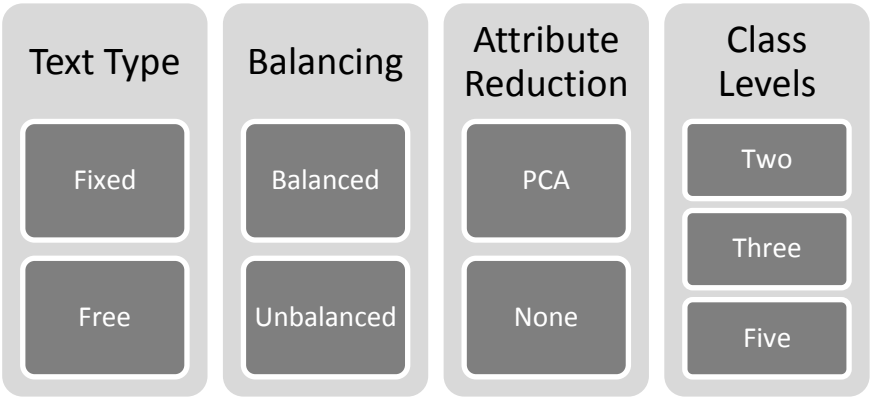

Figure 5.1 Summary of the main categories that were trained.

From Figure 5.1 it should be noted that every possible combination of the categories created a separate variation. In addition, the two and five class-levels have 15 different emotional states whereas the three class-levels have 17 different emotional states due to the additional arousal and valence categories.

5.1.7 Evaluation

We performed ten-fold cross-validation to evaluate the predictive performance of our models, which is standard practice when a data set's size is limited [60]. We randomly divided the data into ten groups with a similar class distribution as the whole data set. Training was performed using nine groups and tested using the group that was held out (tenth group). This happens ten times, once for each group held out, and the results from all the separate training sessions were averaged. All the results that we present were taken from the cross-validation results rather than the training results that were also provided in WEKA's output.

We came up with four different types of categories (as described in Table 5.3) to easily describe the best models from our 376 variations: Bronze, Silver, Gold, and Platinum. Each evaluation category used the classification rate and the Kappa statistic from the ten-fold cross-validation results. The Kappa statistic indicated how much the classification rate was a true reflection of the

model or how much would be attributed to chance alone. Kappa values range from 0 (no agreement other than chance) to 1 (perfect agreement) [60].

Table 5.3 Top evaluation categories. The categories at the top of the table are super sets of rows below.

Type	Description
Bronze	Overall classification rates of 75% and above with a Kappa statistic above 0.4.
Silver	True positive rates of at least 75%, false positive rate less than 25% for each class.
Gold	True positive rates of at least 80%, false positive rate less than 20% for each class.
Platinum	True positive rates of at least 85%, false positive rate less than 15% for each class.

Of these four categories, the Bronze category was different because it takes into account the successful classification rates over *all* of the classes for an emotional state and not individual class classification rates. Alternatively, Silver, Gold, and Platinum are more discriminating, looking at both the true positive (TP) and false positive (FP) rates for each individual class. Using a 3 class-level target as an example, each of the 3 classes would have to have a TP rate greater than 85% and an FP rate less than 15% to be included in the Platinum category. We did this to identify those cases where class skew may have been introducing bias into the overall classification rate. This was important as we wanted to ensure that our models could identify a range of classes (not only one) across one emotional state.

5.2 RESULTS

The remainder of this chapter focuses on the results that we obtained from the analysis described in the previous section. We close this section with an overall summary of the results and by presenting our best performing classifiers as well as others that show potential to create good models given a larger data set.

The different combinations of fixed/free text, balanced/unbalanced, PCA/no-PCA, and class levels resulted in 376 different variations for which we built models. Due to space

considerations, we have only presented the three class-level variations in this chapter; please refer to Appendix D for the results of the two and five class-level variations.

5.2.1 Data Set Attributes

In this section, we briefly describe the different overall attributes that we found in the collected data. We discuss the distribution of the participant responses, the class distribution in the data set, and finally the number of training instances that were used in the analysis.

5.2.1.1 Participant Responses

As was mentioned in Chapter 3, we removed the participants who had fewer than 50 emotional state questionnaires submitted. This resulted in a data set with 12 participants and 1129 samples with the distribution of the number of samples per participant shown in Figure 5.2. The number of instances per participant ranged from 51 to 219 with an average of 94 instances per participant and a standard deviation of 52.7.

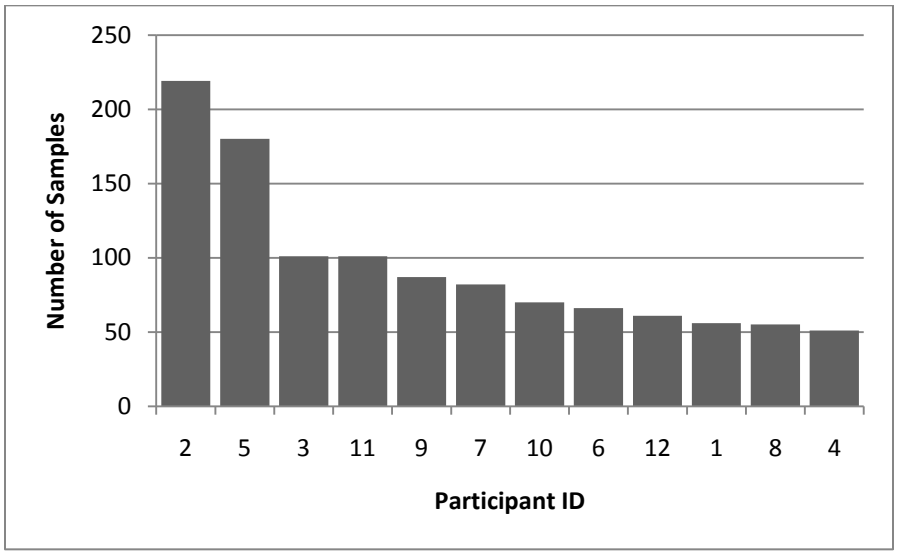

Figure 5.2 Number of samples collected per participant.

From Figure 5.2, we can see that there were two participants who submitted a significantly greater number of samples (more than 75 instances) than the other participants. Although we

have more samples for these two participants, it was still unlikely that this would be enough data to use separate emotional models for each participant. Although this was not part of the original analysis we will revisit its implications later, in our Chapter 6 discussion.

Figure 5.3 and 5.4 illustrate the normalized mean values for each free and fixed text keystroke feature respectfully. From these figures, we can see that many of the values were close to 0 with high standard deviations. In fact, only 3 free text and 4 fixed text features were above 0.2 and many of the features had standard deviations that were higher than the mean values. This suggests that there were strong outliers in the data set that were preventing the data normalization from using the full range from 0 to 1 and clustering the values near 0. In Chapter 6, we discuss modifying our outlier removal as part of our future directions to fix this problem.

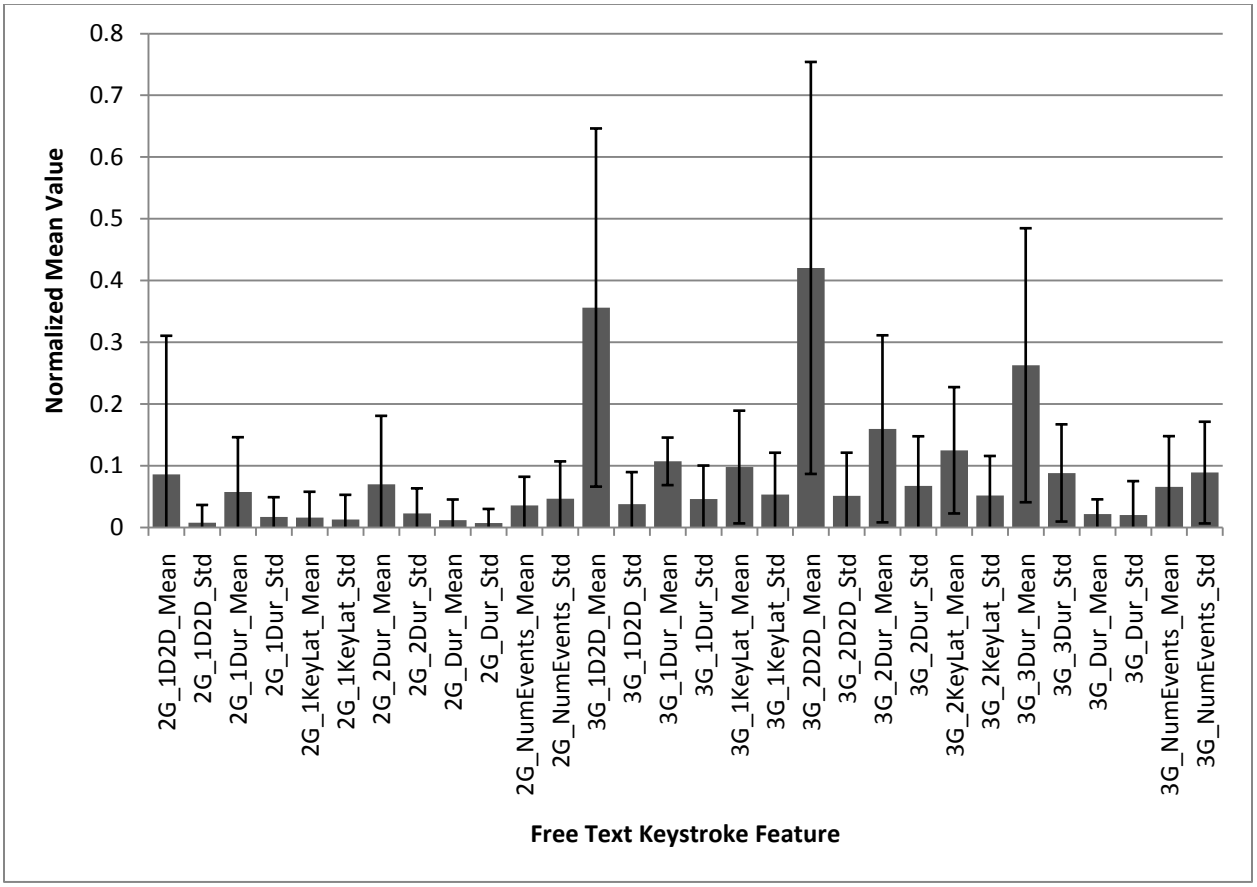

Figure 5.3 Free text keystroke feature variation with standard deviation bars.

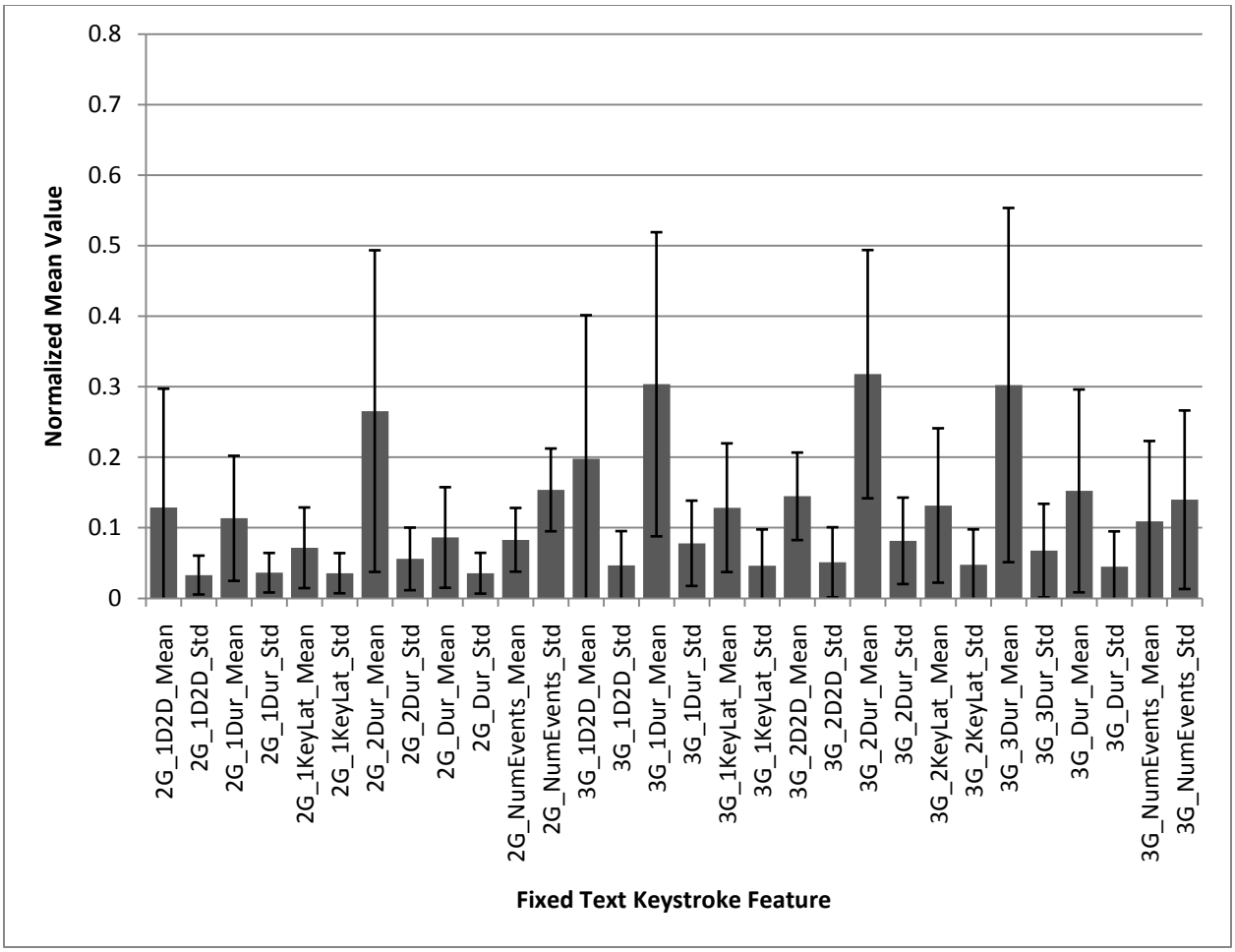

Figure 5.4 Fixed text keystroke feature variation with standard deviation bars.

5.2.1.2 Class Distribution

If we look at the class distribution for each emotional state aggregated across participants (Figure 5.5), we can see that some emotional states had more evenly balanced distributions than others. The distracted, focused, happy, relaxed, and tired states are more balanced than the remainder of the emotional states as they have roughly similar class representations around the neutral class. Anger, boredom, excitement, frustration, hesitance, nervousness, overwhelmed, sadness, and stress were over-represented in the disagree class and under-represented in the agree class. Conversely, the confidence category had an overrepresentation of the agree category and an underrepresentation of the disagree category.

We experienced similar class distribution issues during the pilot and expected this variance in the different emotional categories. This was what led to the addition of the balanced (under sampling) variations in this analysis.

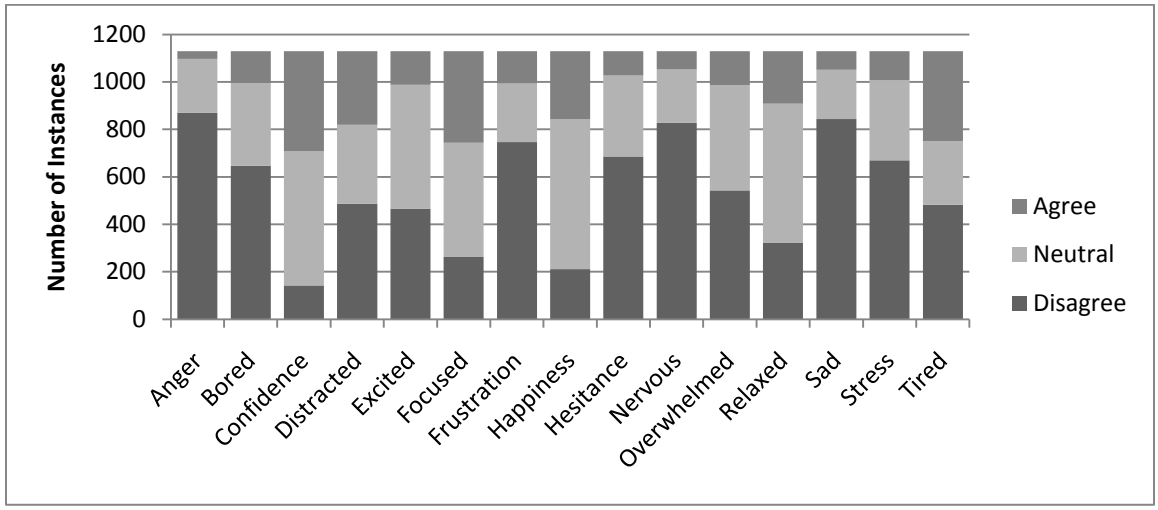

Figure 5.5 Distribution of three class-level unbalanced responses.

Similar distributions for the five and two class-level variations were found, which was not surprising given that these class variations are all derivative of the five class-levels and used the same data set. One small difference could be seen in the two class-level variations because they had fewer instances due to the removal of the neutral class.

Similarly, the balanced variations had considerably fewer instances because data was removed to balance the distribution and handle the class skew problem. The number of instances removed varied greatly based on the number of class levels that were used and how skewed the distribution was. Considering the nervous variation using Figure 5.5, only 76 instances reside in the agree category; the balancing variations would reduce each category to 76 instances. The balanced version of the nervous variation would contain a total of 228 instances when compared to the unbalanced version of 1129 as illustrated in Figure 5.6. This data loss was an unfortunate side-effect of the balancing procedure as it would have been desirable to use the maximum

amount of data possible when building these classifiers. However, the balancing procedure was necessary to ensure that the classification rate would not be affected by class skew.

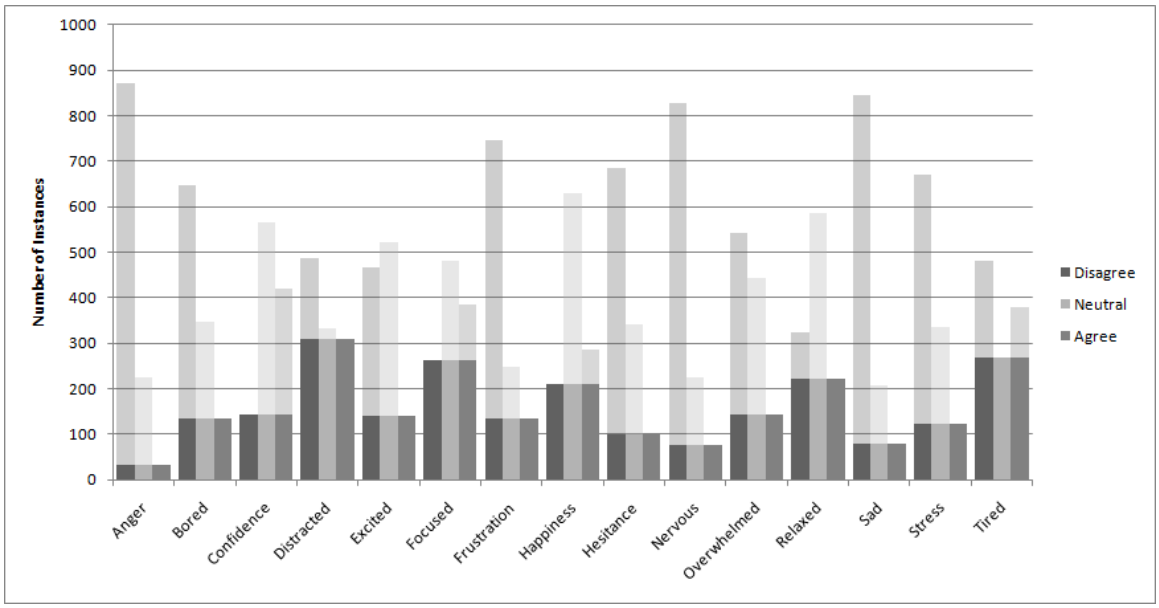

Figure 5.6 Three class balanced distributions compared to original distributions.

The chosen class-level variation can also exacerbate this issue of data loss. This was due to some participants being reluctant to use the full spectrum of responses for the emotional state questions and resulted in the two extreme ends of the scale (strongly disagree and strongly agree) being even more underrepresented than the minority classes in Figure 5.6. For instance, using a five class-level variation with the nervous state, only 27 responses in the strongly agree class existed. This reduced the overall number of instances down to 135 for training. We anticipated these results and determined that the five class-level balanced results would likely not provide good classifiers but we included them in Appendix D for completeness.

From Table 5.4 we saw how both the two class-level variations and the balanced variations affected the number of instances that remained in the training set. Among the balanced variations, we can see that the three class-level variations had the most number of instances when compared to the balanced two and five class-levels. Also, by comparing the two class-level

unbalanced and balanced variations, we saw the compounded reduction on the balanced two class-level variations. We revisit the number of training instances and related implications when we present the overall top results later in the chapter.

Table 5.4 Number of instances used in training after two-class reduction and balanced distributions[10].

Emotional State	Unbalanced		Balanced					
	2-classes		2-classes		3-classes		5-classes	
	Instances	% Used	Instances	% Used	Instances	% Used	Instances	% Used
Anger	903	80.0%	64	5.7%	96	8.5%	35	3.1%
Arousal	n/a	n/a	n/a	n/a	444	39.3%	n/a	n/a
Bored	781	69.2%	268	23.7%	402	35.6%	15	1.3%
Confidence	564	50.0%	286	25.3%	429	38.0%	65	5.8%
Distracted	797	70.6%	620	54.9%	930	82.4%	80	7.1%
Excited	607	53.8%	282	25.0%	423	37.5%	35	3.1%
Focused	647	57.3%	524	46.4%	786	69.6%	150	13.3%
Frustration	882	78.1%	272	24.1%	408	36.1%	45	4.0%
Happiness	498	44.1%	422	37.4%	633	56.1%	55	4.9%
Hesitance	788	69.8%	204	18.1%	306	27.1%	348	30.8%
Nervous	903	80.0%	152	13.5%	228	20.2%	135	12.0%
Overwhelmed	686	60.8%	286	25.3%	429	38.0%	10	0.9%
Relaxed	544	48.2%	442	39.1%	663	58.7%	85	7.5%
Sad	922	81.7%	156	13.8%	234	20.7%	180	15.9%
Stress	793	70.2%	246	21.8%	369	32.7%	25	2.2%
Tired	861	76.3%	758	67.1%	804	71.2%	310	27.5%
Valence	n/a	n/a	n/a	n/a	399	35.3%	n/a	n/a

5.2.2 Cross Validation Results

In this section, we present the results obtained from our 10-fold cross-validation process on the different variations that were trained.

[10] Note that the unbalanced three and five-class variations were not shown in Table 5.4 as there were no instances removed in these cases; the full data set of 1129 instances was used.

For each of the results sets that we present in this section, both the PCA attribute reduction and non-PCA sets are presented separately. In general, the PCA results roughly followed their unreduced counterparts with only slight variation (both positive and negative) depending on the model. The overall classification rate difference was found to be similar for the PCA reduced variations and the non-reduced variations (mean difference of 0.04, standard deviation 3.18). Similar small differences were seen between the Kappa statistics as well (mean difference of 0.01, standard deviation 0.09). For simplicity and due to the slight variation between the PCA and non-PCA results, we refer to the non-PCA results, except in exceptional cases throughout the remainder of this chapter.

Table 5.5 presents the results from our three class-level balanced free text variations. Although these classification rates are fairly low, it should be noted that the classification rate can have different meaning depending on the number of target classes in the model. Classification rates near 50% are only equivalent to chance if there are two target classes. In a three-class scenario, such as the one presented in Table 5.5, 33.3% of the time we would achieve a correct classification by chance alone. Similarly, in a 5-class scenario, any classification rate above 20% would be better than chance.

Table 5.5 Three class-level, balanced, free text results.

Emotional State	No Reduction				PCA Reduction			
	CC %	CC Variance	Kappa	Kappa variance	CC %	CC Variance	Kappa	Kappa Variance
Anger	40.73	25.43	0.11	0.01	45.31	28.75	0.18	0.01
Arousal	41.91	4.72	0.13	0.00	45.99	7.74	0.19	0.00
Bored	40.22	6.79	0.10	0.00	44.43	4.86	0.17	0.00
Confidence	52.38	3.44	0.29	0.00	49.53	7.95	0.24	0.00
Distracted	44.91	2.61	0.17	0.00	48.08	0.85	0.22	0.00
Excited	43.31	9.52	0.15	0.00	46.95	11.64	0.20	0.00
Focused	41.39	4.41	0.12	0.00	47.35	7.83	0.21	0.00
Frustration	46.42	4.70	0.20	0.00	47.89	3.26	0.22	0.00
Happiness	43.14	11.66	0.15	0.00	45.77	5.43	0.19	0.00
Hesitance	50.69	6.80	0.26	0.00	52.58	16.15	0.29	0.00
Nervous	40.00	8.71	0.10	0.00	46.71	12.45	0.20	0.00
Overwhelmed	46.34	9.66	0.20	0.00	46.46	9.12	0.20	0.00
Relaxed	55.94	7.35	0.34	0.00	55.19	6.38	0.33	0.00
Sad	45.64	3.73	0.18	0.00	50.94	13.18	0.26	0.00
Stress	46.42	11.15	0.20	0.00	46.86	6.06	0.20	0.00
Tired	47.18	4.63	0.21	0.00	50.56	4.60	0.26	0.00
Valence	42.51	5.00	0.14	0.00	47.17	4.97	0.21	0.00

All of the free text classification rates listed in Table 5.5 were ‘better than chance’ as we previously described; however, the classification rate variance was high in many of the balanced results and the Kappa statistic too low to consider the classification rates as a valid representation of the predictive performance of the model. The best variation from the balanced free text combinations is the Relaxed variation with a 55.9% correctly classified rate and a Kappa statistic of 0.34; however, this is still too low to qualify for one of the evaluation categories (Bronze, Silver, Gold, and Platinum) that we described in Section 5.1.7.

When looking at the unbalanced free text cross-validation results in Table 5.6, the classification rates were significantly higher than the balanced free text variations seen in Table 5.5. However, the Kappa statistics were again too low to consider these classification rates to be good indicators of the models’ predictive performance. Again, the relaxed model seemed to perform the best

according to the Kappa statistics that were generated, performing slightly better than its balanced counterpart. However, even with this increase, none of the results in Table 5.6 qualified for our evaluation categories.

Table 5.6 Three-class, unbalanced, free text results.

Emotional State	No Reduction		PCA Reduction	
	Correctly Classified %	Kappa	Correctly Classified %	Kappa
Anger	79.63	0.17	79.19	0.15
Arousal	71.48	0.00	71.83	0.10
Bored	59.79	0.07	58.81	0.06
Confidence	60.41	0.26	58.81	0.27
Distracted	43.31	0.03	49.07	0.19
Excited	64.30	0.37	59.61	0.29
Focused	47.83	0.13	50.66	0.21
Frustration	68.82	0.12	68.47	0.10
Happiness	57.75	0.07	57.22	0.17
Hesitance	63.15	0.10	64.22	0.17
Nervous	75.29	0.12	74.67	0.11
Overwhelmed	53.68	0.14	52.88	0.21
Relaxed	67.58	0.41	65.19	0.38
Sad	77.33	0.15	76.26	0.12
Stress	61.56	0.08	58.90	0.12
Tired	57.40	0.29	53.59	0.25
Valence	75.02	0.00	74.67	0.01

Note that variance was reported on the balanced results only because the results for the unbalanced variations were from a single training set and not averaged over 10 different training sets as in the balanced variations.

Reviewing the balanced fixed text results presented in Table 5.7, we saw an average 14.7% increase over the balanced free text results classification rates. Similar increases are seen with the two and five class-level variations shown in Appendix D (both increased by 13.6%). The Kappa statistics are also considerably higher; however, the classification rate variance follows similar patterns with high variance found in the anger, bored, sad, and stress models. Although

the classification rates in these results were greater than chance, no models in this set qualified for our top four evaluation categories.

Table 5.7 Three class-level, balanced, fixed text results.

Emotional State	No Reduction				PCA Reduction			
	CC %	CC Variance	Kappa	Kappa variance	CC %	CC Variance	Kappa	Kappa Variance
Anger	54.58	31.64	0.32	0.01	51.15	27.36	0.27	0.01
Arousal	54.46	7.47	0.32	0.00	51.73	8.65	0.28	0.00
Bored	52.24	11.06	0.28	0.00	54.15	3.81	0.31	0.00
Confidence	66.85	8.47	0.50	0.00	62.42	4.07	0.44	0.00
Distracted	54.79	2.63	0.32	0.00	56.25	2.32	0.34	0.00
Excited	61.84	5.99	0.43	0.00	59.76	7.24	0.40	0.00
Focused	50.95	7.12	0.26	0.00	49.66	2.89	0.24	0.00
Frustration	60.39	6.30	0.41	0.00	58.11	10.91	0.37	0.00
Happiness	54.44	3.18	0.32	0.00	55.34	3.57	0.33	0.00
Hesitance	72.55	3.84	0.59	0.00	67.39	9.06	0.51	0.00
Nervous	61.14	2.36	0.42	0.00	58.16	7.19	0.37	0.00
Overwhelmed	53.43	7.76	0.30	0.00	52.07	4.51	0.28	0.00
Relaxed	66.56	3.38	0.50	0.00	63.79	1.48	0.46	0.00
Sad	71.71	15.70	0.58	0.00	65.68	24.51	0.49	0.01
Stress	59.40	14.39	0.39	0.00	57.45	2.97	0.36	0.00
Tired	63.84	5.76	0.46	0.00	62.51	4.34	0.44	0.00
Valence	59.85	5.40	0.40	0.00	57.02	8.71	0.36	0.00

Finally, looking at the unbalanced fixed text results in Table 5.8, we saw our highest classification rates and Kappa statistics yet. This also holds true for the two and five class-level variations (see Appendix D). Again, we saw an increase in classification rates of 6.9% in the unbalanced fixed text results when compared to the unbalanced free text. Similarly the free text two and five class-level variations see an increase (3.6% for both). The highest classification rates were 85% for both the anger and sad emotional models. Keep in mind that in a three class-level variation that chance is 33%. Furthermore, we saw a number of emotional states (anger, hesitance, nervousness, sadness, and valence) from this result set ranking in the Bronze evaluation category.

Table 5.8 Three class-level, unbalanced, fixed text results.

Emotional State	No Reduction		PCA Reduction	
	Correctly Classified %	Kappa	Correctly Classified %	Kappa
Anger	85.12	0.55	82.82	0.50
Arousal	71.12	0.26	70.77	0.22
Bored	66.61	0.38	66.25	0.37
Confidence	67.58	0.46	65.81	0.42
Distracted	57.66	0.35	54.56	0.31
Excited	68.82	0.48	71.57	0.52
Focused	53.94	0.29	52.52	0.25
Frustration	73.07	0.39	72.28	0.38
Happiness	58.10	0.27	58.99	0.29
Hesitance	77.06	0.56	73.16	0.48
Nervous	83.35	0.59	82.46	0.56
Overwhelmed	64.48	0.40	63.77	0.36
Relaxed	72.45	0.54	70.42	0.50
Sad	84.94	0.62	82.37	0.53
Stress	70.77	0.45	71.66	0.44
Tired	68.29	0.51	67.05	0.48
Valence	78.83	0.43	77.68	0.41

5.2.3 Top Results

This section summarizes the overall best classifiers for each emotional state. We then look into some of the characteristics of these top results identifying possible issues that we found. Through this process we identify the emotional states that produce the best models as well as other emotional states that show potential to create predictive models for future studies that use larger data sets.

The overall top performing results for each emotional state are listed in Table 5.9. We found that 9 of the 17 emotional states have models that were successful in achieving at least one of our standards for good evaluation. From these results we saw that anger, confidence, excitement, hesitance, nervousness, relaxation, sadness, tired and valence states all have models that reach

one of the evaluation categories. Alternatively, we see that arousal, bored, distracted, focused, frustration, happiness, overwhelmed, and stress did not produce classifiers that qualified for our evaluation categories.

Table 5.9 Overall top performing models by number of instances per evaluation category.

Emotional State	Bronze	Silver	Gold	Platinum
Anger	2	0	0	0
Arousal	0	0	0	0
Bored	0	0	0	0
Confidence	4	2	0	0
Distracted	0	0	0	0
Excited	2	1	0	0
Focused	0	0	0	0
Frustration	0	0	0	0
Happiness	0	0	0	0
Hesitance	5	2	2	0
Nervous	6	2	1	0
Overwhelmed	0	0	0	0
Relaxed	4	1	0	0
Sad	6	2	2	1
Stress	0	0	0	0
Tired	4	4	3	0
Valence	2	0	0	0

Table 5.10 describes the number of different variations that made our top evaluation categories. We discuss possible reasons why there were no free text or five class-level models that made our evaluation categories in Chapter 6.

Table 5.10 Detailed breakdown of top models (35 variations in total).

	Text Type		Reduction		Class Level			Balanced		Evaluation			
State	**Fixed**	**Free**	**PCA**	**None**	**2**	**3**	**5**	**True**	**False**	**Bronze**	**Silver**	**Gold**	**Platinum**
Anger	X			X		X			X	X			
Anger	X		X			X			X	X			
Confidence	X			X	X			X		X	X		
Confidence	X		X		X			X		X	X		
Confidence	X			X	X				X	X			
Confidence	X		X		X				X	X			
Excited	X			X	X			X		X	X		
Excited	X			X	X				X	X			
Hesitance	X			X	X				X	X			
Hesitance	X			X		X			X	X			
Hesitance	X		X		X				X	X			
Hesitance	X			X	X			X		X	X	X	
Hesitance	X		X		X			X		X	X	X	
Nervous	X			X	X				X	X			
Nervous	X		X			X			X	X			
Nervous	X		X		X				X	X			
Nervous	X			X		X			X	X			
Nervous	X			X	X			X		X	X	X	
Nervous	X		X		X			X		X	X		
Relaxed	X			X	X				X	X			
Relaxed	X		X		X				X	X			
Relaxed	X			X	X			X		X	X		
Relaxed	X		X		X			X		X			
Sad	X			X	X				X	X			
Sad	X		X			X			X	X			
Sad	X		X		X				X	X			
Sad	X			X		X			X	X			
Sad	X			X	X			X		X	X	X	X
Sad	X		X		X			X		X	X	X	
Tired	X			X	X				X	X	X	X	
Tired	X		X		X				X	X	X		
Tired	X			X	X			X		X	X	X	
Tired	X		X		X			X		X	X	X	
Valence	X		X			X			X	X			
Valence	X			X		X			X	X			
Total	**35**	**0**	**16**	**19**	**26**	**9**	**0**	**13**	**22**	**35**	**14**	**8**	**1**

The two class-level variations created the highest number of successful classifications with 74% of the models while the three class-level variations accounted for the remaining 26%. We take a closer look into how the neutral instance removal in the two class-level variations may have affected these results in the next section.

The PCA reduction variations did appear in some of these top performing models, but seemed to have little effect on the results. Although the PCA successfully decreased the number of attributes before training, the PCA cases performed only slightly poorer (1%-3%) than those cases that had no attribute reduction performed. This difference is small and we generally saw both PCA and no reduction models in the evaluation results. The totals for the attribute reduction columns in Table 5.10 show that there were 16 PCA models that had successful classifications compared to 19 models for the variations that had no reduction performed. This slightly higher number was due to some classification rates being very close to the Bronze evaluation boundary. We discuss the effect of PCA reduction further in Chapter 6.

The results displayed in Table 5.10 show that both unbalanced and balanced variations are represented in the top results. Unbalanced seems to be favored slightly here since unbalanced represents 63% of the models whereas balanced represents the remaining 37%. The reason why there were more unbalanced models likely had to do with the class skew in some of the unbalanced results causing higher classification rates. In the next section, we narrow down these models, identifying the ones that may be biased due to class skew.

5.2.3.1 Narrowing Down the Results

In this section we take a closer look into possible issues with some of the models presented in Table 5.10. We identify possible class skew problems, inspect the classification variance in the balanced variations, and review the number of training instances that were used to create the classifiers. Each step further reduces our results until we reach our best models.

5.2.3.1.1 Identifying Class Skew Problems

Due to the nature of the Bronze category, it is possible for the classification rates to be biased if there exists significant class skew in the data set. For example, the 3 class level anger model appears to have strong class skew when looking at the class distribution in Figure 5.7. From the class distribution we can see that it is heavily weighted towards the disagree category.

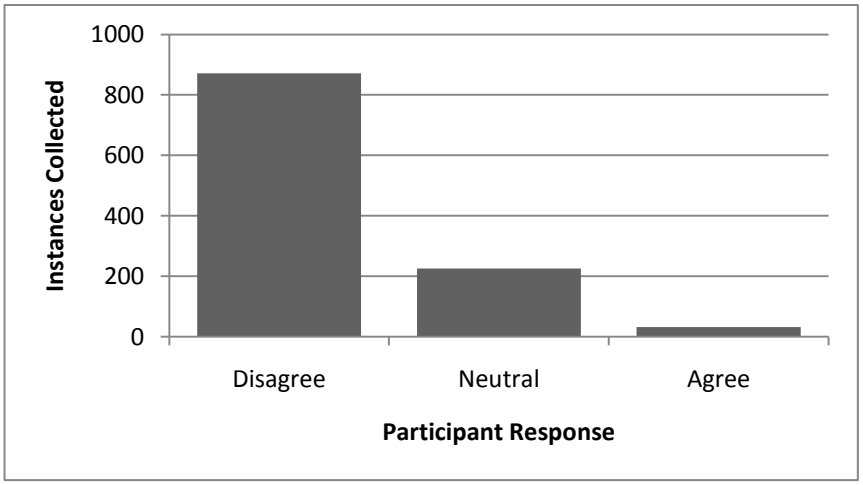

Figure 5.7 Skewed class distribution for the three class level anger data set.

Both of the anger placements in the top results had this skewed class distribution with representations in only the Bronze category. The fact that the Bronze evaluation category only takes the overall classification and Kappa statistic into account and not the individual true positive and false positive values led to some misleading classification rates to be reported. This was seen in a few of the class distributions where the majority of the data was in one class, so it was possible that the classifier could get most of the underrepresented classes incorrect and still result in a good classification rate. Looking further into the TP and FP classification rates for one of the individual classes for the anger classifiers (Table 5.11) we saw that the class skew was biasing the overall performance rates which was demonstrated by the TP rates for the agree category being 0 whereas the disagree TP rate was very high at 0.94.

Table 5.11 True positive and false positive classification rates for the 3-level anger classifier.

Disagree TP	Disagree FP	Neutral TP	Neutral FP	Agree TP	Agree FP
0.94	0.42	0.62	0.07	0.00	0.00

However, this problem was avoided in the balanced variations because we ensured that the class distribution was always uniform. Additionally, the Silver, Gold, and Platinum categories are not affected by this since they only considered the TP and FP rates of each individual class. In these evaluation categories, if any of the classes have a poor classification rate, the entire model is disqualified from the evaluation category being considered.

Although the Bronze category had this potential problem of class skew, it was still important to include because it identified the emotional states that can be studied in future research. If a model appears in this category and not others, it simply indicates that our sample data was heavily skewed and not necessarily that models for that state cannot be successfully trained using keystroke dynamics; this includes both the anger and valence states.

Removing the Bronze only variations from Table 5.10, we are left with 14 variations, only two of which have unbalanced class distributions. Note that these 2 unbalanced variations were for the tired emotional state, which is understandable because the original class distribution was fairly evenly weighted on both the agree and disagree classes.

5.2.3.1.2 Classification Variance in Balanced Variations

Having discussed the problem of class skew in the Bronze evaluation category, we turn our focus to the remaining top three evaluation categories (Silver, Gold, and Platinum), where we noticed abnormalities in the classification rate variance in balanced models. Note that the balance variations were run 10 times and the reported average was used in the evaluation. The classification rate variance gives us an idea of how well this balancing procedure performed over all 10 runs.

Table 5.12 lists the remaining models after removing the Bronze category; the classification rates and reported variance was added to further illustrate the performance of the models. Note that this table consists of only two class-level, fixed text variations. We saw that there is a high degree of variance in some of the classification rates. We further refined our models by removing the classifiers where the lowest classification bound (average – variance) was below 75%. This reduced our set of models to 14 variations.

Table 5.12 Top 3 evaluation categories with classification rates and kappa statistics.

	Reduction		Balanced		Correctly Classified		Kappa		Evaluation		
Emotional State	**PCA**	**None**	**TRUE**	**FALSE**	**%**	**Variance**	**Statistic**	**Variance**	**Silver**	**Gold**	**Platinum**
Confidence		X	X		80.31	2.99	0.61	0.00	X		
Confidence	X		X		79.16	10.90	0.58	0.00	X		
Excited		X	X		76.67	11.84	0.53	0.00	X		
Hesitance		X	X		85.64	8.55	0.71	0.00	X	X	
Hesitance	X		X		82.01	6.09	0.64	0.00	X	X	
Nervous		X	X		83.22	5.70	0.66	0.00	X	X	
Nervous	X		X		78.75	14.82	0.58	0.01	X		
Relaxed		X	X		79.46	2.47	0.59	0.00	X		
Sad		X	X		87.95	13.31	0.76	0.01	X	X	X
Sad	X		X		83.14	20.73	0.66	0.01	X	X	
Tired		X		X	84.20	0.00	0.68	0.00	X	X	
Tired	X			X	81.30	0.00	0.62	0.00	X		
Tired		X	X		83.46	1.16	0.67	0.00	X	X	
Tired	X		X		82.11	0.96	0.64	0.00	X	X	

5.2.3.1.3 Looking Further into the Number of Training Instances

The final characteristic that we investigated was the number of training instances that were used during the model building process. Training models on very few instances could result in a model that is over-fit to the specific training set, thus reducing the model's predictive performance on future data sets. Table 5.13 lists the remaining variations (after the reductions made in the last two sub-sections), along with the number of instances that were used during training. Many of these training sets were reduced drastically from the original 1385 sample

instances. This reduction was seen most with the balanced variations because the under-sampling technique removed excess instances to create a uniform class distribution.

Table 5.13 Top classifiers with number of training instances.

Emotional State	Reduction		Balanced		Training Instances	Evaluation		
	PCA	None	True	False		Silver	Gold	Platinum
Confidence		X	X		286	X		
Confidence	X		X		286	X		
Excited		X	X		282	X		
Hesitance		X	X		204	X	X	
Hesitance	X		X		204	X	X	
Nervous		X	X		152	X	X	
Nervous	X		X		152	X		
Relaxed		X	X		442	X		
Sad		X	X		156	X	X	X
Sad	X		X		156	X	X	
Tired		X		X	861	X	X	
Tired		X	X		758	X	X	
Tired	X			X	861	X		
Tired	X		X		758	X	X	

The variations from Table 5.13 that were less likely to be affected by over-fit were the ones with the larger number of training instances. With this in mind, we considered two different emotional state models, relaxed and tired as our best classifiers from this data set. The relaxed variation was balanced, used two class-levels, with no reductions made, and it had an overall classification rate of 79.5% which is in the silver category and just shy of the Gold boundary. There were four different variations that were left for the tired model with the best result coming from an unbalanced, two class-levels, with no reductions made, and with a classification rate of 84.2% which placed it near the top of the Gold category.

5.2.3.1.4 Decision Tree Structure

In this section, we look at the structure of our best classifier, the unbalanced tired model that uses two class-levels with no further attribute reductions. Figure 5.8 illustrates the entire decision tree for this classifier with 18 leaves and 35 nodes in total. We can see that most of the values that

were used in each branch of the tree are very small. As we indicated in Section 5.2.1.1, this was likely due to strong outliers in the data that prevented the use of the full range between 0 and 1 when normalizing the data.

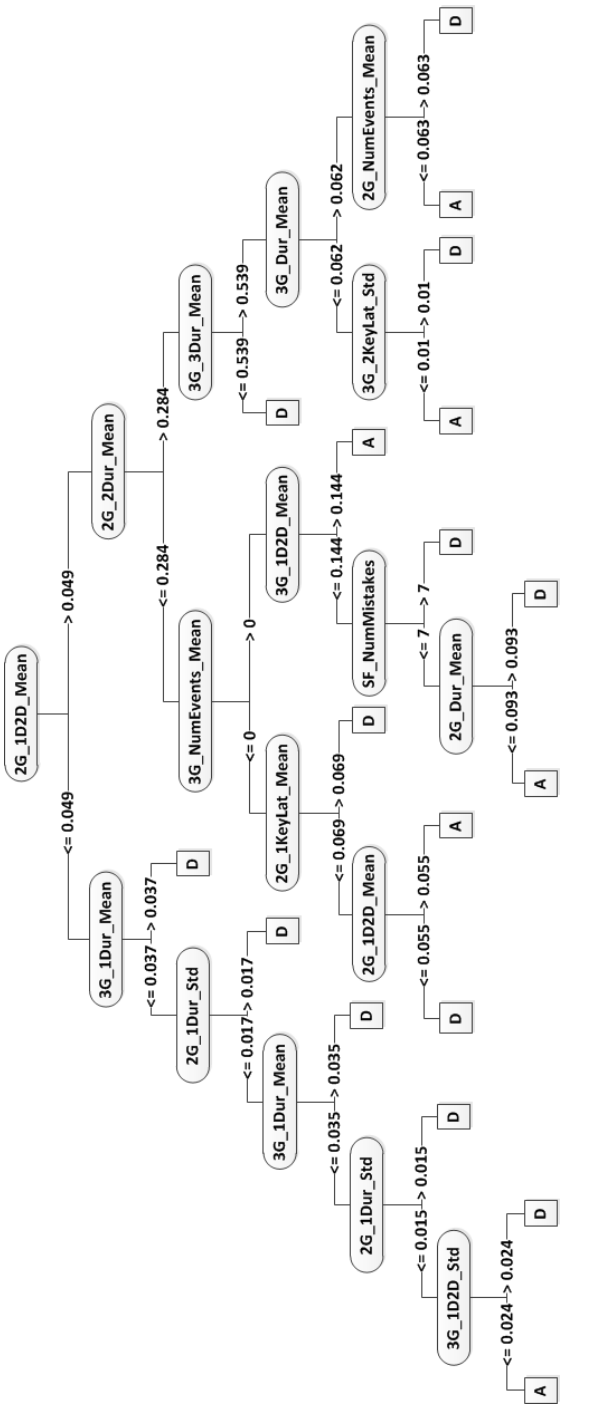

Figure 5.8 Decision tree structure of the fixed text, 2 class-level, unbalanced, non-PCA reduced classifier for the tired emotional state (A=Agree, D=Disagree).

Table 5.14 lists all of the nodes that were used in construction of the decision tree in Figure 5.8. There were a total of 14 different features used from the original set of 31. Most of them consist of the mean values that were calculated of the features, with only 2 of the included classifier features using standard deviation. These features included an approximately even number of digraph and trigraph features. Table 5.15 lists the remaining 17 features that were in the original feature set but were not used in the structure of the final classifier.

Table 5.14 Features included in the tired decision tree classifier.

Feature	Nodes
2G_1D2D_Mean	2
2G_1Dur_Std	2
2G_1KeyLat_Mean	1
2G_2Dur_Mean	1
2G_Dur_Mean	1
2G_NumEvents_Mean	1
3G_1D2D_Mean	1
3G_1D2D_Std	1
3G_1Dur_Mean	2
3G_2KeyLat_Std	1
3G_3Dur_Mean	1
3G_Dur_Mean	1
3G_NumEvents_Mean	1
NumMistakes	1

Table 5.15 Features not used in the tired decision tree classifier.

Feature	Nodes
2G_1D2D_Std	0
2G_1Dur_Mean	0
2G_1KeyLat_Std	0
2G_2Dur_Std	0
2G_Dur_Std	0
2G_NumEvents_Std	0
3G_1Dur_Std	0
3G_1KeyLat_Mean	0
3G_1KeyLat_Std	0
3G_2D2D_Mean	0
3G_2D2D_Std	0
3G_2Dur_Mean	0
3G_2Dur_Std	0
3G_2KeyLat_Mean	0
3G_3Dur_Std	0
3G_Dur_Std	0
3G_NumEvents_Std	0

5.2.4 Summary

In the last section we took our top classification results that were reported in Table 5.13 and reduced them until we came up with two types of models based on the relaxed and tired emotional states. Although we reduced our initial list from 9 different types of models down to 2, the remaining 7 models should still be considered for future studies on keystrokes dynamics and emotional state. These included the states of anger, confidence, excitement, hesitance, nervousness, sadness, and valence. At this point, these emotional states still show potential for creating models using keystroke dynamics given a larger data set.

The remaining models of arousal, boredom, distraction, focus, frustration, happiness, overwhelmed, and stress did not perform well according to the selected features in our dataset. However, there are still a number of other features that can be analyzed in our existing feature set; we leave this as an exercise for future analysis.

CHAPTER 6

DISCUSSION

This chapter summarizes and discuses our findings from the results presented in Chapter 5. We introduce potential applications of this technology and we talk about some of the lessons that we learned from this research. We finish with a discussion on possible future directions and extensions that could be taken.

6.1 SUMMARY OF FINDINGS

In this section, we summarize our findings from our experience-sampling study. We start by reviewing our top-performing models based on the evaluation categories (Bronze, Silver, Gold, and Platinum) described in Chapter 5. We then discuss the best-performing emotional state models based on participants' keystroke rhythms. Following this, we discuss the effects of using PCA reduction versus no attribute reduction, fixed text versus free text keystrokes, variations on class levels, and the class distribution balancing that was performed.

6.1.1 Emotional States

According to our evaluation categories, we had 35 classifier variations in the Bronze category with representations from the anger, confidence, excitement, hesitance, nervousness, relaxation, sadness, tired, and valence states. The Silver category reduced this to 14 variations, which included models for confidence, excitement, hesitance, nervousness, relaxation, sadness, and tired. There were eight variations in the Gold category with models for hesitance, nervousness,

sadness, and tired. Finally, the Platinum category had one model representing the sad emotional state.

Of these models, we further decomposed them by looking at aspects in the data set such as class skew, class variance in the balanced variations, and the number of instances per training variation. From this we concluded that the top two emotional models were the tired and relaxed states with classification rates of 84.2% (0.68 Kappa) and 79.5% (0.59 Kappa) respectively.

We considered the models that qualified for our evaluation categories but not our top two evaluation categories as good candidates for further study given a larger data set. This included the following states: anger, confidence, excitement, hesitance, nervousness, sadness, and valence. In each of these categories our models did have some positive results; however, more data is required due to the class skew or limited data set.

The emotional states that seemed to have weaker classifiers included frustration, focus, happiness, overwhelmed, stress, distraction, and boredom. However, this should not rule out these states from future analysis in keystroke dynamics. It could be that we need features different from those collected or perhaps these particular emotional states have uneven distributions in our data set.

There has been related research with positive results in detecting user stress [57] from keystroke dynamics. Our analysis does not support these findings; however, we used different feature sets with a different data collection methodology that caused class skew as well as small data sets for some of our models. Therefore, these results should not eliminate the possibility that these emotional states can be identified using keystroke dynamics with a different feature set or data that is less skewed for those particular states.

6.1.2 Principle Components Analysis

Half of our variations included attribute reduction using PCA with the other half using no attribute-reduction technique. We did not see consistent differences when comparing the classification rate and kappa statistics for these two approaches. Although the PCA reduction

was successful in reducing the number of attributes, from 31 and 37 to 13 and 15 respectively, the resulting attributes were more difficult to identify, making the decision trees difficult to interpret (the attribute names consisted of all the attributes and their values concatenated into long strings and exceeding the maximum length in WEKA). The impact of PCA reduction on decision tree interpretability depends on the intended use of the final decision tree model. For example, if one wanted to inspect the classifier (in our case a decision tree) at great detail, the PCA may hinder the process as it combines many features into a combination of features represented as a single feature. We did not see a clear advantage to using this attribute-reduction technique for the attributes that we selected for this analysis; however, PCA may still be useful on different attribute sets as an effective way to reduce the dimensionality of a feature set with minimal loss of fidelity.

6.1.3 Fixed versus Free Text Keystrokes

The data set was divided into separate models for both free and fixed text. According to the 10-fold cross-validation results, there were no successful free text classifiers that fit our evaluation category definitions. This may have been a result of the density of the free text data. Although there were very similar mean number of keystrokes per sample period for fixed text (166 characters) and free text (169 characters), there were 265 sample periods where there was no keyboard activity, and thus no free text activity.

There are two situations in which the user could have prompted these sample periods with little or no free text entered. The fixed text interface ensured a minimum number of keystrokes for each collection period; however, the free text may contain a variable amount of data depending on the type of the participant's activity immediately prior to the collection process. Although the mean number of keystrokes were similar, the range of samples varied greatly for free text (standard deviation of 302.8) when compared to the fixed text (standard deviation of 9.9).

Alternatively, there could have been a significant amount of mouse activity causing the activity monitor to trigger the sample collection since the activity monitor was designed to trigger based on both keyboard and mouse activity. Although this activity monitor was designed to ensure that

there was at least some keyboard activity, an error in the data collection software did not enforce this minimum threshold. This bug has been fixed in the data collection software; however, the threshold should be fine tuned for future studies of free text.

6.1.4 Class Breakdown

Our variations were separated into three different class-levels: two, three, and five target classes. Of these groupings, none of the five-class models created classifiers with classification rates high enough to reach our evaluation categories. Although we did have three-class models that made it into the Bronze category, the top three categories (Silver, Gold, and Platinum) consisted of only two-class models. A potential problem with using the two class-level data set was that it reduced the number of instances that we could use to train our models. This may have had a negative impact on some of the models' predictive performance as data mining typically requires large data sets.

6.1.5 Balanced versus Unbalanced Class Distributions

Finally, the remaining variation used an under-sampling technique on the data set that we called balancing. Balancing had a dramatic effect on the number of samples that we could use to train because it removed instances of the majority classes to match the number of instances in the minority class. This exacerbated the existing problem of having a small data set by further reducing the number of instances. Further compounding this problem was the additional reduction for the two-class variations (mentioned in the previous section) where all instances with the neutral target class were removed.

Most of the classifiers in our top three categories were variations that used balancing; however, there were two variations that were created with the 'tired' models that did not use balancing. This was most likely due to the 'tired' class distribution having an approximately even naturally-occurring distribution in the data set. This means that although we did artificially modify the class distribution in some models, there were still successful models that used the original distribution of the data set.

6.2 LESSONS LEARNED

In this section we will discuss what we have learned starting with the limitations of this research. We discuss our analysis approach where we looked at aggregate features and compare it to an individualistic approach where separate models could be tailored to each individual. We then discuss the pros and cons of using the experience-sampling methodology for research in emotional state recognition.

6.2.1 Limitations

As we saw from our results, none of the free text classifiers made it into our top evaluation categories. As we speculated previously in this chapter, this may have been due to the activity monitor getting triggered too quickly for a data collection period; the activity monitor is triggered by both keyboard and mouse events. Although precautions were put in place that required a minimum level of keyboard activity for a collection period to start, the threshold should probably be increased in future studies. This threshold is configurable in the data collection software and could be altered to ensure that enough free text data is provided in order to create classifiers.

Another limitation of this study was the reliance on participants' self reports for the ground-truth of the emotional state. This method is entirely subjective and it is possible for participants to incorrectly identify their emotional states, be it unintentional or otherwise. Objective methods of identifying affect exist; however, most of these methods use specialized expensive equipment that does not lend itself to the experience-sampling methodology.

In some instances, our data set contained very few examples of certain classes and an over abundance of samples for other classes. This caused class skew problems where classifiers that were trained with this original data set were biased to the class with the most instances in it, resulting in poor classification (if at all) of other classes. These types of distributions were expected as we were not inducing emotions in participants, but simply allowing emotions to emerge in an uninfluenced manner. Under these circumstances, uniform distributions would be

unlikely to occur across all of the classes in all of the emotional states. For instance, it would be improbable for most people to be very angry as often as they are neutral or not at all angry.

To alleviate this class skew problem, we employed an under-sampling technique where we adjusted the class distribution by removing instances of the majority class at random. Although this was successful in making many of our classifiers more sensitive to the minority class, this is artificially modifying the distribution of the original data which may lead to unexpected behavior in a real-world situation. However, we did see one classifier in our overall results that performed well without this modification which shows that we are still able to model these emotional states without resorting to under-sampling.

Another limiting factor was the size of our data set both before and after our data processing, which had further reduced our data set in some cases (e.g. class distribution balancing). Data mining typically requires a large amount of training data to be able to create classifiers with any predictive capabilities. This data set size problem was exacerbated by our attempt to handle class skew in the data using under-sampling, which removed a substantial number of instances from our data set in certain variations of our training sets. With a larger sample set and possibly other methods of class skew adjustments, the models created here could have the potential to generate higher and more generalizable classification rates.

Another limitation to using this methodology was the reliance on a model that requires a training period. The model needs a sufficient amount of training examples in order to create a well-performing model. This training period is time consuming, due to the need to collect the training data. Furthermore, behavior biometrics like keystroke rhythms can change over time. In a longer study or in a real-world application, it may be necessary for a solution that is adaptable to accommodate for gradual changes in the user's behavior. However, the difficulty lies not only in the time it takes to train a new model (and when to do it) but also in how to get the new labeled data into in the model with minimal interruption to the user. Keystroke dynamics in authentication systems avoid this issue as it would be clear to the system who would be logged in after the initial authentication phase.

6.2.2 Aggregate Analysis

In order to reduce the vast number of features that we originally extracted and variations that we could look at, we ended up making the decision to use only aggregate features. Looking at specific digraph features would have resulted in too many features and many missing values for graph features that only appeared sporadically in the keystroke data. This meant that we did not use many of the features that we originally extracted during analysis; features that may be strong indicators of particular emotional states.

Additionally, we decided to create emotional state models based on keystrokes from all participants in the study. For example, the anger model was trained on all of the participants' keystrokes and was not specific to individuals. We did this to compensate for the lack of data that we obtained from each individual on average which could have resulted in models with poor predictive capabilities.

6.2.3 Individual Analysis

Even with the same attributes used in our aggregate analysis, an analysis at the participant level could potentially produce even better results than the models described here. Since keystroke dynamics are used in authentication systems to identify particular users' unique keystroke rhythms [3,43], individual models were created for each user to accommodate for these unique keystroke rhythms. It is possible that different emotional states cause different timing changes in different individuals. Given enough samples per participant, an individual-level analysis would be able to create separate models tailored to each participant's keystroke timings and possibly create better-performing classifiers.

The data could also be broken down by some other data point that was extracted from the data set. For instance, there has been recent research that suggests that you can distinguish male and female users based on their keystroke dynamics alone [23]. This could have implications for this research in that creating separate models for the different sexes could lead to better performing models as these models may have less variability. Apart from the analysis here, we had tried to verify this with our data set; however, we found that this had little effect on our overall

classification. This could be because there were only two females in the participant pool that we used for our analysis. A more balanced representation from the two sexes may provide better evidence than what was available in our data set.

6.2.4 Experience-Sampling Methodology

In this section we present the advantages and disadvantages of using an experience-sampling methodology for gathering keystroke and emotional state data.

6.2.4.1 Advantages

There are three main advantages to using the experience-sampling approach.

First, this methodology is less contrived than explicitly inducing emotions into participants as would be done in a laboratory study. It maximized the ecological validity by collecting emotional states in situ and with minimal interruption.

Second, the experience-sampling approach provides the ability to experiment across a larger range of emotions than a traditional mood-induction experiment would allow. In exploratory research such as ours, this advantage allows us to test a wide range of emotional states at once and narrow down the states that provide better models. Testing all 15 different emotional states that we gathered here in a laboratory setting would have been time consuming and taxing on the participants as they would need to be induced into each one of these states (perhaps several times).

Third, this methodology in conjunction with remote data collection makes this study much easier to administer than a laboratory study with the same number of participants and across the same range of emotional states. This required less time to administer and participants were able to avoid taking time to come into the laboratory which can be difficult for participants that have busy schedules.

6.2.4.2 Disadvantages

We found two main disadvantages of using experience sampling to collect data.

First, due to the uncontrolled nature of our study, we could not create a balanced set of conditions from which we could gather a near uniform distribution of classes for each emotional state. As mentioned previously, this resulted in many of our class distributions being skewed towards certain classes and introduced bias into some of our resulting models. This caused secondary problems where we had to adjust for this class skew, which resulted in data loss.

Second, although the experience-sampling methodology was nice to get a broad understanding of a wide variety of emotional states, we did not get the number of instances that we needed in some emotional categories. For this initial phase, experience sampling was beneficial in identifying the emotional states that could create the best classifiers. However, future studies may want to consider using additional alternative techniques on some of the emotional states identified in this research. For example, a controlled mood induction experiment could be used on a single emotional state that produced good results in this work (e.g. tired); a more targeted approach such as this could generate a larger data set that could be used for creating better performing models or individual models for each participant.

6.3 FUTURE WORK

There are many different avenues that can be taken that extend this research. In this section, we discuss some of these different possibilities.

6.3.1 Existing Data Set

There are still many aspects of the current data set that can be explored without the need for an additional study. Although we used decision trees to build our classifiers, there are many other data mining techniques that could be useful and that may better fit the unique characteristics of this data set. For instance, support vector machines have been increasing in popularity recently and could be used for further analysis instead of decision trees. The architecture of the scripts developed for training our models can easily be modified to accommodate other algorithms for

training. In this way, a variety of machine learning algorithms could be attempted and compared against the same data set.

Another approach that could be taken with the current data set is to balance the target classes using an alternative method to under-sampling. The under-sampling technique used in our research was found to drastically reduce the number of instances that were available for use during training. Over-sampling could be used in place of our under-sampling technique as a way to keep as much data as possible for building our classifiers while adjusting for class skew. In over-sampling, the data in the minority classes are replicated in order to create a more even distribution amongst the classes. However, over-sampling may have other unintended side-effects, such as possible over-fitting of the model to the specific data set as there would be more near duplicate instances in the data [13].

Using the current data set, there are also a number of different features and variations that could be explored that were not included as part of this initial analysis. We included only a small portion of the features that we extracted during our data processing; additional combinations of existing features may build better models of the emotional states that we collected. The raw data set could also be used to create additional features not included in this study.

Different levels of analysis could be performed on the existing data set. There may be some overall features and data points (e.g. sex of participant) that could be separated into different models potentially creating better performing classifiers for some states. The application context would be a good candidate for further analysis by filtering out the data that was unlikely to contain English text. For example, video game usage may contain atypical keyboard usage such as extensive use of the directional arrow keys or the ‘w’, ‘a’, ‘s’, ‘d’ keys that are used for commands in many computer games.

Another data point that could be used to split the data set might be the typing proficiency of the user. In previous studies in keystroke dynamics, it was reported that poor typists had great variation in their own typing compared to the minimal variation in expert typists [5]. Splitting out the data set into different levels of typing proficiency may provide better models for each

proficiency level and better global models. The typing proficiency of the user was collected as part of the initial demographic survey, but it could also be determined using the timing values for the fixed text entry as well.

A different outlier removal approach from the one described in section 4.1.1 could also be used to increase the performance of our models. Instead of the single-value threshold that was used in our results, the scripts could be modified to try different threshold values and then plot the classification rates at each level to find the optimum threshold value. Furthermore, the outlier removal process could use a similar approach as in [9,27], where any data that was three standard deviations away from the mean was removed and the mean and standard deviations on the remaining data were recalculated again.

Finally, we could take a different approach to establishing ground truth by using the text data itself. Performing a textual analysis of the free text data to identify the affective state of the participants using a linguistic analysis tool such as the Linguistic Inquiry and Word Count (LIWC) program could establish ground truth. LIWC considers word frequency in text, associating different categories of words along 70 different dimensions of language including positive and negative emotion words [36]. The results extracted from LIWC could then be used to verify the subjective ratings provided by our participants or to use the results as the ground truth when building new models. Alternatively, these linguistic features may also be used as features in the model.

Some additional preprocessing would be necessary in order to accommodate for the data format required by the LIWC program. The text that the user entered would have to be extracted from the keystroke data, taking care to fix any spelling mistakes or to accommodate for any corrections that the user may have performed (e.g. deletions). This process may not be able to be automated due to the requirement of choosing the correct word during the spell-check of the data.

6.3.2 Future Studies

Although we focused on an analysis that aggregated over all participants, we mentioned in the limitations section that an individual-level of analysis could be beneficial and could result in higher classifications due to each person's keystroke having a unique signature. However, more data would be needed for each participant in order to make this option possible. Future studies may want to look at conducting the experiment over a longer period of time or perhaps giving the participants more incentives (or other forms of encouragement) to keep submitting data throughout the study's duration.

Although none of free text models made our top evaluation categories, free text has the most potential benefit for real-world applications. We saw similar conditions (worse results for free text compared to fixed text) as in [43] when we compared the fixed and free text models' classification rates. Monrose and Rubin stated that this variable in keystrokes is likely due to operational conditions during the data collection process, uncooperative participants, or perhaps participants influenced by an 'emotionally charged' situation [43]. If this was the case, this free text may have a greater potential for revealing traces of the users' emotional state.

More data would be needed for any future study that focuses on free text features. To avoid similar problems that existed in this study, the activity monitor that triggers the data collection period would have to be modified, extending the number of keystrokes required before a collection period engaged. The menu option that allowed the participant to initiate a collection period could also be disabled until a threshold of free text keystrokes had been met or exceeded.

Another option that we could use would be to change the activity monitor to trigger only once a certain number of digraphs/trigraphs have been collected. In Bergadano et al. [3], they state that multiple instances of the same digraph did little to improve the performance of their model. However, multiple instances of different digraphs provided better performing classifiers than when using multiple samples for the same digraph. This could reduce the number of times that the participant was interrupted and therefore lead to better compliance.

Future research could focus deeper into the particular emotional states that we identified in our top evaluation categories using a controlled laboratory approach. Mood induction could be used to gather numerous samples for one or two particular emotional states only; gathering more samples to create better-performing models. The class skew problem that we observed could be avoided using such a technique because it should be possible to control the number of instances in each one of the classes as long as the technique used for mood induction is effective. This would avoid some of the problems that we saw in our data set; however, acquiring a sufficient number of instances could be time consuming depending on the induction technique used.

Pressure sensors have been previously shown to indicate emotional states in users and could provide some additional benefit in this research [51]. In a more controlled study, a pressure-sensor keyboard could also be used to gather pressure data from each keystroke. In [20], additional pressure-based features could be extracted and trained with the previous features to achieve better-performing emotional state models. However, this approach introduces a limitation; users must have a pressure-sensitive keyboard. This reduces the applicability of this approach as these keyboards are not widely used.

Regardless of the future directions that this could take, there are ethical issues that need to be addressed which we look at in the next, final section of this chapter.

6.4 POTENTIAL FOR APPLICATION

Affective computing applications, such as the ones that could be derived from this research, have important ethical concerns for user privacy [18,45]. As we pointed out earlier, the furtive aspect of using key loggers to collect data make it a good candidate for determining emotional states because it will likely go undetected and the user's emotional state will thereby be unaffected by the data collection and modeling process. However, the unobtrusiveness of this technology is also a cause for concern when it comes to user privacy.

The ability to use this technology without the user's knowledge provides the possibility of using these techniques to gather information from the user without their consent. This would be effectively be digitizing the user's emotions, creating digital copies of their emotional states at different times when they are using the computer. As with any other type of digital information, it could be easily replicated, stored, or shared indefinitely. Furthermore, once this information is in a digital form, the content creator (the user) no longer has control over what that data can be used for (e.g. targeted advertising or worse).

Due to these ethical considerations, care must be taken in the application of this research in real-world applications. Users should be informed when and how their data is being used and they should have the option to maintain their privacy.

CHAPTER 7

CONCLUSION

The current methods of measuring emotional states of computer users use expensive, specialized, and invasive technologies not found in typical home or office settings. These methods typically use either invasive sensors that affix to the user's body or other technologies such as thermal imaging that use equipment that is expensive and only used by a few select individuals or organizations. These attributes severely limit many real-world applications of Affective Computing solutions.

Our solution involved analyzing users' typing rhythms or keystroke dynamics in order to identify their current emotional state. The main benefit from using this type of affect recognition was the covert nature in which it could be applied. In invasive affective solutions, the user's awareness that they are being recorded may alter their emotional state causing interference in detecting their true emotions.

A secondary benefit of using keystroke dynamics is that the required equipment, any standard keyboard, is inexpensive and already widely used on most computer systems. This provides an excellent opportunity to implement Affective Computing solutions in a technology that is on nearly all computers today.

We created custom keystroke logging software that was used in a field study to collect the data. An experience-sampling methodology was used for sampling, in which participants self-reported on their feelings across 15 different emotional states. These responses were used as the ground-truth when building separate emotional state models based on features extracted from the keystroke data.

6.5 CONTRIBUTIONS

In this section, we briefly recap our four major contributions from this work.

First, we provided a methodology that can be used for creating emotional state models based on their keystroke timings. This methodology involved creating an extensive set of keystroke timing features as well as a number of other data points that can be used during classification. In addition to the 68 features that we used in our analysis, there were over 100 000 other features that could be used in future studies either on the data set collected here or any study on keystroke dynamics. Our solution used decision trees to create our models; however, there are many other forms of supervised machine learning that could be easily incorporated into the system.

Second, we described an ecologically valid approach to gather our keystroke and emotional state data. This involved using an experience-sampling methodology in a field study that we conducted. This was a unique approach to gathering affective state data because we measured affect as the participants perform their daily tasks and participants' emotional states changed naturally (without outside interference from the researchers). This method was very different than the traditional approach in measuring affect where emotions are induced in a laboratory setting.

Third, we created good models of two different affective states: relaxed and tired. Our best resulting model for 2 class-levels of the relaxed state had a classification rate of 79.5% and a Kappa statistic of 0.59. The best model for 2 class-levels of the tired state obtained a classification rate of 84.2% with a Kappa statistic of 0.68.

Fourth, we identified other affective states that show potential for providing good classifiers using our feature set. These include the following emotional states: anger, confidence, excitement, hesitance, nervousness, sadness, and valence. These emotional states show potential for creating promising results given a larger data set because models created for these emotional states achieved strong performance, but were removed due to characteristics of the collected data (e.g. class skew).

6.6 SUMMARY

Emotionally intelligent computers would be able to make better decisions based on the additional information of the emotional context of a situation. The first step in achieving any emotional computer application is to provide computer systems with some capacity to recognize affective state.

To recognize emotional states, we needed a method of measuring affect that is unobtrusive to users so that we had little to no influence over their affective state. It was also desirable to have a system that could be easily implemented using standard computer equipment. Although we had an idea of which emotional states to look at, we needed a data collection methodology that would provide us with data across many different emotional states to reduce the particular emotional states that could be most easily identified through keystroke dynamics.

We suggested that emotional states could be determined by using keystroke features of individual users as input to train models of their emotional states. We created custom key logging software that was used to collect the keystroke data in an unobtrusive manner potentially reducing the effects of the users' knowledge of being recorded. This method of recognizing affect had the added benefit of being widely applicable using inexpensive, standard equipment. Our experience-sampling data collection methodology allowed us to collect data on a wide range of emotional states, aiding the exploration of this emerging area of research.

The classifiers that we created along with our analysis of the performance of these classifiers displays that it is possible to identify user affective states using keystroke dynamics and using technology that is in wide spread use today.

LIST OF REFERENCES

[1] Admit One Security. *http://www.admitonesecurity.com*. Accessed: 03-21-2009.

[2] Bergadano, F., Gunetti, D. et al. 2003. Identity verification through dynamic keystroke analysis. *Intell. Data Anal.* 7, 5 (2003), 469-496.

[3] Bergadano, F., Gunetti, D. et al. 2002. User authentication through keystroke dynamics. *ACM Trans. Inf. Syst. Secur.* 5, 4 (2002), 367-397.

[4] Bird, S., Klein, E. et al. 2009. *Natural Language Processing with Python.* O'Reilly Media Inc.

[5] Bleha, S., Slivinsky, C. et al. 1990. Computer-Access Security Systems Using Keystroke Dynamics. *IEEE Trans. Pattern Anal. Mach. Intell.* 12, 12 (1990), 1217-1222.

[6] Brown, M. and Rogers, S.J. 1993. User identification via keystroke characteristics of typed names using neural networks. *Int. J. Man-Mach. Stud.* 39, 6 (1993), 999-1014.

[7] Cannon, W. 1927. The James-Lange Theory of Emotions: A Critical Examination and an Alternative Theory. *The American Journal of Psychology*. 39, 1 (1927), 106-124.

[8] Carroll, L. 2008. *Alice's Adventures in Wonderland*. The Gutenberg Project.

[9] Clarke, N.L. and Furnell, S.M. 2006. Authenticating mobile phone users using keystroke analysis. *International Journal of Information Security*. 6, 1 (2006), 1-14.

[10] Csikszentmihalyi, M. and Larson, R. Validity and reliability of the Experience-Sampling Method. *The Journal of Nervous and Mental Disease*. 175, 9, 526-536.

[11] De Silva, L. Real-time Facial Feature Extraction and Emotion Recognition.

[12] Dowland, P. and Furnell, S. 2004. A Long-term Trial of Keystroke Profiling Using Digraph, Trigraph, and Keyword Latencies. *IFIP International Federation for Information Processing*. Springer Boston. 275-289.

[13] Drummond, C. and Holte, R. 2003. C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling beats Over-Sampling. (Washington, 2003).

[14] Forgas, J. 1995. Mood and judgement: The affect infusion model (AIM). *Psychological Bulletin*. 117, 1 (1995), 39-66.

[15] Gaines, H.F. 1939. *Cryptanalysis*. Dover Publications.

[16] Gaines, R., Lisowski, W. et al. 1980. Authentication by Keystroke Timing: some preliminary results. Rand Corporation.

[17] Gunetti, D. and Picardi, C. 2005. Keystroke analysis of free text. *ACM Trans. Inf. Syst. Secur.* 8, 3 (2005), 312-347.

[18] Gunetti, D., Picardi, C. et al. 2005. Keystroke Analysis of Different Languages: A Case Study. *Lecture Notes in Computer Science*. Springer Berline / Heidelberg. 133-144.

[19] Gutierrez, F.J., Lerma-Rascon, M.M. et al. 2002. Biometrics and Data Mining: Comparison of Data Mining-Based Keystroke Dynamics Methods for Identity Verification. *Lecture Notes in Computer Science*. 221-245.

[20] Hai-Rong Lv and Wen-Yuan Wang 2006. Biologic verification based on pressure sensor keyboards and classifier fusion techniques. *Consumer Electronics, IEEE Transactions on*. 52, 3 (2006), 1057-1063.

[21] Hancock, J.T., Gee, K. et al. 2008. I'm sad you're sad: emotional contagion in CMC. *Proceedings of the 2008 ACM conference on Computer supported cooperative work* (San Diego, CA, USA, 2008), 295-298.

[22] Hektner, J., Schmidt, J. et al. 2007. *Experience Sampling Method: Measuring the Quality of Everyday Life*. Sage Publications.

[23] indyposted. *http://indyposted.com/14990/to-catch-a-predator-with-keystrokes/*. Accessed: 06-02-2010.

[24] Intille, S.S., Rondoni, J. et al. 2003. A context-aware experience sampling tool. *CHI '03 extended abstracts on Human factors in computing systems* (Ft. Lauderdale, Florida, USA, 2003), 972-973.

[25] Jain, A. and Ross, A. 2008. Introduction to Biometrics. *Handbook of Biometrics*. Springer US.

[26] Jonathan, I., Lazar, J. et al. 2002. Determining Causes and Severity of End-User Frustration. *International Journal of Human-Computer Interaction*. 17, (2002), 333-356.

[27] Joyce, R. and Gupta, G. 1990. Identity authentication based on keystroke latencies. *Commun. ACM*. 33, 2 (1990), 168-176.

[28] Kapoor, A., Burleson, W. et al. 2007. Automatic prediction of frustration. *Int. J. Hum.-Comput. Stud.* 65, 8 (2007), 724-736.

[29] Kapoor, A. and Horvitz, E. 2008. Experience sampling for building predictive user models: a comparative study. *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (Florence, Italy, 2008), 657-666.

[30] Keystroke dynamics - Wikipedia, the free encyclopedia. *http://en.wikipedia.org/wiki/Keystroke_dynamics*. Accessed: 06-05-2010.

[31] Khan, M.M., Ingleby, M. et al. 2006. Automated Facial Expression Classification and affect interpretation using infrared measurement of facial skin temperature variations. *ACM Trans. Auton. Adapt. Syst.* 1, 1 (2006), 91-113.

[32] Lang, P. 1980. Behavioral treatment and bio-behavioral assessment: Computer applications. *Technology in mental health care delivery systems*. (1980), 119-137.

[33] Lang, P. 1995. The emotion probe. *American Psychologist*. 50, 5 (1995), 372-385.

[34] Lazar, J., Jones, A. et al. 2006. Severity and impact of computer user frustration: A comparison of student and workplace users. *Interact. Comput.* 18, 2 (2006), 187-207.

[35] Leggett, J. and Williams, G. 1988. Verifying identity via keystroke characteristics. *Int. J. Man-Mach. Stud.* 28, 1 (1988), 67-76.

[36] LIWC: Linguistic Inquiry and Word Count. *http://www.liwc.net*. Accessed: 05-10-2010.

[37] Mandryk, R., Inkpen, K. et al. 2006. Using psychophysiological techniques to measure user experience with entertainment technologies. *Behaviour & Information Technology (Special Issue on User Experience)*. 25, 2 (2006), 141-158.

[38] Mandryk, R.L. and Atkins, M.S. 2007. A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies. *Int. J. Hum.-Comput. Stud.* 65, 4 (2007), 329-347.

[39] Martin, M. 1990. On the induction of mood. *Clinical Psychology Review*. 10, 6 (1990), 669-697.

[40] McMillan, R. 2006. How Your "Fist" Can Talk. *CIO: Business Technology Leadership*.

[41] Moneta, G. and Csikszentmihalyi, M. 1996. The effect of perceived challenges and skills on the quality of subjective experience. *Journal of Personality*. 64, 2 (1996), 275-310.

[42] Monrose, F. and Rubin, A. 1997. Authentication via keystroke dynamics. *Proceedings of the 4th ACM conference on Computer and communications security* (Zurich, Switzerland, 1997), 48-56.

[43] Monrose, F. and Rubin, A.D. 2000. Keystroke dynamics as a biometric for authentication. *Future Gener. Comput. Syst.* 16, 4 (2000), 351-359.

[44] Partala, T., Surakka, V. et al. 2006. Real-time estimation of emotional experiences from facial expressions. *Interact. Comput.* 18, 2 (2006), 208-226.

[45] Picard, R.W. 2007. *Affective Computing*. MIT Press.

[46] Puri, C., Olson, L. et al. 2005. StressCam: non-contact measurement of users' emotional

states through thermal imaging. *CHI '05 extended abstracts on Human factors in computing systems* (Portland, OR, USA, 2005), 1725-1728.

[47] Russell, J. 2003. Core affect and the psychological construction of emotion. *Psychological Review*. 110, 1 (2003), 145-172.

[48] Schachter, S. 1964. The interaction of cognitive and physiological determinants of emotional state. *Advances in Experiemental Psychology*. 1, (1964), 49-80.

[49] Sheng, Y., Phoha, V. et al. 2005. A Parallel Decision Tree-Based Method for User Authentication Based on Keystroke Patterns. *IEEE Transactions on Systems, Man, and Cybernetics*. 35, 4 (2005), 826-833.

[50] Stern, R.M., Ray, W.J. et al. 2001. *Psychophysiological recording*. Oxford University Press.

[51] Sykes, J. and Brown, S. 2003. Affective gaming: measuring emotion through the gamepad. *CHI '03 extended abstracts on Human factors in computing systems* (Ft. Lauderdale, Florida, USA, 2003), 732-733.

[52] Tsiamyrtzis, P., Dowdall, J. et al. 2007. Imaging Facial Physiology for the Detection of Deceit. *Int. J. Comput. Vision*. 71, 2 (2007), 197-214.

[53] Type Sense. *http://www.deepnetsecurity.com*. Accessed: 03-21-2009.

[54] Vacca, J. 2007. *Biometric technologies and verification systems*. Butterworth-Heinemann.

[55] Velten, E. 1968. A laboratory task for induction of mood states. *Behaviour Research and Therapy*. 6, 4 (1968), 473-482.

[56] Villani, M., Tappert, C. et al. 2006. Keystroke Biometric Recognition Studies on Long-Text Input under Ideal and Application-Oriented Conditions. *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop* (2006), 39.

[57] Vizer, L.M. 2009. Detecting cognitive and physical stress through typing behavior. *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems* (Boston, MA, USA, 2009), 3113-3116.

[58] Vizer, L.M., Zhou, L. et al. 2009. Automated stress detection using keystroke and linguistic features: An exploratory study. *Int. J. Hum.-Comput. Stud.* 67, 10 (2009), 870-886.

[59] Westermann, R., Spies, K. et al. 1996. Relative effectiveness and validity of mood induction procedures: a meta-analysis. *European Journal of Social Psychology*. 26, 4 (1996), 557-580.

[60] Witten, I. and Frank, E. 2005. *Data Mining: Practical machine learning tools and*

techniques. Morgan Kaufmann.

[61] Zimmermann, P., Gomez, P. et al. 2006. Extending usability: putting affect into the user-experience. *Proceedings of NordiCHI'06* (New York, 2006), 27-32.

[62] Zimmermann, P., Guttormsen, S. et al. 2003. Affective computing - a rationale for measuring mood with mouse and keyboard. *International Journal of Occupational Safety and Ergonomics*. 9, 4 (2003), 539-551.

APPENDIX A

CONSENT FORMS

INFORMED CONSENT FORM

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF SASKATCHEWAN
INFORMED CONSENT FORM

Research Project: Using mouse and keystroke dynamics to identify affect

Investigators: Dr. Regan Mandryk, Department of Computer Science (966-4888)

Clayton Epp, Department of Computer Science (966-2327)

Mike Lippold, Department of Computer Science (966-2327)

This consent form is only part of the process of informed consent. Please print off this form for your personal records and reference. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

This study is concerned with detecting a user's affective state in a naturalistic setting. We will utilize experimental software that will run continually on your computer gathering keystroke and mouse data as well as the currently running applications as you go about your daily computer tasks. The goal of the research is to determine whether we can detect patterns in a user's keystrokes and mouse activity that identify a user's affective state.

The session will run for 4 to 8 weeks. During this time you will be asked periodically to fill out a questionnaire on your current mood. At the time of the questionnaire presentation, the program will display the collected keystrokes for the past 10 minutes, as well as the active application names. After viewing this information, you will be given the option to opt out of filling in the questionnaire and having your keystrokes recorded. After filling out the questionnaire, you will also be asked to enter a short text passage. You will again have the option to opt out at this point.

At the end of the session, you will be given more information about the purpose and goals of the study, and there will be time for you to ask questions about the research.

The data collected from this study will be used in articles for publication in journals and conference proceedings.

As one way of thanking you for your time, we will be pleased to make available to you a summary of the results of this study once they have been compiled (usually within two months). This summary will outline the research and discuss our findings and recommendations. If you would like to receive a copy of this summary, please check the box below.

☐ Yes, I would like to receive a copy of a summary of this study.

All personal and identifying data will be kept confidential. If explicit consent has been given, textual excerpts, photographs, or video recordings may be used in the dissemination of research results in scholarly journals or at scholarly conferences. Anonymity will be preserved by using pseudonyms in any presentation of textual data in journals or at conferences. The informed consent form and all research data will be kept in a secure location under confidentiality in accordance with University policy for 5 years post publication. Do you have any questions about this aspect of the study? Please email your questions to mousekeyfieldstudy@cs.usask.ca.

You are free to withdraw from the study at any time without penalty and without losing any advertised benefits.Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed. To withdraw from the study, send an email to mousekeyfieldstudy@cs.usask.caindicating that you would like to withdraw from the study.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your

participation. If you have further questions concerning matters related to this research, please contact:

Dr. Regan Mandryk, Assistant Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca

Clicking on the Accept button on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. If you have further questions about this study or your rights as a participant, please contact:

Dr. Regan Mandryk, Assistant Professor, Dept. of Computer Science, (306) 966-4888, regan@cs.usask.ca

Office of Research Services, University of Saskatchewan, (306) 966-4053

Please print off a copy of this consent form to keep for your records and reference. This research has the ethical approval of the Office of Research Services at the University of Saskatchewan.

Please enter the following information (this information will only be used to contact you regarding this study):

First name: *

Last name: *

Email address: *

TEXTUAL EXCERPT CONSENT FORM

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF SASKATCHEWAN
TEXTUAL EXCERPT CONSENT FORM

Research Project:	Using mouse and keystroke dynamics to identify affect
Investigators:	Dr. Regan Mandryk, Department of Computer Science (966-4888)
	Clayton Epp, Department of Computer Science (966-2327)
	Mike Lippold, Department of Computer Science (966-2327)

TEXTUAL EXCERPTS

"I, test test, agree to allow excerpts of text that I wrote to be used for public presentation of the research results in the manner described in the consent form. However, I understand that I will be given the opportunity to read any excerpts that are intended for public participation and to withdraw consent for them to be reported, if so desired. I also understand that I will receive a copy of any textual excerpts presented publically for my records. I understand that all identifying information will be removed from the excerpts and names will be changed prior to publication."

☐ I agree with the statement above.

Ensure to print a copy of this form for your records.

APPENDIX B

FIXED TEXT EXCERPTS

1	'Here! you may nurse it a bit, if you like!' the Duchess said to Alice, flinging the baby at her as she spoke. 'I must go and get ready to play croquet with the Queen,' and she hurried out of the room.
2	Alice remained looking thoughtfully at the mushroom for a minute, trying to make out which were the two sides of it; and as it was perfectly round, she found this a very difficult question.
3	The players all played at once without waiting for turns, quarrelling all the while, and fighting for the hedgehogs; and in a very short time the Queen was in a furious passion, and went stamping about.
4	Which would NOT be an advantage,' said Alice, who felt very glad to get an opportunity of showing off a little of her knowledge. 'Just think of what work it would make with the day and night!'
5	She was a good deal frightened by this very sudden change, but she felt that there was no time to be lost, as she was shrinking rapidly; so she set to work at once to eat some of the other bit.
6	Luckily for Alice, the little magic bottle was in full effect, and she grew no larger: still it was very uncomfortable, and, as there seemed to be no sort of chance of her ever getting out of the room again.
7	The Hare took the watch and looked at it gloomily: then he dipped it into his cup of tea, and looked at it again: but he could think of nothing better to say than his first remark, 'It was the BEST butter.'
8	This seemed to Alice a good opportunity for making her escape; so she set off at once, and ran till she was quite tired and out of breath, and till the puppy's bark sounded quite faint in the distance.
9	The table was a large one, but the three were all crowded together at one corner of it: 'No room! No room!' they cried out when they saw Alice coming. 'There's plenty of room!' said Alice indignantly.
10	Was I the same when I got up this morning? I almost think I can remember feeling a little different. But if I'm not the same, the next question is, Who in the world am I? Ah, that's the great puzzle!
11	Alice felt dreadfully puzzled. The Hatter's remark seemed to have no sort of meaning in it, and yet it was certainly English. 'I don't quite understand you,' she said, as politely as she could.
12	The great question certainly was, what? Alice looked all round her at the flowers and the blades of grass, but she did not see anything that looked like the right thing to eat or drink under the circumstances.
13	At this moment the door of the house opened, and a large plate came skimming out, straight at the Footman's head: it just grazed his nose, and broke to pieces against one of the trees behind him.
14	'I'm sure those are not the right words,' said poor Alice, and her eyes filled with tears again as she went on, 'I must be Mabel after all, and I shall have to go and live in that poky little house.'
15	So she swallowed one of the cakes, and was delighted to find that she began shrinking directly. As soon as she was small enough to get through the door, she ran out of the house, and found a crowd of animals.
16	Here was another puzzling question; and as Alice could not think of any good reason, and as the Caterpillar seemed to be very unpleasant, she turned away. 'Come back!' the Caterpillar called after her.
17	'That WAS a narrow escape!' said Alice, very frightened at the sudden change, but very glad to find herself still in existence; 'and now for the garden!' and she ran with all speed back to the little door.
18	'Well, then,' the Cat went on, 'you see, a dog growls when it's angry, and wags its tail when it's pleased. Now I growl when I'm pleased, and wag my tail when I'm angry. Therefore I'm mad.'

19 The next thing was to eat the comfits: this caused some noise and confusion, as the large birds complained that they could not taste theirs, and the small ones choked and had to be patted on the back.

20 This was not an encouraging opening for a conversation. Alice replied, rather shyly, I know who I was when I got up this morning, but I think I must have been changed several times since then.'

21 Poor Alice! It was as much as she could do, lying down on one side, to look through into the garden with one eye; but to get through was more hopeless than ever: she sat down and began to cry again.

22 Oh, please mind what you're doing!' cried Alice, jumping up and down in an agony of terror. 'Oh, there goes his PRECIOUS nose'; as an unusually large saucepan flew close by it, and very nearly carried it off.

23 It did so indeed, and much sooner than she had expected: before she had drunk half the bottle, she found her head pressing against the ceiling, and had to stoop to save her neck from being broken.

24 She noticed that one of the trees had a door leading right into it. 'That's very curious!' she thought. 'But everything's curious today. I think I may as well go in at once.' And in she went.

25 Alice was not a bit hurt, and she jumped up on to her feet: she looked up, but it was all dark overhead; before her was another long passage, and the White Rabbit was still in sight, hurrying down it.

26 The Caterpillar and Alice looked at each other for some time in silence: at last the Caterpillar took the hookah out of its mouth, and addressed her in a sleepy voice. 'Who are you?' said the Caterpillar.

27 I've seen hatters before,' she said to herself; 'the March Hare will be much the most interesting, and perhaps as this is May it won't be raving mad-at least not so mad as it was in March.'

28 Still she went on growing, and, as a last resource, she put one arm out of the window, and one foot up the chimney, and said to herself 'Now I can do no more, whatever happens. What WILL become of me?'

29 She was considering in her own mind whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.

30 She tried to look down and make out what she was coming to, but it was too dark to see anything; then she looked at the sides of the well, and noticed that they were filled with cupboards and book-shelves.

31 When the procession came opposite to Alice, they all stopped and looked at her, and the Queen said severely 'Who is this?' She said it to the Knight of Hearts, who only bowed and smiled in reply.

32 Suddenly she came upon a little three-legged table, all made of solid glass; there was nothing on it except a tiny golden key, and Alice's first thought was that it might belong to one of the doors of the hall.

33 The poor little thing was snorting when she caught it, and kept doubling itself up and straightening itself out again, so that altogether, for the first minute or two, it was as much as she could do to hold it.

34 It was the White Rabbit, trotting slowly back again, and looking anxiously about as it went, as if it had lost something; and she heard it muttering to itself 'The Duchess! The Duchess! Oh my dear paws!'

35 They all sat down at once, in a large ring, with the Mouse in the middle. Alice kept her eyes anxiously fixed on it, for she felt sure she would catch a bad cold if she did not get dry very soon.

36 Once more she found herself in the long hall, and close to the little glass table. 'Now, I'll manage better this time,' she said to herself, and began unlocking the door that led into the garden.

37 She drew her foot as far down the chimney as she could, and waited till she heard a little animal (she couldn't guess of what sort it was) scratching and scrambling about in the chimney close above her.

38 The door led right into the kitchen: the Duchess was sitting on a three-legged stool in the middle, nursing a baby; the cook was leaning over the fire, stirring a large cauldron which seemed to be full of soup.

39 Well, perhaps you haven't found it so yet,' said Alice; 'but when you have to turn into a chrysalis and then after that into a butterfly, I should think you'll feel it a little strange, won't you?'

40 And she went on planning to herself how she would manage it. 'They must go by the carrier,' she thought; 'and how funny it'll seem, sending presents to one's own feet! And how odd the directions will look!

41 There seemed to be no use in waiting by the little door, so she went back to the table, half hoping she might find another key on it, or at any rate a book of rules for shutting people up like telescopes.

42 Then they all crowded round her once more, while the Dodo solemnly presented the thimble, saying 'We beg your acceptance of this elegant thimble'; and, when it had finished this short speech, they all cheered.

43 If you're going to turn into a pig,' said Alice, 'I'll have nothing to do with you!' The poor little thing sobbed again (or grunted, it was impossible to say which), and they went on for some while in silence.

44 Alice went on, half to herself, as she swam lazily about in the pool, 'and she sits purring so nicely by the fire, licking her paws and washing her face—and she is such a nice soft thing to nurse.'

45 Alice did not quite know what to say to this: so she helped herself to some tea and bread-and-butter, and then turned to the Dormouse, and repeated her question. 'Why did they live at the bottom of a well?'

46 There could be no doubt that it had a VERY turn-up nose, more of a snout than a real nose; also its eyes were getting extremely small for a baby: altogether Alice did not like the look of the thing at all.

47 'I must be growing small again.' She got up and went to the table to measure herself by it, and found that, as nearly as she could guess, she was now about two feet high, and was going on shrinking rapidly.

48 An enormous puppy was looking down at her with large round eyes, and feebly stretching out one paw, trying to touch her. 'Poor little thing!' said Alice, in a coaxing tone, and she tried hard to whistle to it.

49 'Would it be of any use, now,' thought Alice, 'to speak to this mouse? Everything is so out-of-the-way down here, that I should think very likely it can talk: at any rate, there's no harm in trying.'

50 'And now which is which?' she said to herself, and nibbled a little of the right-hand bit to try the effect: the next moment she felt a violent blow underneath her chin: it had struck her foot!

51 'The first thing I've got to do,' said Alice to herself, as she wandered about in the wood, 'is to grow to my right size again; and the second thing is to find my way into that lovely garden.

52 She heard a little pattering of feet, and she hastily dried her eyes to see what was coming. It was the White Rabbit returning with a pair of white kid gloves in one hand and a large fan in the other.

53 Alice replied eagerly, for she was always ready to talk about her pet: 'Dinah's our cat. And she's such a capital one for catching mice you can't think! And oh, I wish you could see her after the birds!'

54 There was a dead silence instantly, and Alice thought to herself, 'I wonder what they WILL do next! If they had any sense, they'd take the roof off.' After a minute or two, they began moving about again.

55 There was a large mushroom growing near her and when she had looked under it, and on both sides of it, and behind it, it occurred to her that she might as well look and see what was on the top of it.

56 She had not gone much farther before she came in sight of the house of the March Hare: she thought it must be the right house, because the chimneys were shaped like ears and the roof was thatched with fur.

57 She was close behind it when she turned the corner, but the Rabbit was no longer to be seen: she found herself in a long, low hall, which was lit up by a row of lamps hanging from the roof.

58 'We indeed!' cried the Mouse, who was trembling down to the end of his tail. 'As if I would talk on such a subject! Our family always HATED cats: nasty, low, vulgar things! Don't let me hear the name again!'

59 Alice thought this a very curious thing, and she went nearer to watch them, and just as she came up to them she heard one of them say, 'Look out now, Five! Don't go splashing paint over me like that!'

60 'When I used to read fairy-tales, I fancied that kind of thing never happened, and now here I am in the middle of one! There ought to be a book written about me! And when I grow up, I'll write one.'

61 'I wish I hadn't cried so much!' said Alice, as she swam about, trying to find her way out. 'I shall be punished for it now, by being drowned in my own tears! That will be a strange thing, to be sure!'

62 She stretched herself up on tiptoe, and peeped over the edge of the mushroom, and her eyes immediately met those of a large caterpillar, that was sitting on the top with its arms folded, quietly smoking.

63 Hardly knowing what she did, she picked up a little bit of stick, and held it out to the puppy; whereupon the puppy jumped into the air off all its feet at once, with a yelp of delight, and rushed at the stick.

64 'What is a Caucus-race?' said Alice; not that she wanted much to know, but the Dodo had paused as if it thought that somebody ought to speak, and no one else seemed inclined to say anything.

APPENDIX C

ANALYSIS PROCESS FLOW

It would have been difficult to manually execute the large number of variations that we identified in Chapter 5. We decided to automate this process by scripting the procedure using Matlab[11]; review the following overall analysis process for details.

1. The CSV file from the feature extraction result process was used as input to the overall analysis script.
    a. This modularity makes it easy to run the analysis on the entire set of participants or individual participants if needed.
    b. The CSV file is loaded in this step.
2. The script removes all the instances of participants that have fewer than 50 (configurable) instances of data.
3. New participant ids are assigned next.
    a. This was done to make it easier to view the results as GUIDs are 25 characters in length.
    b. New ids are numerical and start at one and increase for each new GUID found.
4. Class variations were created for the 2 and 3 class-level variations from the 5 class-level set.
5. Training input files were saved at this point.
    a. Fixed and free text variations were split.
    b. Each of the class-levels are separated into individual files so that they can be trained separately.
    c. Filenames describe the contents of the file (e.g. _training_fixed_AngerRating_2c.csv).
6. Next the neutral instances are removed from the 2 class-level files.

[11] Matlab `http://www.mathworks.com/`

7. The 10 balanced variations (configurable) are created for each file in the directory.
8. All the CSV training files are then converted into the ARFF file format using the WEKA.core.converters.CSVLoader utility in WEKA.
9. The PCA variations were then created for all files in the directory.
    a. The input features were standardized with a mean of zero using the WEKA.filters.unsupervised.attribute.Standardize filter in WEKA.
    b. PCA was run twice: once to output the results of the PCA and a second time to create new ARFF files to be used in training.
10. J48 training using ten-fold cross-validation was then executed for every file in the directory. Result files were created with similar filenames as their training (source) file.
11. The training results were then extracted from all of the individual output files.
12. The final outcome was the generation of four files:
    a. Everything.csv – includes all features as well as the class variations, the instance reductions, and participant id assignment.
    b. Demographics.csv – includes all the demographic and non-keystroke features as well as the class variations.
    c. Key-analysis.csv – The main results from the classification process.
    d. Balanced-only.csv – Similar to the key-analysis.csv but only the balanced training results are included. These are the results that were obtained before they were averaged and included in the key-analysis.csv file.

APPENDIX D

EXTENDED RESULTS

Here are the results for the two and five-class levels. Note that there is no arousal or valence emotional state here as they only pertain to the three-class level variations.

TWO CLASS-LEVEL RESULTS

Class Distribution

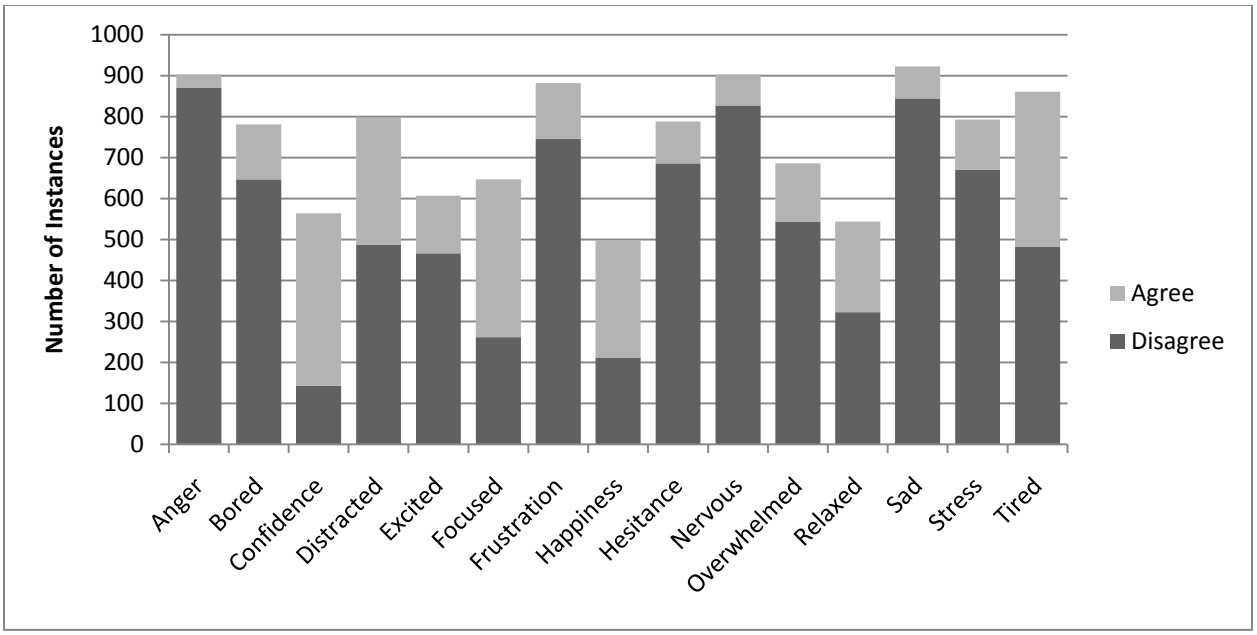

Figure D.1 Two-class distribution for each emotional state.

Cross Validation Results

Table D.1 Two class-level, balanced, free text results.

Emotional State	No Reduction				PCA Reduction			
	CC Rate	CC Variance	Kappa	Kappa variance	CC Rate	CC Variance	Kappa	Kappa Variance
Anger	57.66	52.33	0.15	0.02	57.97	36.05	0.16	0.01
Bored	57.76	6.55	0.16	0.00	59.51	11.46	0.19	0.00
Confidence	66.12	4.11	0.32	0.00	66.33	7.80	0.33	0.00
Distracted	61.95	1.80	0.24	0.00	63.02	7.25	0.26	0.00
Excited	59.68	3.69	0.19	0.00	68.26	8.36	0.37	0.00
Focused	56.34	1.03	0.13	0.00	59.73	3.15	0.19	0.00
Frustration	57.54	4.99	0.15	0.00	54.41	9.91	0.09	0.00
Happiness	58.98	10.44	0.18	0.00	65.00	6.68	0.30	0.00
Hesitance	70.25	6.94	0.40	0.00	73.19	11.27	0.46	0.00
Nervous	54.80	10.30	0.10	0.00	66.32	17.58	0.33	0.01
Overwhelmed	59.55	10.03	0.19	0.00	56.92	10.59	0.14	0.00
Relaxed	71.00	1.09	0.42	0.00	68.64	5.77	0.37	0.00
Sad	59.62	10.41	0.19	0.00	65.51	18.15	0.31	0.01
Stress	65.16	7.35	0.30	0.00	62.20	12.41	0.24	0.00
Tired	67.08	2.45	0.34	0.00	70.21	2.48	0.40	0.00

Table D.2 Two class-level, unbalanced, free text results.

Emotional State	No Reduction		PCA Reduction	
	Correctly Classified	Kappa	Correctly Classified	Kappa
Anger	96.46	0.00	96.46	0.00
Bored	82.84	0.00	82.84	0.00
Confidence	79.79	0.33	79.79	0.32
Distracted	61.98	0.06	69.01	0.32
Excited	76.77	0.01	78.42	0.16
Focused	61.51	0.08	62.13	0.12
Frustration	84.58	0.00	84.58	0.00
Happiness	58.03	0.02	65.46	0.28
Hesitance	87.06	0.00	89.85	0.37
Nervous	91.58	0.00	91.58	0.00
Overwhelmed	79.15	0.00	79.45	0.09
Relaxed	72.43	0.38	67.83	0.30
Sad	91.54	0.00	91.54	0.00
Stress	84.49	0.00	83.98	0.20
Tired	72.94	0.43	71.78	0.41

Table D.3 Two class-level, balanced, fixed text results.

Emotional State	No Reduction				PCA Reduction			
	CC Rate	CC Variance	Kappa	Kappa variance	CC Rate	CC Variance	Kappa	Kappa Variance
Anger	68.28	59.71	0.37	0.02	66.25	42.97	0.33	0.02
Bored	69.93	8.33	0.40	0.00	69.66	8.94	0.39	0.00
Confidence	80.31	2.99	0.61	0.00	79.16	10.90	0.58	0.00
Distracted	71.42	2.59	0.43	0.00	69.58	3.73	0.39	0.00
Excited	76.67	11.84	0.53	0.00	74.68	4.09	0.49	0.00
Focused	64.05	4.30	0.28	0.00	62.27	6.96	0.25	0.00
Frustration	63.24	12.92	0.26	0.01	63.42	19.44	0.27	0.01
Happiness	71.71	6.12	0.43	0.00	69.27	2.72	0.39	0.00
Hesitance	85.64	8.55	0.71	0.00	82.01	6.09	0.64	0.00
Nervous	83.22	5.70	0.66	0.00	78.75	14.82	0.58	0.01
Overwhelmed	67.41	12.06	0.35	0.00	69.76	16.82	0.40	0.01
Relaxed	79.46	2.47	0.59	0.00	75.02	1.59	0.50	0.00
Sad	87.95	13.31	0.76	0.01	83.14	20.73	0.66	0.01
Stress	74.51	26.44	0.49	0.01	72.07	8.45	0.44	0.00
Tired	83.46	1.16	0.67	0.00	82.11	0.96	0.64	0.00

Table D.4 Two class-level, unbalanced, fixed text results.

Emotional State	No Reduction		PCA Reduction	
	Correctly Classified	Kappa	Correctly Classified	Kappa
Anger	96.46	0.00	96.46	0.00
Bored	82.07	0.30	83.99	0.28
Confidence	83.69	0.56	81.56	0.50
Distracted	67.88	0.34	70.77	0.43
Excited	82.70	0.49	79.08	0.40
Focused	63.68	0.25	65.38	0.26
Frustration	83.79	0.00	84.47	0.06
Happiness	71.69	0.44	68.07	0.36
Hesitance	91.62	0.62	90.10	0.53
Nervous	93.91	0.57	93.36	0.54
Overwhelmed	76.38	0.16	80.32	0.22
Relaxed	78.86	0.56	77.21	0.52
Sad	93.49	0.56	92.62	0.42
Stress	85.12	0.33	85.25	0.29
Tired	84.20	0.68	81.30	0.62

FIVE CLASS-LEVEL RESULTS

Class Distribution

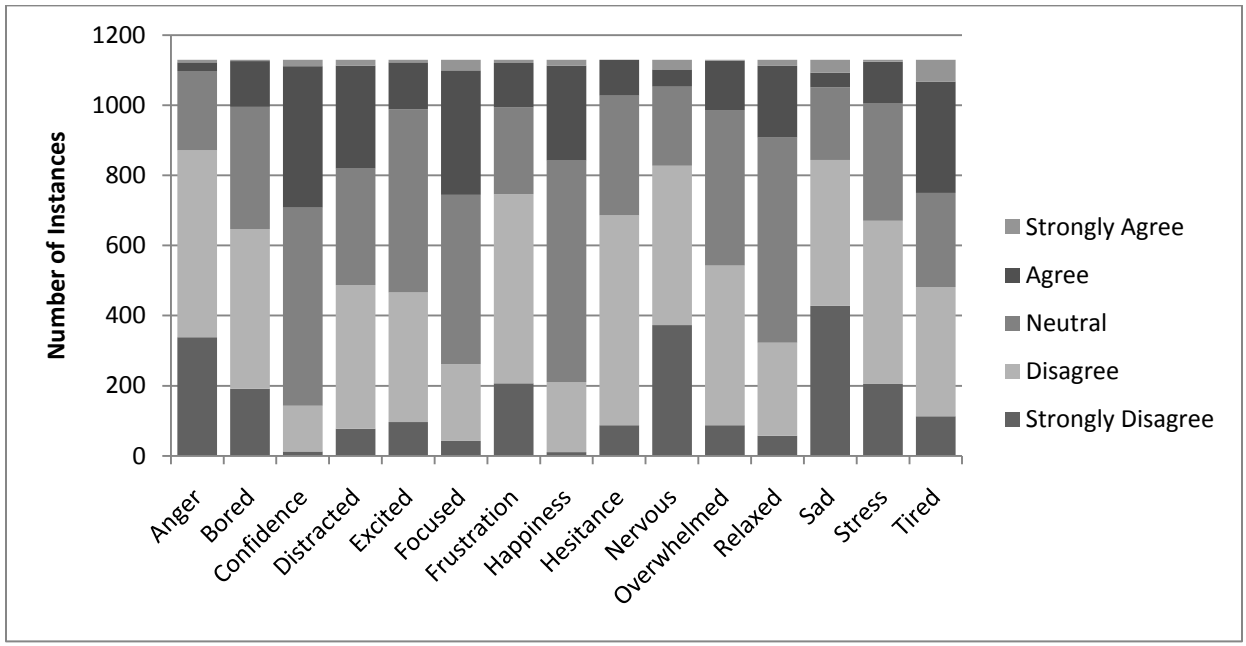

Figure D.2 Five class-level distribution for each emotional state.

Cross Validation Results

Table D.5 Five class-level, balanced, free text results.

Emotional State	No Reduction				PCA Reduction			
	CC Rate	CC Variance	Kappa	Kappa variance	CC Rate	CC Variance	Kappa	Kappa Variance
Anger	57.66	52.33	0.15	0.02	57.97	36.05	0.16	0.01
Bored	57.76	6.55	0.16	0.00	59.51	11.46	0.19	0.00
Confidence	66.12	4.11	0.32	0.00	66.33	7.80	0.33	0.00
Distracted	61.95	1.80	0.24	0.00	63.02	7.25	0.26	0.00
Excited	59.68	3.69	0.19	0.00	68.26	8.36	0.37	0.00
Focused	56.34	1.03	0.13	0.00	59.73	3.15	0.19	0.00
Frustration	57.54	4.99	0.15	0.00	54.41	9.91	0.09	0.00
Happiness	58.98	10.44	0.18	0.00	65.00	6.68	0.30	0.00
Hesitance	70.25	6.94	0.40	0.00	73.19	11.27	0.46	0.00
Nervous	54.80	10.30	0.10	0.00	66.32	17.58	0.33	0.01
Overwhelmed	59.55	10.03	0.19	0.00	56.92	10.59	0.14	0.00
Relaxed	71.00	1.09	0.42	0.00	68.64	5.77	0.37	0.00
Sad	59.62	10.41	0.19	0.00	65.51	18.15	0.31	0.01
Stress	65.16	7.35	0.30	0.00	62.20	12.41	0.24	0.00
Tired	67.08	2.45	0.34	0.00	70.21	2.48	0.40	0.00

Table D.6 Five class-level, unbalanced, free text results.

Emotional State	No Reduction		PCA Reduction	
	Correctly Classified	Kappa	Correctly Classified	Kappa
Anger	96.46	0.00	96.46	0.00
Bored	82.84	0.00	82.84	0.00
Confidence	79.79	0.33	79.79	0.32
Distracted	61.98	0.06	69.01	0.32
Excited	76.77	0.01	78.42	0.16
Focused	61.51	0.08	62.13	0.12
Frustration	84.58	0.00	84.58	0.00
Happiness	58.03	0.02	65.46	0.28
Hesitance	87.06	0.00	89.85	0.37
Nervous	91.58	0.00	91.58	0.00
Overwhelmed	79.15	0.00	79.45	0.09
Relaxed	72.43	0.38	67.83	0.30
Sad	91.54	0.00	91.54	0.00
Stress	84.49	0.00	83.98	0.20
Tired	72.94	0.43	71.78	0.41

Table D.7 Five class-level, balanced, fixed text results.

Emotional State	No Reduction				PCA Reduction			
	CC Rate	CC Variance	Kappa	Kappa variance	CC Rate	CC Variance	Kappa	Kappa Variance
Anger	68.28	59.71	0.37	0.02	66.25	42.97	0.33	0.02
Bored	69.93	8.33	0.40	0.00	69.66	8.94	0.39	0.00
Confidence	80.31	2.99	0.61	0.00	79.16	10.90	0.58	0.00
Distracted	71.42	2.59	0.43	0.00	69.58	3.73	0.39	0.00
Excited	76.67	11.84	0.53	0.00	74.68	4.09	0.49	0.00
Focused	64.05	4.30	0.28	0.00	62.27	6.96	0.25	0.00
Frustration	63.24	12.92	0.26	0.01	63.42	19.44	0.27	0.01
Happiness	71.71	6.12	0.43	0.00	69.27	2.72	0.39	0.00
Hesitance	85.64	8.55	0.71	0.00	82.01	6.09	0.64	0.00
Nervous	83.22	5.70	0.66	0.00	78.75	14.82	0.58	0.01
Overwhelmed	67.41	12.06	0.35	0.00	69.76	16.82	0.40	0.01
Relaxed	79.46	2.47	0.59	0.00	75.02	1.59	0.50	0.00
Sad	87.95	13.31	0.76	0.01	83.14	20.73	0.66	0.01
Stress	74.51	26.44	0.49	0.01	72.07	8.45	0.44	0.00
Tired	83.46	1.16	0.67	0.00	82.11	0.96	0.64	0.00

Table D.8 Five class-level, unbalanced, fixed text results.

Emotional State	No Reduction		PCA Reduction	
	Correctly Classified	Kappa	Correctly Classified	Kappa
Anger	96.46	0.00	96.46	0.00
Bored	82.07	0.30	83.99	0.28
Confidence	83.69	0.56	81.56	0.50
Distracted	67.88	0.34	70.77	0.43
Excited	82.70	0.49	79.08	0.40
Focused	63.68	0.25	65.38	0.26
Frustration	83.79	0.00	84.47	0.06
Happiness	71.69	0.44	68.07	0.36
Hesitance	91.62	0.62	90.10	0.53
Nervous	93.91	0.57	93.36	0.54
Overwhelmed	76.38	0.16	80.32	0.22
Relaxed	78.86	0.56	77.21	0.52
Sad	93.49	0.56	92.62	0.42
Stress	85.12	0.33	85.25	0.29
Tired	84.20	0.68	81.30	0.62

APPENDIX E

LOG FILE EXAMPLES

KEYBOARDEVENTS.LOG

```
Timestamp      Milliseconds   ParticipantId  VkCodeInt      VkCode  KeyEventType
        ForegroundWindowTitleBarText  ScanCode       Char    None    LButton RButton Cancel  MButton
        XButton1       XButton2       LButton, XButton2      Back    Tab     LineFeed
        LButton, LineFeed      Clear   Return  RButton, Clear RButton, Return        ShiftKey
        ControlKey     Menu    Pause   Capital KanaMode       RButton, Capital       JunjaMode
        FinalMode      HanjaMode      RButton, FinalMode     Escape  IMEConvert      IMENonconvert
        IMEAceept      IMEModeChange  Space   PageUp  Next    End     Home    Left    Up      Right
        Down    Select  Print   Execute PrintScreen    Insert  Delete  Help    D0      D1      D2
        D3      D4      D5      D6      D7      D8      D9      RButton, D8    RButton,             D9
        MButton, D8    MButton, D9    XButton2, D8    XButton2, D9   64      A       B       C
        D       E       F       G       H       I       J       K       L       M       N       O
        P       Q       R       S       T       U       V       W       X       Y       Z       LWin
        RWin    Apps    RButton, RWin  Sleep   NumPad0 NumPad1 NumPad2 NumPad3 NumPad4 NumPad5 NumPad6
        NumPad7 NumPad8 NumPad9 Multiply        Add     Separator       Subtract        Decimal Divide
        F1      F2      F3      F4      F5      F6      F7      F8      F9      F10     F11     F12
        F13     F14     F15     F16     F17     F18     F19     F20     F21     F22     F23     F24
        Back, F17      Back, F18      Back, F19      Back, F20      Back, F21      Back,        F22
        Back, F23      Back, F24      NumLock Scroll  RButton, NumLock       RButton,         Scroll
        MButton, NumLock       MButton, Scroll        XButton2, NumLock      XButton2,        Scroll
        Back, NumLock  Back, Scroll   LineFeed, NumLock      LineFeed, Scroll       Clear,  NumLock
        Clear, Scroll  RButton, Clear, NumLock       RButton, Clear, Scroll LShiftKey
        RShiftKey      LControlKey    RControlKey    LMenu   RMenu   BrowserBack    BrowserForward
        BrowserRefresh BrowserStop    BrowserSearch  BrowserFavorites       BrowserHome
        VolumeMute     VolumeDown     VolumeUp       MediaNextTrack MediaPreviousTrack
        MediaStop      MediaPlayPause LaunchMail     SelectMedia    LaunchApplication1
        LaunchApplication2     Back, MediaNextTrack   Back, MediaPreviousTrack      Oem1    Oemplus
        Oemcomma       OemMinus       OemPeriod      OemQuestion    Oemtilde       LButton,
        Oemtilde       RButton, Oemtilde      Cancel, Oemtilde       MButton,             Oemtilde
        XButton1, Oemtilde     XButton2, Oemtilde     LButton, XButton2, Oemtilde   Back,  Oemtilde
        Tab, Oemtilde  LineFeed, Oemtilde     LButton, LineFeed, Oemtilde   Clear,        Oemtilde
        Return, Oemtilde       RButton, Clear, Oemtilde       RButton,       Return,       Oemtilde
```

ShiftKey, Oemtilde ControlKey, Oemtilde Menu, Oemtilde Pause, Oemtilde
Capital, Oemtilde KanaMode, Oemtilde RButton, Capital, Oemtilde JunjaMode,
Oemtilde FinalMode, Oemtilde HanjaMode, Oemtilde RButton, FinalMode, Oemtilde
OemOpenBrackets Oem5 Oem6 Oem7 Oem8 Space, Oemtilde PageUp,
Oemtilde OemBackslash LButton, OemBackslash Home, Oemtilde ProcessKey
MButton, OemBackslash Packet Down, Oemtilde Select, Oemtilde Back, OemBackslash
Tab, OemBackslash PrintScreen, Oemtilde Back, ProcessKey Clear, OemBackslash
Back, Packet D0, Oemtilde D1, Oemtilde ShiftKey, OemBackslash ControlKey,
OemBackslash D4, Oemtilde ShiftKey, ProcessKey Attn Crsel Exsel EraseEof
Play Zoom NoName Pa1 OemClear LButton, OemClear

2009-08-22 14:38:41.270 63386548721270 11f61436-0a7f-1648-4895-5a603da64b21 65 A
KEYDOWN firefox 30 a 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 0 0
0 0 0 0 0 0 0 0 0 1 0 1
1 0 1 0 1 0 0 0 1 1 0 1
1 0 0 0 0 0 0 128 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 1 0 0 1 0
0 0 0 1 0 0 0 0

2009-08-22 14:38:41.426 63386548721426 11f61436-0a7f-1648-4895-5a603da64b21 65 A
KEYUP firefox 30 a 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 0 0
0 0 0 0 0 0 0 0 0 128 0 1
1 0 1 0 1 0 0 0 1 1 0 1
1 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0

```
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1	0	0	1	0
0	0	0	1	0	0	0	0
```

WINDOWEVENTS.LOG

```
Timestamp	Milliseconds	ParticipantId	WindowTitles

2009-08-22 14:51:26.246	63386549486246	11f61436-0a7f-1648-4895-5a603da64b21	firefox
	explorer

2009-08-22 14:51:36.269	63386549496269	11f61436-0a7f-1648-4895-5a603da64b21	firefox
	explorer	winword

2009-08-22 14:51:46.311	63386549506311	11f61436-0a7f-1648-4895-5a603da64b21	firefox
	explorer	winword	excel
```

QUESTIONNAIREEVENTS.LOG

```
Timestamp	Milliseconds	ParticipantId	QuestionnaireId	SampleTextDisplayed
	FrustrationRating	AngerRating	HappinessRating	ConfidenceRating
	HesitanceRating	StressRating	RelaxedRating	ExcitedRating	BoredRating
	SadRating	NervousRating	TiredRating	FocusedRating	DistractedRating
	OverwhelmedRating

2010-06-21 13:50:20.687	63386549456715	11f61436-0a7f-1648-4895-5a603da64b21	2	4
	2	3	5	5	2	3	4	5	2	1	1	5
	2	1	2
```

DEMOGRAPHICS.LOG

```
Timestamp      ParticipantId  Sex     Age     Occupation     WhereInstalled WhereInstalledOther
       FirstLanguage  FirstLanguageLCID      TypedLanguage  TypedLanguageLCID      DominateHand
       MouseHand      PointingDevice MouseButtons   TypingAbilities        ComputerTime
       VideoGameTime  TypingSoftwareTime     VirtualMachine LaptopDesktopInstall
       LaptopDesktopInstallOther     PercentageTimeonThisMachine   IPAddresses

2010-06-21 13:50:20.687       63386549456715 11f61436-0a7f-1648-4895-5a603da64b21  Male    31
       Graduate Student       Work           English (Canada)       4105    English        (Canada)
       4105    Right   Right   Mechanical mouse       2 buttons      Average 2 - 4 hours    Less
       than 3 hours a         week    3-7 hours a week       No      Desktop        About   half
       128.234.54.120
```

SYSTEMINFORMATION.LOG

```
Timestamp      Milliseconds   ParticipantId  MonitorCount   PrimaryMonitorSizeWidth
       PrimaryMonitorSizeHeight      VirtualScreenWidth     VirtualScreenHeight
       MouseButtonCount       MouseButtonsSwapped    MouseSpeed     MouseWheelPresent
       MouseWheelScrollDelta  MouseWheelScrollLines  KeyboardDelay  KeyboardSpeed
       DoubleClickTime        DoubleClickSizeWidth   DoubleClickSizeHeight  CursorSizeWidth
       CursorSizeHeight       DragSizeWidth  DragSizeHeight CultureEnglishName
       CultureKeyboardLayoutId       CultureLCID    CultureName
       CultureThreeLetterISOLanguageName

2010-06-21 13:50:20.687       63386651240358 11f61436-0a7f-1648-4895-5a603da64b21  1       1280
       1050    1280    1050    3       True    10      False   120     3       1       31      500
       4       4       32      32      4       4       English (United States)       1033    1033
       en-US   eng
```

APPLICATION.LOG

```
2009-08-22 14:00:27.474 - Creating the input manager
2009-08-22 14:02:26.527 - Demographics saved.
2009-08-22 14:02:26.542 - Starting the BasicUserInputManager
2009-08-22 14:02:26.542 - Start the broadcaster
2009-08-22 14:02:26.558 - Starting mouse hook...
2009-08-22 14:02:26.558 - Starting keyboard hook...
2009-08-22 14:02:26.558 - Start the purge timer
2009-08-22 14:02:26.558 - Start the window timer
2009-08-22 14:02:26.558 - Start the write timer
```

ERROR.LOG

```
2009-08-24 09:33:20.039 - Unable to broadcast event to listener. Message: Thread was being aborted.
```