Managing, mining and visualizing multi-modal data for stress awareness

Kurniawan, H.

Award date:
2012

Eindhoven University of Technology
Department of Mathematics and Computer Science

Managing, Mining and Visualizing Multi-Modal Data for Stress Awareness

Master Thesis

Hindra Kurniawan

Supervisor:
dr. Mykola Pechenizkiy

Submitted in partial fulfillment of the
requirements for the degree of Master of Science
August 19, 2012

Abstract

The stress problem has become the major dilemma affecting many people's lives and professions. Nowadays, stress is often unrecognized, and people accept stress as a normal condition. Short term stress is not necessarily bad, as on some occasions it can help us to meet challenges. On the other hand, prolonged stress should be avoided because it has been shown to cause physical breakdowns and makes our body vulnerable to diseases.

In this thesis, we propose a framework for stress analytics, which focuses on management and analysis of multi-modal affective data captured in text, speech, facial expression and physiological signals, such as Galvanic Skin Response (GSR). The framework allows for automatic stress detection based on multimodal data, for instance, from GSR and speech. We investigate the discriminating power of speech and GSR in distinguishing two different stress levels in the controlled experiment environment. A collective of 10 subjects voluntarily participated in the psychological study for stress elicitation. The stress was induced by using the Stroop-Word color test and solving mental arithmetic problems. During the experiment, the speech was recorded and the homemade GSR device was used to monitor the skin conductance.

Four different machine learning classifiers were investigated regarding their ability to discriminate between two different stress levels. The state-of-the-art classifier, Support Vector Machine (SVM), outperformed the other classifiers. The reasonable accuracy of 70% was achieved by using individual GSR data as an input to SVM classifier. On the other hand, using the speech signal as an input to an SVM classifier yields a maximum accuracy of 92%. Furthermore, combining both GSR and speech models does not improve the performance in significant ways.

Acknowledgement

I would like to thank all those who helped me to carry out this thesis. First and foremost, I would like to thank my supervisor Prof. Mykola Pechenizkiy, for his constant support and invaluable guidance during the course of this project.

I would like to thank Rafal Kocielnik, who has helped me in conducting the psychological stress elicitation experiment. I am also thankful to those who participated either as a subject or as an evaluator during the psychological experiment.

I express my gratitude to Prof. Paul De Bra and Prof. Natalia Sidorova, members of the assessment committee, who have read, evaluated my work and providing feedbacks.

I would also like to thank Louis Wearden for his effort in checking the language in this thesis.

Last but not least, I would like to thank my parents, brothers, and all my friends, who provided invaluable moral support.

Eindhoven, The Netherlands
August 2012

Hindra Kurniawan

List of Figures

List of Tables

List of Acronyms

ACF Auto-Correlation Function

ANS Autonomic Nervous System

API Application Programming Interface

BP Blood Pressure

BVP Blood Volume Pulse

DCT Discrete Cosine Transform

DTW Dynamic Time Warping

ED Euclidean Distance

EDA Electrodermal Activity

EM Expectation Maximization

FACS Facial Action Coding System

FFT Fast Fourier Transform

GAS General Adaptation Syndrome

GMM Gaussian Mixture Model

GSR Galvanic Skin Response

HOLAP Hybrid OLAP

HRV Heart Rate Variability

JDBC Java Database Connectivity

MAS Manifest Anxiety Scale

MDX Multi-Dimensional eXpressions

MFCC Mel Frequency Cepstral Coefficients

MOLAP Multidimensional OLAP

OLAP Online Analytical Processing

PLP Perceptual Linear Perception

POS Part-of-Speech

QBE Query-by-Example

RAPT Robust Algorithm for Pitch Tracking

RASTA Relative Spectral Transform

RBF Radial Basis Function

RDBMS Relational DataBase Management System

ROLAP Relational OLAP

SDS Social Desirability Scale

SQL Structured Query Language

ST Skin Temperature

SVM Support Vector Machine

TF-IDF Term-Frequency and Inverse Document Frequency

VSA Voice Stress Analysis

XML eXtensible Markup Language

Contents

Chapter 1

Introduction

1.1 Motivation

Stress is a psychological response to the emotional, mental and physical way in which we respond to pressure. Immediate dangers provoke the '*fight or flight*' stress response, which causes the hormones adrenaline and cortisol to be released into the blood stream to prepare the body to either battle or leave the conflict. This instant burst of energy may help us in critical or emergency situations, such as giving us the ability to run beyond our normal capability in a dangerous situation. Not only are certain hormones released during this period, but changes in heart rate, sweat, breathing and contraction of muscles can also be observed. If the body is healthy, after this period it should restore balance and return to the normal state.

Unfortunately, nowadays stress has become a common phenomenon that can hardly be avoided in daily life situations. Short term stress, at some points, may help us to meet the challenges but not the prolonged stress. Prolonged stress has been known to be able to cause a physiological breakdown that makes the body vulnerable to diseases, such as hypertension and coronary artery disease [1][2]. Not only physical diseases may be caused by stress but also mental illness, such as generalized anxiety disorder or depression [3].

People nowadays have difficulty recognizing stress and accept stress as a normal condition. Therefore, the technology for recognizing, analyzing and detecting stress is important as a preemptive way to alleviate stress. This technology at

least could help the individual to become aware of stress and gain insight into the condition that regularly arouses a stress response. Moreover, it can also become an automatic tool to prevent stress by means of intervention.

1.2 Thesis Objective and Methodology

The objective of this thesis is two-fold:

1. *Develop a framework for stress analytics, which can help the user (or domain experts) to analyze interesting stress patterns, gain insight in what is happening and explore the stress-related data.*
2. *Develop a model which can automatically differentiate between two different stress levels from multi-modal affective data.*

The stress analytics framework, in essence, is a multidisciplinary approach involving four main components: data management, data mining from multimodal affective data, OLAP support and visualization. Data management handles how the multimodal data is stored and aligned. Data mining employs supervised machine learning for automatic stress detection. Raw data can be summarized and presented by means of basic OLAP exploration, which allows a multidimensional query. Finally, the visualization addresses the presentation of the previous components for the end user. The system can visualize the raw evidence (e.g. Speech audio playback and actual GSR time series) to make users more aware of their stress level. Last but not least, a time series similarity search is provided in our framework by means of a shape-based query-by-example to help the domain expert to analyze various stress patterns that may occur across different individuals.

In order to accomplish our second objective, we conduct a study of existing approaches for automatic stress recognition. The problems of stress detection have been investigated by researchers, and many types of solutions have been proposed. Stress can be detected from the changes in physiological signals, such as heart rate, body temperature, skin conductance, body acceleration, pupil diameter, blood pressure, etc. Besides physiological signals, stress can be detected from other cues, such as speech and facial expression. In this thesis, we used two signals obtained from Galvanic Skin Response (GSR) and speech for an automatic stress detection task. The rationale behind choosing these two signals is due to the fact

that both measurements are unobtrusive and, to our knowledge, there is no existing work yet which has investigated these signals for stress detection. As for a real-life application, the stress detection from speech and GSR can be applied in many areas, such as in call-centers and public relation jobs.

The detection of stressful events is a challenging task and far from trivial due to several reasons. First, there is the unavailability of the general stress model. Therefore, the interpretation of stress itself is still ambiguous. Second, it is unclear which features and classifier should be used for detecting stress within literatures. Third, the stress experiments are hardly reproducible; hence, the results of the experiment are incomparable.

However, we used supervised machine learning techniques to differentiate between two different stress levels. Supervised machine learning required preexisting labeled dataset for training and evaluation. Unfortunately, we cannot find an openly available benchmark which consists of speech and GSR. Therefore, we conducted a psychological stress elicitation experiments to obtain the dataset.

1.3 Main Result

The result of this thesis is a framework for stress analytics, and the automatic stress classification using multi-modal data. Specifically, stress analytics was implemented as a web-based system that enabled the summarization view of stress level over different periods of times, OLAP exploration, zoom-in to the level of evidences and zoom-out to the grand total summary of stress level, searching the most similar time series and many more.

During this thesis, we conducted the psychological controlled stress experiment to elicit a certain stress level to the individuals and have collected 10 hours of affective data including GSR, speech, facial expression and subjective annotation. This dataset was used as a labeled dataset for supervised machine learning. The experiment for detecting two different stress levels using supervised machine learning showed a promising result, and it indicates that both GSR and speech are indeed a good indicator for stress detection.

1.4 Thesis Structure

The structure of this thesis is organized as follows. Chapter 2 describes the framework for stress analytics. In Chapter 3, the techniques behind automatic stress detection from the speech and GSR are elaborated. Chapter 4 presents the evaluation of automatic stress detection. Finally, in Chapter 5, we draw the conclusions from our work, and the possible improvement is presented for future work.

Chapter 2

Stress Analytics Framework

Stress Analytics is the system for management, analysis and automated stress classification of multi-modal affective data captured from text, speech, facial expression and physiological signals. One of the principal goals of the system is to enable users (or domain experts) to analyze interesting stress patterns and visualize various evidence of stress so that they are able to manage stress in a better way.

The rest of this chapter is organized as follows. Section 2.1 reviews previous studies on the multimodal analysis and similarity search on time-series. In Section 2.3, we explain how to store and align the raw data. Section 2.4 introduces our design of stress cube and OLAP. In Section 2.5 we briefly describe the Shape-Based Query-by-Example functionality. Finally, in Section 2.6 we present the visualization of stress analytics.

2.1 Related Work

2.1.1 Multimodal Analysis

Multi-modal analysis is a rapidly expanding interdisciplinary field in linguistics and language-related fields of study, including education [4]. This is a field of study that concerns combining multiple resources (e.g. text, image, audio, language, and video) to create meaning in different contexts, such as in movie, digital media, and daily life situations.

The multi-modal analysis has been applied not only in the science field, but also in other disciplines, such as the social sciences and linguistics. In linguistic [5], it has been studied in order to understand the integration and interaction of two or more data to convey the meaning within texts. For instance, language, page layout and image-text relation may affect the way the individual perceives the meaning. In the social sciences, for example in [6], multi-modal science has been used to deconstruct Myspace's social network in order to understand how different communicative data (e.g. text, graphics, pictures and music) can be used to create a user's stereotype and engagement for sociable purposes.

On a different body of work, multi-modal analysis has been demonstrated as a digital tool that can link low-level features in different media (text, image and video) to higher-order semantic information using social semiotic theory and computer-based techniques of analysis [7]. The demonstrator utilized the recorded video's lecture as a study case for the analysis. The aim of this tool is to provide multi-modal analysis of the media recorder in digital forms (e.g. recorded video). More precisely, the following functionality and features are supported within the tool:

- Create time-stamped tier-based annotations and overlay.
- Create text, image and sound annotations.
- Searching functionality to locate a pattern of interest defined with respect to all types of annotation.
- Media analytics, which provide the automatic detection of music classification, silence audio detection, face detection, tracking objects in videos, optical character recognition and basic image filtering.
- Providing gesture and movement analysis.

The analysis, then, can be stored in the database for further retrieval and visualization of the results.

2.1.2 Similarity Search on Time-Series

Similarity measure is an active area of time-series data mining research, since all tasks, including classification and clustering, require that the notion of similarity be defined. Generally, similarity measure can be categorized into two groups:

shape-based and structure-based similarity. The former, determines the similarity of two time series based on the distance between individual points, whereas, the latter looks at the higher structural level.

Measures have been proposed for shape-based similarity, including but not limited to: Euclidean distance, Dynamic Time Warping (DTW) [8], Longest Common Sub-Sequence (LCSS) [9], Edit Distance with Real Penalty (ERP) [10], and Time-Warp Edit Distance (TWED) [11]. In [12], the performance comparisons of these measures using five time series datasets are reported. The results showed that Euclidean distance, given the simplest and the fastest method, is indeed as good as other more complex measures. Therefore, they recommended Euclidean as a valid and computationally inexpensive option for measuring similarity. In a different study, we found that DTW has been demonstrated by several authors [13][14][15] to be more superior compared to Euclidean distance in many data mining applications, including rule discovery, clustering and classification.

Shape based similarities works well for short time-series but produces a poor result for long time-series [12]. A structure-based similarity measure is an alternative way to determine similarity for long time series based on the higher-level structure. This is known also as a model-based similarity, as this approach extracts global features such as auto-correlation, skewness and model parameter from data. In [16], the author proposed using the ARMA model for learning time-series data then using the model coefficients as a feature. A different approach, using a representation similar to the one used for classifying text documents, was proposed by [17]. They used a histogram-based representation that allows computation of similarity between data based on the high-level structure. The technique is called as 'bag of patterns', similar to the text-based 'bag of words'.

Several approaches to speed-up the similarity queries for time-series data have been proposed recently. The basic idea is to index the time series, by mapping the sequences to the high level representation, then store this index in an efficient data structure for fast retrieval. Agrawal et al. [18] proposed an indexing method by using the Discrete Fourier Transform (DFT) to map the time sequences to the frequency domain, and then using only the first few of the frequencies. These coefficients thus stored using an R*-Trees [19] data structure for fast indexing and retrieval. This approach works well for similarities queries, given the query and the sequences to be searched have the same length. Faloutsos et al. [20] present an

efficient indexing method to locate subsequences within a collection of sequences. In other words, this method can handle the query which has different lengths of sequence. They used a similar idea as proposed by [18]. The novel idea of this approach was using a sliding window over the data sequences and extracting its features (e.g. frequency representation of time series). The result is a trail in feature space. This trail is divided into several sub-trails and, subsequently, is represented by their Minimum Bounding Rectangle (MBR). Afterwards, these MBR are stored using spatial method data structure like R*-tree for fast indexing and retrieval. Besides DFT, many high-level time-series representations have been proposed, including Discrete Wavelet Transform (DWT) [21], Piecewise Aggregate Approximation (PAA)[22] and Symbolic Aggregate approXimation (SAX) [23].

2.2 Framework Overview

The model of stress analytics consists of four central parts: storing and aligning multiple sources of raw data, data mining and pattern mining, Online Analytical Processing (OLAP) support for stress exploration, and visualization as shown in Figure 2.1. In this thesis, we only address and implement the following tasks: storing raw data, data mining and classification from raw data, OLAP interactive support, shape-based Query-by-Example support, pointer to evidences and visualization for the end user. Nevertheless, we did not implement the data mining or pattern mining from OLAP result, and the data mining from raw data was performed in an offline instead of online setting.

Figure 2.2 shows a detailed overview of the framework for stress analytics. The bottom-most level contains information of raw data structures. The data are originated from different heterogeneous sources, including but not limited to textual data, physiological signals and metadata. For example, we used the existing textual data which have been collected by Erik Tromp during his thesis [24]. Tromp had collected and performed a multilingual sentiment analysis of textual data from social media, email and electronic agenda.

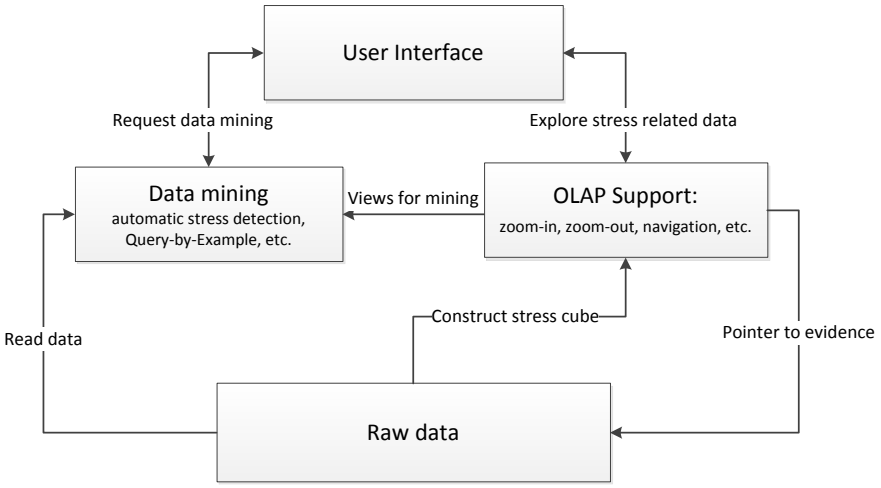


Figure 2.1: A high-level overview of the stress analytics framework.

After the raw data is stored, it will be preprocessed, and several distinguishing features will be extracted afterwards. The datasets must be represented by features for learning. The performance of the learned system is affected by the definition of its feature. In fact, the choice of features is more important than the choice of the learning algorithm itself [25]. The feature often has to be preprocessed before it is used. The preprocessing steps for textual data include but are not limited to removing stop words, assigning parts-of-speech (POS) and lemmatization. The features which are commonly utilized for textual data are frequency of words, Term-Frequency and Inverse Document Frequency (tf-idf), and occurrences of a word. We refer to [24] for a detailed explanation of preprocessing and feature extraction of textual data.

Raw physiological signals which are collected have to be preprocessed. The preprocessing may include but is not limited to removing noise, extracting voiced sound from speech, smoothing, and discretization. Several features afterwards are extracted for learning. The preprocessing and feature extraction step is going to be elaborated more in Chapter 3.

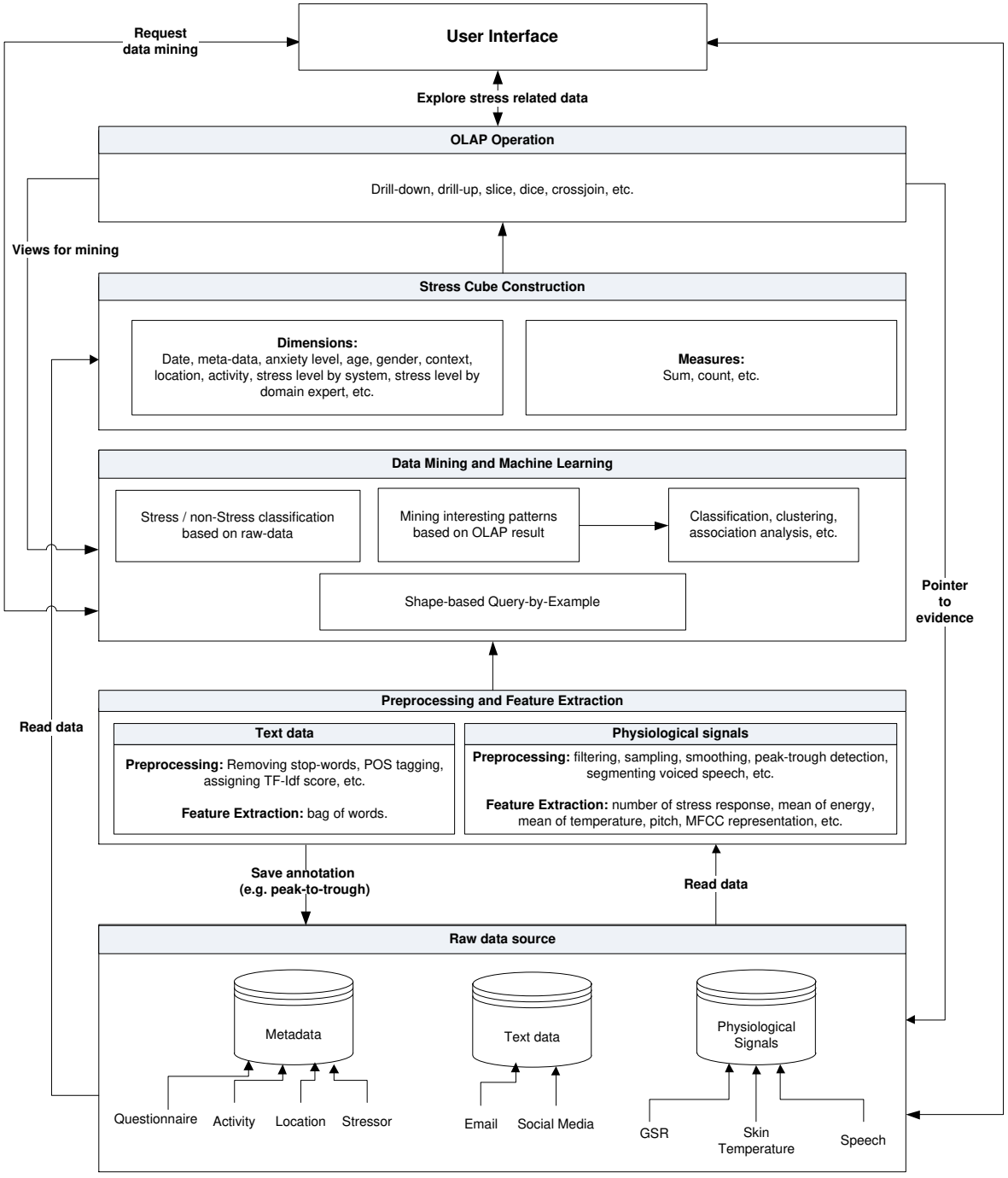


Figure 2.2: A low-level overview of the stress analytics framework.

The data mining and machine learning can be used for two purposes in this framework. First, we can use data mining as a classification tool for both descriptive or predictive modeling. Descriptive modeling differentiates objects from different classes. On the other hand, the predictive modeling predicts a label of unknown records. We refer to Chapter 3 for a detailed explanation of how to build a model to differentiate between two different stress states based on the raw data. Second, we envision that data mining, together with machine learning, can be applied for finding interesting patterns from the OLAP result. For instance:

- Finding clusters of subjects who have the similar stress level during the specified time with respect to the same gender or age.
- Finding a sequence of tasks (stressors) which can induce a high stress level on the subject.
- Predicting the general subject's stress level in the future based on their past data.

The output of the data mining block, together with raw data, is used to construct a stress cube. A stress cube is a set of data organized in such a way to facilitate non-predetermined queries for OLAP. OLAP is commonly used for analyzing business data by aggregating information on multi-dimensional data. We utilized an OLAP support for interactive visualization of stress based on predefined multi-dimensional data. Based on this cube and raw data, a simple yet interactive user interface is provided to allow the user to navigate and explore the cube.

Finding the most similar time-series pattern (e.g. GSR or skin temperature) from the database is a beneficial feature which can enable the domain expert to retrieve, classify and study the particular stress pattern across different subjects, time and other dimensions. This functionality was implemented inside our framework and is hereafter called as shape-based Query-by-Example (QBE).

2.3 Storing and Alignment of Raw Data

2.3.1 Storing Raw Data

The raw data originating from different sources, such as physiological signals, speech, textual data and metadata, were stored using a relational database. Figure

2.3 illustrates the Entity-Relationship diagram for storing the raw data. Physiological signals, such as Galvanic Skin Response (GSR), Skin Temperature (ST), heart rate and pupil diameter are commonly used to measure the dynamic changes of the Autonomic Nervous System (ANS)[26]. In this thesis, we only address how to store GSR and skin temperature data. Nevertheless, due to the nature of this database structure, more data can be added without jeopardizing the existing one.

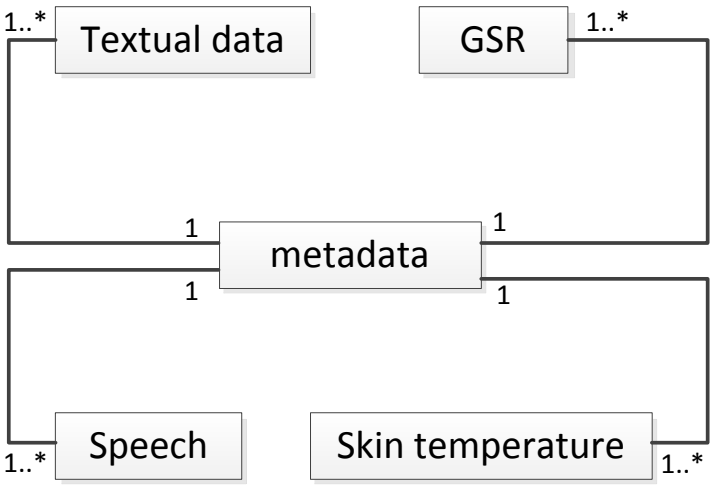


Figure 2.3: ER-diagram for storing the four different raw data.

The speech can be stored as a binary file in the relational database for easy retrieval, viewing and audio playback purposes. The textual data may originate from various sources, including but not limited to email, personal diary, agenda, and social media content. Senticorr: Multilingual sentiment analysis of personal correspondence is an interactive, extensible system for automated sentiment analysis on multilingual user-generated content from various social media and emails [27]. One of the goals of this sentiment analysis is to make people aware of how much positive and negative content they read and write. Therefore, each email is annotated with the number of positive, negative and objective sentences. The email and its annotations, then, were stored in the database as an evidence of textual content to support the analysis. As for metadata, it contains additional descriptive information about the data which have been collected. For example, the metadata may contain information about who the subject is (e.g. age, gender and anxiety), where the data was collected (e.g. campus, home or street), what

activity the subject performed (e.g. walking, standing or running), and when the data was taken. The technical detail on how to store the raw data can be found in Appendix A.

2.3.2 Alignment of Raw Data

Due to the fact that the raw data obtained from heterogeneous sources have to be aligned one with another, we opted for using a global clock system to synchronize these data because of the simplicity and the ease of implementation. In other words, using a global clock system implies all data must have the same reference of date and time.

2.3.3 Storing Other Raw Data

Due to the nature of this database structure, another raw data can be added without jeopardizing the entire framework. For example, it is easy to add a video recording of facial expression to the framework by following the diagram structure as shown in Figure 2.3. Furthermore, the extension of the diagram for storing another metadata is also straightforward.

2.4 OLAP and Stress Cube

2.4.1 Online Analytical Processing (OLAP)

Data warehouses are the collection of historical, summarized, non-volatile data accumulated from transactional databases. They are optimized for OLAP and have proven to be valuable for decision making [28]. The data in a warehouse are conceptually modeled as data cubes. The size of the data warehouse is typically huge, and OLAP queries are complex. As for the term itself, OLAP usually refers to analysing large quantities of data in real time. Operation in OLAP typically involves read-only with raw data in bulk. In other words, OLAP operations do not change the content of raw data. The term online refers to the ability of the system to respond to queries fast enough to allow an interactive exploration of the data regardless of the huge size of the relational database, which is usually up to

several gigabytes.

OLAP operations help the analyst to understand the meaning contained in the databases using multi-dimensional analysis. The analysts can navigate through the database, filter particular subsets of data, changing the data's orientation and defining analytical calculation [29]. A common operation of OLAP includes but is not limited to slicing, dicing, drill-down and roll-up. A slice is a subset of a multi-dimensional array corresponding to a single value for one or more members of the dimensions not in the subset. A dice operation is a slice on more than two dimensions of a data cube. Drill-up or down is a specific analytical technique whereby the user navigates among levels of data ranging from the most summarized (up) to the most detailed (down). In essence, each operation is equivalent to adding a "WHERE" clause in the SQL statement.

OLAP systems can be mainly classified into three categories by the way they access the data [30]. These categories are multidimensional, relational, and hybrid. The first one, Multidimensional OLAP, abbreviated as MOLAP, stores both the source data and the aggregations in a multi-dimensional array storage, rather than in a relational database. MOLAP is the fastest option for data retrieval, but it requires the most disk space. The second one, Relational OLAP, abbreviated as ROLAP, works directly with a relational database. The base data and dimension tables are stored as a relational table and the aggregated information is stored in a new table. The third one is Hybrid OLAP, abbreviated as HOLAP, and it is a combination of both MOLAP and ROLAP. HOLAP divides the data storage between relational and specialized storage. Some data, mostly the aggregated one, is stored in special storage while the rest is in relational.

We used Mondrian[1], an open-source OLAP server, which is able to analyze large quantities of data in real time. Mondrian only supports the ROLAP model as its storage method. In addition, Mondrian does not store aggregated data on disk, but on cache (memory) once it has read a piece of data once. The engine executes queries written in the Multi-Dimensional eXpressions (MDX) language by reading data from a relational database (RDBMS), and it presents the results in a multi-dimensional format via Java Application Programming Interface (API). MDX was the first query language created by Microsoft and dedicated for multi-dimensional analysis. Nowadays, MDX language is considered the new standard

[1] http://mondrian.pentaho.com/

for multi-dimensional analysis.

Olap4j[2], a Java API which resembles JDBC (Java Database Connectivity) for relational database, is used for accessing multi-dimensional data. It is designed as an abstraction layer to handle database connection, query model abstraction, executing and obtaining the OLAP results.

2.4.2 Stress Cube

The core of any OLAP system is an OLAP cube. The cube can be multi-dimensional or an n-dimensional cube. The cube metadata is typically created based on the star or snowflake schema. The idea of star schema is quite simple: All data are categorized as dimensions or measures. Measures are derived from the records in the fact table, and dimensions are derived from the dimension tables. Therefore, the star schema is represented by a centralized fact table which is connected to multiple dimensions. Each tuple in the fact table consists of the pointer to each of the dimensions. The snowflake schema, on the other hand, is quite similar to the star schema with the exception that child tables may have multiple parent tables, hence resembling a snowflake shape. The star schema is a special case of snowflake schema. We refer to [30] for detailed explanation about data warehouse, OLAP, the star schema, and the snowflake schema.

The stress cube metadata structure was implemented as a star schema. The fact table contains numerical measures for indicating a stress level. As for the dimensions, it consists of attributes that we wish to be aggregated, such as subject id, date, activity, stressor (task), location and email. Figure 2.4 illustrates the star schema model. Technical detail of stress cube construction can be found in Appendix A.

[2]http://www.olap4j.org/

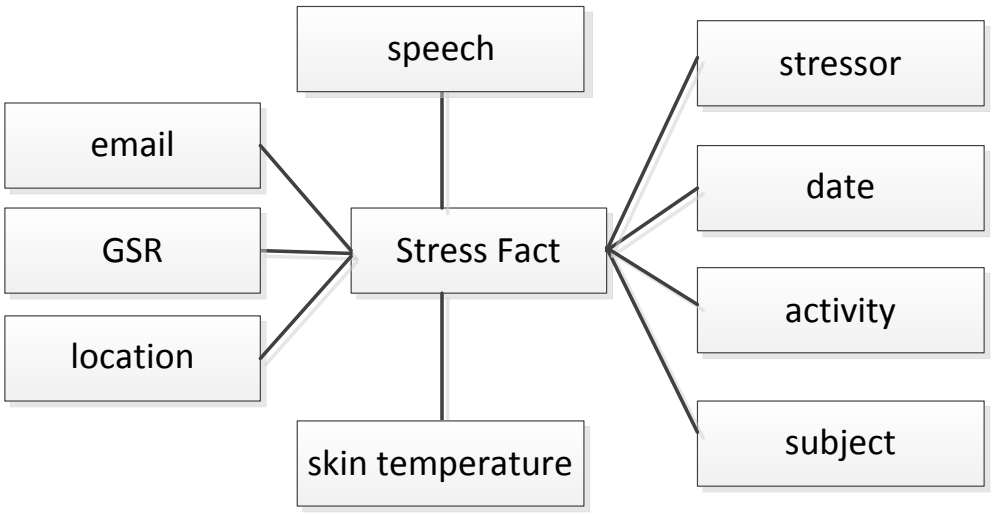


Figure 2.4: Stress cube star schema. Stress fact is a fact table, while the rest is dimension.

2.4.3 Extension of Stress Cube

As we used the ROLAP model as a storage method for OLAP, the extension or modification of the fact table (or dimensions) has to be done in a relational database. The star schema as depicted in Figure 2.4 is straightforward to extend. For example, adding another metadata such as accelerometer or room temperature can be achieved by creating a new dimension and linking it directly to the fact table.

2.5 Shape-Based Query-by-Example

Query-by-example is a search mechanism which allows a user to search for similar documents based on the example of some particular document or list of documents. The document content itself could be any of the following: textual, graphical, time series, or multimedia. In this thesis, we consider a query-by-example based on shape for time-series data. The problem is formulated as follows: given a subsequence query C, we wish to find the most similar (1-Nearest Neighbor) shape-based subsequence R from T, where $|R| = |C|$ and T is the sequences of

time-series in the database.

We utilized the existing UCR-Suite algorithm [31], the current state-of-the-art, for searching subsequences time series under Dynamic Time Warping (DTW) [32] and Euclidean distance. The original source code of UCR-suite algorithm[3] was written in C language. In order to conform to our core application, we rewrote the code by using Java language and presented both similarity results found by using Euclidean distance and by DTW. We did not conduct any objective experiments and evaluations to measure the performance differences between Euclidean distance and DTW. Though based on an anecdotal evaluation, the result found by using DTW is preferable to Euclidean Distance as shown in Figure 2.5. More elaborate explanation about shape-based QBE can be found in Appendix A.

2.6 Stress Analytics Visualization

The summary view of the stress level over different dimensions is presented via an interface providing a basic OLAP-style exploration. The interface allows the user to zoom-in to the level of evidence and zoom-out to the grand total summary of the stress level. Furthermore, the feature of shape-based query-by-example is provided to enable the user (or domain expert) to search for a similar stress pattern (e.g. from GSR or skin temperature) across different dimensionalities.

We implemented the visualization of stress analytics using a web-based system which used Apache Tomcat 7.0[4] as an HTTP web server together with MySQL 5.0[5] as a storage engine. The complete list of all open-source libraries, tools, software, and script that we used for developing the stress analytics can be found in Appendix D.

[3] http://www.cs.ucr.edu/∼eamonn/UCRsuite.html
[4] http://tomcat.apache.org/
[5] http://dev.mysql.com/

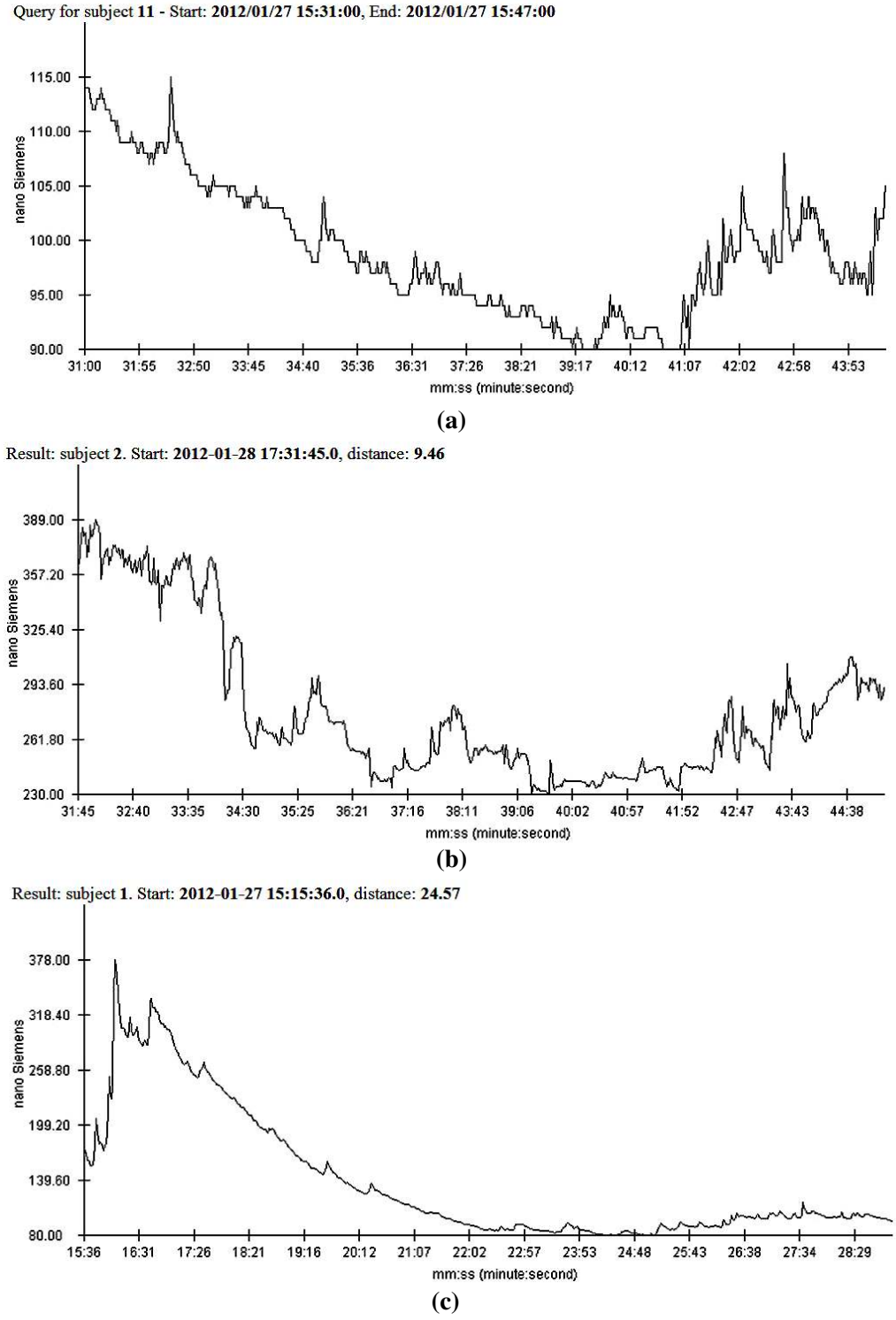


Figure 2.5: Shape-based Query-by-Example (QBE). (a) Query time-series. (b) The most similar time-series found by using DTW. (c) The most similar time-series found by using Euclidean distance.

In a nutshell, the visualization provides three different functionalities, interactive OLAP exploration, showing evidence (e.g. stress related physiological signals) or stress related events (e.g. email), and search functionality (e.g. shape-based query-by-example). Figure 2.6 depicts the sample graph obtained by using OLAP explorations. Apart from the visualization by OLAP, we present a visualization of raw data as an evidence to make an individual more aware of what was actually happening. Four different evidences: GSR, skin temperature, email and speech (both audio recording and its waveform) are shown, and is illustrated in Figure 2.7. The shape-based QBE is depicted Figure 2.5. The interactive overall diagram of stress analytics visualization can be found in Appendix A.

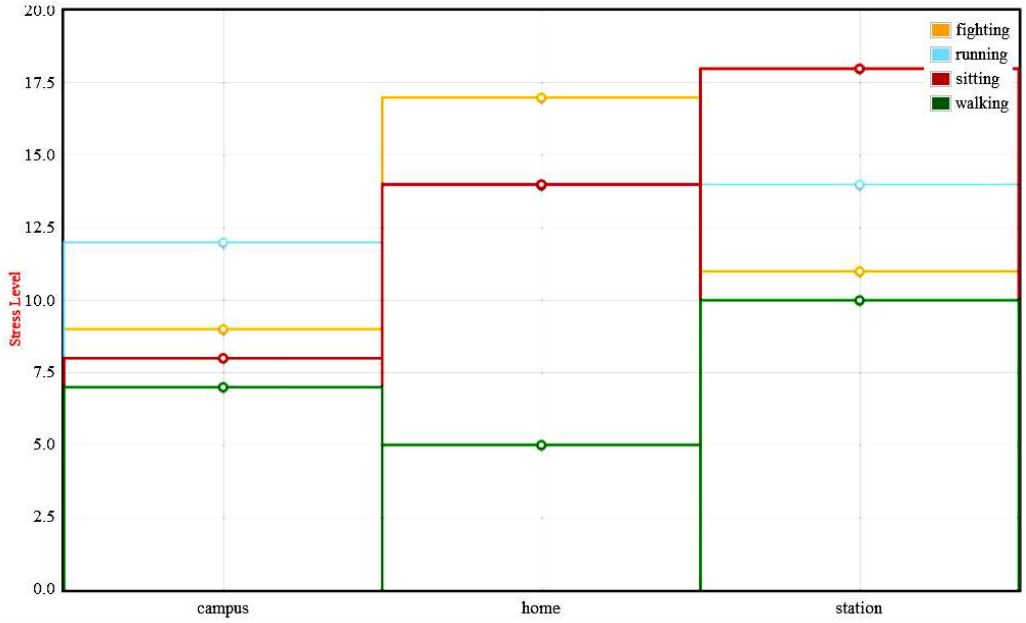
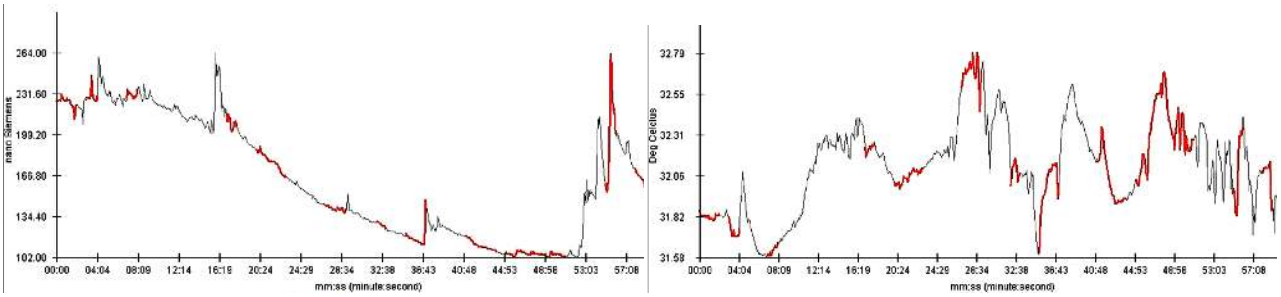
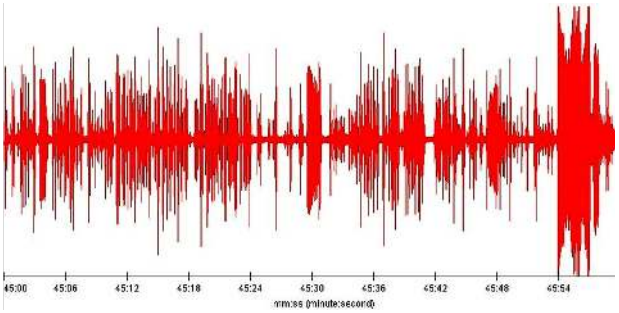


Figure 2.6: Visual exploration of the stress cube: Aggregated stress level for different locations and activities.



(a) **(b)**



(c)

Folder and subject: [Inbox] -- Test email
Created at: 2012-01-27 14:20:00.0
Senticor analysis: Objective: 7, Positive: 1, Negative: 1

From: SATestEmail84(satestemail84@gmail.com) **To:** sa2011test@hotmail.com
CC: **BCC:**

Dear Paul I am happy to hear that you are doing well For me the situation is not that nice I am in the middle of a big crisis I recently have too much work that I cannot handle Perhaps I am taking too many responsibilities but on the other hand I feel that everyone around me is doing just fine having the same amount of work as me Because of all that I am a bit in the middle of despair Perhaps I need to think about organizing my things better I hope that things for you will continue to be fine and that I will hear from you soon All the best John

(d)

Figure 2.7: Four different raw data evidences. Red and black lines in time series graph represent stress and non-stress periods respectively. (a) GSR. (b) Skin temperature. (c) Speech waveform representation. The system also provides the spectrogram representation and audio player for playing back the speech waveform. (d) Email raw data. The black, green, and red sentences represent objective, positive and negative respectively.

Chapter 3

Automatic Stress Detection from Speech and Galvanic Skin Response (GSR)

Chronic stress has become a serious problem affecting different life situations and carrying a wide range of health-related diseases, including cardiovascular disease, cerebrovascular disease, diabetes, and immune deficiencies [33]. Thus, stress management is an utmost importance for preventing it becoming chronic. Technologies that automatically recognize stress can become a powerful tool to motivate people to adjust their behavior and lifestyle to achieve a better stress balance. Technology such as sensors can be used for obtaining an objective measurement of stress level, which afterwards, machine learning technique is employed to build a model for stress detection and recognition. The objective, therefore, is to detect the changes in GSR and speech, and discriminate them based on the binary labeling of stress. Stress detection in this section was conducted in an offline setting.

The rest of this chapter is organized as follows. In Section 3.1, we review the previous studies on the stress model, stress in speech detection and stress detection using physiological signals. We describe the preprocessing and feature extraction for speech and GSR signals in Section 3.2. In Section 3.3, several machine learning techniques for classification are discussed. Section 3.4 provides an explanation of combining both GSR and speech for stress detection.

3.1 Background and Previous Work

3.1.1 Stress Model

The concept of stress remains elusive, very broad and used differently in a number of domains. It is poorly defined, and no single agreed definition of stress is in existence. However, two models which are commonly used today come from Selye and Lazarus. Selye [34] proposed General Adaptation Syndrome (GAS) model, which identifies various stages of stress response. Selye argues that stress is a disruption of homeostatis by physical or psychological stimuli. Physical factors, such as noise, excessive heat or cold, and psychological factors, such as extreme emotion, frustration and sleep deprivation, alter the internal equilibrium of the body which causes a stress response. There are three stages in this model. The first stage is the alarm stage; the body identifies the danger or threat and goes into a state of alarm. In this stage, the hormone adrenaline and cortisol are released to provide instant energy. Adrenaline is produced in order to prepare the body for 'fight' or 'flight'. The secretion of adrenaline causes the blood flows to the large muscle of the body, as the body prepares to run away or fight. The cortisol hormone, known as 'stress hormone', increases the blood pressure and blood sugar in order to restore body homeostasis after the stress period. The second stage is the resistance stage, where the body focuses all of its resources to restore balance, and the recovery for repair and renewal takes place. In this stage, possibly the source of stress might be resolved. If the stressful condition is not resolved at this stage, then it moves to the next stage. The last stage is the exhaustion stage, when the body can no longer resist the stressor and psychological breakdown begins. In this stage, the body is vulnerable to disease and even death.

Lazarus's model [35] is slightly different from Selye's. Lazarus emphasises his theory on two central concepts: appraisal, i.e., individuals' evaluation and interpretation of their circumstances, and coping, i.e., individual efforts to handle and solve the situation. Lazarus argues that neither the stressor nor one's response is sufficient for defining stress, rather it would be one's perception and appraisal of the stressor that would determine if it creates stress. The first stage is the primary appraisal, where the subject analyzes the stressor and determines if it is positive or negative, exciting or dangerous, etc. During this stage, emotions are generated by the appraisal. The second stage is the secondary appraisal, where the subject decides if he or she can cope with the given stressor. If one can cope with the

stressor, then the stress might be resolved and could be kept at a minimum level. On the other hand, if one cannot cope with the stressor, he or she will experience a high degree of stress.

3.1.2 Stress Detection in Speech

Hansen et al. [36] define stress in the scope of recognition from the voice signal as any condition which causes a speaker to vary speech production from "natural" conditions. The speech is considered as neutral when a speaker is in a quiet room with no task obligations. Moreover, Hansen divides the stressor into two groups: perceptual and physiological. The perceptual stressor is a condition whenever a speaker perceives his or her environment to be different from natural, such as his or her intention to produce speech varies from neutral condition. The perceptual stressor can be induced by emotion, environmental noise (e.g. Lombard's effect), high cognitive workload, and frustration over contradictory information. On the other hand, the physiological stressor causes the speaker to deviate from neutral speech production despite his or her own intention due to the physical impact on one's body. Causes of physiological stressors include, but are not limited to, G-force, vibration, drug interactions, and sickness.

The effect of stress in relation to speech production has been well studied over the past decades. Respiration has been found to correlate with certain emotional situations. When an individual experiences a stressful situation, his respiration rate increases. This presumably will increase subglottal pressure during speech, which is known to increase pitch (or fundamental frequency) during voiced section [37]. Moreover, an increase in the respiration rate causes a shorter duration of speech between breaths which, in turn, affects the speech articulation rate. In a different body of work, Scherer [38] confirmed the previous finding, by showing that voice is indeed a good indicator of stress and found a high correlation between stress and the increase of fundamental frequency. Further details of analysis, modeling, and recognition for speech under stress can be found in [39].

In general, the speech classification approach involves extracting discriminatory features from the audio, then, the features will be utilized as an input to a pattern classifier. The audio features can be categorized into prosodic and spectral features. Pitch, loudness, speaking rate, duration, pause and rhythm are all perceived characteristics of prosody. These measurements are usually encoded in terms of

statistical measures like means, medians, standard deviations, and averages over the whole utterances. On the other hand, several spectral features which are commonly used for emotion detection include, but are not limited to, Mel Frequency Cepstral Coefficients (MFCC), Linear Predictive Cepstral Coefficient (LPCC), Mel Filter Bank (MFB), and Perceptual Linear Prediction (PLP). The spectral feature is usually extracted by using a frame window over the whole utterances. The efficient number of frame size per speech utterance is investigated in [40].

After features have been defined, the machine learning technique is carried out to classify instances of different classes. Cichosz et al. [41], investigated the emotion recognition using binary-tree based classifier, where consecutive emotions are extracted at each node, based on an assessment of feature triplets. A different body of work using a hidden semi-continuous Markov model has been investigated by Nogueiras et al. [42]. They showed that the result obtained from the classifier is very similar to that obtained with the same database in a subjective evaluation by human judges. Shah [43] investigated Support Vector Machine and K-Means to classify opposing emotions. He used statistics related to the pitch, Mel Frequency Cepstral Coefficients (MFCCs) and a formant as an input to classification algorithms. Gaussian Mixture Model (GMM) has been shown by [44] to be a competitive model for recognizing emotion and speech. In addition, a detailed exploration of affective expression in speech could be found in Shikler [45] work.

Another controversial model for detecting stress in speech is by using the Voice Stress Analysis (VSA) [46] technique, which measures fluctuations in the physiological microtremors present in speech. Microtremors are present in every muscle in the body, including vocal cords and have a frequency around 8-9 Hz. During stress, this range is shifted to 8-12 Hz range. Stress thus can be detected by analyzing the changes in the microtremors' frequency of an individual voice. Furthermore, VSA is known also as a controversial technology for lie detection.

3.1.3 Stress Detection using Physiological Signals

The human nervous systems can be divided into two main divisions, the voluntary and the Autonomic Nervous Systems (ANS). The voluntary nervous system is concerned with the control of body movement via muscles, motor and sensory neurons. On the other hand, the ANS is an involuntary division which has less conscious control. It controls the organs of our body such as the heart, stomach

and intestines.

Furthermore, the ANS can be divided into two divisions, sympathetic and parasympathetic nervous systems. The parasympathetic nervous system is responsible for nourishing, calming the nerves to return to the regular function, healing, and regeneration. On the contrary, the sympathetic nervous system is accountable for activating the glands and organs for defending the body from the threat. It is called 'fight' or 'flight' response. The activation of sympathetic nervous system might be accompanied by many bodily reactions, such as an increase in the heart rate, rapid blood flow to the muscle, activation of sweat glands, and increase in the respiration rate.

These physiological changes, such as the electrical activity on the scalp, Blood Pressure (BP), Blood Volume Pulse (BVP), electrical activity of the heart over a period of time, electrical activity produced by skeletal muscles, Galvanic Skin Response (GSR), Skin Temperature (ST), eye's blinking rate, pupil dilation, and Heart Rate Variability (HRV), can be measured objectively by using modern technology sensors. These psychological variables can be monitored in non-invasive ways and have been investigated by many researchers over the past decades. The researcher often uses multimodal signals to obtain more precise information about the state of mind.

The research provided in [47][26][48][49][50] proposes a multimodal system for detecting stress, which consists of Blood Volume Pulse, Galvanic Skin Response, Skin Temperature, and Pupil Dilation. They used these signals to differentiate between two different affective states in a computer user and found a strong correlation between these signals and stress variation.

A real-life study for automatic stress recognition involving call center employees can be found in [51]. They modified the Support Vector Machine (SVM) loss function to incorporate the participant specific information. The similar work for personalized stress detection was presented in [52] as well.

In [53], an activity-aware mental stress detection scheme using Electrocardiogram (ECG), GSR, and an accelerometer is proposed. They studied the correlation between the user's activity and physiological measurements while one was subjected

to mental stressors. Another interesting work on the stress detection field was presented by Healey and Picard [54] in 2005. They conducted a physiological study on a real-world driving task where the objective is to determine a driver's relative stress level. Several signals, such as ECG, Electromyogram (EMG), GSR, and HRV, were recorded continuously while drivers followed a predefined route through an open road in the Boston area. They could distinguish between three different driving conditions with a great accuracy across multiple drivers and multiple days.

A different body of works for stress detection by using facial expression together with physiological signals has been proposed in [55][56]. They used the Bayesian network for modeling stress and the associated factors and showed that the inferred user stress level is consistent with the psychological theory.

In general, the machine learning technique is employed for building a model for differentiating between different levels of stress. The popular machine learning methods which are used within literature include, but are not limited to, Fuzzy Logic [57], Support Vector Machine (SVM) [48][26][53], Decision Tree Classifier [26][53], Naïve Bayes Classifier [26], and Bayesian network [56][55][53].

3.2 Feature Selection

3.2.1 Speech

Several speech features which are investigated are described as follows:

1. **Energy**
   The basic feature of audio signals for human auditory perception is loudness. In general, several terms that are commonly used to describe the loudness of audio signals are volume, intensity and energy. The energy of the signal X for each sample i in time is described as:

   $$Energy_i = X_i^2$$

   The most common method to compute energy is using the overlapping time frames as in the fundamental energy calculation. This method results in a more continuous signal. Let $X_1, ..., X_N$ defines the signal samples in a frame, and then the smoothed energy in each frame is given as:

$$SmoothedEnergy_{frame} = \sum_{i=1}^{N} X_i^2$$

Figure 3.1 shows a speech signal and the results of different speech signal calculations.

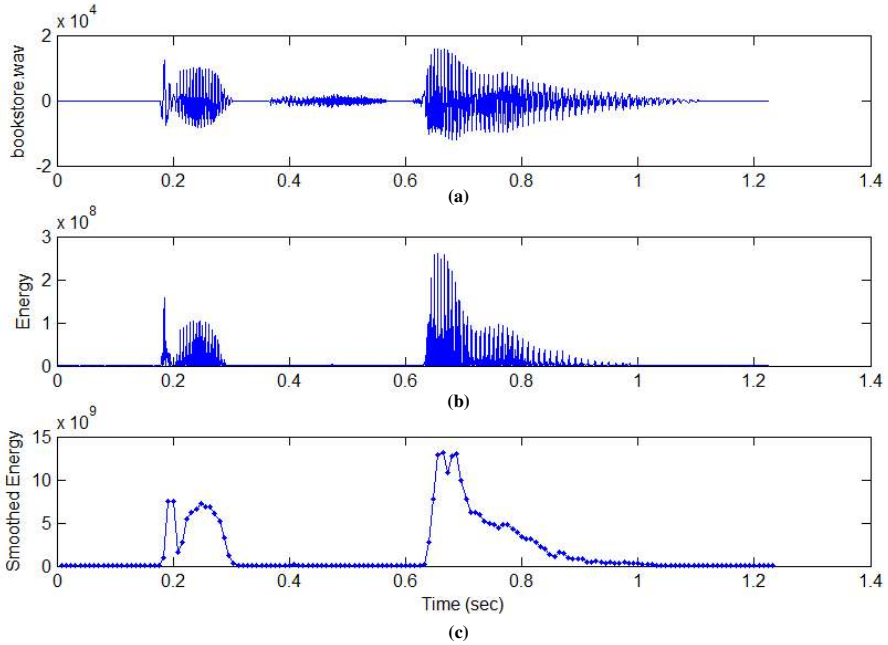


Figure 3.1: (a) Speech signal of the utterance of the word "bookstore". (b) The energy of each sample. (c) The average energy for each frame, using frame-size = 256 and overlap = 128.

2. **Voiced and Unvoiced Speech**
   The audio signals are generally referred to as signals that are audible to humans, which is typically from 20Hz to 20KHz. There are many sources of audio signals: human voices, sound from animals, periodic sound, aperiodic sound, etc. In this thesis, we only considered human voices. Basically, we can divide each short segment of human voices into two categories: voiced and unvoiced sound. Voiced sounds are produced by the vibration of the vocal cord, hence we can observe the fundamental periods in a frame. On the other hand, unvoiced sounds are not produced by the vibration of the vocal cord. Instead, they are produced by the rapid flow of air through the nose or teeth. Hence we cannot perceive any fundamental periods in a frame, due to the

fact that there is no vibration in the vocal cord. Figure 3.2 shows a difference between voiced and unvoiced sounds.

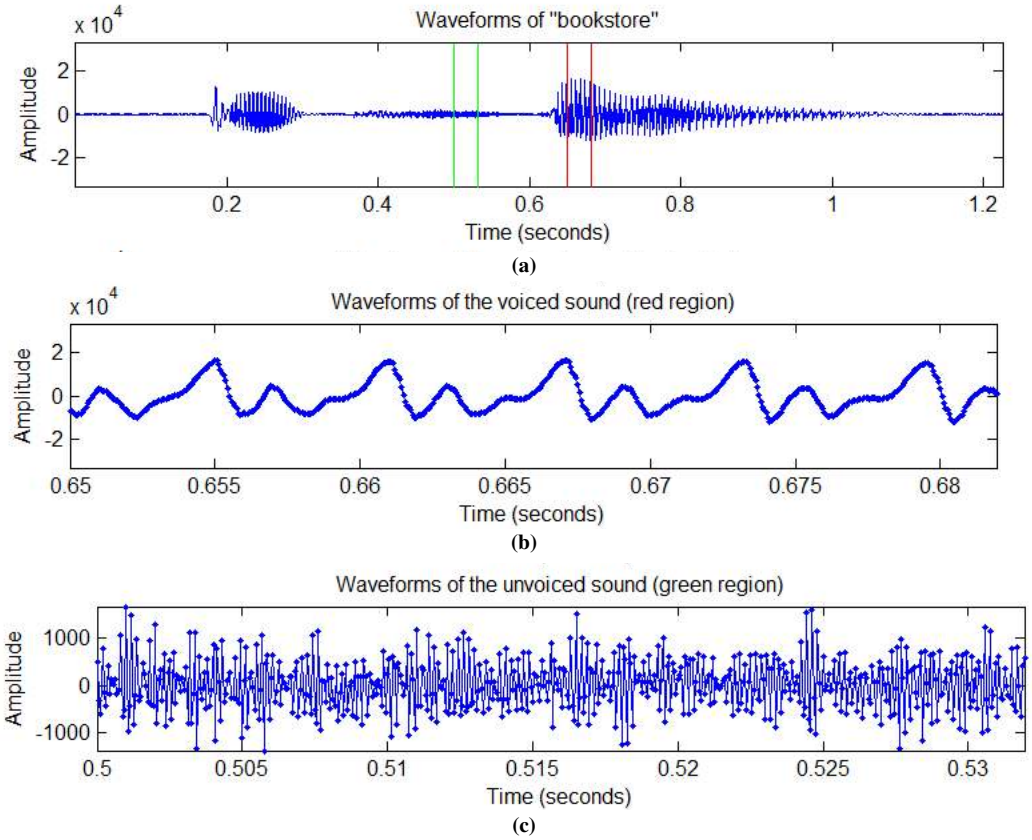


Figure 3.2: (a) Speech signal of the utterance "bookstore". (b) Waveform of the voiced sound, as shown in the red region in (a). (c) Waveform of the unvoiced sound, as shown in the green region in (a).

3. **Pitch**

   Pitch is the fundamental frequency of audio signals, which represent the vibration rate of the sound source. In other words, pitch is the perceptual correlation of the rate of vibration of the vocal cord. People tend to use an intonation to convey meaning when speaking, and its perceptual correlation is called pitch. Much research [58][59] has shown that pitch conveys considerable information about emotional status.

   There are many different extraction algorithms for the fundamental frequency. One of the basic algorithms is an auto-correlation function (ACF). The auto-

correlation is the cross-correlation of the signal with itself, or how well the signal matches a time-shifted version of itself. This method is useful for finding patterns in a signal. The ACF function is given as:

$$acf(\tau) = \sum_{i=0}^{n-1-\tau} s(i)s(i+\tau)$$

where τ is the time lag in terms of sample points. The value of τ which maximize $acf(\tau)$ over a specified range is selected as the pitch period in sample points. Figure 3.3 illustrates the computation of the auto-correlation function.

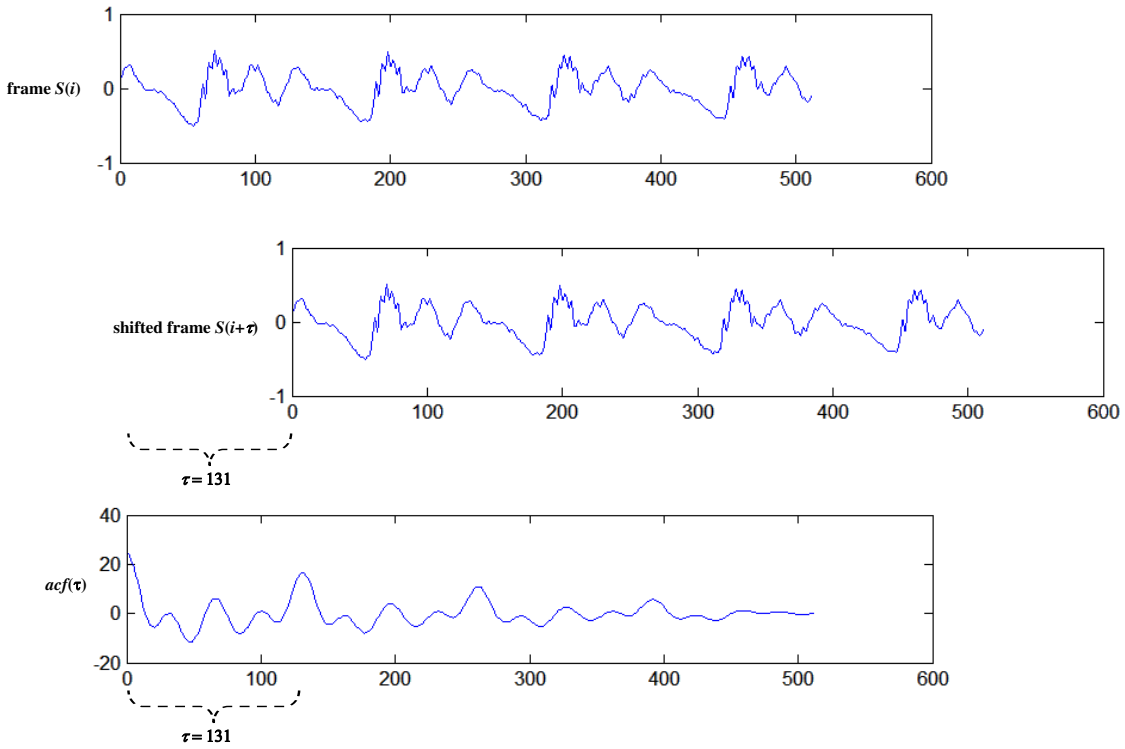


Figure 3.3: The auto-correlation function illustration. The maximum of ACF (we omit the first one) happens at index around 131, hence the corresponding pitch is $fs/(131-1) = 16000/130 = 123.08Hz$, where fs denotes the frame rate (frame per second).

The auto-correlation function, in fact, is not a good choice for pitch tracking, since it used sampling and windowing for determining the maximum of auto-

correlation and this turned out to cause many problems [60]. Numerous algorithms for pitch tracking have been suggested over the years. Among these, the most popular algorithm is the Robust Algorithm for Pitch Tracking (RAPT), proposed by Talkin [61]. In this thesis, we used the `VOICEBOX`[1] speech processing toolbox which contains the RAPT implementation.

4. **Mel Frequency Cepstral Coefficients (MFCCs)**
   MFCCs are coefficients that represent human audio perception. These coefficients have been shown to have a great success in speaker recognition application. In addition, MFCCs are the most widely used spectral representation of speech in many applications, including speech and speaker recognition [62]. The application is not only limited to speaker recognition, but also to speech and emotion recognition. The method of MFCCs can be summarized as follows:

   - Transform a raw signal to frequency-based using Fast Fourier Transform (FFT).
   - The scale of frequencies is corrected to Mel's frequency that approximates the human system auditory response more closely.
   - Take the logs powers at each of the Mel frequencies.
   - Take the Discrete Cosine Transform (DCT) of log-Mel spectrum.
   - Store only few high-order coefficients. For instance, store the first 13 components.
   - For each coefficient, calculate its mean, variance, maximum, and minimum value.

   We used the `VOICEBOX` speech processing toolbox which contains the MFCC implementation.

5. **RASTA-PLP**
   Another popular method for speech feature representation is by using Relative Spectral Transform (RASTA) - Perceptual Linear Perception (PLP) [63]. PLP was originally purposed to minimize the differences between speakers while maintaining the important speech information. RASTA is a technique which involves several filtering to make PLP more robust to linear spectral distortions. For example, in [63] RASTA-PLP has been demonstrated to show a significant error rate improvement for recognition using a telephone line

[1] http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

which contains static noise. In this thesis, we used the existing RASTA-PLP[2] implementation in Matlab.

3.2.2 Galvanic Skin Response

Galvanic Skin Response (GSR) is a measure of electrical conductance of the skin, which is proportional to sweat secretion [64]. When an individual experiences stress, the sweat glands which are controlled by the sympathetic nervous system are activated. Therefore, skin conductance may act as an indicator of stress arousal. The skin conductance measurement is usually placed on hands or feet, where the density of sweat gland is the highest.

In general, GSR has a typical startle response, which is a fast change of the GSR signal in response to a sudden stimulus. Features which are used to characterize this response include the amplitude and rising time of the signal. Figure 3.4 illustrates the startle response in GSR signal.

[2]http://labrosa.ee.columbia.edu/matlab/rastamat/

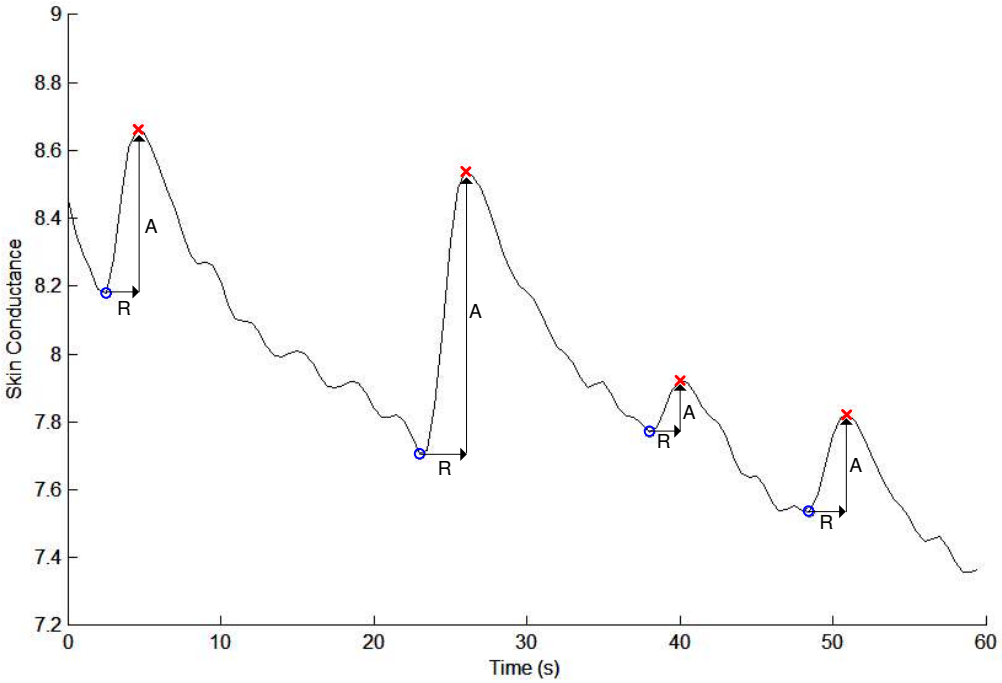


Figure 3.4: Four GSR startle responses. The peak which is detected by the algorithm is marked with 'x' and the onset is marked with 'o'. The amplitude and rising time of the response are denoted by A and R respectively.

Boucsein has demonstrated that skin conductance is subject to inter-person variability, with differences in age, gender, ethnicity, and hormonal cycle contributing to individual differences [65]. Due to these differences; we normalized the skin conductance signals by subtracting the baseline minimum and dividing by baseline range [66]. More precisely; it is given as:

$$GSR_{normalized} = \frac{GSR - min(GSR_{baseline})}{max(GSR_{baseline}) - min(GSR_{baseline})}$$

where $GSR_{baseline}$ corresponds to the values of the GSR when the user is supposed to be relaxed and measured at the beginning of experiment.

Several GSR features which are investigated are described as follows:

- Mean, minimum, and maximum of skin conductance.
- Standard Deviation of skin conductance.

- Total number of startle response in segment.
- The sum of startle amplitude ($\sum A$).
- The sum of rising time response ($\sum R$).
- The sum of energy response. It is estimated as the areas under the response $\sum \left(\frac{1}{2}A \times R\right)$.
- Mean, minimum, and maximum of peak height.

Picard et al. [54] have demonstrated that the total number of startle responses, the sum of startle magnitude, the sum of response's duration, and the sum of response's areas are a reliable feature for detecting stress. In this thesis, automatic detection of GSR responses was carried out by using EDA toolbox [3].

3.3 Classification Methods

We formulate the automatic stress detection as a supervised learning task. Given some past stress data, in which we know a label for each data (e.g. stress and non-stress), the aim is to learn a model such that given some previously unseen instance we can determine as accurately as possible to which group this instance belongs to. All classification was carried out in offline settings. The GSR and speech instances are constructed by segmenting the data using one-minute non-overlapping window. Furthermore, first the GSR data and the speech data are aligned, then, the instances which only contain the voiced sound of the subject are stored. Further details about how the instances are created are discussed more in Chapter 4.

We utilized four different machine learning techniques, including K-Means classifier using vector quantization as our baseline due to its simplicity, binary decision tree classifier, Gaussian Mixture Model (GMM) which has been shown to work well for speaker or music recognition and the Support Vector Machine (SVM), as the state-of-art for classification.

[3] https://github.com/mateusjoffily/EDA/wiki

3.3.1 K-Means

K-means clustering algorithm can be used for classification by using a Vector Quantization technique. The algorithm works by dividing a large set of vectors into groups having the same number of points closest to them. Each group is then represented by its centroid points. More precisely, the training algorithm works as follows:

- Choose the number of centroid n, for representing instances' distribution for each class.
- Run standard K-means algorithm for each class, with the number of centroid n. Hence, after performing K-means algorithm, we can use its centroid as a representation of all instances' distribution in that class.

After the training phase, the averaging Euclidean distance is used for classifying un-seen instances. The algorithm calculates the average distance of points to its closest (1-nearest-neighbor) centroids. The test instance is then classified to the class who minimizes this measure. More precisely, the pseudo code is illustrated in Algorithm 1.

Algorithm 1 K-Means Vector Quantization Algorithm

```
{Let T be a single testing instance to be classified}
{Let C be a set of classes. (e.g. 'stress' or 'non-stress')}
for i=1 to |C| do
    {Get centroid representation for class C_i}
    Centroid ← getCentroid(i)
    sumDistance ← 0
    for j=1 to |T| do
        minDist ← Infinite
        for k=1 to |Centroid| do
            t ← euclideanDistance(Centroid[k],T[j])
            if t < minDist then
                minDist ← t
            end if
        end for
        sumDistance ← sumDistance + minDist
    end for
    distanceClass[i] ← sumDistance
end for
return arg min_{1≤k≤|C|}(distanceClass[k])
```

3.3.2 Decision Tree Classifier

Decision tree classifier [67] is a simple and widely used method for classification. The classifier is based on flowchart-like binary tree structure, where each internal (non-leaf) node denotes a test on attribute, each branch represents an outcome of the test, and each leaf node holds a class label. First, the decision tree model has to be created from the training set and once has been constructed; classifying a test set is straightforward. Starting from the root node, we apply the test condition of the attribute and follow the branch. This process is repeated until we arrive at the leaf node which contains the outcome of the class label. We used the existing solution of decision tree classifier from Matlab Statistics Toolbox [4].

3.3.3 Gaussian Mixture Model (GMM)

A Gaussian Mixture Model (GMM) [68] is a parametric probability density function represented as a weighted sum of Gaussian component densities as given by the equation:

$$P(x|\lambda) = \sum_{i=1}^{M} w_i \; g(x|\mu_i, \Sigma_i)$$

where x is D-dimensional continuous-valued features, w_i are the mixture weights, and $g(x|\mu_i, \Sigma_i)$ are the component Gaussian densities. Each component density is a D-variate Gaussian function of the form:

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu_i)' \; \Sigma_i^{-1}(x-\mu_i)\right\}$$

where μ_i and Σ_i are a mean vector and covariance matrix respectively. The mean vectors, covariance matrix and mixture weights are generally represented by the notation:

$$\lambda = \{w_i, \mu_i, \Sigma_i\}, \;\; i = 1, ..., M$$

Given training vectors and a GMM configuration, the GMM parameter λ has to be estimated in order to match the distribution of the training vectors. The most popular algorithm to estimate this parameter λ is Maximum Likelihood (ML)

[4] http://www.mathworks.nl/products/statistics/

estimation. We refer to [69] for a detailed discussion about this algorithm. Figure 3.5 illustrates the mixture of Gaussian found by this algorithm which best fits the arbitrary histogram data.

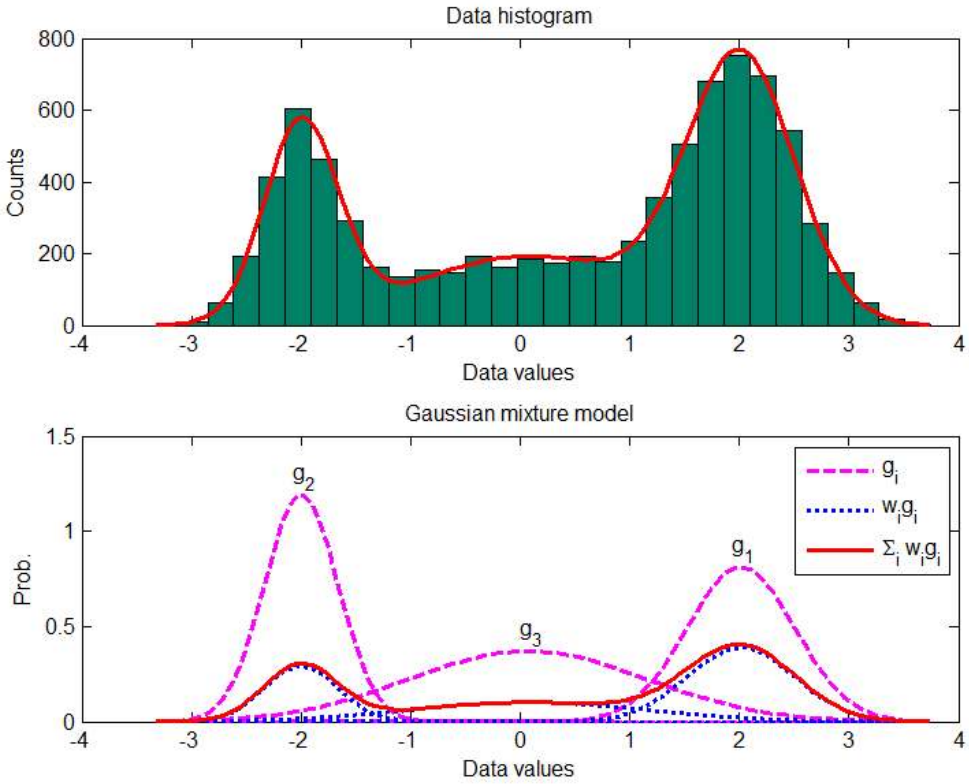


Figure 3.5: Top: Arbitrary histogram data. The GMM distribution fit is shown by the red line. Down: The Gaussian parameter (weight and densities) components.

The basic idea of GMM for pattern classification is similar to the K-Means algorithm. First, we find the best parameter GMM λ for each class given the training vectors (e.g. $\forall_{i=1...|C|}\lambda_i$). Next, once the Gaussian parameter has been found, determining a test vector's class is straightforward. According to the posterior Bayes inference:

$$P(\lambda_i|x) \propto P(x|\lambda_i)P(\lambda_i)$$

and assumption that each class has an equal *a priori* probability (e.g. $P(\lambda_i) = 1/C$), a test vector x' therefore is assigned to the class i which maximizes $P(x'|\lambda_i)$.

Depending on the choice of covariance matrices, GMM can have several different forms, including nodal covariance, grand covariance and global covariance. The first is a model which has one covariance matrix per Gaussian component. The second has a single covariance matrix for all Gaussian components in a model. The last has a single covariance matrix shared by all models. In addition, the covariance matrix type can be full or diagonal. In this thesis, we opted to use nodal, diagonal covariance matrix due to the fact that this configuration has been shown by [70] to result in better identification performance compared to the nodal and grand full covariance matrix for the speaker identification task. We used the Machine Learning Toolbox[5] [71] which contains the GMM implementation.

3.3.4 Support Vector Machine (SVM)

Support Vector Machines (SVMs) are the state-of-the-art supervised machine learning technique for data classification, which behaves robustly over a variety of different learning tasks. The basic idea behind SVM is to build a classifier model based on training data for which it not only separates the vector instances in one class from those in the others, but for which the separation, or margin, is as large as possible.

We used the LibSVM tool [72], an integrated library for Support Vector classification, regression, and distribution estimation. In order to classify the data using this tool, the guidelines from [73] are followed and are summarized as follows:

1. Transform data to the format of an SVM package.
2. Conduct simple scaling on the data.
3. Choose the Radial Basis Function (RBF) kernel $K(x, y) = e^{-\gamma|x-y|^2}$.
4. Select the random sample from training data and use cross-validation to find the best parameter C and γ.
5. Use the best parameter C and γ to train the whole training set and obtain the model.
6. Test the test set based on this model.

[5] http://neural.cs.nthu.edu.tw/jang/matlab/toolbox/machineLearning/

3.4 Stress Detection Using Fusion of GSR and Speech

There are many options to combine the result of two different models. One option is to enrich feature space and construct a model which could separate the instances in the best possible way. Another option is to build an individual model and combine them using ensemble learning. Figure 3.6 illustrates the feature enrichment and ensemble learning methods.

Ensemble learning is generally used to combine prediction decision from multiple models and aggregates their results into a single class label. There are many ensembles learning methods, which are typically used, including voting, adaptive weighting, stacking, logistic regression, fuzzy integrals, co-training, bagging, boosting, random subspace, and many more. We used logistic regression method for combining the result from two different models (i.e. speech and GSR) into one single regression value. The rationale behind choosing this method is that it is simple, and has been shown to work well for combining two different models in [74]. Logistic regression requires that the GSR and speech data are available at the same time instance and have been aligned with each other. We carried out a manual segmentation and alignment of speech data in offline settings. However, in the operational settings, this technique requires a real-time segmentation and alignment of speech and GSR.

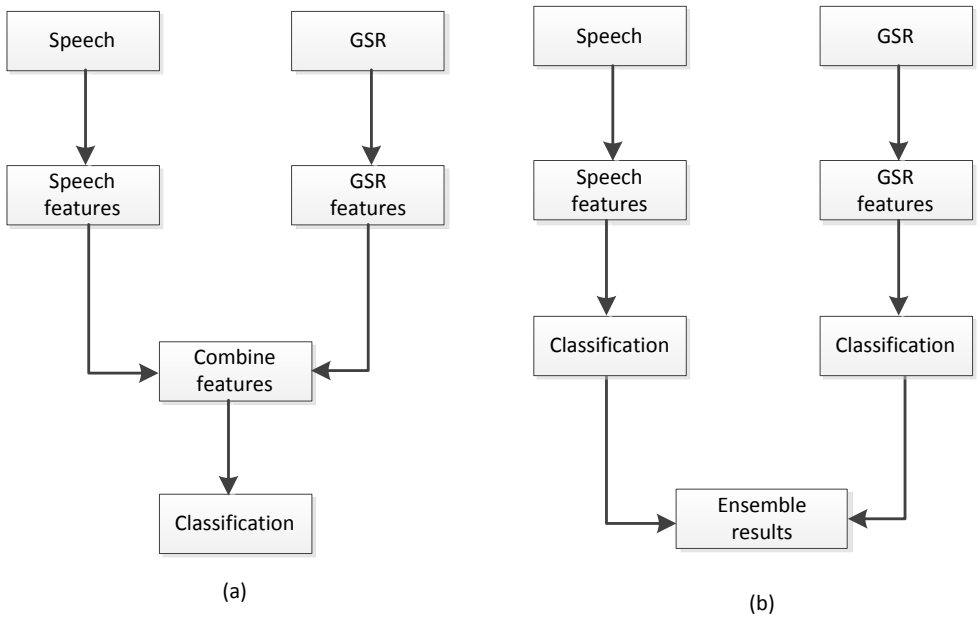


Figure 3.6: (a) Enrichment of feature space. (b) Ensemble learning approach.

Logistic regression [75] is a standard method for addressing binary classification problems. For instance, logistic regression might be used for predicting whether a patient has a diabetes disease based on the observed characteristics of patients (e.g. age, gender, body mass index and blood test). This method is useful owing to its ability to map any arbitrary input values from negative infinity to positive infinity to the outcome values between zero and one. Logistic regression is deemed more appropriate for modeling binary classification problems, compared to linear regression. The most obvious reason is that the linear regression model has no bounds on what the outcome values will be. Hence, the outcome might have a value lower than zero or greater than one, which is unsuitable to our binary classification purpose.

Let's assume that we have N observed instances and each instance i consists of M independent variables. Let the outcome of associated binary-valued denoted as y_i is the probability of two possible values (e.g. 0 and 1). The goal of logistic regression is to model the relationship between the independent variables and the outcome, so it can correctly predict an unseen instance for which only the independent

variables are available. Logistic regression used the same mechanism as in linear regression by modeling the probability p_i, the probability of the outcome of 1 (e.g. "success", "yes", etc), with a linear combination of the independent variables and the set of regression coefficients. More precisely, the linear predictor function $f(i)$ is given as:

$$f(i) = c_0 + c_1.x_{1,i} + ... + c_M.x_{M,i}$$

where $c_0, ..., c_M$ are regression coefficients and $x_{1,i}, ..., x_{M,i}$ are the input of instance i which consists of M independent variables. After defining the linear predictor function, we linked it with the probability of binary-valued outcomes using this equation:

$$ln\left(\frac{p_i}{1-p_i}\right) = c_0 + c_1.x_{1,i} + ... + c_M.x_{M,i}$$

where ln is the natural logarithm. Coefficients $c_0, ..., c_M$ are estimated from the training instances to best match its outcome by utilizing the maximum likelihood algorithm [75]. Once these coefficients have been found, we can use them to predict an unseen (testing) instance by using the above equation. In this thesis, we used the Matlab Statistics Toolbox [6] for finding these regression coefficients.

[6] http://www.mathworks.nl/products/statistics/

Chapter 4

Evaluation of Stress Detection

In this section, we conduct a series of experiments in order to examine the performance of different models for detecting two distinct states of stress level: recovery phase with workloads phase and light workload with high workload phase. Four different machine learning classifiers, K-Means, Decision Tree Classifier, Gaussian Mixture Model (GMM) and Support Vector Machine (SVM), will be investigated, and their performance will be compared. We opt to choose the simplest K-Means algorithm as our baseline throughout the whole experiments.

The remainder of this chapter is organized as follows. In Section 4.1, we describe the experimental settings, such as the evaluation metrics, cross-validation, statistical significance test and Kappa inter-annotator agreement. Section 4.2 describes the dataset for the experiment, together with its preprocessing and analysis. Finally, in Section 4.3, we present and discuss the experimental results.

4.1 Experimental Setting

4.1.1 Evaluation Metrics

The classification evaluation is utilized to measure the performance of the classifier in the classification task. The best classifier which gives the finest evaluation in turns will be chosen as a final classifier. The most common metrics used for evaluation are accuracy, precision and recall. In order to compute these metrics for binary classification, we need to calculate the number of true positive, false

positive, true negative and false negative as depicted in Table 4.1. In our context, the positive class would be a stress class, and the negative class would be a non-stress class.

Actual Class	**Predicted Class**	
	Positive	**Negative**
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

Table 4.1: Classification confusion matrix.

The accuracy, precision and recall are defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

4.1.2 Cross-Validation

10-fold Cross-Validation

Cross-validation is a statistical method for validating a predictive model which is mainly used to estimate how accurately a model will perform in practice. In 10-fold cross-validation, the dataset is randomly divided into 10 subsets. For each subset, one will be used as testing data and the remainder as training data. The process is then repeated 10 times. Afterwards, the 10 results from the folds can be averaged to produce a single estimation.

Figure 4.1 illustrates the training and testing procedure using the first fold as testing and the rest as training data. First, the best parameter is searched from training data (fold 2 to fold 10) by using a 5-fold cross-validation. The rationale behind choosing 5-fold lies in the limitation of the data size. After the best parameter is obtained, it will then be used to construct the model. The testing data

from the first fold is then evaluated using this model, and its accuracy is reported. This process is repeated 10-times and accuracies are averaged to produce a single estimation.

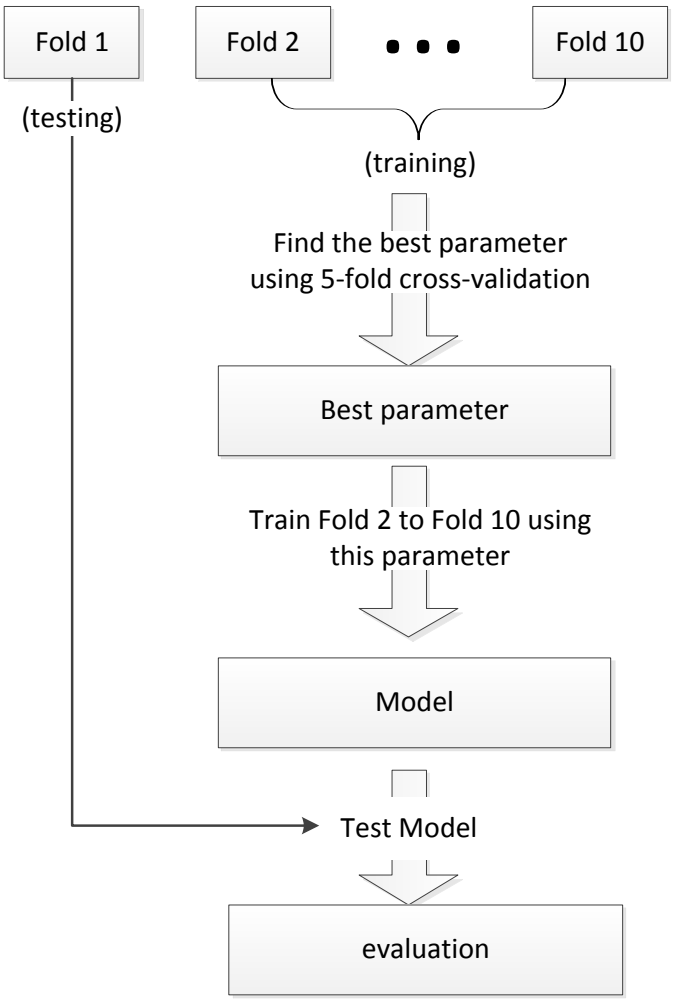


Figure 4.1: 10-fold cross validation illustration using fold 1 as testing and the rest as training data.

1-Subject-Leave-Out Cross-Validation

1-subject-leave-out cross-validation is used to evaluate the model performance for the subject independent case. This method, in nutshell, works as follows. The

dataset is divided into n subset, where n denotes the number of the different subject. The data from the same subject cannot be both present in the testing and training data. For each subset, it will be used for testing and the rest as training data. The process is then repeated n times, and the result can be averaged to produce a single estimation.

4.1.3 Statistical Significance Test

Statistical significance is an assessment based on statistics to know whether the observation reflects a pattern rather than just a chance. We utilized the permutation test [76] (also called exact tests, randomization tests, or re-randomization tests) for testing the significance of data. The permutation test concept is relatively simple and intuitive, which involves permutation (or randomization) of the data label to test the statistical significance.

Algorithm 2 illustrates the permutation test. If the calculated `p-value` is below a certain threshold chosen for statistical significance, usually 0.05, then the `null-hypothesis` is rejected. In essence, the `null-hypothesis` states that two distinct states of stress level (e.g. stress and non-stress) are not differentially dependent on the data, in the sense that if we used different data or changed the order of data, we would have observed exactly the same results. One of the permutation test benefits is that it does not rely on the data distribution. In other words, this test is a non-parametric test. However, a drawback of this test is that it requires a very expensive computation time.

Algorithm 2 Permutation Test

```
1: {Let c be the best accuracy obtained from one model.}
2: p-value ← 0
3: for i=1 to n do
4:     Permute the label of data (both training and testing).
5:     {The same setup as the one that was utilized to obtain c, is used in modelRun().}
6:     {modelRun() returns the evaluation performance, i.e. an accuracy.}
7:     c' ← modelRun()
8:     if c' ≥ c then
9:         p-value ← p-value + 1
10:    end if
11: end for
12: return (p-value / n)
```

4.1.4 Kappa Inter-Annotator Agreement

Cohen's Kappa coefficient is a statistical measure for inter-annotator agreement [77]. We utilized this measure to investigate the agreement between two different models (e.g. model based on GSR and model based on speech) on the same dataset. The Kappa coefficient is given as:

$$\kappa = \frac{pr(a) - pr(e)}{1 - pr(e)}$$

where $pr(a)$ is the observed relative agreement between raters and $pr(e)$ is the hypothetical probability of chance agreement. The κ has value 1 if the raters are in complete agreement. If there is no agreement between the raters, κ has value 0. For the sake of clarity, let's consider the example given in Table 4.2, which depicts the agreement between model A and B.

		B	
		Yes	**No**
A	**Yes**	30	5
	No	10	25

Table 4.2: Inter-annotator confusion matrix example.

Model A and B both said "Yes" on 30 instances and both said "No" on 25 instances. Thus, the observed percentage of agreement is $pr(a) = (30+25)/70 = 0.79$. Model A said "Yes" 35 times and "No" 35 times, hence the probability model A said "Yes" is 50% of the time. Model B said "Yes" 40 times and "No" 30 times, hence the probability model B said "Yes" is 57.1%. Thus, the probability that both models said "Yes" randomly is $0.5 \times 0.571 = 0.29$, and that both of them said "No" randomly is $0.5 \times 0.43 = 0.22$. The overall probability of random agreement then becomes $pr(e) = 0.29 + 0.22 = 0.51$. Applying the κ formula above, we get the following inter-annotator agreement:

$$\kappa = \frac{0.79 - 0.51}{1 - 0.51} = 0.57$$

4.2 Dataset Description and Evaluation

Due to the unavailability of a free stress dataset which incorporates both speech and GSR, we conduct a controlled psychological stress elicitation experiment. The

main goal of this experiment is to obtain a labeled stress dataset. The whole experiment is elaborated in Appendix B.

The stress elicitation experiment was conducted inside TU/e by employing 10 volunteer graduate students from the Mathematics and Computer Science department. The experiment itself, in summary, consists of three different tasks which have to be performed: recovery (relaxation), light workload, and heavy workload session. In total, we collected approximately 10 hours of stress-related data from the experiment.

The GSR patterns we observed during the whole experiment can be grouped into three categories. The first pattern is in line with our initial expectation. See Figure 4.2 (a) for illustration. The average mean GSR's value during light workload is lower than the mean value during the heavy workload session. Moreover, we observed that the GSR patterns decrease in the recovery period as the subject probably felt less stress. In the second pattern (see Figure 4.2(b)), there are no statistically significant differences between the average mean value of the light and heavy workload session. The subject was presumably easy to get aroused even with the task which was very weightless, hence he or she was most likely more susceptible to stress. The third pattern (see Figure 4.2 (c)) is completely different from the other two. The GSR patterns decrease in almost all of the tasks as if the subject performed the relaxation task. This might be caused by the subject ability to cope with the stress or the fact that the subject was not motivated at all in participating in this test. We found this pattern in only 1 out of 10 instances.

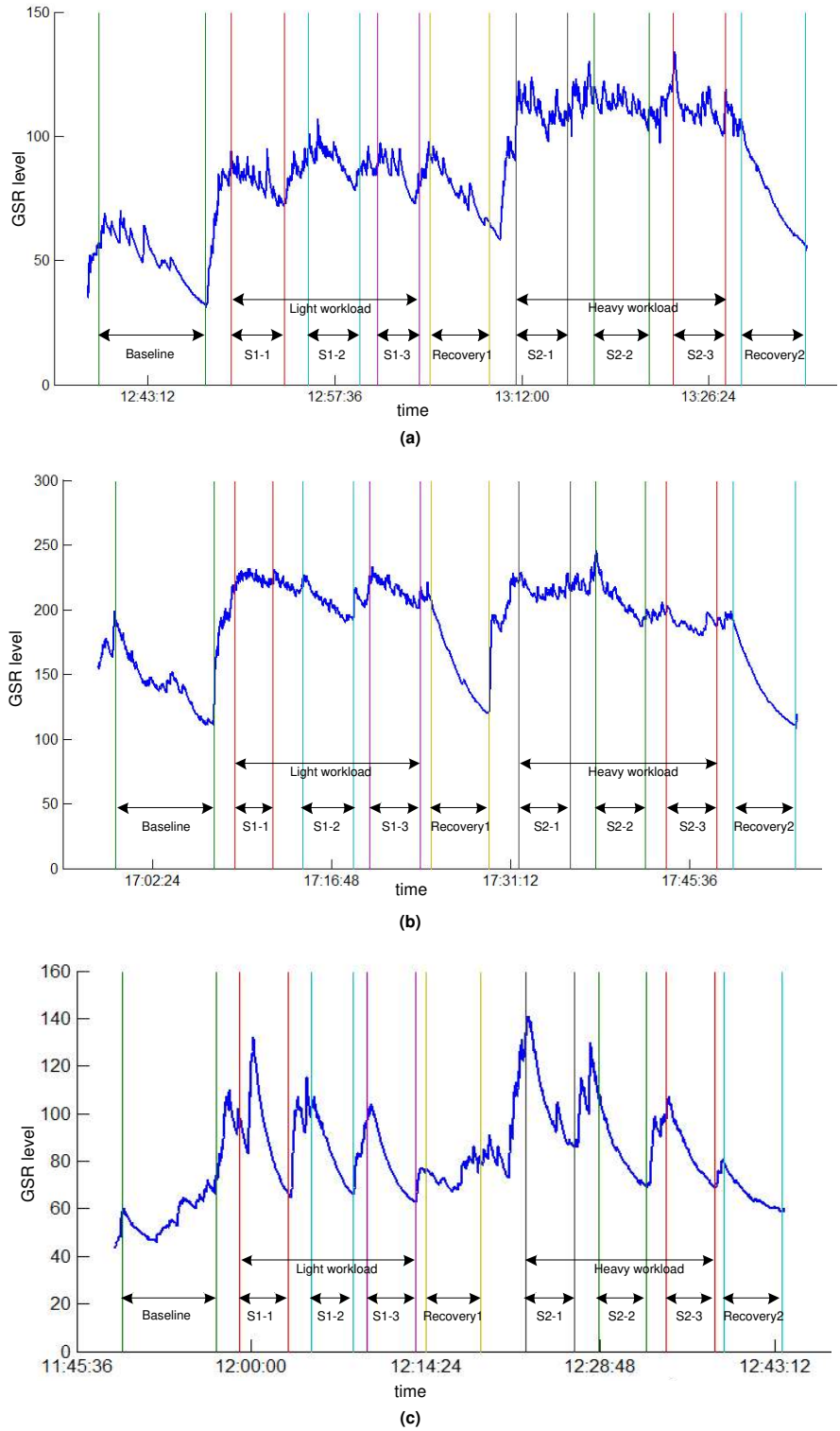


Figure 4.2: Three types of GSR patterns. (a) The first type. (b) The second type. (c) The third type.

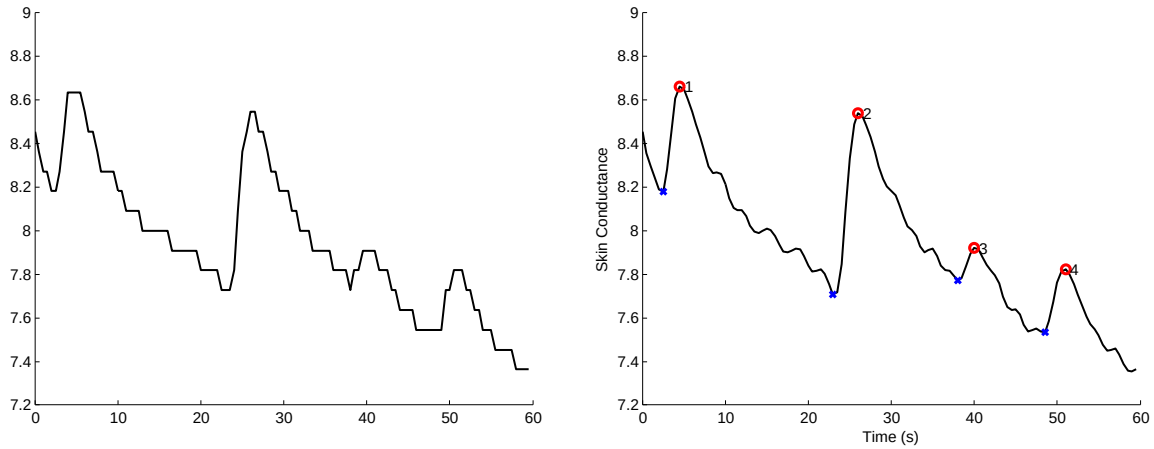
The data which has been collected contains approximately 10 hours of GSR and speech, measured on 10 subjects. We segmented the GSR and speech raw values by using a moving 60-second non-overlapping time window. The rationale behind using a 60-second window is that we can observe a reasonable GSR startle response and sufficient utterances of voiced sound within this period. The GSR and speech data are synchronized using a global clock system approach. More precisely, the computer clock system was stored side by side with the GSR data (e.g. "2012/06/28 13:10:25.100 40", representing the GSR value of 40 at 28-June-2012 13:10:25.100 hours). The information about the start and end time of each task was stored separately in our system. Whenever the system stored these values (e.g. start and end time), the short burst of beep sound was played from the computer and was recorded by the speech recorder. Therefore, the speech and GSR data can be aligned together. Table 4.3 summarizes the total instances obtained from recovery, light workload, and heavy workload session.

Phase	**GSR**	**speech**
Recovery session	100	-
Light workload session	110	110
Heavy workload session	120	120

Table 4.3: Total GSR and speech instances.

Three preprocessing steps were carried out for the GSR data. First, the data was normalized by using its baseline to minimize the impact of individuality (e.g. GSR measurement varies among individuals). Secondly, the GSR time series were filtered by using a low pass Butterworth filter with a cut-off frequency of 0.5 Hz. Finally, the GSR response was localized by using the EDA toolbox. Figure 4.3 illustrates the GSR raw time-series data and its shape after having been preprocessed.

We used Praat software [78] for a manual speech segmentation. The speech was segmented for each task, based on two short burst of beep sounds. After the specific task had been localized, we segmented the speech into several instances using a 60-second non-overlapping time window. Finally, we removed the other sounds apart from the subject's utterances from this instance (e.g. we removed the evaluator speech, environmental noise, etc.).



(a) (b)

Figure 4.3: One minute GSR instance. (a) Raw GSR graph. (b) GSR graph after having been preprocessed. The GSR responses found by the EDA toolbox are shown in the picture.

Figure 4.4 shows the plot of three different instances (e.g. recovery, light workload, and heavy workload session) using three distinct GSR features: mean, number of responses and energy of response. It is evident in this graph that the recovery phase has fewer numbers of responses and lower energy of response, compared to the light and heavy workload phases. This finding is in line with our first hypothesis (see Appendix B), which states that the number of GSR startle response during the recovery period should be lower than the workloads. However, it seems there is no clear cut, which could separate these instances based only on these three features.

We also investigate the relation between GSR and speech by plotting their mean value together (See Figure 4.5 (a)). It is obvious from this graph that the heavy workload and the light workload instances are hardly separable by using only two features. Figure 4.5 (b) depicts the distribution of the same instances aggregated based on the subject. From this figure, we can infer that different individuals show distinct GSR and pitch characteristics. For example, subject 9 shows a significant difference of GSR and pitch value between the light workload and the heavy workload, which is not the case for subject 8. The mean of GSR and pitch during the heavy workload is higher than that of the light workload for several subjects. This finding supports our second and third hypotheses (see Appendix

B): The mean of skin conductance during the heavy workload should be higher than the light workload, and the mean of pitch should increase under a stressful condition, respectively.

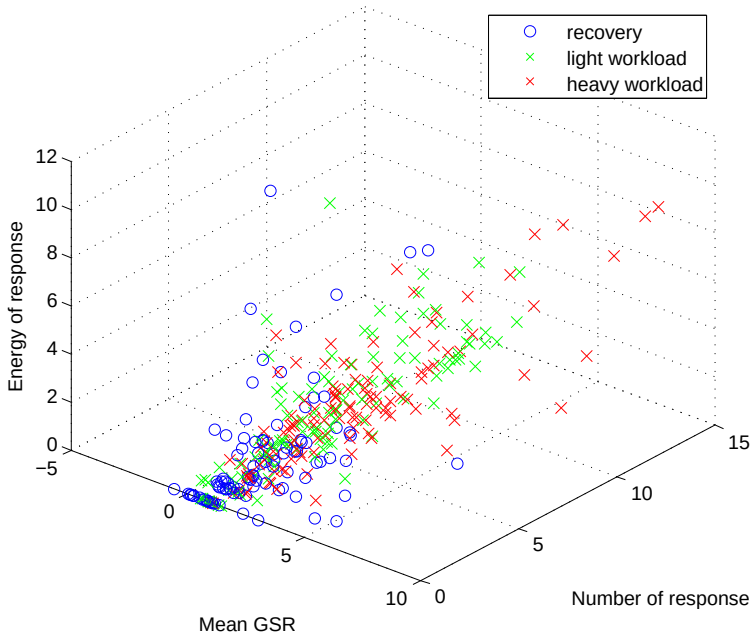
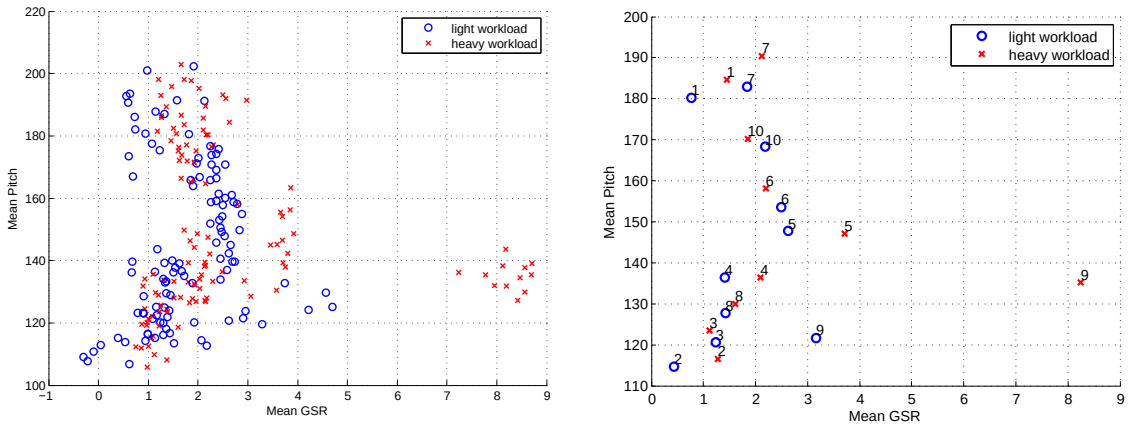


Figure 4.4: Three different instances: recovery, light workload and heavy workload.



(a) (b)

Figure 4.5: (a) The distribution of all instances. (b) The average mean of GSR and speech aggregated with respect to the subject. The number on top of the symbol represents the subject id.

4.3 Experimental Setups and Results

4.3.1 Subject Dependent Model

The experiments are conducted to find the best classifier model to detect two different states of stress level. In this section, the experiments were conducted using 10-times 10-fold cross-validation since the dataset size is small. This method works by repeating a 10-fold cross-validation 10 times, and the average statistics are returned for the evaluations. We called this model subject dependent since the data from the same subject can be present both in the training and testing sets.

Stress Model using GSR features

In this section, the binary classification models for three different scenarios are investigated. The first one is a model involving recovery and workloads (light and heavy) session. The second one is between recovery and heavy workload. The last one is between light workload and heavy workload session. The dataset distribution for these three different scenarios is depicted in Table 4.4.

Recovery vs workloads	**#recovery**	100
	#workloads	230
Recovery vs heavy workload	**#recovery**	100
	#heavy workload	120
Light vs heavy workload	**#light workload**	110
	#heavy workload	120

Table 4.4: The distribution of dataset.

The following features were used in this GSR experiment: mean GSR, maximum GSR, minimum GSR, maximum GSR - minimum GSR, number of GSR's response, mean peak, maximum peak, minimum peak, maximum - minimum peak, amplitude of response, rising time response and energy of response. Table 4.5 depicts the average classifier accuracy using 10-times 10-fold cross-validation. More detailed evaluations involving accuracy, precision and recall can be found in Appendix C.

It is evident in the result that the GMM, SVM and decision tree method outperformed the baseline (K-Means). The SVM outperforms the other classifiers and

	K-Means	GMM	SVM	Decision Tree
Recovery vs workloads (light & heavy)	46.12±2.25	70.51±0.49	**79.66±0.77**	73.45 ± 1.27
Recovery vs heavy workload	55.54±2.64	74.90±0.79	**80.72±0.61**	77.81±1.31
Light vs heavy workload	53.21±1.00	66.82±0.46	**70.60±1.10**	62.52±1.79

Table 4.5: Binary classification accuracy (in percent) using 10-times 10-fold cross-validation scheme. Boldface: the best accuracy for a given setting (row).

can reach accuracy around 80.72% for classifying recovery versus heavy workload session. The result indicates also that differentiating the stress level in the light versus heavy workload setting is harder than in the recovery versus heavy workload setting.

Stress Model using Speech features

In this section, the speech features are used to build a model which can classify an instance into binary classes: light workload and heavy workload. The distribution of the dataset is depicted in the third row in Table 4.4. The features which are investigated include pitch, MFCC, MFCC-Pitch and RASTA PLP. A total of 12 pitch features are used, including mean, minimum, maximum, median, standard deviation, range (maximum - minimum) of pitch and its first derivation. As for the MFCC features, we used 144 features, which consist of mean, variance, minimum and maximum of the first 12 cepstral coefficients (excluding the 0-th coefficient), delta coefficients (the first derivative of coefficients) and delta-delta coefficient (the second derivative of coefficients). The feature MFCC-Pitch represents the direct concatenation of MFCC and pitch features. In total, 108 RASTA PLP features were used, which consist of statistics such as mean, variance, minimum and maximum of RASTA PLP coefficients, its first derivative and the second derivative. The experimental result using these features with 10-times 10-fold cross-validation is depicted in Table 4.6. More detailed evaluations involving accuracy, precision and recall can be found in Appendix C.

K-means results in the lowest accuracy and is unsuitable for stress detection. The GMM classifier outperforms K-Means (baseline) with insignificant differences. In general, the GMM accuracies using speech features are lower than using GSR features. In contrast, the SVM method, which is not based on the distribution fit technique, gives a high accuracy in this setting. This is most likely due to the high dimensionality of speech (e.g. up to 144 dimensions for MFCC) which makes the Gaussian distribution sparse, thus, making the Expectation-Maximization al-

	K-Means	GMM	SVM	Decision Tree
Pitch	49.65±2.28	58.82±1.46	**62.08±1.57**	55.60±2.75
MFCC	55.39±1.92	56.78±1.76	**92.39±0.58**	68.86±3.07
MFCC-Pitch	49.17±2.34	59.08±0.94	**92.56±1.63**	70.69±1.33
RASTA PLP	50.60±0.42	52.30±2.78	**91.69±0.94**	71.47±2.97

Table 4.6: Speech classification accuracy (in percent) using 10-times 10-fold cross-validation scheme. Boldface: the best accuracy for a given setting (row).

gorithm fail to cluster and fit the data. One possible way to address this issue is by considering the sparseness of data in the Gaussian EM algorithm as shown in [79]. The decision tree classifier, despite its simplicity, performs quite well in this setting and can reach a reasonable accuracy up to 70%. The SVM model results in the best accuracies, compared to the rest of classifiers. The SVM reached the accuracy of 92.39% using the MFCC. By using the concatenation of MFCC and pitch, the SVM reached the highest accuracy of 92.56%, though the improvement is insignificant. Furthermore, it is evident that pitch alone is not a good indicator for stress classification, as it gives the worst result, compared to other features.

Stress Model using Fusion of GSR and Speech

There are many approaches for combining the GSR and speech. One of them is by using feature enrichment. Both features from speech and GSR are combined together. Afterwards, the classifier is utilized to predict the class output. Another approach is by using ensemble learning such as the logistic regression technique. First, an individual model is built separately using the best parameter which had been found in the previous section. Afterwards, the result of both models is combined using logistic regression to produce a new regression of class output. All classifications in this section were carried out by using the SVM classifier. The result of classification is shown in Table 4.7. The accuracy, precision and recall for each feature can be found in Appendix C.

	Enriching Feature Space	Logistic Regression
MFCC and GSR	90.73±1.19	**92.43±0.77**
MFCC-Pitch and GSR	91.34±1.07	**92.47±1.37**
Pitch and GSR	69.04±1.24	**70.17±2.36**

Table 4.7: Fusion of Speech and GSR classification accuracy (in percent) using 10-times 10-fold cross-validation scheme. Boldface: best accuracy for a given setting (row).

The logistic regression method outperforms feature enrichment with negligible improvement. Figure 4.6 illustrates the best overall results obtained using individual model and fusion of models. We can conclude from this result that combining two different models, both using logistic regression and feature enrichment, does not improve the performance in a significant way.

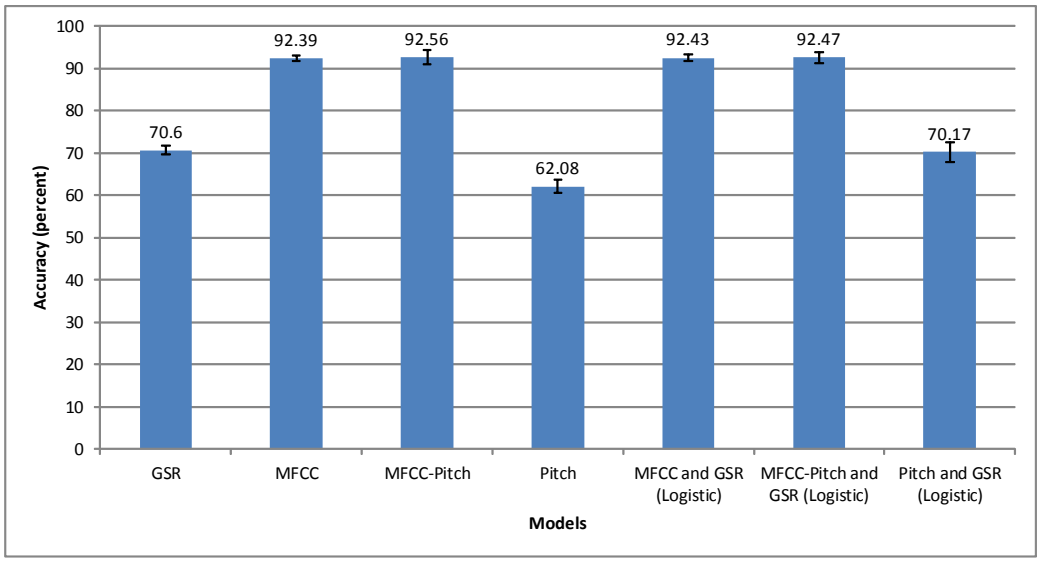
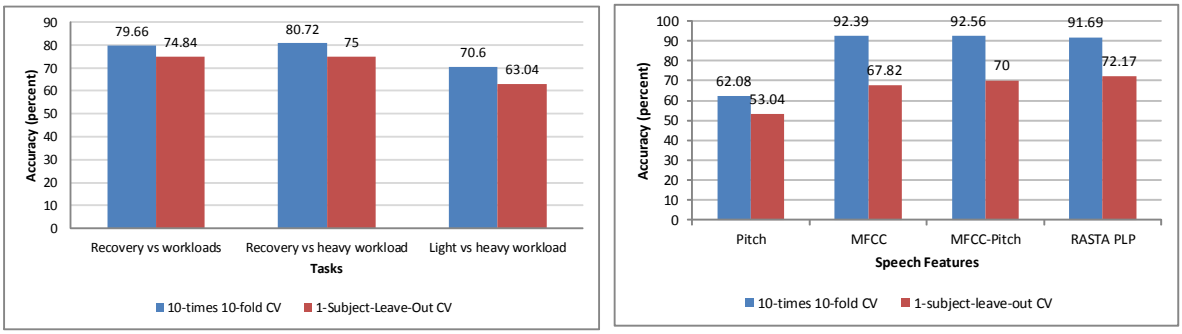


Figure 4.6: Overall classification accuracies of individual and combined models.

4.3.2 Subject Independent Model

The 1-subject-leave-out cross-validation approach was studied to evaluate the model performance for the subject independent case. Figure 4.7 shows a comparison of the model obtained using 10-times 10-fold cross-validation against 1-subject-leave-out cross-validation. The accuracies using 10-times 10-fold cross-validation

are obtained from the best result of the previous section. It is obvious from this graph that the accuracies of the classifiers are dropped when using the subject independent model. Hence, it is better to address the stress classification problem as a subject dependent model.



(a) (b)

Figure 4.7: Comparison of two different evaluations: 10-times-10-fold CV and 1-subject-leave-out CV. (a) GSR. (b) Speech.

4.3.3 Statistical Significance Test

The permutation test with the number of iteration $n = 100$ was run to test the statistical significance for each model in GSR and speech. We chose the number of repetition 100 owing to the fact that it took around three hours to complete each experiment using this value. All results gave a zero `p-value`, therefore, the null-hypothesis can be rejected. In other words, the dataset is statistically significant for determining two different stress levels.

4.3.4 Disagreement Test between GSR and Speech Model

Cohen's Kappa test was utilized to measure the agreement between speech and GSR model for classifying the same instances. The value $\kappa = 0$ means no agreement between models, while $\kappa = 1$ means they are in complete agreement. The inter-annotator agreement result is depicted in Table 4.8. It is evident from this result that the models gave a poor agreement, especially for pitch and GSR that result in $\kappa = 0.13$. These results indicate that both models may have a high diversity or independence in the ensemble. Thus, it can be exploited to achieve a higher accuracy by finding the optimal strategy for ensemble. For example, the

techniques such as using a dynamic integration of an ensemble classifier [80][81] may give a promising outcome.

	Kappa
MFCC and GSR	0.32 ± 0.18
MFCC-Pitch and GSR	0.31 ± 0.21
Pitch and GSR	0.19 ± 0.12

Table 4.8: Kappa Inter-Annotator agreement result.

Chapter 5

Conclusions and Future Work

In this work, we have studied the problem of managing stress-related data, visualization, analysis and multi-modal data mining for stress detection. In this chapter, we summarize, in brief, the contribution, formulate the conclusion which can be drawn from the result and finally present the directions for future work.

5.1 Main Contribution

The main results and contribution of this thesis can be divided into two categories: the stress analytics framework and multi-modal data mining for stress detection. The framework for stress analytics proposed solutions for management of stress related data, basic analysis using OLAP for stress explorations, query-by-example analysis, visualization and automated stress classification using multimodal affective data captured from text, speech, GSR, facial expression and other physiological signals. The framework itself, by nature, is easy to extend and modify. This framework enables the user or domain expert to analyze the interesting stress patterns, mining raw data for analysis, visualize various pieces of evidence of stress, gain an insight into the potential causes of stress and make people aware of this information, so they can cope with stress in the best possible way. In addition, the framework also enables the possibility for data mining or pattern mining based on OLAP result, which may explain the relationship between stress and other factors even better.

Due to the modern technologies, the objective measurement of stress level is be-

coming possible by means of sensor, such as a GSR device. Several features of GSR and speech can be extracted. Afterwards, the supervised machine learning classifier was employed to classify the instances. Supervised machine learning required an openly available benchmark for evaluations. Unfortunately, we cannot find a free dataset which consists of both speech and GSR. Hence, we conducted a controlled psychological experiment to obtain the labeled dataset.

The experiment itself was conducted inside TU/e by employing 10 volunteer graduate students from the Mathematics and Computer Science department. The objective of this experiment was to elicit a certain stress level on the subject, while taking the objective and subjective measurement at the same time. The objective measurement, such as speech, facial expression and GSR, were taken. The results of the experiment show that different individuals exhibit distinctive characteristics of both GSR (e.g. mean of GSR) and speech (e.g. mean of pitch) patterns. As for the subjective measurement, we used questionnaires. In total, we obtained 10 hours of affective data which was used in the stress detection experiment.

We investigated different models for detecting stress using four classifiers: K-means, GMM, SVM and decision tree. The SVM outperformed the other classifiers and can reach an accuracy of up to 92% by using speech features. We conclude that these experiments show that speech and GSR provide a viable method of measuring stress level in laboratory settings. The results showed that speech is indeed a good indicator for determining stress. On the other hand, it turns out that using only GSR data is not sufficient for determining the stress level as it at most can reach an accuracy of 70% for differentiating between light and heavy workload. Furthermore, it has been demonstrated in [82] that the GSR signal is varied not only from person to person but also e.g. from day to another for the same person. Therefore, including another measurement instead only from GSR will be more reliable. Combining both GSR and speech using feature enrichment or logistic regression does not improve the performance in a significant way. Finally, the permutation test revealed that our dataset showed a statistical significance of stress level instead of chance.

5.2 Future Work

In our current work, we have developed all parts of the framework for stress analytics, except the data and pattern mining from the OLAP result. It will be interesting to incorporate data mining or pattern mining from the OLAP result into the framework, such that the finer grained analysis of stress can be conducted. The framework is straight forward to extend, hence other raw data, such as facial expression, can be added to the system. As for stress level, the framework which we have developed can only recognize two different stress levels, e.g. stress and non-stress. However, it should be noticed that stress can also be positive or negative depending on the context. For instance, positive stress can be caused by the excitement or an intrinsic motivation of the individual. Therefore, the system which can detect this is promising. One possible way to detect positive and negative stress is to utilize context-aware data mining by using an additional source of information, such as personal diary, email, speech and facial expression, to infer the subject's emotion at that particular time.

We have collected the labeled stress data from 10 subjects by means of physiological stress elicitation experiment. Although we got the high accuracies from this data, it will be more interesting to collect more data from the experiment to enable better analysis. Furthermore, we have collected the facial expression during the experiment which was not used for creating the stress model. In essence, the facial expression can be analyzed using computer vision techniques based on Paul Ekman's model of Facial Action Coding System (FACS). FACS is a system which taxonomizes human facial expression and is commonly used to categorize the physical expression of emotions. As the muscle in facial expression is controlled by the autonomic nervous system, we argue that stress can be detected from facial expression as well.

The supervised machine learning can classify binary states of stress level up to 92% accuracy for the experiment in laboratory settings. However, detecting stress levels in an operational setting will be a challenging task and more difficult than in a rigorous laboratory environment. In a real-life setting, there is no guarantee that the signals will be free of noise. For instance, the speech may contain background noise, while the GSR signals may contain artifacts owing to the exact placement of the sensor and the physical activity. The stress detection model can be extended to handle this issue, for example, by introducing a filter to remove static background

noise, which may occur during the recording. Furthermore, in an operational setting, the speech and GSR may not be available at the same time, hence it is necessary to design a model which can recognize and solve this situation. It has been known that stress is influenced by various factors, including, but are not limited to environment, personality, motivation, emotion and activity. Hence, we suspect the model performance for real-life settings will be worse than a laboratory environment.

Bibliography

[1] S. Holmes, D. Krantz, H. Rogers, J. Gottdiener, and R. J. Contrad, "Mental stress and coronary artery disease: A multidisciplinary guide.," *Progress in Cardiovascular Disease*, no. 49, pp. 106–122, 2006.

[2] T. Pickering, "Mental stress as a causal factor in the development of hypertension and cardiovascular disease," *Current Hypertension Report*, no. 3, pp. 249–254, 2001.

[3] J. Herbert, "Fortnighly review. Stress, the brain, and mental illness.," *BMJ (Clinical research ed.)*, vol. 315, pp. 530–535, Aug. 1997.

[4] C. Jewitt, ed., *The Routledge Handbook of Multimodal Analysis.* New York: Routledge (Taylor and Francis), 2011.

[5] K. L. O. Halloran and B. A. Smith, "Multimodal Text Analysis,"

[6] K. Young, "Applying Multimodal Analysis to MySpace: An Instructional Framework to Develop Students' Digital Literacy," *Advances in Communications and Media Research (Volume 8)*, 2011.

[7] K. L. O'Halloran, A. Podlasov, A. Chua, C.-L. Tisse, and F. V. Lim, "Challenges and Solutions to Multimodal Analysis: Technology, Theory and Practice," *Developing Systemic Functional Linguistics: Theory and Application.*

[8] E. J. Keogh, "Exact indexing of dynamic time warping," pp. 406–417, 2002.

[9] M. Vlachos, D. Gunopoulos, and G. Kollios, "Discovering similar multidimensional trajectories," *Data Engineering, International Conference on*, vol. 0, p. 0673, 2002.

[10] L. Chen and R. Ng, "On the marriage of lp-norms and edit distance," in *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, VLDB '04, pp. 792–803, VLDB Endowment, 2004.

[11] P.-F. Marteau, "Time warp edit distance with stiffness adjustment for time series matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 306–318, Feb. 2009.

[12] J. Lin, S. Williamson, K. Borne, and D. DeBarr, *Pattern recognition in time series*. 2012.

[13] K. Gollmer and C. Posten, *Detection of distorted pattern using dynamic time warping algorithm and application for supervision of bioprocesses*. 1995.

[14] B. kee Yi, H. V. Jagadish, and C. Faloutsos, "Efficient retrieval of similar time sequences under time warping," pp. 201–208, 1997.

[15] S. Chu, E. Keogh, D. Hart, M. Pazzani, and Michael, "Iterative deepening dynamic time warping for time series," in *In Proc 2 nd SIAM International Conference on Data Mining*, 2002.

[16] K. Deng, A. W. Moore, and M. C. Nechyba, "Learning to recognize time series: Combining arma models with memory-based learning," in *In IEEE Int. Symp. on Computational Intelligence in Robotics and Automation*, pp. 246–250, IEEE Press, 1997.

[17] J. Lin and Y. Li, "Finding structural similarity in time series data using bag-of-patterns representation," in *Proceedings of the 21st International Conference on Scientific and Statistical Database Management*, SSDBM 2009, (Berlin, Heidelberg), pp. 461–477, Springer-Verlag, 2009.

[18] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient similarity search in sequence databases," pp. 69–84, Springer Verlag, 1993.

[19] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The r*-tree: an efficient and robust access method for points and rectangles," in *Proceedings of the 1990 ACM SIGMOD international conference on Management of data*, SIGMOD '90, (New York, NY, USA), pp. 322–331, ACM, 1990.

[20] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," in *Proceedings of the 1994 ACM SIGMOD international conference on Management of data*, SIGMOD '94, (New York, NY, USA), pp. 419–429, ACM, 1994.

[21] K.-P. Chan and A. W. chee Fu, "Efficient time series matching by wavelets," in *In ICDE*, pp. 126–133, 1999.

[22] E. Keogh, K. Chakrabarti, S. Mehrotra, and M. Pazzani, "Locally adaptive dimensionality reduction for indexing large time series databases," *ACM Trans. Database Syst.*, vol. 27, pp. 188–228, June 2002.

[23] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing sax: a novel symbolic representation of time series," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 107–144, 2007.

[24] E. Tromp, "Multilingual sentiment analysis on social media," Master's thesis, Eindhoven University of Technology, the Netherlands, 2011.

[25] G. Mishne, "Experiments with mood classification in blog posts," in *1st Workshop on Stylistic Analysis Of Text For Information Access*, 2005.

[26] J. Zhai and A. Barreto, "Stress detection in computer users through noninvasive monitoring of physiological signals.," *Biomedical Sciences Instrumentation*, vol. 42, pp. 495–500, 2006.

[27] E. Tromp and M. Pechenizkiy, "Senticorr: Multilingual sentiment analysis of personal correspondence," *Data Mining Workshops, International Conference on*, vol. 0, pp. 1247–1250, 2011.

[28] E. F. Codd, S. B. Codd, and C. T. Salley, "Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate," *Codd and Date*, vol. 32, pp. 3–5, 1993.

[29] T. O. Council, "OLAP and OLAP Server Definitions," 1995.

[30] S. Chaudhuri and U. Dayal, "An overview of data warehousing and OLAP technology," *SIGMOD Rec.*, vol. 26, pp. 65–74, Mar. 1997.

[31] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, M. B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Searching and Mining Trillions of Time Series Subsequences under Dynamic Time Warping," in *Proceedings of ACM SIGKDD 2012*, SIGKDD 2012, 2012.

[32] M. Muller, "Information Retrieval for Music and Motion," pp. 69–84, 2007.

[33] J. Cacioppo, L. Tassinary, and G. Berntson, *Handbook of Psychophysiology*. Cambridge University Press, 2000.

[34] H. Selye, "Stress and The General Adaptation Syndrome," *The British Medical Journal*, vol. 1, June 1950.

[35] R. Lazarus and S. Folkman, *Stress, Appraisal, and Coping*. Springer Series, Springer Publishing Company, 1984.

[36] B. Womack and J. Hansen, "N-channel hidden markov models for combined stressed speech classification and recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 7, pp. 668 –677, nov 1999.

[37] P. Rajasekaran, G. Doddington, and J. Picone, "Recognition of speech under stress and in noise," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86.*, vol. 11, pp. 733 – 736, apr 1986.

[38] K. R. Scherer, *Voice, stress, and emotion*, pp. 159–181. New York: Plenum, m.h. appley, & r. trumbull, (eds.) ed., 1986.

[39] J. H. Hansen and S. Patil, "Speaker Classification I," ch. Speech Under Stress: Analysis, Modeling and Recognition, pp. 108–137, Berlin, Heidelberg: Springer-Verlag, 2007.

[40] D. Datcu and L. Rothkrantz, "The recognition of emotions from speech using GentleBoost classifier - A comparison approach," 2006.

[41] J. Cichosz and K. Slot, "Emotion recognition in speech signal using emotion-extracting binary decision trees," 2007.

[42] T. L. Nwe, S. W. Foo, and L. C. D. Silva, "Speech emotion recognition using hidden markov models," *Speech Communication*, vol. 41, no. 4, pp. 603 – 623, 2003.

[43] S. Hewlett, "Emotion detection from speech."

[44] S. A. Patil and J. H. L. Hansen, "Detection of speech under physical stress: model development, sensor selection, and feature fusion," in *INTERSPEECH*, pp. 817–820, 2008.

[45] T. Sobol-Shikler, *Analysis of affective expression in speech*. PhD thesis, University of Cambridge, Computer Laboratory, 2008.

[46] J. Z. Zhang, N. Mbitiru, P. C. Tay, and R. D. Adams, "Analysis of stress in speech using adaptive empirical mode decomposition," in *Proceedings of the 43rd Asilomar conference on Signals, systems and computers*, Asilomar'09, (Piscataway, NJ, USA), pp. 361–365, IEEE Press, 2009.

[47] J. Zhai, A. Barreto, C. Chin, and C. Li, "Realization of stress detection using psychophysiological signals for improvement of human-computer interactions," in *SoutheastCon, 2005. Proceedings. IEEE*, pp. 415 – 420, april 2005.

[48] J. Zhai and A. Barreto, "Stress Detection in Computer Users Based on Digital Signal Processing of Noninvasive Physiological Variables," 2006.

[49] A. B. Barreto and J. Zhai, "Physiologic instrumentation for real-time monitoring of affective state of computer users," *Blood*, vol. 3, p. 6, 2003.

[50] A. Barreto, J. Zhai, N. Rishe, and Y. Gao, "Significance of Pupil Diameter Measurements for the Assessment of Affective State in Computer Users," in *Advances and Innovations in Systems, Computing Sciences and Software Engineering* (K. Elleithy, ed.), pp. 59–64, Springer Netherlands, 2007.

[51] J. Hernandez, R. R. Morris, and R. W. Picard, "Call center stress recognition with person-specific models," in *Proceedings of the 4th international conference on Affective computing and intelligent interaction - Volume Part I*, ACII'11, (Berlin, Heidelberg), pp. 125–134, Springer-Verlag, 2011.

[52] Y. Shi, M. H. Nguyen, P. Blitz, B. French, S. Fisk, F. D. Torre, A. Smailagic, D. P. Siewiorek, M. Absi, E. Ertin, and et al., "Personalized stress detection from physiological measurements," *Analysis*, 2010.

[53] F.-T. Sun, C. Kuo, H.-T. Cheng, S. Buthpitiya, P. Collins, and M. L. Griss, "Activity-aware mental stress detection using physiological sensors," *Technology*, no. 23, pp. 1–20, 2010.

[54] J. Healey and R. Picard, "Detecting stress during real-world driving tasks using physiological sensors," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 6, pp. 156 – 166, june 2005.

[55] A. Benoit, L. Bonnaud, A. Caplier, P. Ngo, L. Lawson, D. G. Trevisan, V. Levacic, C. Mancas, and G. Chanel, "Multimodal focus attention and stress detection and feedback in an augmented driver simulator," *Personal Ubiquitous Comput.*, vol. 13, pp. 33–41, Jan. 2009.

[56] W. Liao, W. Zhang, Z. Zhu, and Q. Ji, "A real-time human stress monitoring system using dynamic bayesian network," in *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, p. 70, june 2005.

[57] S. Begum, M. U. Ahmed, P. Funk, N. Xiong, and B. von Schéele, "Using calibration and fuzzification of cases for improved diagnosis and treatment of stress," in *8th European Conference on Case-based Reasoning workshop proceedings* (M. Minor, ed.), pp. 113–122, September 2006.

[58] T. Banziger and K. Scherer, "The role of intonation in emotional expressions," *Speech Communication*, vol. 46, no. 3-4, pp. 252–267, 2005.

[59] F. Yu, E. Chang, Y. qing Xu, and H. yeung Shum, "Emotion detection from speech to enrich multimedia content," in *Second IEEE Pacific-Rim Conference on Multimedia*, pp. 550–557, 2001.

[60] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," in *IFA Proceedings 17*, pp. 97–110, 1993.

[61] D. Talkin, "A robust algorithm for pitch tracking (rapt)," *Speech coding and synthesis*, vol. 495, pp. 495–518, 1995.

[62] S. B. Davis and P. Mermelstein, "Readings in speech recognition," ch. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, pp. 65–74, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990.

[63] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, "Rasta-plp speech analysis," 1991.

[64] C. Darrow, "The rationale for treating the change in Galvanic Skin Response as a change in conductance.," pp. 31–38, 1964.

[65] W. Boucsein, *Electrodermal Activity*. The Springer series in behavioral psychophysiology and medicine, Springer, 2011.

[66] L. G. Tassinary, "Inferring psychological significance from physiological signals," *American Psychologist*, vol. 45, pp. 16–28, 1990.

[67] J. Han and M. Kamber, "Data mining: Concepts and techniques," The Morgan Kaufmann Series in Data Management Systems, pp. 291–310, Elsevier, 2006.

[68] D. A. Reynolds, "Gaussian mixture models," in *Encyclopedia of Biometrics*, pp. 659–663, 2009.

[69] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, vol. 39, no. 1, pp. 1–38, 1977.

[70] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 72–83, 1995.

[71] J.-S. R. Jang, "Machine learning toolbox," Software available at `http://mirlab.org/jang/matlab/toolbox/machineLearning`.

[72] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[73] C.-w. Hsu, C.-c. Chang, and C.-j. Lin, "A practical guide to support vector classification," *Bioinformatics*, vol. 1, no. 1, pp. 1–16, 2010.

[74] I. Lefter, L. J. M. Rothkrantz, D. A. van Leeuwen, and P. Wiggers, "Automatic stress detection in emergency (telephone) calls," *IJIDSS*, vol. 4, no. 2, pp. 148–168, 2011.

[75] S. Menard, *Applied Logistic Regression Analysis*. No. v. 106;v. 2002 in Quantitative Applications in the Social Sciences, Sage Publications, 2002.

[76] T. E. Nichols and A. P. Holmes, "Nonparametric permutation tests for functional neuroimaging: A primer with examples. human brain mapping," 2001.

[77] J. Carletta, "Assessing agreement on classification tasks: the kappa statistic," *Comput. Linguist.*, vol. 22, pp. 249–254, June 1996.

[78] P. Boersma and D. Weenink, "Praat: doing phonetics by computer [Computer program] Version 5.3.19," 2012. Software available at `http://www.praat.org/`.

[79] A. Krishnamurthy, "High-dimensional clustering with sparse gaussian mixture models," 2011.

[80] S. Puuronen, V. Y. Terziyan, and A. Tsymbal, "A dynamic integration algorithm for an ensemble of classifiers," in *Proceedings of the 11th International Symposium on Foundations of Intelligent Systems*, ISMIS '99, (London, UK, UK), pp. 592–600, Springer-Verlag, 1999.

[81] A. Tsymbal, M. Pechenizkiy, and P. Cunningham, "Diversity in search strategies for ensemble feature selection," *Information Fusion*, vol. 6, no. 1, pp. 83–98, 2005.

[82] J. Bakker, L. Holenderski, R. Kocielnik, M. Pechenizkiy, and N. Sidorova, "Stess@work: from measuring stress to its understanding, prediction and handling with personalized coaching," in *IHI*, pp. 673–678, 2012.

[83] J. A. Taylor, "A personality scale of manifest anxiety," *Journal of Abnormal Psychology*, vol. 48, no. 2, pp. 285–90, 1953.

[84] D. P. Crowne and D. Marlowe, "A new scale of social desirability independent of psychopathology.," *Journal of Consulting Psychology*, vol. 24, no. 4, pp. 349–354, 1960.

[85] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[86] S. wook Kim, S. Park, and W. W. Chu, "An index-based approach for similarity search supporting time warping in large sequence databases," in *In ICDE*, pp. 607–614, 2001.

[87] B.-K. Yi, H. V. Jagadish, and C. Faloutsos, "Efficient retrieval of similar time sequences under time warping," in *Proceedings of the Fourteenth International Conference on Data Engineering*, ICDE '98, (Washington, DC, USA), pp. 201–208, IEEE Computer Society, 1998.

[88] J. Stroop, "Interference in serial verbal reactions," *J. Exp. Psychol*, vol. 18, pp. 643 – 661, 1935.

[89] D. Kahneman and D. Chajczyk, "Tests of the automaticity of reading: Dilution of stroop effects by color-irrelevant stimuli," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 9, no. 4, pp. 497–508, 1993.

[90] P. Renaud and J.-P. Blondin, "The stress of stroop performance: physiological and emotional responses to color-word interference, task pacing, and pacing speed," *International Journal of Psychophysiology*, vol. 27, no. 2, pp. 87 – 97, 1997.

[91] C. Kirschbaum, K. M. Pirke, and D. H. Hellhammer, "The 'Trier Social Stress Test' – a tool for investigating psychobiological stress responses in a laboratory setting," *Neuropsychobiology*, vol. 28, pp. 76–81, 1993.

[92] C. Kirschbaum, O. Diedrich, J. Gehrke, S. Wuest, and D. Hellhammer, "Cortisol and behavior: The 'Trier Mental Challenge Test' (TMCT): First evaluation of a new psychological stress test," *Perspectives and Promises of Clinical Psychology. Applied Clinical Psychology*, pp. 67–78, 1992.

[93] K. Dedovic, R. Renwick, N. K. Mahani, V. Engert, S. J. Lupien, and J. C. Pruessner, "The montreal imaging stress task: using functional imaging to investigate the effects of perceiving and processing psychosocial stress in the human brain," *Journal of Psychiatry & Neuroscience*, vol. 30, no. 5, pp. 319 – 325, 2005.

[94] H. G. Wallbott and K. R. Scherer, "Stress specificities: Differential effects of coping style, gender, and type of stressor on autonomic arousal, facial expression, and subjective feeling," *Journal of Personality and Social Psychology*, vol. 61, pp. 147–156, 1991.

[95] P. Ekman and W. V. Friesen, "Detecting deception from the body or face," *Journal of Personality and Social Psychology*, vol. 29, no. 3, pp. 288–298, 1974.

[96] A. D. S. Sierra, C. Snchez-Avila, J. Guerra-Casanova, and G. B.-D. Pozo, *Real-Time Stress Detection by Means of Physiological Signals*. Recent Application in Biometrics, Jucheng Yang and Norman Poh (Ed.), InTech, 2011.

Appendix A

Technical Detail for Stress Analytics Framework

A.1 Storing Raw Data

Figure A.1 depicts the relational database diagram for storing four different types of raw data: physiological signals, speech, textual data (e.g. Email) and metadata.

A.1.1 Physiological signals

The GSR and skin temperature data were stored in the same table `GSR_device`, owing to the assumption that both were collected by using the same device. The table `GSR_device` can store GSR and skin temperature instances for any sampling frequency. The field time and millisecond in the table are used to indicate the actual date and time the instances were taken. Table A.1 illustrates the instances which were sampled using 2Hz frequency.

subject_id	time	milisecond	gsr	skin_temp
1	2012-04-01 13:01:05	100	1240	1310
1	2012-04-01 13:01:05	400	1250	1330

Table A.1: Two instances of `GSR_device`. The `gsr` and `skin_temp` are referring to the GSR and skin temperature level respectively.

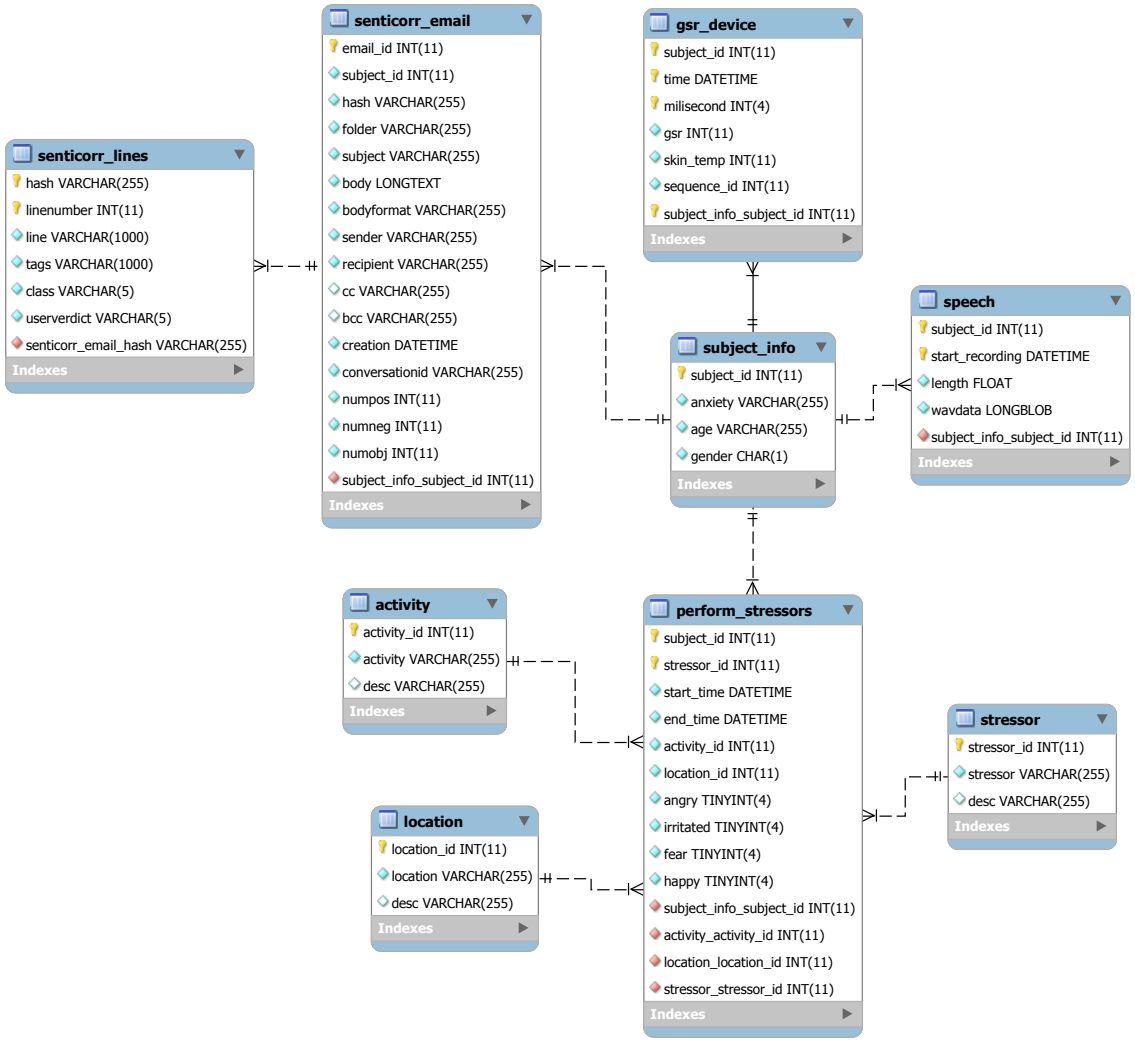


Figure A.1: Database diagram for storing raw data.

A.1.2 Speech

The audio speech was stored in `speech` table as a binary wav file. The binary file, start time, and length of the recording are stored in `wavdata`, `start_recording` and `length` field respectively, as shown in Table A.2. The start time of recording is stored into the database for enabling an alignment with other raw data, and will be explained in the upcoming section. On the other hand, the length of recording is stored for efficiency and speeding up purposes, as it is expensive to query and retrieve the wav file directly.

subject_id	start_recording	length	wavdata
1	2012-04-01 13:01:05	34.82	recording1.wav
1	2012-04-01 21:10:00	90.82	recording2.wav

Table A.2: Two instances of `speech`. The `length` is in second.

A.1.3 Textual data

We used the Senticorr plug-in to obtain the textual data from the email together with its annotation and stored it into our database. The original `senticorr_email` table structure is exactly the same as shown in Figure A.1 without the `subject_id` field. The original structure does not conform with our current database, since our convention required a table to have at least `subject_id` and a date time field. Table `senticorr_email` has already had a date time notion in a `creation` field. Therefore, we showed that it is possible to integrate the Senticorr email with this project just by adding a `subject_id` field into the original table `senticorr_email`.

A.1.4 Metadata

Metadata contains additional descriptive information about the data content, which is depicted in Figure A.1, consists of `subject_info`, `perform_stressors`, `stressor`, `activity` and `location` table.

The table `subject_info` contains information regarding the subject, including age, gender, and the general anxiety level. The anxiety level is a subjective personality test, obtained using a questionnaire, to assess the subject's manifest anxiety scale. The tests, which are commonly used in the experiment, include Manifest Anxiety Scale (MAS) [83] and Social Desirability Scale (SDS) [84]. The methods of these tests are explained more in Appendix B. The `anxiety` field has three options, namely "low anxiety", "high anxiety" and "anxiety deniers". Anxiety deniers closely related to whom, which has the social desirability type of "faking good" and "faking bad".

We stored other metadata that provides additional information, such as where is the location (campus, home, etc.), what is the activity (walking, standing, etc.), what is the task (e.g. driving, teaching, etc.), when is the time and the subjective assessments of the task itself. This is illustrated in Table A.3.

subject_id	stressor_id	start_time	end_time	activity_id	location_id	angry	irritated	fear	happy
1	1	2012-04-20 10:29:00	2012-04-20 11:05:00	1	2	3	4	5	1
2	1	2012-04-20 13:01:00	2012-04-20 13:20:00	1	2	2	1	3	5

Table A.3: Two instances of `perform_stressors` table. The integer number in the field `stressor_id`, `activity_id`, and `location_id` is a key which refers to the `stressor`, `activity` and `location` table respectively. The subjective assessment of the task is stored in the field `angry`, `irritated`, `fear` and `happy`. Each of them may have an integer value ranging from 1 to 5, where 1 denotes the less susceptible and 5 the most susceptible.

A.2 Stress Cube

The stress cube metadata structure was implemented as a star schema. This is depicted in Figure A.2. The `facts_stress` is a fact table, whereas the dimensions are `subject_info`, `date`, `activity`, `senticorr_email`, `location` and `stressor`. The numeric measures are stored in field `by_system`, `by_expert` and `by_user`, which may have an integer value be either 0 (non-stress), 1 (stress), or `null` (no-data). The field `by_system` means the value is calculated automatically from the stress model. The field `by_expert` means that the value is annotated manually by a domain expert by analyzing the raw data. Finally, `by_user` means the user manually assesses the stress level of a certain task. This is illustrated in Table A.4 and A.5.

Mondrian engine necessitates that the schema which defines the multi-dimensional database should be provided in an XML (eXtensible Markup Language) file. The schema should contain a logical model, consisting of cubes, hierarchies, and members, and a mapping of this model onto a physical model. The illustration of this schema is given in Figure A.3.

subject_id	time_id	by_system	by_expert	by_user
1	1	1	1	null
2	2	1	0	0

Table A.4: The illustration of `fact_stress` table. Note that for the sake of presentation, we omit certain fields from the table. `time_id` is a key that points to Table A.5.

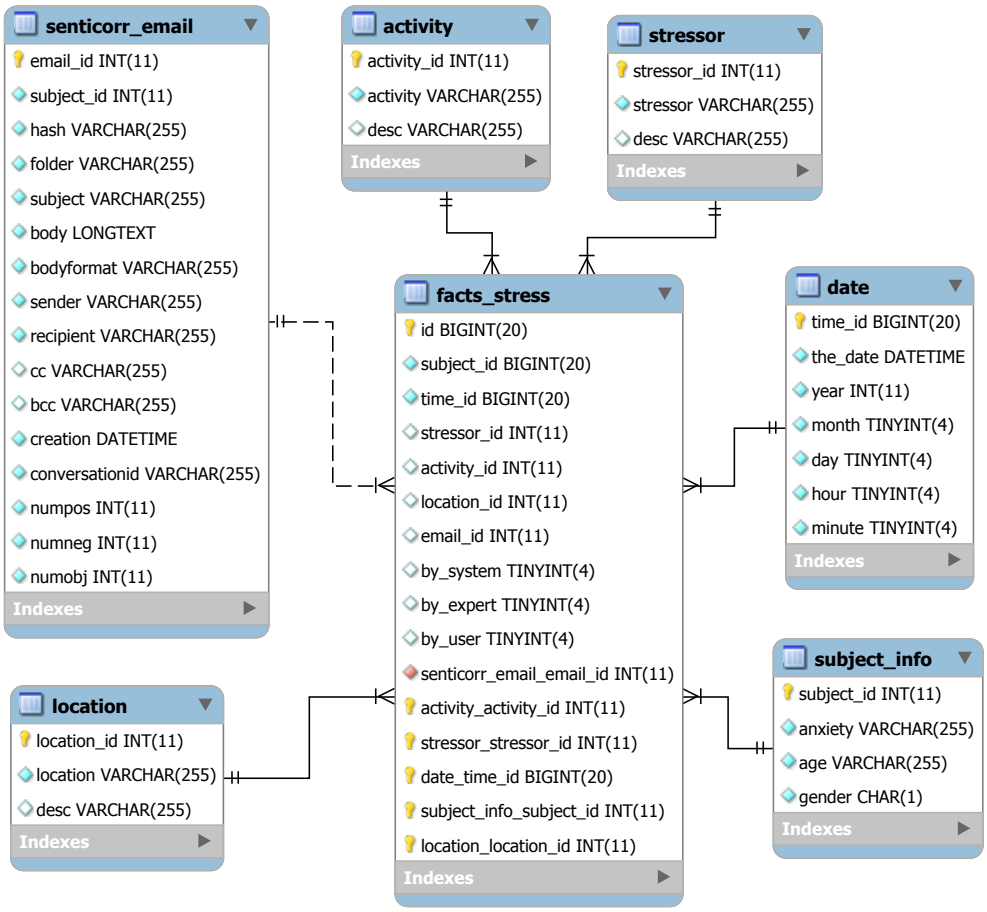


Figure A.2: Stress cube star schema.

time_id	the_date	year	month	day	hour	minute
1	2012-04-21 17:53:00	2012	04	21	17	53
2	2012-04-21 17:54:00	2012	04	21	17	54

Table A.5: Two instances of `date` table. The smallest granularity of time is in minute.

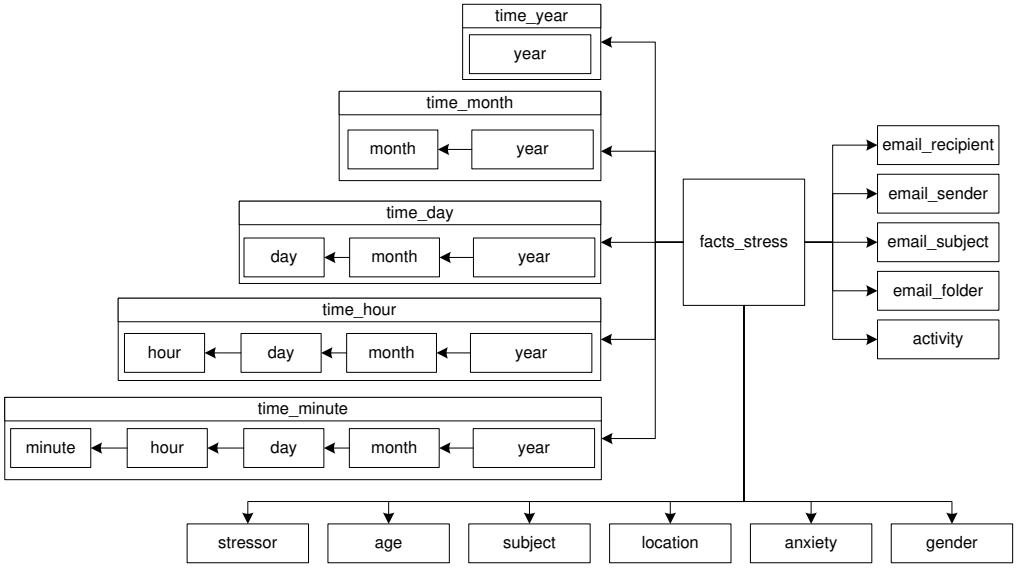


Figure A.3: Mondrian XML schema.

The detail information about XML schema is given as follow:

```
<?xml version="1.0"?>
<Schema name="ACMESchema">
    <Cube name="Stress">
        <Table name="facts_stress"/>
        <Dimension name="anxiety" foreignKey="subject_id">
            <Hierarchy hasAll="true" allMemberName="all" primaryKey="subject_id">
                <Table name="subject_info"/>
                <Level name="anxiety" column="anxiety" uniqueMembers="true"/>
            </Hierarchy>
        </Dimension>
        <Dimension name="gender" foreignKey="subject_id">
            <Hierarchy hasAll="true" allMemberName="all" primaryKey="subject_id">
                <Table name="subject_info"/>
                <Level name="gender" column="gender" uniqueMembers="true"/>
            </Hierarchy>
        </Dimension>
        <Dimension name="age" foreignKey="subject_id">
            <Hierarchy hasAll="true" allMemberName="all" primaryKey="subject_id">
                <Table name="subject_info"/>
                <Level name="age" column="age" uniqueMembers="true"/>
            </Hierarchy>
        </Dimension>
        <Dimension name="stressor" foreignKey="stressor_id">
            <Hierarchy hasAll="true" allMemberName="all" primaryKey="stressor_id">
                <Table name="stressor"/>
                <Level name="stressor" column="stressor" uniqueMembers="true"/>
            </Hierarchy>
        </Dimension>
        <Dimension name="activity" foreignKey="activity_id">
            <Hierarchy hasAll="true" allMemberName="all" primaryKey="activity_id">
                <Table name="activity"/>
                <Level name="activity" column="activity" uniqueMembers="true"/>
            </Hierarchy>
        </Dimension>
        <Dimension name="location" foreignKey="location_id">
            <Hierarchy hasAll="true" allMemberName="all" primaryKey="location_id">
                <Table name="location"/>
                <Level name="location" column="location" uniqueMembers="true"/>
            </Hierarchy>
        </Dimension>
        <Dimension name="subject" foreignKey="subject_id">
            <Hierarchy hasAll="true" allMemberName="all" primaryKey="subject_id">
                <Table name="subject_info"/>
                <Level name="subject" column="subject_id" uniqueMembers="true"/>
            </Hierarchy>
        </Dimension>

      <Dimension name="timeminute" foreignKey="time_id">
         <Hierarchy hasAll="true" allMemberName="all" primaryKey="time_id">
            <Table name="date"/>
            <Level name="year" column="year" uniqueMembers="true"/>
            <Level name="month" column="month" uniqueMembers="false"/>
            <Level name="day" column="day" uniqueMembers="false"/>
            <Level name="hour" column="hour" uniqueMembers="false"/>
            <Level name="minute" column="minute" uniqueMembers="false"/>
```

```
    </Hierarchy>
</Dimension>

<Dimension name="timehour" foreignKey="time_id">
    <Hierarchy hasAll="true" allMemberName="all" primaryKey="time_id">
        <Table name="date"/>
        <Level name="year" column="year" uniqueMembers="true"/>
        <Level name="month" column="month" uniqueMembers="false"/>
        <Level name="day" column="day" uniqueMembers="false"/>
        <Level name="hour" column="hour" uniqueMembers="false"/>
    </Hierarchy>
</Dimension>

<Dimension name="timeday" foreignKey="time_id">
    <Hierarchy hasAll="true" allMemberName="all" primaryKey="time_id">
        <Table name="date"/>
        <Level name="year" column="year" uniqueMembers="true"/>
        <Level name="month" column="month" uniqueMembers="false"/>
        <Level name="day" column="day" uniqueMembers="false"/>
    </Hierarchy>
</Dimension>

<Dimension name="timemonth" foreignKey="time_id">
    <Hierarchy hasAll="true" allMemberName="all" primaryKey="time_id">
        <Table name="date"/>
        <Level name="year" column="year" uniqueMembers="true"/>
        <Level name="month" column="month" uniqueMembers="false"/>
    </Hierarchy>
</Dimension>

<Dimension name="timeyear" foreignKey="time_id">
    <Hierarchy hasAll="true" allMemberName="all" primaryKey="time_id">
        <Table name="date"/>
        <Level name="year" column="year" uniqueMembers="true"/>
    </Hierarchy>
</Dimension>

<Dimension name="emailfolder" foreignKey="email_id">
    <Hierarchy hasAll="true" allMemberName="all" primaryKey="email_id">
        <Table name="senticor_email"/>
        <Level name="folder" column="folder" uniqueMembers="true"/>
    </Hierarchy>
</Dimension>

<Dimension name="emailsubject" foreignKey="email_id">
    <Hierarchy hasAll="true" allMemberName="all" primaryKey="email_id">
        <Table name="senticor_email"/>
        <Level name="subject" column="subject" uniqueMembers="true"/>
    </Hierarchy>
</Dimension>

<Dimension name="emailsender" foreignKey="email_id">
    <Hierarchy hasAll="true" allMemberName="all" primaryKey="email_id">
        <Table name="senticor_email"/>
        <Level name="sender" column="sender" uniqueMembers="true"/>
    </Hierarchy>
```

```
        </Dimension>

        <Dimension name="emailrecipient" foreignKey="email_id">
            <Hierarchy hasAll="true" allMemberName="all" primaryKey="email_id">
                <Table name="senticor_email"/>
                <Level name="recipient" column="recipient" uniqueMembers="true"/>
            </Hierarchy>
        </Dimension>

        <Measure name="by_system" column="by_system" aggregator="sum" formatString="
            Standard"/>
        <Measure name="by_expert" column="by_expert" aggregator="sum" formatString="
            Standard"/>
        <Measure name="by_user" column="by_user" aggregator="sum" formatString="Standard"
            />
    </Cube>
</Schema>
```

A.3 Shape-Based Query-by-Example

The definition of time series is as follows. A time series $T = t_1, ..., t_m$ is an ordered set of m real-valued variables. Next, we introduce a definition of subsequence: Given a time series T with length m, the subsequence C of T is a shorter sequence of length $n < m$ of contiguous position from T. Formally, $C = t_p, ..., t_{p+n-1}$, for $1 \leq p \leq m - n + 1$.

Therefore, we formulate the problem as: given a subsequence query C, we wish to find the most similar (1-Nearest Neighbor) shape-based subsequence R from T, where $|R| = |C|$.

A.3.1 Distance Metric

The distance metrics, which are commonly used for measuring similarity between two time series, are Euclidean distance and Dynamic Time Warping.

Euclidean Distance

Euclidean distance is the most common and popular distance measure in data mining. Let Q and C be two time series with an equal length $|Q| = |C| = n$. The Euclidean distance between Q and C is defined as:

$$ED(Q,C) = \sqrt{\sum_{1}^{n}(q_i - c_i)^2}$$

Dynamic Time Warping

Dynamic Time Warping [32] is a method to measure similarity between two sequences, which may vary in time or speed. For instance, similarities in speech patterns would be detected, even if in one recording, the person was speaking fast and if in another, he or she was speaking slowly. This method was originally developed for automatic speech recognition, to cope with different speaking speeds. These days, DTW has been applied for video, audio, and graphics as well. Indeed, any data which could be represented as a linear representation can be analyzed using DTW.

The algorithm finds the best warping path by creating a 2-dimensional matrix n-by-m. Each element (i-th,j-th) contains the distance d between element q_i and c_j: $(q_i - c_j)^2$. The best alignment between two sequences is satisfied by using the following Dynamic Programming equation:

$$dtw(i,j) = d(q[i], c[j]) + min(dtw(i-1,j), dtw(i,j-1), dtw(i-1,j-1))$$

where dtw is the global distance up to (i,j) and $d(i,j)$ is the squared Euclidean Distance between two points. Figure A.4 illustrates the warping path found by Dynamic Time Warping.

The DTW algorithm has a complexity of $O(nr)$ time, where n is the maximum length between two time series and r is the size of warping window (e.g. Sakoe-Chiba band and Itakura Parallelogram [85]). The basic DTW algorithm is expensive to compute, therefore, in order to speed up the DTW computation, one usually used several optimizations such as using lower bounds. Lower bounds are used first to prune sequences that could not possibly match before the actual sequences are compared. This lower bound apparently should be fast to compute and almost linear at time complexity. Several lower bounds which are usually used include Kim[86], Yi [87], and Keogh[8] lower bound.

A.3.2 UCR-Suite Algorithm

The UCR-Suite algorithm is the current state-of-the-art for searching subsequence time series under DTW. The algorithm exploited several optimizations such as early abandoning Z-normalization, reordering early abandoning, reversing the query or data role in Keogh's lower bound and cascading lower bound for speeding up the computational time. We refer to [31] for a detailed explanation about this algorithm.

A.3.3 Integration Into Stress Analytics

The integration of the UCR-Suite algorithm to stress analytics is straight-forward. Stress analytics uses the similarity search functionality for finding a similar GSR (or skin temperature) shape in the database. The user first selects a particular query subsequence, and then the system determines the collection of candidate sequences, for which the query subsequence will be compared to. Candidate sequences are all contiguous sequences in the database excluding the query subsequence. More formally, let $S = \{S_1, S_2, S_3, ..., S_n\}$ be the collection of sequences, and $S_k(j:m)$ be the query subsequence which the user selects, where $1 \leq k \leq n$, and $1 \leq j < m \leq |S_k|$. Then the candidate sequence C is determined as a col-

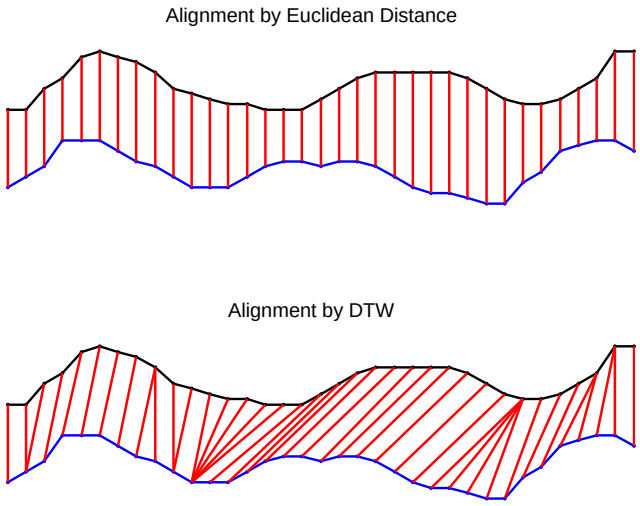


Figure A.4: Euclidean distance and Dynamic Time Warping (DTW) alignment.

lection of $(S - S_k) \cup S_k(1 : j - 1) \cup S_k(m + 1 : |S_k|)$. Afterwards, the UCR-suite algorithm is used to find the most similar subsequence (1-Nearest Neighbor) R from C, where $|R| = m - j + 1$.

A.4 Stress Analytics Visualization

The interactive overall diagram of stress analytics is depicted in Figure A.5. In a nutshell, the system provides three different functionality, interactive OLAP exploration, showing evidence (e.g. stress-related physiological signals) or stress-related events (e.g. email), and search functionality (e.g. shape-based query-by-example). The data which have been processed by OLAP is visualized as a two-dimensional graph. The user may interact and explore the cube by using a graphical user interface. Moreover, several filtering options are included for more fine-grained analysis.

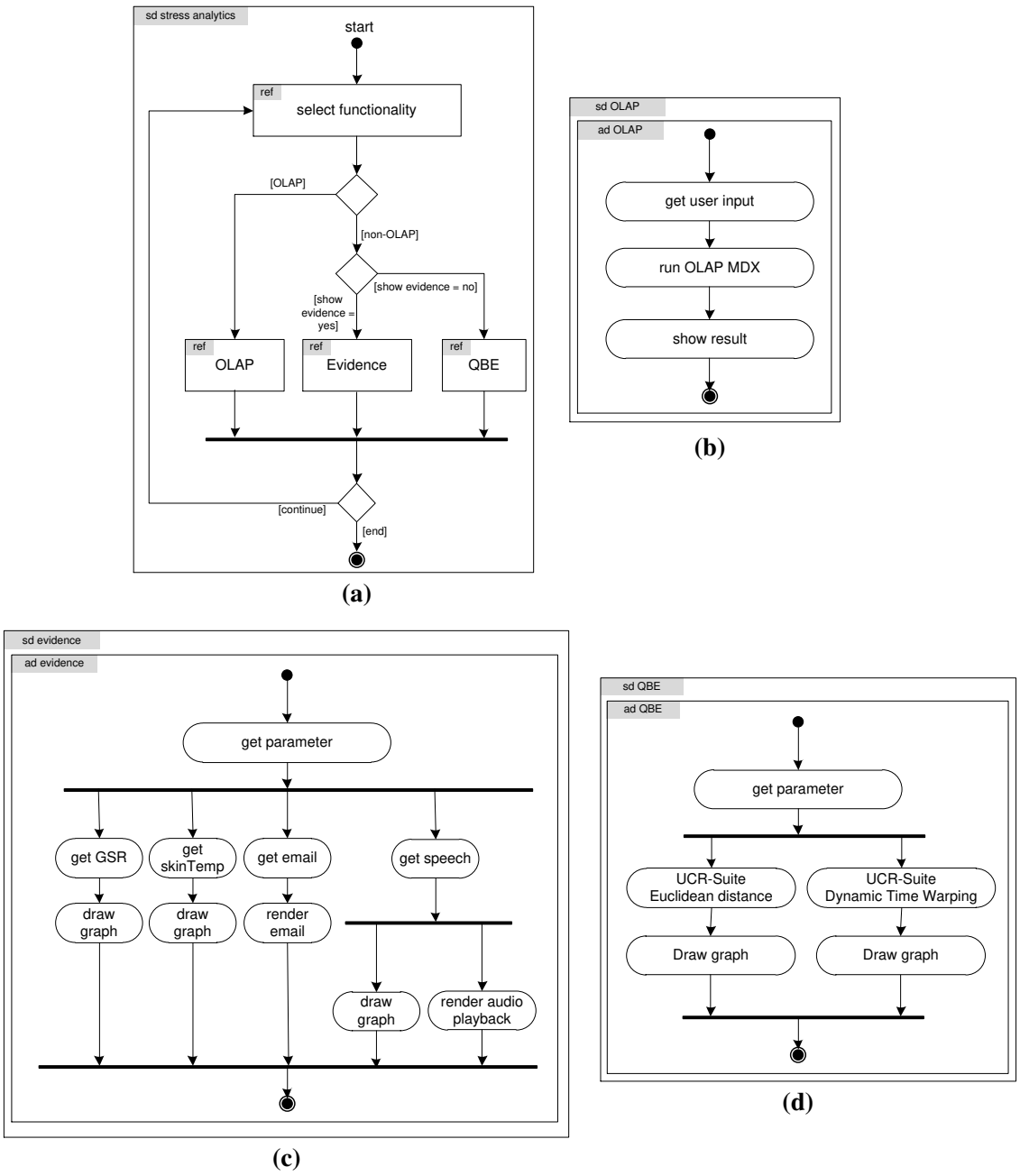


Figure A.5: Interactive overview diagram of stress analytics. (a) Main diagram. (b) OLAP diagram. (c) Evidence diagram. (d) Query-by-Example (QBE) diagram.

Appendix B

Psychological Stress Elicitation Experiment

In general, stress can be both physical and psychological. Physical stress may arise as a result of the contact of our body with the inconvenient physical environment (e.g. noisy environment, vibration, sickness, cold temperature, etc.) which will induce stress to the individual. On the other hand, psychological stress may occur when the individual cannot cope with special challenge or task (e.g. workload which is beyond one's capability, job's pressure, etc.). In real life, the physical stress is often present together with psychological stress. For instance, one most likely cannot do things which usually he could when he is in a sick condition, therefore, this may lead to frustration and stress. In this chapter, we discussed in detail the psychological stress elicitation experiment in laboratory settings.

The remainder of this appendix is organized as follows. In Section B.1, we present our motivation and goal for conducting this psychological experiment. Section B.2 briefly describes related works. Section B.3 explains in detail the data which we collected during the experiment. The method and protocol of the experiment are discussed in Section B.4.

B.1 Motivation, Goals and Hypothesis

We conduct this experiment, mainly owing to the unavailability of a free stress dataset which incorporates both speech and GSR. This labeled dataset will be used to create a stress model which can differentiate between two different stress levels, which was discussed in Chapter 3. Another reason is that we would like to investigate several claims about GSR and speech characteristics under stress and non-stress conditions as demonstrated in literatures.

The hypotheses that we made in this experiment are as follows:

- The number of GSR startle responses during the relaxation period should be lower, compared to the light or heavy workload, as the GSR level should increase during the stress period which, in turn, increases the number of startle responses.
- The mean of skin conductance during the heavy workload should be higher than during the light workload. We assume that the heavy workload will induce more stress to the subjects.
- When the subject experiences stress, his respiration rate increases. This will increase Subglotall pressure during speech, which, in turn, increases the fundamental frequency F_0 (pitch). Hence, we expect that the mean of F_0 will increase under a stressful (heavy workload) condition.

B.2 Related Works

There have been numerous methods proposed for stress elicitation experiments within literatures. In 1935, Stroop proposed a psychological test called The Stroop Color-Word Interference Test [88]. The test itself demands that the color of a word designating a different color to be named. Stroop found that it took a longer time to read the words printed in a different color than name the same words printed in black. This task, widely known as Stroop effect or Stroop task, is widely used as a tool to understand our cognitive-perceptual process [89]. In a different area of research, Stroop effect has been widely utilized as a cognitive stressor able to induce a heightened level of physiological arousal. The reliability of Stroop task to induce a certain level of stress has been demonstrated in a lot of studies [90][47].

The 'Trier Social Stress Test' (TSST) [91] is the standardized test for the induction of moderate psychological stress in a laboratory setting. This test consists of certain protocols, which have to be performed by the subject of the experiment. In a nutshell, first the subject is asked to take the role as a job applicant, and they should introduce themselves to three managers in a free speech of five-minute duration. This task corresponds to the public speaking stressor. Following this task, the subject is instructed to serially subtract number 13 from 1,022 as fast and as accurate as possible. On every failure, the subject has to restart at 1,022. This task corresponds to the mental arithmetic stressor. In this study, this protocol has been found to induce considerable changes in the concentration of adrenocorticotropin (ACTH), cortisol, prolactin and heart-rate in six independent studies.

Kirschbaum et al. [92] have demonstrated that the cortisol levels of the subject increased when they performed The 'Trier Mental Challenge Test' (TMCT) in a group setting. This test demands the subject to solve mental arithmetic tasks under time pressure. The arithmetic task is divided into different categories, ranging from the simplest (e.g. $1+1=?$) to the toughest (e.g. $9+10\times2-18/2=?$). After the subject finishes each session of the task, they have to report their outcome in front of the group. The other test derived from TMCT is The 'Montreal Imaging Stress Task' (MIST) [93], which consists of computerized mental arithmetic challenges combined with social evaluative threat components. In this condition, the difficulty and time limit of the tasks are manipulated to be beyond the subject's mental capacity. In addition, when the subject is performing the mental arithmetic tasks, the average and expected performance information is displayed on the monitor. Upon completion of each task, the performance evaluation is given to further increase the social evaluative threat of the situation.

Other procedures which are known to be able to elicit stress include solving the Raven Standard (and Advanced) Progressive Matrices test [94], watching the slides that contain extreme emotional conditions [94], asking the subjects to lie about their feeling after watching an unpleasant surgery movie [95], playing a video game [55], real-world driving task [54], and a hyperventilation task, which consists of deep and fast breaths every three seconds [96].

B.3 Data Collections

B.3.1 Objective Measurements

During the experiment, several signals were recorded including speech, facial expression, and skin conductance. We used an ordinary speech recorder to record the subject's voice when he or she was performing tasks. The speech was sampled at a sampling rate of 44,100 Hz by using two channels. Facial expression was recorded using Handycam Camcorders with High Definition (HD) resolution at $1,440 \times 1,080$ pixels.

We used a homemade GSR sensor to measure the changes in skin conductance. This was carried out by using the LEGO Mindstorms NXT [1] and an RCX wire connector sensor. The LEGO Mindstorms NXT and RCX wire connector are shown in Figure B.1. Stress causes the activation of sweat glands, which, in turn, affects the amount of sweat produced. The changes of sweat affect the skin conductance. The more relaxed the individual, the dryer the skin will be, hence the skin conductance is lower. In contrast, when an individual is in stress, the sweat in the hand increases, which, in turn, increases the skin conductance.

Figure B.1: (a) LEGO NXT Mindstorms. (b) RCX wire connector sensor.

[1] http://mindstorms.lego.com/en-us/Default.aspx

The RXC connector is just a plain Analog to Digital Converter (ADC) sensor, which converts the analog reading to digital raw values in the range of 0 to 1,023. We cut one end of the RCX connector which connects to the brick and soldered it to a copper. On top of this copper, we attached aluminum foil and glued it to a stripped Velcro for a wrapper around the finger. This homemade device is known as dry type electrodes, while the professional one would utilize a conductive paste (gel) for more stable and repeatable readings. The reading returns 0 if the two wires are separated (e.g. not touch each other) and returns 1,023 if the two wires are connected directly. Figure B.2 illustrates the modified RCX connector, and the homemade GSR device used in the experiment.

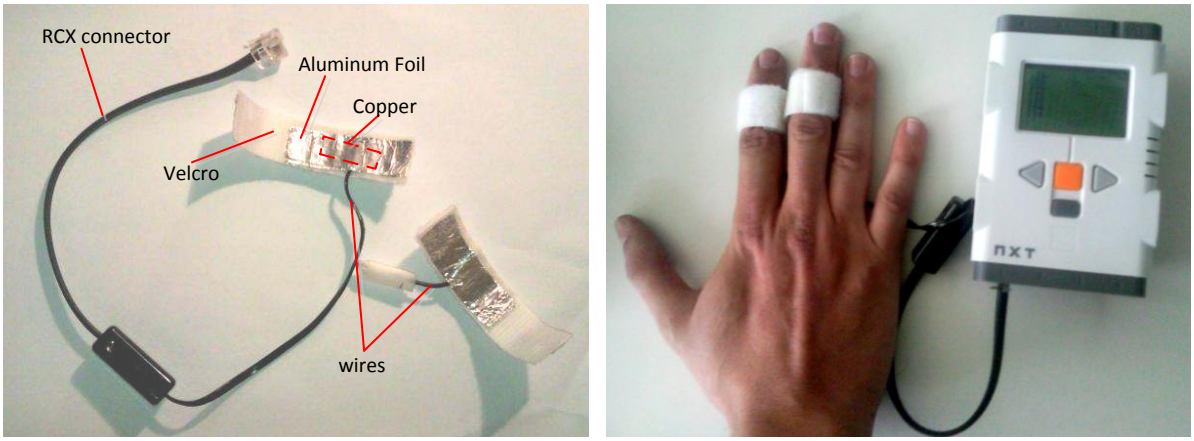


Figure B.2: (a) The modified RCX connector. (b) Homemade GSR device.

The measurements were sampled with 2Hz frequency by using LEGO NXT. Next, the raw value was sent in real time to the computer by means of Bluetooth's connection. We used LEJOS - Java for LEGO Mindstorms [2] open source framework for handling this connection.

B.3.2 Subjective Measurements

We collected not only objective measurement's data but also subjective measurements by means of questionnaires. The subject's anxiety characteristics could be grouped into four different categories based on two-dimensional scales [38]. The

[2] http://lejos.sourceforge.net/

scales were obtained by using two questionnaires, including Taylor Manifest Anxiety Scale (MAS) [83] and Crowne-Marlowe Social Desirability Scale (SDS) [84]. Using a certain cut-off threshold, the subjects can be grouped into four extremes [38]: anxiety-deniers (low MAS, high SDS), high-anxiety (high MAS, low SDS), defensive high-anxiety (high MAS, high SDS), and low-anxiety (low MAS, low SDS). These scales were commonly used for participant selections before the experiment began [94].

Upon completion of each task, the subjects were asked to complete a free scale questionnaire assessing their emotional feeling. The questionnaire ask the subjects to rate from number "1" (less susceptible) to "5" (most susceptible) for the following emotions: angry, irritated, happy and satisfied. All results of the subjective measurements will not be utilized for building a stress model, but will be employed as an additional annotation to enrich our stress analytics system.

B.4 Experiment Methods

B.4.1 Locations and Subjects

The experiment was conducted in room 7.86, HG main building at Technical University Eindhoven (TU/e). 10 graduate students (8 males, 2 females) from the department of Mathematics and Computer Science participated in this study. The mean age, Body Mass Index (BMI), MAS, and SDS of the subject are 26.2 ± 2.6, 22.9 ± 2.7, 18.3 ± 7.4, and 19.2 ± 5.2 respectively.

B.4.2 Control Settings

There are five parameters, which were controlled during the experiments. First, the room temperature was made constant by means of an air conditioner. Second, no type of physical stressor (e.g. no noisy environment) was applied to the subject. Third, the subject performed all tasks in the standing position. Fourth, the GSR measurement was collected from the right hand, second phalanx of index and middle fingers. Fifth, the order of the task was made random in each session.

B.4.3 Procedure

The stress experiment lasted for approximately one hour and consisted of three sessions, including baseline, light workload and heavy workload. The overall timeline diagram of the experiment is illustrated in Figure B.3. At the time 0 minute, the subject filled in a personal questionnaire about age, body weight, height, Taylor Manifest Anxiety Scale (MAS) and Social Desirability Scale (SDS) for roughly 10 minutes. At the time +10 minutes, the subject was seated in the room after having had a GSR sensor attached, turning on the speech recorder and positioning the video camera to allow a close-up recording of the face. The subject was asked to relax while watching a movie about nature clips (e.g. mountain, forest, beach, etc.) accompanied by relaxation instrumental music in the background. The movie itself was projected to the screen straight in front of the subject. There were no other persons in the room, except the subject and the operator. The operator was sitting at a table behind the subject and operated the entire apparatus during the whole experiment.

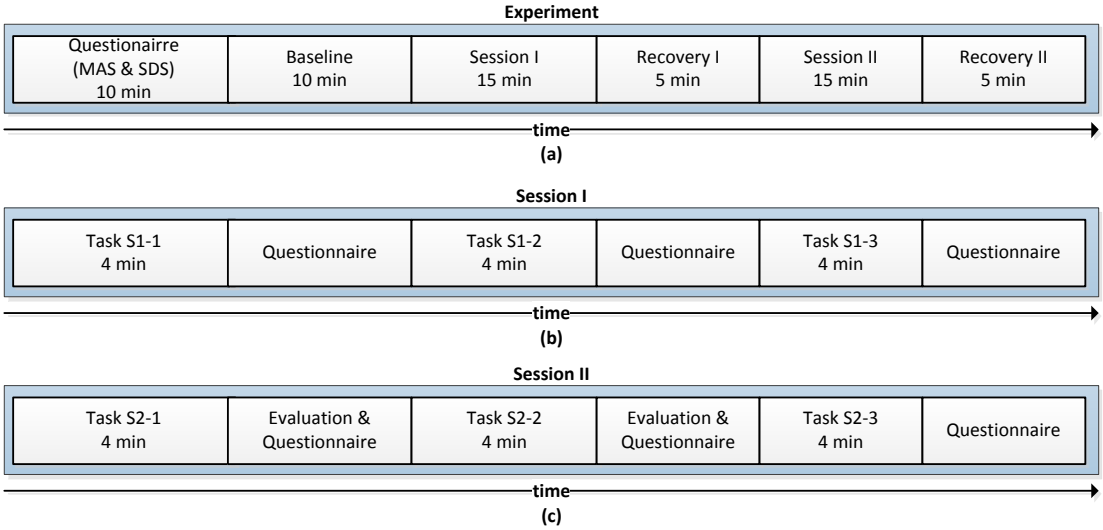


Figure B.3: (a) The whole experiment timeline. (b) Session I timeline. (c) Session II timeline.

After completing the baseline session, the subject at time +20 minutes performed three different tasks in the first session. The first session corresponds to the light

workload, where the subject performed easy tasks without time limitation, pressure, social threat and comparisons with other individuals. The order of tasks in this session was made random to minimize the influence of the order itself on the stress level. In this session, the subject performed all tasks in the standing position and answered each trial verbally. There were no other persons in the room during this session, except the subject and the operator. Upon completion of each task, the subject gave a subjective evaluation by rating the questionnaire assessing their personal emotions. The first session timeline is illustrated in Figure B.3 (b).

The three tasks in the first session consist of a Stroop-Word congruent color test, an easy mental arithmetic test, and an easy mental subtraction test. The reason behind using three different kinds of stressors is two-fold. The first is the necessities to collect sufficient instances with limited participants. The second is to avoid using the same stressor so as to prevent the habituation effect. An individual who is carrying out the same task, even after an undefined period of time, would be prepared to face the task, and the response of his (or her) psychological signals will not be certainly the same [96].

The Stroop-Word congruent color test lasted for approximately four minutes. The subject was instructed to verbally name the font color of the given words presented on the screen. The word itself is designating a color name. The word's designation and its font color always match. Therefore, this situation corresponds to the spontaneous action, and we expect no stress is evoked in this task. In total, there are five color names which were used, including red, green, yellow, blue and white. Each trial lasted for two seconds. In case the subject cannot produce a decision within two seconds, the screen automatically changed to the next trial. Figure B.4 depicts the Stroop-Word congruent color test.

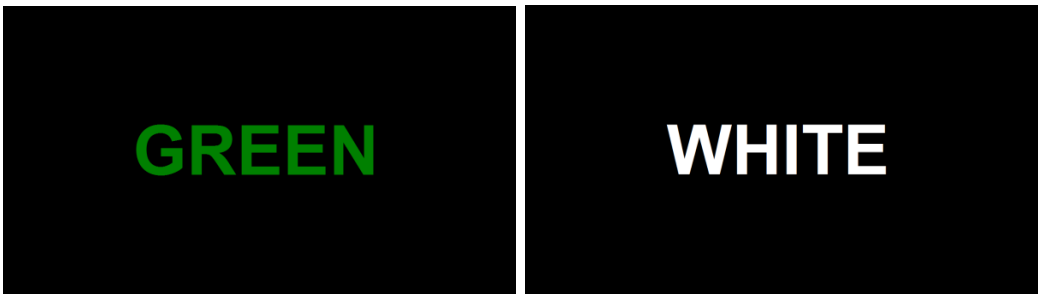


Figure B.4: Stroop-Word congruent color test.

The easy mental arithmetic test lasted for approximately 3.5 minutes. The subject was instructed to give an answer involving a simple addition and subtraction with two to three numbers. There was no time limitation for answering the question, and the correct (or incorrect) feedback was displayed on the screen. This is illustrated in Figure B.5.

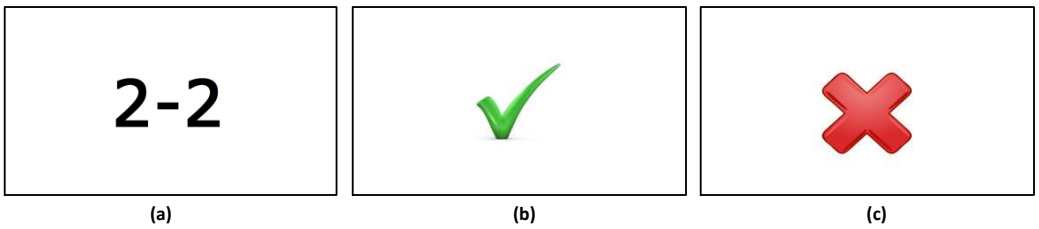


Figure B.5: Easy mental arithmetic. (a) The question. (b) Right answer. (c) Wrong answer.

In the easy mental subtraction test, the subject was instructed to serially subtract number 1 from 300. This test lasted for four minutes. There was no imposition to do the task as quickly as possible and there was no intervention in case the subject made a mistake.

It has been known that GSR has a characteristic to react quickly to an event (stressor) but has a very slow decreasing response to go back to the previous baseline [55]. Hence, based on this priori information, the subject at the time +35 minutes was asked to sit in the relaxation chair for watching a natural clips movie with instrumental relaxation music. This relaxation lasted for approximately five minutes.

The second session consisted of three different tasks, including a Stroop-Word incongruent color test, a hard mental arithmetic test, and a hard mental subtraction test. The order of these tasks was made random and different from the first session to minimize the habituation effect. The second session timeline is depicted in Figure B.3 (c). In essence, the tasks in the second session correspond to the heavy workload (stressful situation) in which the subject was instructed to do tasks beyond his (or her) ability, with imposition of time limitation, pressure, social evaluative threat and comparison with other populations. The subject performed all tasks in this session in the standing position and answered each trial verbally. The setting in the room was the same as the first session, except, there was a presence of one evaluator. The evaluator was sitting at a table in front of

the subject. Furthermore, the subject was informed that his (or her) performance, articulation, poise, voice frequency and facial expression would be evaluated by the evaluator. There were two evaluations given in this session. The first one was given after the subject completed the first task. The evaluator informed the subject that his (or her) performance was not as good as expected and beyond the average population. Furthermore, the evaluator imposed a social evaluative threat by informing the subject that the experiment will fail if his (or her) performance did not improve in the next task. The other evaluation was given after the subject completed the second task. The evaluator gave a negative evaluation in this phase by saying that the performance was still not improved. A question related to the individual personality was given, such as "You look tired today. Did you have a sleep problem last night?"

The Stroop-Word incongruent color test lasted for 4 minutes. The subject was instructed to name the font color of the word verbally, in which the word's designation and the font color were made to be mismatched. Five color names were used, including red, green, blue, yellow and white. Each trial lasted for 1.3 seconds. In case the subject could not produce a decision within a time limit, the screen was automatically changed to the next trial. Figure B.6 illustrates this idea.

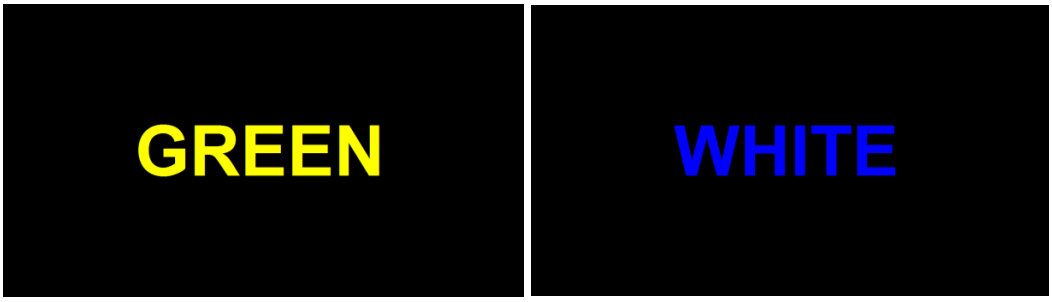


Figure B.6: Stroop-Word incongruent color test.

The hard mental arithmetic test consisted of multiple addition, subtraction, multiplication and division with precedence. The result of the arithmetic is always in an integer form within a range of 0 to 10. This task lasted for approximately four minutes and consisted of several trials (questions). Each trial only lasted for five seconds and the timer was shown at the top right of the screen. In case the subject could not produce an answer within a time limit, he (or she) was asked to guess the

number from 0 to 10. Each correct and incorrect answer increased and decreased the subject's score respectively. The test itself was designed to be adaptive in such a way that the subject can only solve 50 to 60 percent of the questions correctly regardless of their ability. This was accomplished by preparing three groups of questions, including easy, medium and hard. The easy level involved arithmetic with three to four numbers. The medium level involved intricate arithmetic with four to six numbers. The hard level involved an extreme arithmetic up to nine numbers with elaborate precedence and negative numbers. The questions at this level are hardly solvable within a five-second time limit. The system showed a question depending on the subject's score (e.g. if the subject's score drops below a certain threshold, then present an easy level question). The subject's score and the population score were shown as two different bars on top of the screen. The population score was a fictitious score and always made greater than the subject's score to introduce a social evaluative threat and the notion of self's inferior. Figure B.7 illustrates the hard arithmetic test.

In the hard mental subtraction test, the subject was instructed to serially subtract number 13 from 1,010 as fast and as accurately as possible. On every failure, the evaluator interrupted and instructed the subject to restart the calculation from 1,010 by saying, "Stop please! Start again from 1,010!" This task lasted for four minutes.

At the time +40 minutes, after the subject completed the first recovery, the operator called an evaluator, who immediately entered the room and sat at a table in front of the subject. Afterwards, the subject completed three different tasks in the second session for approximately 15 minutes. At the time +55 minutes, the evaluator left the room, and the subject was seated in a relaxation chair for watching natural clip movies with instrumental relaxation music in the background. This relaxation lasted for approximately five minutes. At the time +60 minutes, the operator called back the evaluator, and the subject was debriefed about the real purpose of the experiment.

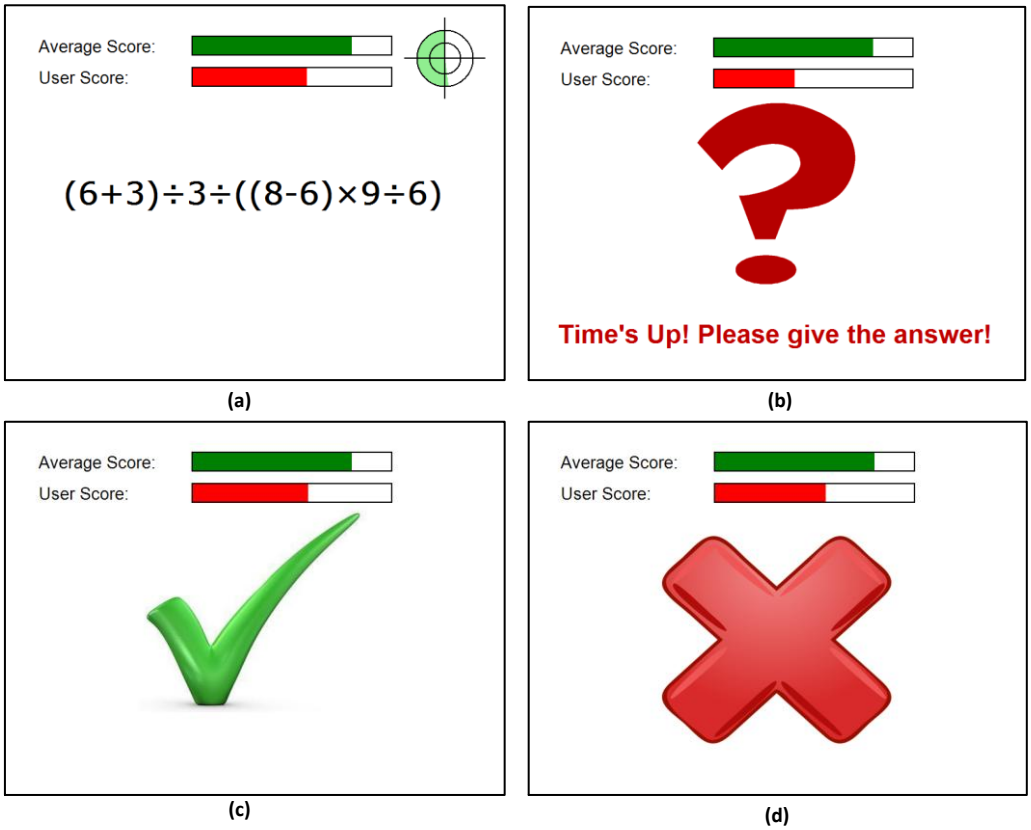


Figure B.7: Hard mental arithmetic test. (a) The question. Top right: Five seconds countdown timer. (b) Time limit exceeded. (c) Correct answer. (d) Incorrect answer. A loud wrong answer buzz sound is played.

Appendix C

Evaluation Detail for Stress Detection

This appendix presents an evaluation detail for Chapter 4.

C.1 Stress Model using GSR features

All results in this section were obtained using 10-times-10-folds cross-validation.

1. **Recovery vs Workloads (light & heavy)**

	Accuracy	Precision	Recall
K-Means	46.12 ± 2.25	35.47 ± 1.62	87.16 ± 2.19
GMM	70.51 ± 0.49	50.99 ± 1.16	72.61 ± 1.52
SVM	79.66 ± 0.77	87.75 ± 0.89	82.41 ± 0.73
Decision Tree	73.45 ± 1.27	81.28 ± 1.01	80.67 ± 1.22

2. **Recovery vs Heavy Workload**

	Accuracy	Precision	Recall
K-Means	55.54 ± 2.64	51.07 ± 2.38	90.27 ± 3.26
GMM	74.90 ± 0.79	72.29 ± 1.50	72.64 ± 1.43
SVM	80.72 ± 0.61	85.29 ± 1.06	78.20 ± 1.19
Decision Tree	77.81 ± 1.31	77.18 ± 2.36	77.18 ± 2.36

3. **Light vs Heavy Workload**

	Accuracy	**Precision**	**Recall**
K-Means	53.21 ± 1.00	50.53 ± 0.68	50.53 ± 0.68
GMM	66.82 ± 0.46	65.59 ± 0.95	66.59 ± 1.20
SVM	70.60 ± 1.10	72.20 ± 1.87	72.02 ± 2.26
Decision Tree	62.52 ± 1.79	67.12 ± 3.67	63.58 ± 1.96

C.2 Stress Model using Speech features

All results in this section were obtained using 10-times-10-folds cross-validation.

1. **Pitch feature**

	Accuracy	**Precision**	**Recall**
K-Means	49.65 ± 2.28	–	57.21 ± 7.18
GMM	58.82 ± 1.46	58.50 ± 2.81	50.71 ± 2.24
SVM	62.08 ± 1.57	65.04 ± 1.93	61.80 ± 3.23
Decision Tree	55.60 ± 2.75	59.74 ± 3.59	57.11 ± 3.03

2. **MFCC feature**

	Accuracy	**Precision**	**Recall**
K-Means	55.39 ± 1.92	–	29.78 ± 6.04
GMM	56.78 ± 1.76	53.90 ± 1.62	63.60 ± 3.68
SVM	92.39 ± 0.58	92.09 ± 1.09	93.79 ± 0.84
Decision Tree	68.86 ± 3.07	70.95 ± 4.51	70.31 ± 2.91

3. **MFCC-Pitch feature**

	Accuracy	Precision	Recall
K-Means	49.17 ± 2.34	–	52.12 ± 4.56
GMM	59.08 ± 0.94	57.90 ± 1.35	51.70 ± 2.15
SVM	92.56 ± 1.63	91.45 ± 1.09	94.57 ± 2.47
Decision Tree	70.69 ± 1.33	73.73 ± 1.62	71.71 ± 2.19

4. **RASTA PLP feature**

	Accuracy	Precision	Recall
K-Means	50.60 ± 0.42	50.60 ± 0.42	99.90 ± 0.31
GMM	52.30 ± 2.78	49.50 ± 2.55	62.49 ± 5.71
SVM	91.69 ± 0.94	92.11 ± 1.61	92.21 ± 1.27
Decision Tree	71.47 ± 2.97	73.84 ± 2.88	73.22 ± 2.59

C.3 Stress Model using fusion of GSR and Speech

All results in this section were obtained using 10-times-10-folds cross-validation and SVM as a classifier.

1. **Enrich Feature Space**

	Accuracy	Precision	Recall
MFCC and GSR	90.73 ± 1.19	90.76 ± 1.67	91.74 ± 1.54
MFCC-Pitch and GSR	91.34 ± 1.07	93.08 ± 1.07	90.45 ± 1.65
Pitch and GSR	69.04 ± 1.24	71.17 ± 1.37	68.65 ± 2.12

2. **Logistic Regression**

	Accuracy	Precision	Recall
MFCC and GSR	92.43 ± 0.77	92.75 ± 1.47	92.73 ± 1.35
MFCC-Pitch and GSR	92.47 ± 1.37	92.40 ± 1.61	93.32 ± 1.27
Pitch and GSR	70.17 ± 2.36	71.94 ± 2.82	70.71 ± 2.33

C.4 Subject Independent Model

All results in this section were obtained using 1-subject-leave-out cross-validation and SVM as a classifier.

1. **GSR features**

	Accuracy	Precision	Recall
Recovery vs workloads	74.84	83.93	80.00
Recovery vs heavy workload	75.00	79.66	75.83
Light vs heavy workload	63.04	69.01	66.66

2. **Speech features**

	Accuracy	Precision	Recall
Pitch	53.04	54.55	65.83
MFCC	67.82	67.01	80.00
MFCC-Pitch	70.00	71.72	79.16
RASTA PLP	72.17	75.10	79.16

Appendix D

Open Source Library / Toolbox / Software / Script

The following are the library, software, engine and script which were used for developing the web-based stress analytics:

1. **Senticorr: Multilingual sentiment analysis of personal correspondence**
   http://www.win.tue.nl/∼mpechen/projects/senticorr/
2. **Mondrian OLAP**
   An open source OLAP server, which implemented ROLAP model as its storage method.
   http://mondrian.pentaho.com/
3. **OLAP4j API**
   olap4j is an open Java API for OLAP.
   http://www.olap4j.org/
4. **Apache Tomcat 7.0**
   Apache Tomcat is an open source software implementation of the Java Servlet and JavaServer Pages technologies.
   http://tomcat.apache.org/
5. **MySql 5.0**
   Relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases.
   http://dev.mysql.com/

6. **Flot**
   Flot is a pure Javascript plotting library for jQuery. It produces graphical plots of arbitrary datasets on-the-fly client-side.
   http://code.google.com/p/flot/

7. **Trentrichardson Timepicker**
   The timepicker addon adds a timepicker to jQuery UI Datepicker.
   http://trentrichardson.com/examples/timepicker/

8. **jPlayer**
   HTML5 Audio & Video for JQuery.
   http://www.jplayer.org/

9. **musicg**
   Lightweight Java API for audio analysing. This API allows developers to extract audio features and operate audio data like reading, cutting and trimming easily from an inputstream. It also provides tools for digital signal processing, renders the wavform or spectrogram for research and development purpose.
   http://code.google.com/p/musicg/

10. **UCR-Suite**
    The software that enables ultrafast subsequence search under both Dynamic Time Warping (DTW) and Euclidean Distance (ED).
    http://www.cs.ucr.edu/∼eamonn/UCRsuite.html

The following are the library, software and toolbox which were used for data-mining and experiments:

1. **Robert Jang Machine Learning Toolbox**
   This toolbox (MLT, or Machine Learning Toolbox) provides a number of essential functions for machine learning, especially for data clustering and pattern recognition. We used this toolbox for the implementation of Gaussian Mixture Model (GMM).
   http://neural.cs.nthu.edu.tw/jang/matlab/toolbox/machineLearning/

2. **Praat: doing phonetics by computer**
   We used this free software for manual segmentation of speech audio.
   http://www.fon.hum.uva.nl/praat/

3. **VOICEBOX: Speech Processing Toolbox for MATLAB**
   VOICEBOX is a speech processing toolbox consists of MATLAB routines for

speech processing. The robust pitch tracking (RAPT) and MFCC representation were taken from this toolbox.
http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

4. **LibSVM: A Library for Support Vector Machines**
   LIBSVM is an integrated software for support vector classification, (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR) and distribution estimation (one-class SVM). It supports multi-class classification.
   http://www.csie.ntu.edu.tw/ cjlin/libsvm/

5. **EDA Toolbox**
   EDA is a Matlab toolbox for Electrodermal Activity (EDA) processing and analysis.
   https://github.com/mateusjoffily/EDA/wiki

6. **RASTA-PLP**
   Relative Spectral Transform - Perceptual Linear Prediction.
   http://labrosa.ee.columbia.edu/matlab/rastamat/

7. **Matlab Statistics Toolbox**
   Statistics Toolbox provides algorithms and tools for organizing, analyzing, and modeling data. We used this toolbox for logistic regression and decision tree classifier.
   http://www.mathworks.nl/products/statistics/

8. **LEJOS - Java for LEGO Mindstorms**
   LeJOS is a Java based replacement firmware for the Lego Mindstorms RCX microcontroller and NXJ is a Java based replacement firmware for the Lego.
   http://lejos.sourceforge.net/