TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

BIKRAM THAPA

CORRELATION OF STRESS AND PHYSIOLOGICAL DATA

Master of Science thesis

Examiner: Prof. Hannu-Matti Jarvinen
Examiner and topic approved by the
Faculty Council of the Faculty of
Computing and Electrical Engineering
28th February 2018

ABSTRACT

BIKRAM THAPA: Correlation of stress and physiological data
Tampere University of Technology
Master of Science thesis, 63 pages
June 2018
Master's Degree Programme in Information Technology
Major: Pervasive System
Examiner: Prof. Hannu-Matti Jarvinen
Keywords: stress and physiological data correlation, Pearson correlation between stress and physiological data, physiological behavior during stressful programming, measurement of mood during programming

Stress is a mental pressure caused by demanding circumstances, tasks or environment that we live in our daily life. Long-term stress like Chronic stress has a longer negative emotional effect and in a long-term uncontrolled situation, it could damage health and prone to the huge risk of mental and cardiac diseases.

Workplace stress is one of the major stress factors that affect the young working people. According to the World Organization for Stress and the American Institute of Stress, the number of patients with stress-related diseases has been increasing at a drastic rate. Among all people, adults and working people have been reported as being highly affected by stress diseases.

In this thesis, the stress of computer programmers is researched with the participants from software development professionals and students at the university. Their physiological data is examined to find the existence of such features that can signal the different stress levels which can be useful in developing stress aware systems. The physiological activity data is collected using an existing computer peripheral like mouse and keyboards whereas a popular statistical analysis method called Pearson correlation is used to inspect the correlation between stress and physiological data. Such features can be used to model a stress classifier in future which can help in the prediction of stress and provide assistance in mental and psychological well-being.

As a process of organizing and conducting the research successfully, the research proceeds through series of phases like planning, research on related fields, design an experiment, data collection and finally data analysis and interpreting the result.

PREFACE

I am thankful to Professor Hannu-Matti Jarvinen for supervising through the final the phase of Masters thesis and his guidance on writing scientific papers. Special thanks to Professor Petri Ihantola and Researcher Mikko Nurminen for providing the relevant thesis topic and supporting in various technical and instructive ways during the research phase.

Great appreciation and admiration to the laboratory unit of the Pervasive department for providing facilities required for the research purpose and providing financial support. Grateful to all participants who voluntarily participated in experiment session and thanks to all my friends for inviting in parties and events.

Finally, always deepest feelings and love for my parents and girlfriend, Dipti, for inspiring and motivating me with their warm love and always encouraging to pursue Masters degree.

Bikram Thapa
Helsinki, 27.03.2018

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

RQ	Research Question
EDA	Electrodermal Activity
GSR	Galvanic Skin Response
ECG	Electrocardiogram
DNA	Deoxyribonucleic Acid
HR	Heart Rate
IO	Input Output
BIOS	Basic Input Output System
GUI	Graphical User Interface
OOP	Object Oriented Programming
JNI	JAVA Native Interface
JNH	JAVA Native Hook
JVM	JAVA Virtual Machine
3D	3 Dimension
IOS	iPhone Operating System
HCI	Human Computer Interaction
SD	Secure Digital
JSON	JavaScript Object Notation
ASCII	American Standard Code for Information Interchange,
OS	Operating System
IDE	Integrated Development Environment
NASA	National Aeronautics and Space Administration
TLX	Task Load Index
RQ	Research Question
IBM	International Business Machine Corporation
SPSS	Statistical Package for the Social Sciences
UI	User interface
TMC	Test My Code platform

1. INTRODUCTION

Stress is a mental strain caused by demanding circumstances that could be long term or short term. Short-term stress is sometimes beneficial like in flight mode in a dangerous situation while long-term stress is harmful to health that causes chronic diseases in future. Stress creates metabolic or hormonal changes which are unobservable or behavioral changes that are observable through our eyes. Stressors are present in everyday life activities that we perform and depend on the ability of a person to handle it. In our daily life, stressors originate from social judgment, competitiveness, time pressure, inability to handle information overload and many other sources [8, 37].

Human stress can be monitored and controlled using modern technology. Finding stress symptom is possible through monitoring of physiological and behavioral pattern. Continuous monitoring of one's personal stress can help in understanding psychological effect, managing the stressful situation and assisting healthy well being [64].

Systems aware of human stress and cognitive load has huge potential in the development of automated support systems. Stress recognition system can be used for monitoring health, remote learning, automated guidance, automated tutoring and optimizing the workload process [33]. Further, these systems can assist in understanding the learning performance of computer users, teaching outcomes, development of self-adapting systems, development of human emotion aware robots etc.

Stress measurement is one of the hot topic and interest among many researchers in the field of science, technology, psychology and health [59].

At professional workplaces helping employees to balance stress, work, health and work quality is also one of the great interest of employers and researchers. Researchers have studied various method for understanding employee's behavioral and psychological patterns related to stress at workplaces like software development, call centers, academic places etc.[58, 48, 49, 1, 2]. Stress-related diseases can cost huge loss for employee's health and economy of the company[72]. Benefits of stress measuring systems at workplaces include regular monitoring employee's health status, measuring project performance, assisting in balancing work and life etc. In industries like software development, programmer's code quality, expertise, performance, the difficulty of the task, ability to take a quick decision at the deadline can be predicted using their stress and biometric data[47, 37, 20, 61, 18]. Programmers need to solve various technical obstacles to achieve the perceived goal where the failure might cause a negative impact on learning and lose interest towards the subject[16].

There are two underlying approaches in measuring stress [12]

- invasive methods.
- non-invasive methods.

The invasive method uses external sensors that are attached to the human body which measure the biological data and require continuous attention [7]. For example, the biological data could be Electro Dermal Activity(EDA), Galvanic Skin Response (GSR), Heart Rate Variability(HRV), eye movement data etc. Contrary to the invasive method, non-invasive method eliminates the uses of sensors without need to alter the environment or invasion to the human body. The non-invasive method might also use some devices or hardware component that can record human behavioral pattern based on usage. Some example of the computer-related behavioral pattern includes typing rhythm, mouse clicks, facial expression, the pressure of handling input devices etc.

Both methods have pros and cons based on their features. In comparison to economic cost, non-invasive method is cheaper and reachable to a wider range of users since such hardware or devices are owned by the people for personal computing like mobile phones, computer tablets etc. The invasive method is mostly designed for specific task that might require some specific knowledge to interpret the data, should be wearable to in body parts constantly etc.

Since the research interest is on measuring cognitive stress related to programming, a non-invasive method is a suitable method for collecting stress-related behavioral data. Existing computer hardware components and software can be utilized for capturing data. Most research in finding stress for computer-related jobs used existing hardware like a mouse, keyboard, software and human-computer-interaction patterns for stress detection which maintained its accuracies ranging from 77 to 88% [12].

The goal of this research is measuring the stress of programmers during short-term programming session using non-invasive method and finding correlation of captured variables using statistical models.

The research examines keystroke dynamics and mouse dynamics data to find stress pattern related to programming and their correlation. The keystroke dynamics is a unique typing rhythm of a person that makes unique identification of person which can be used for authentication[29]. Similarly, mouse dynamics is also a behavioral mouse usage behavior of a person.

The research tries to address the following research questions (RQs) :

- RQ1. Is keystroke dynamics variable before and after the occurrence of compilation errors?
- RQ2. How does the keystroke vary with timing parameter?
- RQ3. Can stress be predicted based on programmers short term programming session, past experiences, and programmers coding activity?
  - How often errors are generated and correction is done when solving easy and complex questions?
  - How often the user is active in coding environment depending on tasks difficulty level?
  - How does perceived stress correlate to keystroke and mouse dynamics based on their answers to stress survey?
- RQ4. Could Moodmetric ring, a GSR sensor equipped ring to measure stress, be a helpful tool for this research?

The RQ1 is an analysis of how a typing pattern changes before and after the compilation errors are generated. Cryptic error log is one of the factor that effect novice programmers to struggle in learning and slows the performance causing stressful situation[3, 15]. The research question studies the association of variation in keystroke and effect of compilation error logs.

The RQ2 studies how typing behavioral changes when there is time pressure. Some of research have concluded that time pressure has a negative effect on performance, decision taking, creativity, causes stress at workplaces, creates discomfort and anomalous behavior[30, 20, 39, 42]. Miikka Kuutila and his group have reviewed several research papers related to time pressure in software development and found that time creates high number of error at deadlines, experts are not effected as much as novices, tendency of focusing on more technical things increases, time pressure might increase productivity [37].

The RQ3 is a measure of how efficient the short term programming sessions are in finding stress pattern with small data sets. A research by Nandita and Tom concludes that some parameters for stress prediction may not be feasible over certain time segments which reduce the quality and efficiency of the model, however, based on small crucial time segments and appropriate parameters, the prediction results in better accuracy [62]. The recording of programmers activities like text selection, mouse movement, cursor position gives an insight of activeness and difficulties during programming session [67].

The main objective is to analyze which parameters listed in the RQ3 form a good combination for stress measurement or how well these parameters perform. The parameters include keyboard dynamics, mouse dynamics, application usage data, timestamps and GSR data.

Lastly, RQ4 is an evaluation of a stress measuring ring named Moodmetric by a Finnish company called VigoFere Oy. The company claims that their Moodmetric ring measures mood of users based on their galvanic skin response data. In this research, the data from the ring will be used with other parameters for finding the correlations.

This research is carried out in four different phases and organized in following different structures:

- Development of behavioral data collection application :
  This phase includes the development of a software application to log users behavioral data as listed in RQ3. Section 3.1 explains the implementation procedures and different parameters that are logged during the programming session.
- Conducting short term programming sessions:
  A short one-hour programming session will be conducted for each participant where various questions with different difficulty levels will be solved by the subject. The Sections 3.3 and 3.4 describes the procedure for tasks design and sampling participants.
- Data collection and preprocessing:
  Sections 3.1, 3.5 explains the method for collecting various data like survey,

user's behavioral data. Section 3.2 illustrates data preprocessing algorithm and collecting fine-grained data.

- Analysis of data with statistical models and interpretation of result: This phase includes data analysis and evaluation process using different statistical methods. Chapter 4 illustrates the analysis and interpretation of results.

2. BACKGROUND

This chapter explains the state of art in the research area. The sections in this chapter include various background information related to stress, different types of stress, scientific research method and technologies that are relevant for the implementation phase of this research work.

2.1 Stress, effect and types of stress

Stress can be defined as a mental, emotional and physical strain which is caused by the demanding circumstances. The demanding circumstances or stressors can origin from various sources like relationship, job, money, challenge, health etc. which can impact negatively on health [45, 52, 14, 28]. Stress has various effect on different age groups and gender but the influence of regular and uncontrolled stress on health is always negative that can damage the health and causes illness. [65, 59, 53]. Long-term stress is the main cause for chronic diseases that cause damage to internal organs like heart, brain, respiratory system, blood circulation etc. which is difficult to cure or rehabilitate and as well causes huge economic loses[59, 28].

While the damage and effect on health by uncontrolled long-term stress is severe, researchers have also found that not all stress is harmful, but some stress like short-term stress is beneficial in times when there are flight or fight situations to protect themselves or quickly respond to stimuli[44].

In general, stress can be defined mainly in two categories [60]:

- Acute Stress: Stress that has effect for shorter duration and lasts after a short moment. For example, stress before exam or job interview.
- Chronic Stress: Stress that has long prolonged effect and needs special care and long medication practice to rehabilitate or recover. For example, stress caused by the presence of diseases, poverty etc.

2.2 Stress from the biological aspect

Stress has been studied from a various perspective in different fields like science, biology, psychology etc. Various phenomenon is being taken into account while studying stress factors by different researchers. This section illustrates the different aspects and factors associated with stress and the different methodologies used.

Researchers have applied different intrusive and non-intrusive methods along with machine learning or statistical models to measure the stress of their subjects. Those methods vary with one another due to difference in research objectives and fields. In the past, researchers used mainly sensors to capture body signals and measure stress whereas, in the newer research methods, researchers used machine learning techniques with regularly monitored health data along with sensor data. Therefore the methods could be categorized into measurement vs diagnostic approach[21, 7].

Figure 2.2 represents various method and body signals classification obtained to understand human stress, causes and it's pattern [21, 7]. The figure shows the uses of multi-model parameters. As shown in the figure, stress measurement can be grouped into two approaches:

- Diagnostic approach: The Diagnostic approach is based on measurement of changes in physiology, behavior or related activity that may be observable and can be captured by sensors.
- Predictive approach: The predictive approach is based on the information gained from the person's monitored data.

The new approach of measuring signal contradicts to older methods with main difference on the usage of personal profile data. The new approach includes psychological information, background knowledge, performance and behavioral pattern etc. that might be useful for validation of stress signal obtained from sensors. Relying on information obtained by sensors from the human body might be correct. For example, during anger or excitement or physical workout the heartbeat may be faster which means that this data should be used along with personal information. An example of personal information could be a record of the person who goes to workout in the morning and during that time the higher heart rate is not related to stress.

Although there might be differences in term stress, physiological pattern, and measurable signals by researchers, there is no standardized definition and principles. There should be standardization in the emotional model, stimuli for physiological pattern identification, physiological measures, features extraction and model for emotion classification [41].

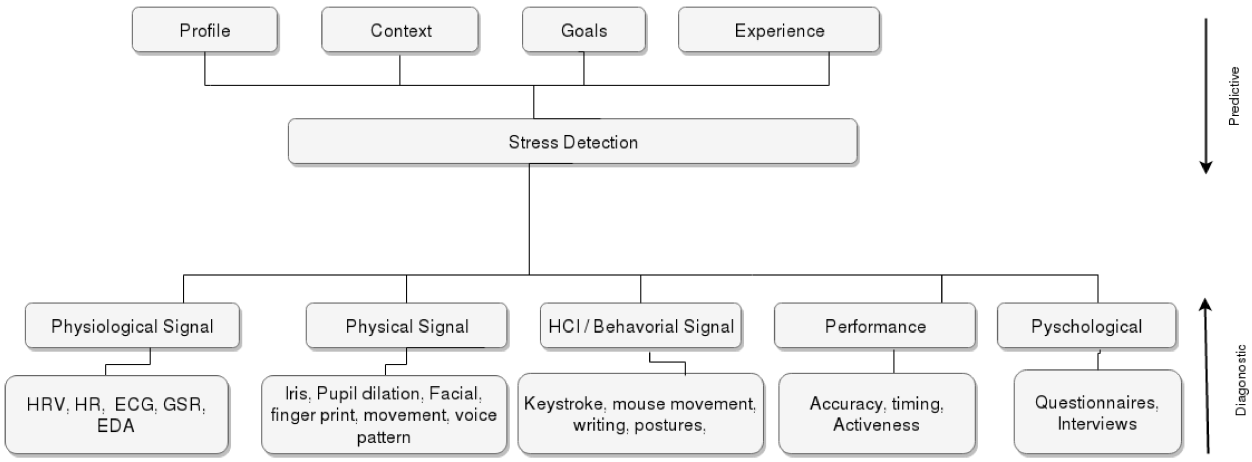


Figure 2.1 *Human stress detection based on Predictive and Diagnostic approach*

The human physiological response is generated by the psychological command from the brain where the nervous system plays main role in receiving and delivering such commands to act on an external stimuli [13]. Nervous system provides linkage between stressor and response thus the nervous system acts as the main origin place for human stress.

Figure 2.1 shows the basic categorization of human nervous system. Our nervous system can be divided into following two group:

- Peripheral nervous system: The peripheral nervous system is associated with receiving sensory information outside the body, transfer information to central nervous system and give commands to parts of the body to respond.
- Central nervous system: The central nervous system consists of brain and spinal cord responsible for thoughts, imagination and information processing. Central nervous system is associated with processing internal sensory information received from peripheral nervous system.

Without peripheral nervous system the central nervous system cannot function properly to respond to stimuli as it is only through peripheral system the central nervous system get messages.

The peripheral nervous system is further divided into two groups.

- Somatic nervous system: Somatic nervous system originates from spinal cord which is responsible for the movement of body with skeletal muscles and stimuli sensed from the environment.

- Autonomic nervous system: Autonomic nervous system is responsible for muscle movement except the body movement like neurons of the gastrointestinal tract, and cardiac and smooth muscles etc. [54]. Autonomic nervous system consist of parasympathetic nervous system and sympathetic nervous system. The parasympathetic controls human homeostasis and body at rest, calm and sleepy stage where as sympathetic is responsible for flight or fight action on a dangerous situation. Parasympathetic controls pupil constricts, increased salivary gland, low heartbeats, lungs respiration constricts, less movements etc. whereas the same opposite happens in sympathetic nervous system.

The stress effects on the regular functioning of organs such as skin color, temperature, blood pressure, heartbeats, cognitive load, body movement etc. These changes are associated with sympathetic responses which are possible to measure using external sensors like heart beat sensor, respiration sensor, thermometer, EDA, ECG, camera etc. To understand better about stress and different methods for measuring stress, the Section 2.4 explains the different method being used to measure stress by many researchers.

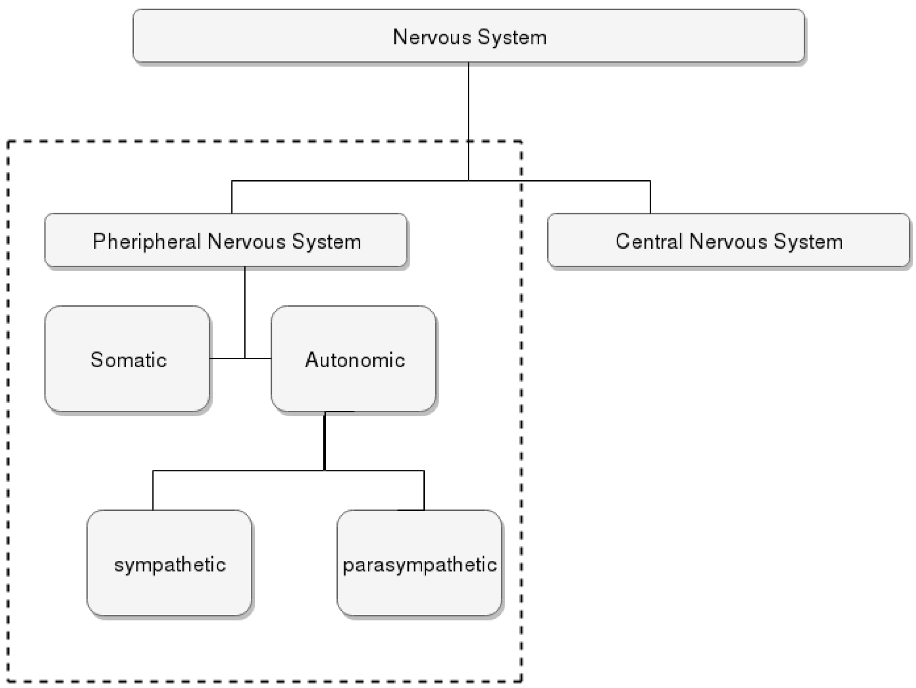


Figure *2.2 Human Nervous system categorization related to human emotion*

2.3 Related research on stress detection during programming

Stress detection and prevention is an interest of many researchers in various fields like science, health, psychology etc. Mainly in the health sector, stress is taken as one of the major factor that exacerbates the health and immune system which results in chronic diseases like cardiovascular diseases, cognitive memory problem, regular illness, respiratory diseases etc. [10].

A research in "Using Psycho-physiological measures to asses difficulty in software development" by Andrew Beagle and Sebastian used eye-tracking sensor, electrodermal and electroencephalogram sensors to measure the physiological data of professional programmers during programming. Their main finding in the research was more than 60% accuracy on the prediction of the difficulty of task based on physiological data and applied machine learning algorithms. Using naive Bayes classifier they were able to find more than 80% accuracy on novice programmer on prediction about the situation the novice programmer feel stressed and find the task difficult.

Another research on detection of frustration of novice programmers, Fwa Hua Leong used contextual modalities and keystroke analysis to create a model for automatic stress prediction. According, to the paper "Automatic Detection of Frustration of Novice Programmers from contextual and keystroke logs" they used keystroke analysis as non-obtrusive method to find the stress in novice programmers. The term stress is mentioned as frustration in the paper. According to Hua Leaong, the prediction model was able to get 0.67 accuracy level and recall of 0.833 which is a positive result in detection of stress. Their method used logistic regression with lasso regularization for modeling stress and prevent data overfitting whereas their data included keystroke data collected during programming.

Similarly, related to non-obtrusive method, the research by Andre Pimenta, Davide Carneira, Jose Neves in their research paper "A neural network to classify fatigue from human-computer interaction" they reported their accuracy to detect stress was above 80% using an artificial neural network and data captured with repeated experiment. Keystroke, mouse movements and clicks data were logged while participants were performing human interaction based exercises on stressful and non-stressful situation. Additionally, they used NASA TLX which allowed the participant to reflect their mental, physical demand related to tasks performed.

The research paper "Detecting Emotional stress during typing task with time pressure" by Yee Mei Lim, Alaaddin Ayesh and Martin present their research based on time pressure. The research analyses stress and effect on mouse and keystroke dynamics affected by time pressure. They explain that there can be huge potential to develop an adaptive e-learning system by detecting e-learners emotional stress based on keystroke and mouse dynamics. Their findings show that unfamiliarity with task increases stress in e-learners.

Lastly, related researches but not using the same technique are from Seothwa Lee, Danial Hooshyar, Hyesung Ji on paper called "Mining biometric data to predict programmers expertise and task difficulty" where they present their findings on prediction of programmers expertise based on data obtained from psycho-physiological sensors. With experiment with 38 novice and expert programmers, their data was analyzed with Pearson correlation and NASA TLX. The result showed that their model could predict task difficulty and programmers expertise level with 64% precision and 97% precision and 68% and 96% recall respectively. The research paper on "Time Pressure: A Controlled Experiment of Test case Development and Requirements Review" Mika V. Mantyla, Kai Petersen, Timo O.A Lehtinen and Casper Lassenius used time pressure to understand productivity on developing test cases and reviews. They used controlled experiment to understand the productivity with professionals in the software industry. Their result showed that there is significant productivity increased when the deadline approaches but found no significant evidence that time pressure decreases productivity. In research to modeling and improving pass-fail classifier by Kevin Casey in his paper "Using Keystroke Analytics to improve pass-fail classifier" he presents his research to find the early point when a student needs special intervene to assist them. He used digraph latency data to model pass fail classifier. The result shows that when student learns more depth into programming and is writing a complex program, it could be an ideal early indicator for pass-fail classier to use those dimensions to improve classifier. The paper also concludes that the programming languages skills also plays a significant role in prediction and accuracy.

2.4 Biometrics introduction

The scientific research on using biological data started in the 19th century as computer power became more powerful and proved to be a reliable way to identify criminals [32]. However this wasn't new topic since people in America, Europe and Asia used some physical characteristics like typing signatures to verify people but the revolution for using in computing and scientific purpose started in the 1960s [6, 9]. Later all applications, measurements and integrating of biological and behavioral features in computing and scientific research was termed biometrics.

The word biometric is derived from the Greek word 'bio' and 'metron'. 'Bio' relates to the meaning life and 'metron' relates to measure. In other words, the statistical measurement of biological and physical features of the human body [27]. In ancient periods biometrics still existed but without the use of computer technology. The use of fingerprints, hand signatures in historical periods are evidence that biometric existed previously. Biometrics revolutionized in the mid 1960's as a security measure in network and software authentication [9].

Each individual is unique to his/her physiological or physical conditions which can be taken as an additional feature to enhance security layer over authentication process. But the loss of such physical structures or features due to accidents, change on the behavioral pattern may also cause the loss of control or access to such systems. Therefore, biometrics cannot replace the existing security systems like PIN code, passwords, swipe cards etc. but they can be used for enhancing the current security system [27] . Based on the principle measurement of characteristics, the biometric features can be categorized into two types [5, 57]:

- Physiological biometric: Physiology is a term used to define the characteristics of a human body and biometric that deals with such characteristics are known as physiological biometric[35]. Such physiological characteristics are bonded the with human body since their birth [57] and can be measured using external devices like wearable sensors, laboratory tests, ECG etc. The common physiology based biometrics are iris recognition, DNA analysis, hand geometry, fingerprint recognition etc.
- Behavioral biometrics: Behavioral biometric measures the human behavioral patterns that are reflected to outside world and occurs repeatedly in daily life which forms a distinguishable pattern of a person [35]. Examples are typing pattern, gait analysis, body gestures, hand signatures etc.

It is also good to explain that the term physiological and behavioral biometrics have some common and different features. Some researchers have used the term "affect" to explain the phenomenon rather than pointing directly to physiology [33].

Although physiological and behavioral classifications may have some differences, the following four general qualities are important in order to be accepted as valid biometric features[32].

- Universal: Universal explains the term that every individual must have some characteristics to be usable in biometric. However, some specific features like scars, spots on skins are not considerable universal.
- Persistent: Explains that selected biometric feature should not alter over time. For example - fingerprint, researcher Anil K. Jain and his fellow group identified that fingerprint of a child after 2.5 years of birth serves his/her identification throughout the life [26].

- Unique: Uniqueness defines that the feature of biometric should be unique in order to distinguish one person from another person.
- Distinctiveness: This quality explains that biometric features should be distinctive although some characteristics might not be unique. The distinctive property should be sufficient enough to separate the individuals. Hand geometry feature is an example of distinctiveness in biometric.

2.5 Key-loggers and Keystroke dynamics

Although the term key logger and keystroke dynamics seem to have similar meaning and functionality in a way they capture data from computer keyboard or mobile screen, there are certain differences between them. This section explains key loggers, keystroke dynamics and features of keystroke dynamics.

2.5.1 Key-logger and types

Key-logger is a malware program that maliciously records user's keyboard's and touch screen's input as well as activity information to gain personal information[73]. The key-logger is designed to record personal data and transfer it though network when the computer devices have an Internet connection. Therefore, a key-logger is taken as a major security threat to the computer users and has a bad reputation as it can be used for illegal purposes. But there are also good uses of key-loggers like monitoring illegal uses of software and application, keeping track of information for verification process etc.

Key-loggers can be divided into two types as

- Software key-loggers: Software key-loggers are programs that run in the background being invisible in a computer and spies on input data. The software key-loggers can be classified into two types as [73]
  - User Level: User level key-loggers are easiest to construct and to detect as well. User level key-loggers have an access to user's account and have global hooks to the keyboard's events. Such key-loggers are transferred and executed through website widgets, advertisement illusions etc. and can replicate themselves when activated.
  - Kernel level: Kernel level key-loggers requires special administrative access and privileges and usually operate during operating system boot

process. This kind of key-logger might exist at network computers or servers and is able to replicate. They have a hook to kernel.

- Hardware key-loggers: Hardware key-loggers consist of hardware component connected between the keyboard and I/O processing unit. Hardware level key-loggers can also have access to BIOS level and do not need any installation drivers or such software to activate it.

2.5.2 Keystroke Dynamics

The evolution of keystroke dynamics started in 19th century as it proved to be a reliable method for authentication while telegraph was a popular method for messaging [66].

Keystroke dynamics records detailed, timed typing rhythm of a person based on keyboard events like key presses and releases, duration of keypress etc. while typing using keyboard [76]. Thus keystroke dynamics differs to key-loggers in a way that it stores detailed timing information and forms as digital footprint. Keystroke dynamics is a cheap behavioral non-intrusive biometric widely used for authentication that requires only software running on the background without additional hardware [76, 74]. Since the success of using keystroke of authentication, during last decades there has been increasing research in using keystroke biometrics for understanding the human psychology and physiological reactions for development of automated self-adapting systems [40, 7, 21, 67, 43, 33, 66].

2.5.3 Keystroke dynamics measurement Process

Keystroke dynamics can be applied into two different aspects[50]:

- Static text: The static text relates to fixed words which are predetermined or saved like passwords and used in static period like login [46]. Static text keystroke dynamics provides better verification than using simple passwords but cannot be used in replacement of user's cognitive password.

- Free or dynamic text: Dynamic text is based on non-fixed free words typed by the user without knowing in prior. Dynamic text keystroke monitors the keystroke during the entire session for better verification but the accuracy is less than static keystroke dynamics [76].

Researchers have used keyloggers for recording the keystroke pattern which is the easiest and non-intrusive method in data collection [40, 38, 19, 46, 55, 38, 21, 12, 31]. However, in some new research method, different novel approaches are used like sensing keystroke pressure during typing, free text linguistic analysis and keystroke acoustics [25, 51, 71, 56]. In Microsoft Research, Hernandez, Pablo and his team induced a pressure sensor beneath the keyboard for sensing pressure and found that pressure amount increases significantly as stress increases which was revealed in their measurement from more than 79% candidate's data [25]. In linguistic feature based analysis, the author used the spontaneous free typed text by user to compare with Cognitive emotion related database to assess the emotional state [71]. Similarly, Joseph Roth used a novel approach of using keystroke sound for authentication but the result from their experiment did not show better results [56].Despite the variation in keystroke measurement, different experiments were conducted based on the objective of research like whether authenticating a user or sensing the stress level.

There are two phases in keystroke dynamics 1) training 2) recognition. In the training phase, typing parameters are obtained and a model is trained based on the typing behavioral data. The recognition phase uses stored information and checks match against new input data using the classification method.

Figure 2.3 shows the general flow chart of keystroke training and testing using keystroke dynamics during the authentication process.

2.5.4 Keystroke dynamics features collection

Keystroke dynamics is based on the timing and frequency of keys pressed, released, hold and paused events [76, 34]. Timestamp is an important parameter in keystroke dynamics. There are various terms used to represent the measurable keystroke dynamics features by researchers but many of them share common properties [34, 36, 50, 46, 40]. Although there are differences in the term for keystroke features representation, the following lists describes the commonly used keystroke dynamic features [34, 40, 76, 70]:

- Latency Time: Time between first the key is released full upwards and full depression of the second key. Also called "Flight" time or "Up-Down" time.
- Dwell Time: The amount of time spent after key is pressed and the key is not released. Also called "Duration" or "Hold" or "Press-Hold" time.

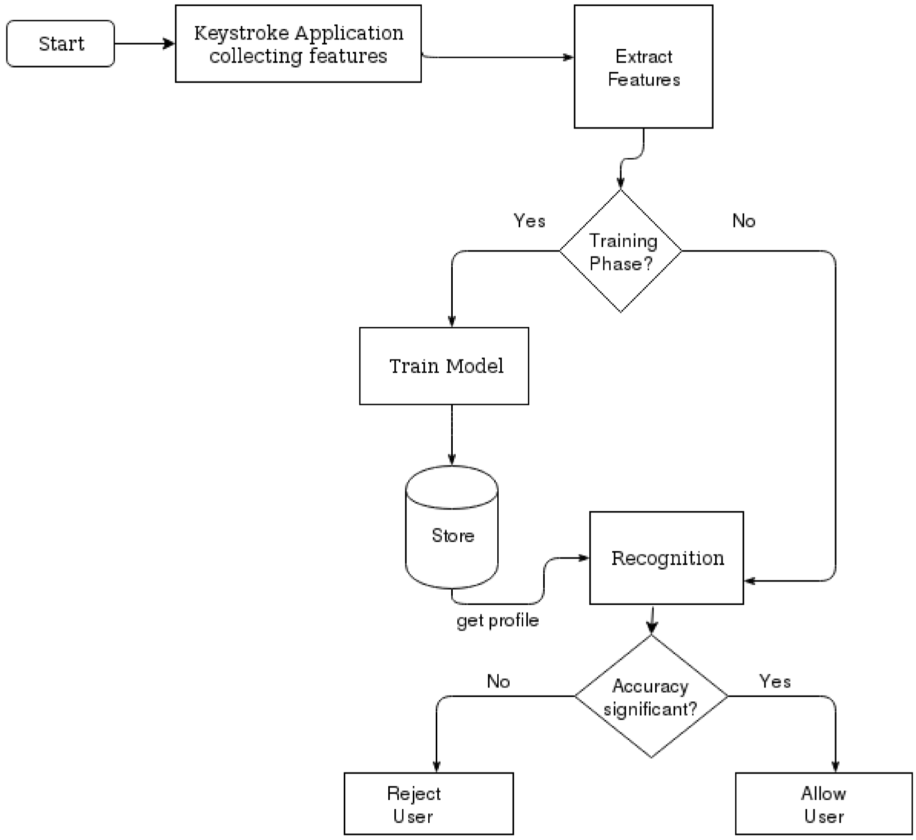


***Figure* 2.3** *Flow chart of keystroke based authentication phases*

- Seek Time: The mean time between the last pressed key and a new key is pressed for some particular key.
- Digraph: This is less appearing keystroke event. Timestamps when a first key is not fully released the next key is pressed or first key is pushed down and also the second key is pushed down while the first key has not been depressed completely. Also called overlapping time. Normally this happens when the keyboard is typed fast.
- Trigraph Time: Time between the first key is pressed and the third key is released up. It combines three keys event.

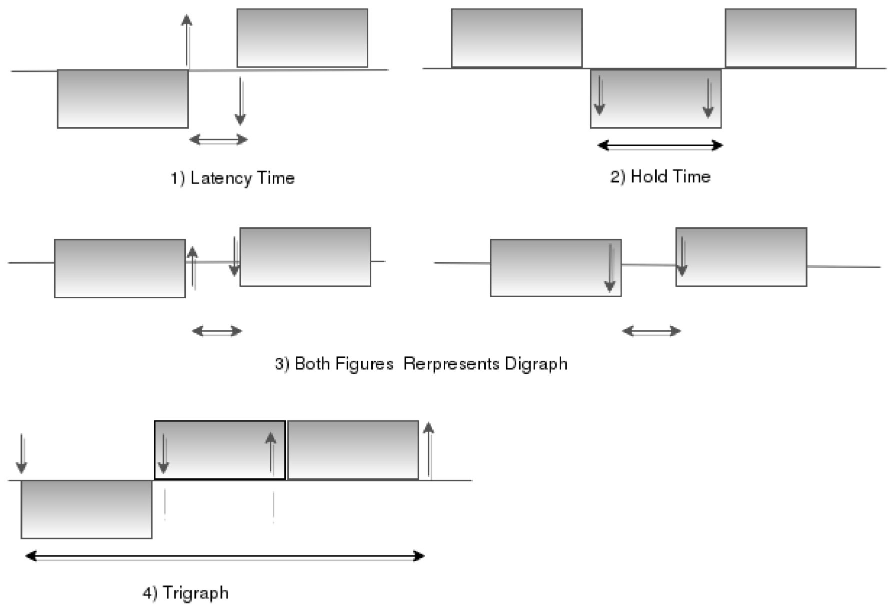


Figure 2.4 *Keystroke Dynamics Features*

In Figure 2.4, horizontal lines with an arrow on both sides represent the keystroke timestamps. The plain horizontal line represents the surface baseline of the keyboard, up and down arrows represent if keys are fully released or depressed and up-down arrows crossing the mid horizontal plain line represents partial depression or release of keys.

Except key press and release timing, there are other properties measured as part of keystroke dynamics:

- Frequency: Frequency is another characteristic used in keystroke dynamics. Keystroke frequency is a count of a number of particular keys pressed within some window time frame like minute or hour. The key pressed for correction and deletion are the counted separately and processed later in analysis [34]. Alternatively called keystroke verbosity [4].
- Pauses: The number of pauses within the given interval or the time duration when there is no keypresses or releases.
- Session timing: The time between start to end of the certain task or whole job session.

2.6 Mouse dynamics

The mouse dynamic is user's mouse usage behavioral pattern during interaction with GUI components. Many computer mouses share similar features whether it is notebook touchpad or external physical mouses. Most common mouse related behavior features are cursor movements, clicks, scroll etc. Mouse trackers provide real-time rich, the valuable behavioral insight of human psychological state [17, 24]. Studies by David Sun on HIS paper has shown that mouse dynamics provides better stress detection than using other physiological sensors [64]. Despite psychology, Business is another sector that has benefits from using mouse dynamics. Some commercial companies have used mouse dynamics to understand customer engagement and behavior with products on their websites. A company named kissmetrics (kissmetrics.com) claims their services provide customer analytics with mouse tracking to better understand the consumer behavior.

The following list explains the general keystroke features that can be extracted from the mouse events

- Mouse clicked: Pressing or releasing of mouse left or right button.
- Mouse cursor movement: Mouse cursor moved from one place to another place.
- Mouse Application Focus: Application gets focused by mouse events.
- Mouse Application Out Focus: Mouse cursor moved away from tested application.
- Mouse Dragged: Data or object moved by mouse like dragging pictures, GUI components (widget) in editor etc.
- Mouse Scrolled: Mouse wheel is scrolled.
- Mouse silence: There is no event with the mouse. Mouse cursor stays idle.
- Mouse hover: Mouse pointer is hovered over some graphical component.
- Mouse Selection: Mouse is used to select texts or other objects like files etc.
- Mouse Acceleration: The acceleration of cursor at a given time.
- Mouse velocity: The velocity of cursor movement.
- Mouse Distance: The distance measured as high and low peak or high and low distance traveled by the mouse during a given time.

3. RESEARCH METHOD AND IMPLEMENTATION

This chapter describes the experimental settings, method and tools used for capturing keystrokes, mouse dynamics, and recording application usage and webcam video data using through computer peripherals and software. The section also reviews the data filtering process and a Pearson statistical model for analyzing the correlation of captured variables.

3.1 Participants selection and motivation

In this research, a total of 10 subjects participated who had different programming skills and knowledge of data structures. The controlled experiment needed each participants to solve different programming tasks and have good programming knowledge and skills in prior. The main reason to have skilled subjects was to obtain maximum keystrokes data related to programming rather than novices who would generate fewer keystrokes data which could not be abundant for data analysis. Therefore, volunteers without programming experience in past were excluded.

Subjects were from different countries having different native languages and used different keyboard layouts. Most of the subjects were affiliated to either University profession or software development profession in industry. Also, some of the subjects were academic software engineering student motivated by some incentive as rewards at the end. Their names and only background of programming skills were taken into account when inviting them for an experiment session.

3.2 Programming task design

The research experiment included one hour of programming session where seven different Java programs had to be written by each participant. Although having the experience in programming most subjects did not have much experience with Java. But because of their prior experience and knowledge with another programming languages and familiarity with data structures and algorithms, it made reasonable to include them in experiment session.

The Java questions were designed to have a different level of difficulty and ordered in easiest to most difficult ones. First two question were the easy ones which required only basic programming concept like loops, conditional checking etc. while the rest five questions needed efficient data structure and object-oriented programming knowledge. In addition, those five difficult questions also required good performance in terms of running algorithmic complexity.

Difficulty settings were applied based on an assumption that each participant would be able to solve at least one or two easy questions and probably would try to solve the difficult ones. Thus this would provide an opportunity to collect keystroke and mouse dynamics data that could be useful for examining the difference in easier and difficult ones.

Table 3.1 shows the design of questions with difficulty level and required Java skills.

Table 3.1 *Programming skills and difficulty levels of programming tasks*

Question	Required Programming Skills	Difficulty	Difficulty level
1	Basic Java operators and Syntax	Easy	1
2	Loop and conditional checking	Easy	2
3	OOP, algorithms, performance	Difficult	3
4	Data structure and algorithms, performance	Difficult	4
5	Data Structure and algorithms, performance	Difficult	5
6	Data structure and algorithms, performance	Difficult	6
7	Data structure and algorithms, performance	Difficult	7

3.3 Coding environment

The research was conducted in the Laboratory of pervasive computing with preselected and configured computer which would collect the physiological data. Instead of conducting the experiments of all participants at once, the experiment was conducted in different sessions as a suitable time for all participants did not match. Also, the other reason was that the laboratory had only one computer installed with required data collecting software running on it.

The default system used for the experiment was equipped it Linux environment and most of the participants were familiar with it. Since participants were from different

Figure 3.1 *Finnish Keyboard layout*

countries, the tasks were made available in multiple languages. Finnish and English were the two languages available. During the experiment, tasks were explained in English and materials were translated in English as well.

Some participants were having issues in using the keyboard layout and language in the computer as the system used for experiment had Finnish layout which was not familiar to some participants. The Finnish keyboard layout varies slightly with the wild-card characters which is necessary in programming.

Figure 3.1 shows the basic Finnish keyboard layout being used in the experiment session.

3.4 Self workload reporting

Self workload reporting is a set questionnaire to be filled by every participant after completing each task. Basically, it contains stress related questions which is used in this research as an alternative to Moodmetric ring where participants specify their stress level rather than by measuring with the sensor. The self work load reporting is used for studying the statistical correlation of captured physiological data experienced by subjects. The self workload assessment is performed using NASA task load Index survey.

The NASA TLX is a subjective multidimensional workload assessment method based on the average of six subscale ratings provided by the operators during the task performance[23]. NASA TLX was originally used in aviation which later was adopted in various application like military, driving, robotics operation, computer usages etc. [22]

Figure 3.2 *NASA TLX Questionnaire*

The six dimensions include sub scale ratings for mental, physical, temporal, perceived performance, effort, and frustration demands. However, in this research, a slight modification is made by adding new dimension named 'Familiarity' to obtain information about past experience of similar to the current task. The physical demand is also removed from the original NASA TLX since the task eliminates the physical activities. The NASA TLX questionnaire form is displayed in NetBeans IDE when the task is completed successfully and passes all test cases set in the server. But in case failure and unfinished task, a physical hard copy of NASA TLX is given to the subject so that they can still rate their workload experience.

3.5 Construction of Keyboard, mouse and Application logger

Since the research's interest is on finding stress-related behavioral pattern using non-invasive technique, behavior logging applications are used to capture the mouse, keyboard, application, screen and webcam video data. The data capturing process leverages embedded computer hardware components like touchpad, webcam, keyboard etc. An open source logging software named Java Native Hook(JNH) is used for recoding the user's behavioral data which is available from a github repository(https://github.com/kwhat/jnativehook/). The software is slightly customized to capture data related to research questions using custom parameters like timestamp, events etc.

The selection of JNH over other software is due to its support on various platform and its capability to provide low-level system-wide hook to listen to keyboard and mouse events. Most programming languages provide basic keyboard and mouse events information however they require specific access to hardware component due to OS security issues. Another problem with the basic keylogger listeners is loss of the data when a window loses active focus state, eg. when the window is minimized. The JNH makes it possible through the use of Java Native Interface(JNI).

The JNI is a framework that facilitates Java code running on a Java Virtual Machine (JVM) to be called or call another native program that have access to hardware. Thus JNI acts as a bridge between low-level language or assembly language. The JNH leverages the platform dependent native code like c++ or c through JNI. Although multi-platform support is one of the good features of JNH, it also requires programmers to code different codes for different platforms.

3.5.1 Key logger data collection

Keywords consist of alphabets, numbers, symbols and Unicode characters depending on locale languages and keyboard settings. Although keyboards might have different keys based on key's position or locale, JNH captures key events like the key press, key release, key hold etc. and defines specific hex representation for each key. There is no representation for Unicode characters. In the context of this research, Unicode characters are not used and even Java syntax does not contain Unicode characters. Such Unicode characters are eliminated from the programming tasks. Additionally JNH is capable of handling modifiers key that changes the value of keys capitalizing letters, selecting texts, printing symbols etc. Shift, Ctrl, alt keys are examples of modifier keys.

The following list describes the keyboard events that are supported by JNH.

- Key Press Events - Event triggered when a key is pressed but hasn't reached the bottom.
- Key Release Events - A key pressed at an earlier time was released.
- Key Typed Events - Key reaches to bottom and actual key value is realized.

```
#define org_jnativehook_keyboard_NativeKeyEvent_VC_MINUS          0x000C  // '-'
#define org_jnativehook_keyboard_NativeKeyEvent_VC_EQUALS         0x000D  // '='
#define org_jnativehook_keyboard_NativeKeyEvent_VC_BACKSPACE    0x000E

#define org_jnativehook_keyboard_NativeKeyEvent_VC_TAB            0x000F
#define org_jnativehook_keyboard_NativeKeyEvent_VC_CAPS_LOCK    0x003A

/** VC_A thru VC_Z */
#define org_jnativehook_keyboard_NativeKeyEvent_VC_A              0x001E
#define org_jnativehook_keyboard_NativeKeyEvent_VC_B              0x0030
#define org_jnativehook_keyboard_NativeKeyEvent_VC_C              0x002E
#define org_jnativehook_keyboard_NativeKeyEvent_VC_D              0x0020
#define org_jnativehook_keyboard_NativeKeyEvent_VC_E              0x0012
#define org_jnativehook_keyboard_NativeKeyEvent_VC_F              0x0021
```

Figure 3.3 *JNI Keycode representation for keys in keyboard*

Table 3.2 *Keyboard events and data captured during different event*

Event	**Data captured**
NATIVE_KEY_PRESSED	keyCode=Hex code representation value keyText= Pressed keys value [a-z / 1-9 / Symbols] keyChar=Undefined - to represent nothing is typed yet
NATIVE_KEY_RELEASED	keyCode=Hex code representation value keyText=Pressed keys value [a-z / 1-9 / Symbols] keyChar=Pressed keys value [a-z / 1-9 / Symbols]
NATIVE_KEY_TYPED	keyCode=Hex code representation value keyText=Pressed keys value [a-z / 1-9 / Symbols] keyChar=Pressed keys value [a-z / 1-9 / Symbols]
timestamp	Timestamps of event triggered.

Figure 3.3 shows the Java code for different keycode values. Those key codes shown in the figure are refined virtual constants for native use. The virtual constants include representation for all alphabets (a-z / A-Z), numbers (0-9), symbols, function keys(F1 - F12) and other keys function like power, escape keys etc. There are some instances when multiple keys are pressed to type some Unicode character or used as a modifier to trigger some event. Shift key, up, down etc. are an example of modifier keywords which is also represented by predefined constants.

Figure 3.2 shows the common key events and the data that are captured during those events. NATIVE_KEY_PRESSED, NATIVE_KEY_TYPED, NATIVE_KEY_RELEASED are different keyboard related events that are triggered during typing activity. Also as seen in the figure, the data captured during key events also varies slightly. The value of keychar for NATIVE_KEY_PRESSED event is always undefined as it is unknown what value is typed unless NATIVE_KEY_TYPED is triggered. Timestamp is another custom Linux's time stamp value logged along with other data in all events. The timestamps are recorded to analyze the correlation of these variables with the timing parameter as stated in our research question.

```java
/**
 * @see org.jnativehook.keyboard.NativeKeyListener#nativeKeyPressed(org.jnativehook.keyboard.NativeKeyEvent)
 */
public void nativeKeyPressed(NativeKeyEvent e) {
    displayEventInfo(e);
}

/**
 * @see org.jnativehook.keyboard.NativeKeyListener#nativeKeyReleased(org.jnativehook.keyboard.NativeKeyEvent)
 */
public void nativeKeyReleased(NativeKeyEvent e) {
    displayEventInfo(e);
}

/**
 * @see org.jnativehook.keyboard.NativeKeyListener#nativeKeyTyped(org.jnativehook.keyboard.NativeKeyEvent)
 */
public void nativeKeyTyped(NativeKeyEvent e) {
    displayEventInfo(e);
}
```

Figure 3.4 *Java functions to handle various keyboard events*

Figure 3.4 shows the Java methods that get called when keyboard events trigger. Similarly, Figure 3.5 shows the sample data logged from keyboard events by JNH.


```
","eventData":"NATIVE_KEY_TYPED,keyCode=0,keyText=Undefined,keyChar='n',keyLocation=KEY_LOCATION_UNKNOWN,rawCode=110","tim
","eventData":"NATIVE_KEY_PRESSED,keyCode=57424,keyText=Down,keyChar=Undefined,modifiers=Shift,keyLocation=KEY_LOCATION_ST
","eventData":"NATIVE_KEY_PRESSED,keyCode=14,keyText=Backspace,keyChar=Undefined,keyLocation=KEY_LOCATION_STANDARD,rawCode
","eventData":"NATIVE_KEY_TYPED,keyCode=0,keyText=Undefined,keyChar='\b',keyLocation=KEY_LOCATION_UNKNOWN,rawCode=65288","
","eventData":"NATIVE_MOUSE_MOVED,(988,284),button=0,clickCount=0","timeStamp":1505814150776,"activeWindow":"Bonus-tiraB.1
","eventData":"NATIVE_MOUSE_MOVED,(990,402),button=0,clickCount=8","timeStamp":1505814161636,"activeWindow":"Bonus-tiraB.1
","eventData":"NATIVE_KEY_PRESSED,keyCode=57416,keyText=Up,keyChar=Undefined,keyLocation=KEY_LOCATION_STANDARD,rawCode=653
```


Figure 3.5 *Sample of keylogger data captured by JNH*

3.5.2 Mouse logger data collection

Besides keystroke logging, JNH also captures mouse events which are provided within the JNH package. The mouse event works with both external mouse or touch pads in laptops. JNH also has representation for a 3D mouse having more than two buttons. Below is a list of mouse events captured by JNH

- Mouse Down Events: Left or Right button pressed down
- Mouse Up Events: Pressed button is released
- Mouse Click Events: Clicked mouse button reached to the bottom
- Mouse Move Events: Mouse is moved
- Mouse Drag Events: Mouse is dragged

- Mouse Wheel Events: Wheel is scrolled or scroll event is triggered by touch-pads

The mouse events and data obtained from the mouse is shown in table 3.3. Those constants for keyboard events are predefined in a base class file named "Native-MouseEvent.java" which is shown in figure 3.4.

Table 3.3 *Mouse events and data captured during different event*

Event	**Data Captured**
NATIVE_MOUSE_PRESSED	location = pixels (x, y) of screen button= 1 or 2 [1=left button, 2 = right button] modifiers=Button1 [button pressed] clickCount= n [n number of counts] timestamps = time stamp of event
NATIVE_MOUSE_MOVED	location = pixels (x, y) of screen button=0 [0 = no button pressed during move event] clickCounts = 0 [0 = no clicks during move event] timestamps = time stamp of event
NATIVE_MOUSE_RELEASED	location = pixels (x, y) of screen button= 1 or 2 [1=left button, 2 = right button] modifiers=Button1 [button pressed] clickCount= n [n number of counts] timestamps = timestamps of event
NATIVE_MOUSE_WHEEL	location = pixels (x, y) of screen button= 1 or 2 [1=left button, 2 = right button] clickCount=n[n number of counts] scrollType=WHEEL_UNIT_SCROLL scrollAmount= n [n number of scrolls] wheelRotation=1 or 2 [up or down] wheelDirection=WHEEL_VERTICAL_DIRECTION timestamp = timestamp of event
NATIVE_MOUSE_CLICKED	location= button= 1 or 2 [1=left button, 2 = right button] clickCount=n[n number of counts] timestamps = timestamps of event
NATIVE_MOUSE_DRAGGED	location = pixels (x, y) of screen button= 1 or 2 [1=left button, 2 = right button] modifiers=Button1 or Button2 [button pressed] clickCount= n [n number of counts] timestamp = timestamp of event

```
#define org_jnativehook_mouse_NativeMouseEvent_NATIVE_MOUSE_CLICKED              org_jnativehook_mouse_NativeMouseEvent_
#define org_jnativehook_mouse_NativeMouseEvent_NATIVE_MOUSE_PRESSED            1 + org_jnativehook_mouse_NativeMouseEvent_NATI
#define org_jnativehook_mouse_NativeMouseEvent_NATIVE_MOUSE_RELEASED   2 + org_jnativehook_mouse_NativeMouseEvent_NATIVE_MOUSE
#define org_jnativehook_mouse_NativeMouseEvent_NATIVE_MOUSE_MOVED              3 + org_jnativehook_mouse_NativeMouseEvent_NATI
#define org_jnativehook_mouse_NativeMouseEvent_NATIVE_MOUSE_DRAGGED            4 + org_jnativehook_mouse_NativeMouseEvent_NATI
#define org_jnativehook_mouse_NativeMouseEvent_NATIVE_MOUSE_WHEEL              5 + org_jnativehook_mouse_NativeMouseEvent_NATI
```

Figure 3.6 *Java KeyCode representation in NativeMouseEvent class*

Similarly, figure 3.7 shows the mouse event methods that are called during the mouse events triggers and calls the same handler function as keyboard called displayEventInfo(). The sample of the data captured by JNH is shown in figure 3.8. Mouse logger also logs the custom timestamps data along with all the mouse events data.

```java
/**
 * @see org.jnativehook.mouse.NativeMouseListener#nativeMouseClicked(org.jnativehook.mouse.NativeMouseEvent)
 */
public void nativeMouseClicked(NativeMouseEvent e) {
    displayEventInfo(e);
}

/**
 * @see org.jnativehook.mouse.NativeMouseListener#nativeMousePressed(org.jnativehook.mouse.NativeMouseEvent)
 */
public void nativeMousePressed(NativeMouseEvent e) {
    displayEventInfo(e);
}

/**
 * @see org.jnativehook.mouse.NativeMouseListener#nativeMouseReleased(org.jnativehook.mouse.NativeMouseEvent)
 */
public void nativeMouseReleased(NativeMouseEvent e) {
    displayEventInfo(e);
}

/**
 * @see org.jnativehook.mouse.NativeMouseMotionListener#nativeMouseMoved(org.jnativehook.mouse.NativeMouseEvent)
 */
public void nativeMouseMoved(NativeMouseEvent e) {
    displayEventInfo(e);
}
```

Figure 3.7 *Sample of Keystroke data captured*


```
"eventData":"NATIVE_MOUSE_DRAGGED,(1470,16),button=0,modifiers=Button1,clickCount=0","timeStamp":1505816184760,"activeWindow":"Key, mouse & windowlogging.
"eventData":"NATIVE_MOUSE_MOVED,(446,556),button=0,clickCount=0","timeStamp":1505814103132,"activeWindow":"Netbeans with TMC 1.1.7"}},
"eventData":"NATIVE_MOUSE_CLICKED,(348,188),button=1,clickCount=1","timeStamp":1505812774892,"activeWindow":"Mozilla Firefox"}},
"eventData":"NATIVE_MOUSE_MOVED,(1389,50),button=0,clickCount=0","timeStamp":1505814111196,"activeWindow":" "}},
"eventData":"NATIVE_MOUSE_PRESSED,(812,392),button=0,modifiers=Button1,clickCount=0","timeStamp":1505816195500,"activeWindow":"Pääte"}},
```


Figure 3.8 *Sample of Keystroke data captured*

3.5.3 Application logger data collection

The application logging is another feature of JNH which captures the active software window the user is engaged with. The application logger captures the name of the

```
"NATIVE_MOUSE_MOVED,(1430,383),button=0,clickCount=0","timeStamp":1505814101888,"activeWindow":"Netbeans with TMC 1.1.7"}},
"NATIVE_MOUSE_MOVED,(292,838),button=0,clickCount=0","timeStamp":1505815963924,"activeWindow":"array in Java - Mozilla Firefox"}},
"NATIVE_MOUSE_MOVED,(1379,37),button=0,clickCount=0","timeStamp":1505814090860,"activeWindow":"Google - Mozilla Firefox"}},
"NATIVE_MOUSE_MOVED,(975,1079),button=0,clickCount=0","timeStamp":1505815933820,"activeWindow":"Netbeans with TMC 1.1.7"}},
"NATIVE_MOUSE_MOVED,(314,1067),button=0,clickCount=0","timeStamp":1505815981176,"activeWindow":"Netbeans with TMC 1.1.7"}},
```


Figure 3.9 *Sample of application logger data captured by JNH*

software or window title of active application. The application logger data is added in every log of the mouse and keyboard events which is separated by a name called "activeWindow".

The application logger provides useful information to analyze the application usage behavior like browsers, focus on coding environment etc. If something is typed in an active software input fields the JNH also records the text from it. For example, the text retrieved from application keylogger during typing in browser might be a good source to understanding the help-seeking and searching behavior. Free text analysis from application logger might be good source to classify help-seeking words related to stress and programming.

Figure 3.9 shows the sample data of application logger. The "activeWindow" captures actively focused application's name and window title.

3.5.4 Video logger data collection

Video logger is a custom shell script created for webcam video recording for this research to capture facial expression. The application is not a fully coded package or class, however, it just uses shell script through the console to call the service of an open source video software called ffmpeg (https://www.ffmpeg.org/).

In short introduction, ffmpeg is cross platform solution to record and stream audio and video data. The ffmpeg has wide range of features like recording and converting video and audio, streaming media files etc. The ffmpeg is powered by codec library called 'libavcodec' which uses variety of video and image codecs [75]. The selection of ffmpeg for this research is based on the feature like it requires low computation power, has smooth functioning, good documentation and provides simple commands to record audio and video data.

Video logger only records the video of subjects when they are participating in an experiment. A timestamp is added as an overlay on the video so that it can be known what subjects are doing during the given time interval. The video data is recorded as an intention to utilize it with machine learning algorithms in future for stress prediction research.

Figure 3.10 shows the ffmpeg command that gets executed by bash shell script through console. The shell script stores the captured audio, video from webcam to a local hard drive.

```
#!/bin/bash
ffmpeg -f v4l2 -video_size 1920x1080 -i /dev/video0 -vf drawtext="fontsize=24:
fontfile=/usr/share/fonts/truetype/liberation/LiberationSans-Regular.ttf
:text='Time %{localtime} Framenumber %{frame_num}  pts %{pts \\: hms}':
x=(w-text_w)-30: y=(h-text_h)-30:box=1:boxcolor=black@0.5:boxborderw=5:
fontcolor=white"
/home/tmc-testi/TTY/user_testing-coding-logged/latest/webcam_output`date '+%x-%X'`.mkv
```

Figure 3.10 *ffmpeg command run through bash script*

3.5.5 GSR and Moodmetric Data collection

One of interest in finding correlation of stress and physiological data in this research is done by measuring galvanic skin response (GSR) data which is obtained from a wearable ring named Moodmetric.

Moodmetric a commercial wearable ring by Vigofere Oy that uses EDA/GSR sensor to measure the skin electric conductance resulting from the human autonomic nervous system. The skin response or electrodermal is used for measuring the mood of a user like activeness or calm or relaxation state. Moodmetric ring co-works with mobile application and is available in IOS and Android store which performs the actual stress calculation after receiving data via Bluetooth from ring and generates the analytics in an interactive graphical presentation. Moodmetric provides non-intrusive way to monitor the mood of user which makes it applicable for wide range of research and understanding HCI for personalized systems [11, 68].

Additionally, Moodmetric company also provides an open source R programmed analytical tool and source code for research purpose. The data can be exported from ring to mobile Secure Digital card (SD card). The actual calculation of mood is done by an application running in the device while the ring only measures the data and transfers the device. The mood calculation is performed on data based on the interval of 60 seconds.

Figure 3.11 shows the Moodmetric ring and an application running on the mobile device and figure 3.12 shows the data sample exported from the ring to SD card which is viewed using SQLite viewer tool. Column ”mm” represents the mood metric value, while other values are used for calculation of mood captured by GSR

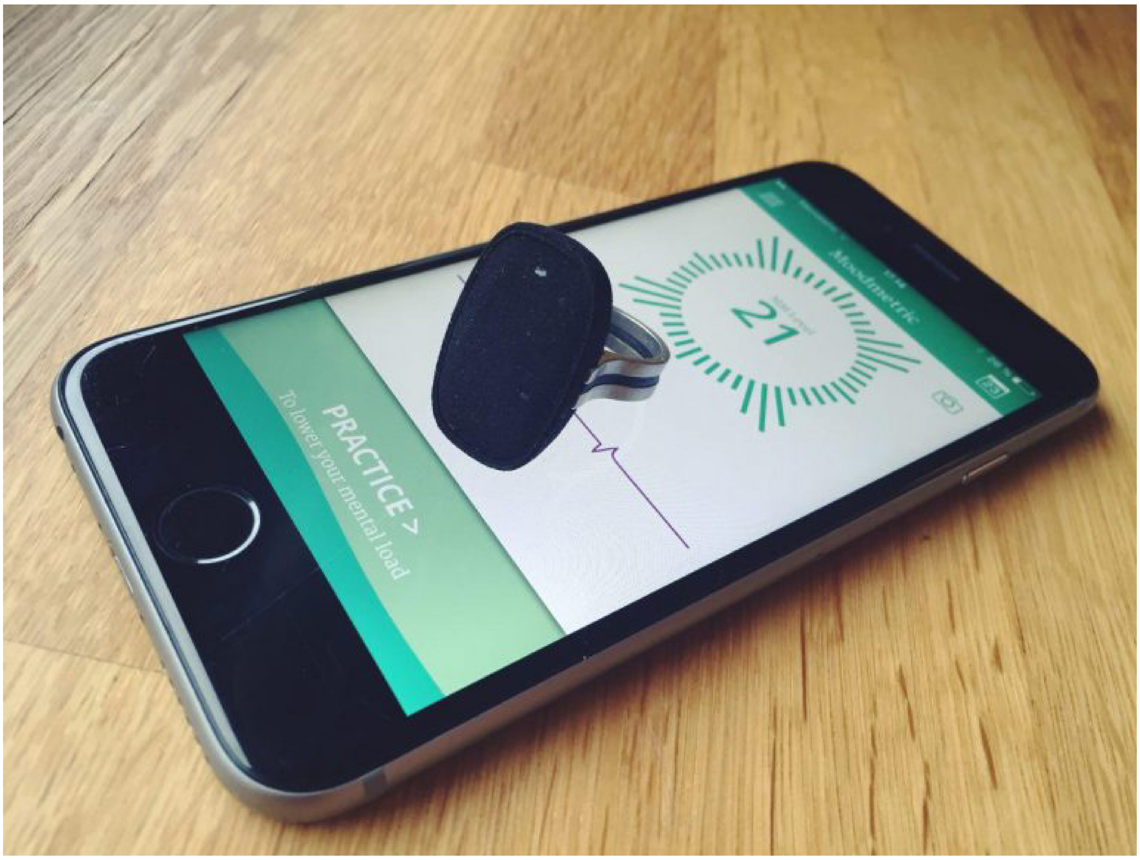


***Figure* 3.11** *Moodmetric ring and mobile application running on Iphone*

id	dt	time	scrn	mm	sci	steps	aa
12	12:27:00 PM	1503404820	12	63	40	14	238
13	12:28:00 PM	1503404880	0	49	40	0	238
14	12:29:00 PM	1503404940	0	33	40	0	238
15	12:30:00 PM	1503405000	1	36	41	0	238
16	12:31:00 PM	1503405060	6	38	37	0	238

***Figure* 3.12** *Moodmetric ring and mobile application running on Iphone*

Sensor.

4. DATA PREPROCESSING

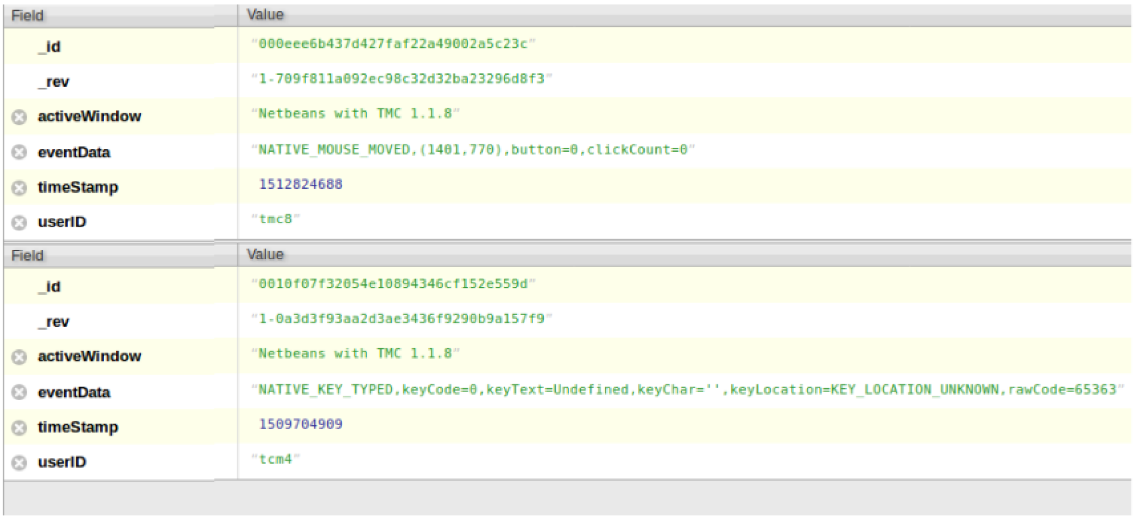
This chapter reviews the preprocessing of captured data. The sections in this chapter explain the data filtering process and the elimination of unnecessary data to reduce the noises in data. Data filtering algorithm for statistical correlation analysis method is also discussed.

4.1 Data storage and retrieval

Huge amount of data is generated by mouse, keyboard and webcam events in every millisecond. So, it is necessary to log every event along with time stamp which helps in understanding the correlation between captured data and time series. Basically, mouse generates a large number of data even when a cursor in screen is moved from a certain point (x,y) to another point(x1, x2). The movement along axis happens so rapidly that it logs hundreds of mouse movement data per every second. More than 25 thousand data was collected from each participant related to mouse and keyboard activity. This describes the necessity that a fast processing database was necessary for the storage. Therefore Couch database was used to store the data which can store and fetch data at very high speed.

Couch database is a scalable multi-platform support flat file database suitable for big data. The software is distributed as an open source software by Apache Foundation which is developer friendly and provides an easily scalable architecture. Unlike relational databases like mysql, posgre, sql etc. Couch database uses Javascript Object Notation (JSON) which can be processed by software that can parse and consume it.

Figure 4.1 shows the JSON data model used by couch database for storage of captured by JNH data.



Field	Value
_id	"000eee6b437d427faf22a49002a5c23c"
_rev	"1-709f811a092ec98c32d32ba23296d8f3"
activeWindow	"Netbeans with TMC 1.1.8"
eventData	"NATIVE_MOUSE_MOVED,(1401,770),button=0,clickCount=0"
timeStamp	1512824688
userID	"tmc8"

Field	Value
_id	"0010f07f32054e10894346cf152e559d"
_rev	"1-0a3d3f93aa2d3ae3436f9290b9a157f9"
activeWindow	"Netbeans with TMC 1.1.8"
eventData	"NATIVE_KEY_TYPED,keyCode=0,keyText=Undefined,keyChar='',keyLocation=KEY_LOCATION_UNKNOWN,rawCode=65363"
timeStamp	1509704909
userID	"tcm4"

Figure 4.1 *Captured data as JSON format in couch database web administration panel*

4.2 Key logger data preprocessing

As mentioned in the previous section, key and mouse logger data is generated rapidly, therefore a fixed interval of 1 minute is taken as a baseline for time series unit. So every keystroke data was grouped based on the same minute interval. Similarly as subjects were allowed to search references and seek help from online materials, data was separated based on the focus of active window used by the subject. For example, data generated during typing in Netbeans IDE versus data generated during the usage of other tools like browsers, other utilities etc.

Result: Grouped data per minute interval.
All_KeyLogs_PerMinute = dict(dict());
while *fetchData* **do**
  hr_Min = getHourMin(fetchData.timeStamp);
  data_in_minute = {"timestamp": fetchData.timestamp,
   "event":fetchData.eventName};
  All_KeyLogs_PerMinutes[hr_Min].update(data_in_minute)
end

Algorithm 1: Grouping data based on one minute interval

Algorithm 1 illustrates the grouping of key logger data based in 1 minute interval. The key events are grouped as a set or array if an hour and minutes obtained from timestamps is same.

Result: Calculation of error keys per minute related to programming in IDE.
while *All_KeyLogs_PerMinute* **do**
  TKEPM = $\sum$(KB $\in$ NBD) + $\sum$(KD $\in$ NBD)
end

Algorithm 2: Total Error correction keys per minute related to IDE.

Result: Total Errors per Task Interval
while *All_KeyLogs_PerTaskInterval* **do**
  TKEPT = $\sum$ (KE $\in$ NBD + KE $\notin$ NBD)
  WHERE (taskStartTime $\leq=$ timestamp $\geq$ taskEndTime)
end

Algorithm 3: Total Errors per Task interval.

Result: Calculation of Keys typed per minute related to programming in IDE.
while *All_KeyLogs_PerMinute* **do**
  TKTPM = $\sum$ (KP $\in$ NBD)
end

Algorithm 4: Total Keys pressed per minute related to programming in IDE

Result: Total Keys per minute (IDE +Non IDE)
while *All_KeyLogs_PerMinute* **do**
  TAKTPM = $\sum$ (KP $\in$ NBD + KP $\notin$ NBD)
end

Algorithm 5: Total Key presses per minute.

Result: Total Key pressed per Task Interval
while *All_KeyLogs_PerTaskInterval* **do**
  TKPT = $\sum$ (KP $\in$ NBD + KP $\notin$ NBD)
  WHERE (taskStartTime $\leq=$ timestamp $\geq$ taskEndTime)
end

Algorithm 6: Total Key presses per Task.

Result: Calculation of Idle Time per Task
while *All_KeyLogs_PerTaskInterval* **do**
  TITPT = $\sum$ (Time_Without_Keyboard_Activity)
end

Algorithm 7: Total Idle time per task

Result: Total key hold-time per minute
while *All_KeyLogs_PerMinute* **do**
  TKLPT = $\sum$ (KR $\in$ NBD + KT $\notin$ NBD)
end

Algorithm 8: Total Key presses per Task.

Algorithm 2 represents the process to extract the total correction keys pressed in every one-minute interval related to programming activity in Netbeans IDE whereas Algorithm 3 is the calculation of total error keys typed during each task. The correction keys are "backspace" and "delete" keys which are represented by KB and KD.

Algorithm 3 represents the key errors per task interval. Unlike Algorithm 2, backspace and delete keys are examined with the time taken per task interval.

Algorithm 4 represents total keys typed per minute in IDE, whereas Algorithm 5 is a representation of total keys pressed per minute without the constraint of either Netbeans IDE or browser. TKPM is an abbreviation for Total keys Pressed Per Minute.

Similarly, Algorithms 5 and 6 represent total keypresses per minute and per task respectively. For calculation of time taken per task, the timestamp of task start and completion is saved automatically by TMC plugin in the server.

Algorithm 7 represents the total idle time the user spends per task without any keyboard activity. The idle time is measured by summing up time when keyboard activity does not happen.

Finally, Algorithm 8 represents the key hold time. In other words, a key is pressed and held for a few seconds. The key event is also called down to down time or press to press time.

4.3 Mouse logger data preprocessing

Mouse dynamics data is also grouped according to the time interval. Mouse dynamics data is also distinguishable as Netbeans or non-Netbeans related data based on active window property which is captured along with the mouse dynamics data.

The mouse data can be categorized based on the events like clicks, duration of movement, the distance of clicks etc. Like keyboard data filtering, mouse data are separated and grouped if they happen in the same minute as shown in Algorithm 9. The hour and minute in timestamp of every event is used as a key to group data.

Algorithm 10 represents the total clicks made in every minute in Netbeans IDE and non-Netbeans application like browser.

Result: Grouped data per minute interval.
All_KeyLogs_PerMinute = dict(dict());
while *fetchData* **do**
  hr_Min = getHourMin(fetchData.timeStamp);
  data_in_minute = {"timestamp": fetchData.timestamp,
   "event":fetchData.eventName};
  All_MouseLogs_PerMinutes[hr_Min].update(data_in_minute)
end

Algorithm 9: Grouping data based on one minute interval

Result: Total Mouse clicks per minutes
while *All_MouseLogs_PerMinute* **do**
  TMC = $\sum$ (MC $\in$ NBD + MC $\notin$ NBD)
end

Algorithm 10: Total mouse button pressed per Minute

4.4 Moodmetric GSR data preprocessing

Another part of data analysis is GSR mood data analysis collected by Moodmetric. The ring has a Bluetooth connectivity feature to transfer data to the mobile devices. However, Moodmetric data was lost due to the technical issues. The Moodmetric ring has problem with data transfer and longer duration power supply. Moodmetric uses Bluetooth technology for data transfer and there is no such plug and play features like USB port for copying data. The main problem with data transfer with Bluetooth was an interruption with device connection and non-compatibility with all kind of mobile devices. The interruption in connection causes data to be erased from the ring and only partial data is transfered to the mobile device. The erasing of data from ring after the closing of connection was designed as a default feature.

Figure 4.2 and 4.3 show the partial data collected in the mobile application. The first participant started the experiment at 14:00 Helsinki time while the data is lost during the time of experiment as shown in Figure 4.2. Figure 4.3 shows loss of data after interruption on Bluetooth connection.

The consecutive failures in data collection from more than 4-5 experiment concluded in a decision to eliminate the usage of Moodmetric ring. Therefore Moodmetric ring was not used with the rest of the participants. This also concluded that the next process to analyze stress and correlation with physiological data would be based on stress related survey data collected during each experiment session.

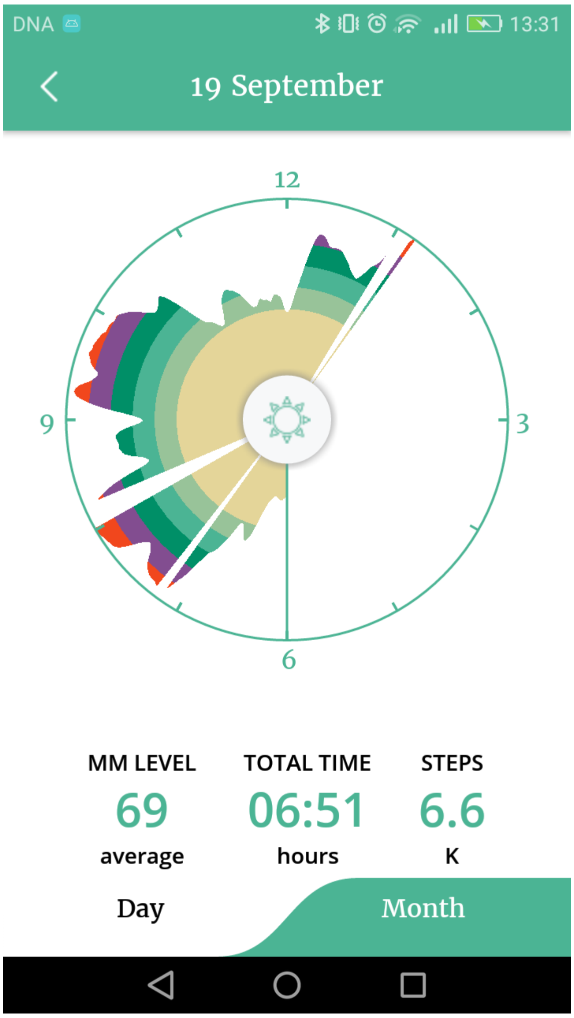


Figure 4.2 *Moodmetric data captured during day time*

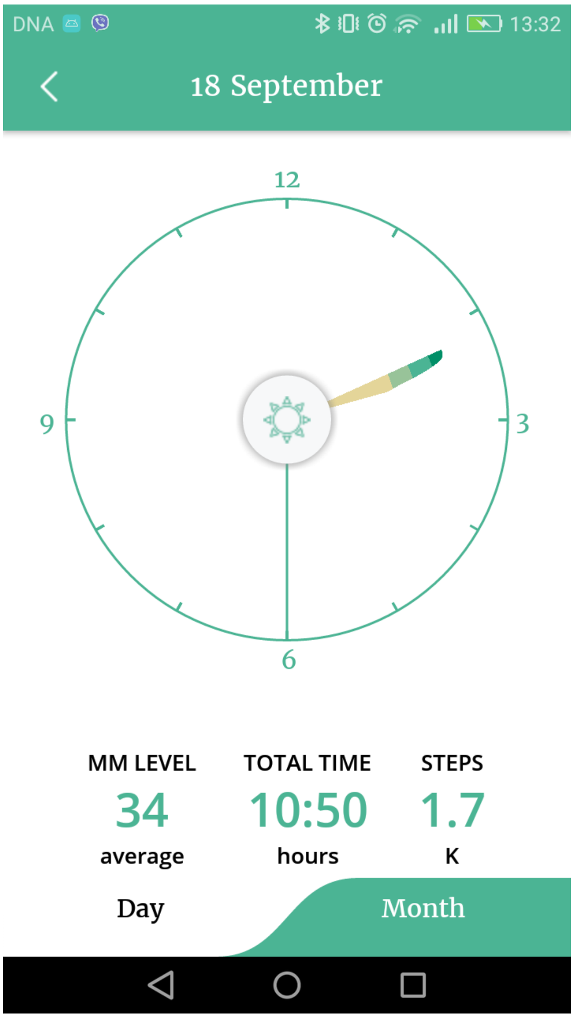


Figure 4.3 *Moodmetric data lost during transfer via Bluetooth to mobile device*

4.5 Survey data preprocessing

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
User	Question	Difficulty level	Mental	Familiarity	Hurried	Successful	Hardship	Stressed	Started Time	Submited time	Time Taken	K -L In minutes	Task Completed	Errors per Task	Netbeans Errors	Reason-givingup
TMC3	1	1	8	1	9	8	2	1	13:29	13:34	0:05	5	1	21	21	
	2	2	2	9	3	9	1	3	13:35	13:51	0:16	16	1	100	63	
	3	3	5	2	4	5	4	5	13:52	14:17	0:25	25	1	152	141	
	4	4	6	4	4	4	6	6	14:18	14:29	0:11	11	0	56	49	Time ende
TMC4	1	1	3	8	1	9	3	2	11:43	11:58	0:15	15	1	69	69	
	2	2	3	8	1	8	4	3	11:59	12:16	0:17	17	1	159	96	
	3	3	8	4	3	2	7	9	12:17	12:43	0:26	26	0	84	84	Time ende
TMC5	1	1	2	0	1	7	0	6	13:54	14:12	0:18	18	1	23	23	
	2	2	0	0	0	8	1	3	14:13	14:19	0:06	6	1	31	23	
	3	3	7	0	2	4	6	4	14:20	14:58	0:38	38	1	128	123	
	4	4	2	0	5	6	1	0	14:59	15:04	0:05	5	1	11	11	
TMC6	1	1	1	5	1	10	3	1	11:30	11:46	0:16	16	1	53	53	
	2	2	1	9	2	10	3	3	11:47	12:02	0:15	15	1	6	2	
	3	3	6	4	4	4	7	6	12:03	12:30	0:27	27	0	55	18	Time ende
TMC7	1	1	1	6	6	9	1	4	13:35	13:55	0:20	20	1	256	213	
	2	2	2	2	4	4	2	5	13:56	14:31	0:35	35	1	338	326	
	3	3	6	3	5	1	7	6	14:32	14:35	0:03	3	0	0	0	
TMC8	1	1	0	0	0	10	0	0	14:28	14:34	0:06	6	1	40	40	
	2	2	0	0	0	10	0	0	14:35	14:39	0:04	4	1	19	0	
	3	3	1	6	0	10	0	0	14:40	14:51	0:11	11	1	84	72	
	4	4	3	3	5	3	8	4	14:52	15:28	0:36	36	0	88	88	Time ende
TMC9	1	1	3	8	5	5	4	2	17:12	17:30	0:18	18	1	23	23	
	2	2	3	9	5	10	4	3	17:31	17:33	0:02	2	1	0	0	
	3	3	8	0	5	10	10	8	17:34	18:09	0:35	35	1	50	49	
	4	4	10	0	7	4	8	8	18:10	18:10	0:00	20	0	0	0	time ende
TMC10	1	1	2	2	3	7	5	1	11:10	11:19	0:09	9	1	67	67	
	2	2	3	2	2	3	4	1	11:20	11:28	0:08	8	1	35	30	
	3	3	5	1	5	5	6	2	11:29	11:46	0:17	17	1	6	2	
	4	4	3	3	3	8	6	3	11:47	11:57	0:10	10	1	48	48	
	5	5	6	5	4	7	6	3	11:58	12:08	0:10	10	1	12	3	
	6	6	6	3	6	7	7	4	12:09	12:10	0:01	1	0	0	0	Time ende
TMC 11	1	1	3	8	2	10	3	6	13:40	13:52	0:12	12	1	55	51	
	2	2	2	2	2	10	4	4	13:53	14:06	0:13	13	1	9	4	
	3	3	6	2	3	3	6	6	14:07	14:40	0:33	33	0	82	63	Time ende

Figure 4.4 *Survey data collected from each participant for each questions attempt*

Figure 4.4 shows the survey data collected from each participant where users, coding test questions and difficulty levels are represented by columns A, B and C respectively. The columns D-I represent NASA TLX survey data and columns J-N represent time taken to complete each task.

Most participants did not complete the last programming task they had attempted which can be seen in column N with a binary value representing either 1 for successful completion or 0 for failure. The reason for not completing is given in column Q. Some participant's data were invalid and was not significant to calculate the correlation as their data did not show any changes. Highlighted rows in column I represents such invalid data.

4.6 Data Analysis with Pearson Correlation Coefficient

Pearson correlation is one of the popular statistical methods to find the linear correlation of variables and their associations. It is also named as bivariate correlation.

The Pearson correlation coefficient is given by following formula -

$$\rho x, y = \frac{\operatorname{cov}(X, Y)}{\sigma_x \sigma_y} \tag{4.1}$$

- cov - covariance of variables
- σ_x - standard deviation of X
- σ_y - standard deviation of Y

Whereas, ρ can be written as

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 (y_i - \overline{y})^2}} \tag{4.2}$$

The Pearson correlation has a value between 1 and -1 where the value less than zero or negative number represents the negative correlation whereas value closer to 1 represents the strong correlation. Figure 4.5 as shows different correlations drawn in a graph.

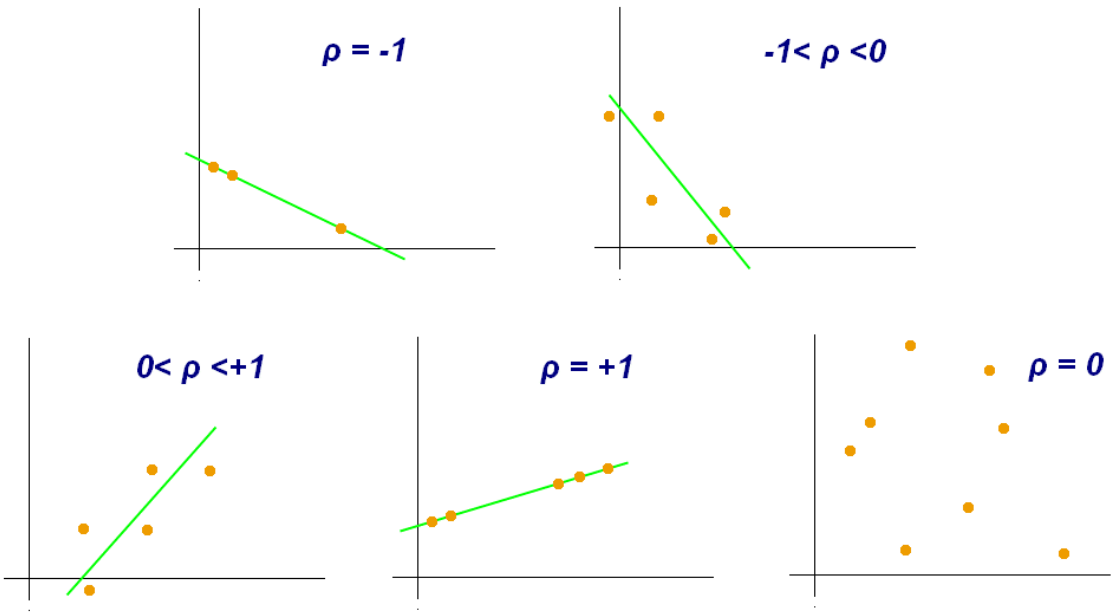


Figure 4.5 *Various Pearson Correlation plots in graph with different values*

As shown in Figure 4.5 the plots 1 and 2 in the first row represents the negative relationship whereas the first two plots on second row on left represents positive correlation and the last plot on the rightmost side of second row represents neutral correlation or no relation.

5. RESULTS AND EVALUATION

This chapter describes the result and interpretation of data analysis conducted using Pearson correlation coefficient approach as discussed in chapter 4. The correlation is examined along with the research questions listed in chapter 1.

5.1 RQ1. Analysis of keystroke dynamics before and after compilation errors

This research question was designed to study the effect of compilation errors on physiological activities like typing behavior, error keys presses etc. The code compilation logs and errors on code functionality are logged in TMC server where preset coding test cases are stored for every question. Every submitted solution is checked against those test cases and compared for the correctness.

However, the log from every participant's submission and code test result did not show enough information to study the related research question. Most participants submitted their code in TMC test server only when they had confirmed that their code functionality works properly in local machine without compiling directly in TMC Server. This also indicates that they were familiar with such coding platform. Only two participant's log showed their attempts when their test cases failed and still they were trying to solve it. Table 5.1 shows the test case result of two participants whose tests failed and was logged in server for their attempts.

As shown in Table 5.1, very few logs were collected in the server related to successful submission or failed submission. The submitted tasks were tested against the pre-set test cases in TMC server.

Table 5.1 *TMC Server test case logs for each participant's code submission*

Submission Time	Participant	Question	Test case result
29.12.2017 14:41	tmc11	Q4	Fail
29.12.2017 14:40	tmc11	Q3	Fail
29.12.2017 14:36	tmc11	Q3	Fail
29.12.2017 14:06	tmc11	Q2	Ok
29.12.2017 14:03	tmc11	Q2	Fail
29.12.2017 13:52	tmc11	Q1	Ok
29.12.2017 13:51	tmc11	Q1	Fail
29.12.2017 13:51	tmc11	Q1	Fail
26.11.2017 14:49	tmc7	Q2	Fail
26.11.2017 14:31	tmc7	Q2	Ok
26.11.2017 14:29	tmc7	Q2	Fail
26.11.2017 14:22	tmc7	Q1	Fail
26.11.2017 13:55	tmc7	Q1	Ok
26.11.2017 13:49	tmc7	Q1	Fail
26.11.2017 13:49	tmc7	Q1	Fail

5.2 Analysis of Moodmetric data with timing parameters

As stated in Research question 2, the goal of measurement of timed data is to evaluate the stress level affected by timing factor like the deadline, timed competition etc. For this purpose, the Moodmetric ring was used to measure the mood of the user in an interval of every second or minute then correlate with physiological data.

However, the technical problem of Moodmetric ring on data transfer and power shortage caused loss of data. Consecutive failures on data collection with Moodmetric ring turned to a decision of eliminating it and rather use the survey data. Figure 5.1 and 5.2 shows the data captured for a participant with time parameter.

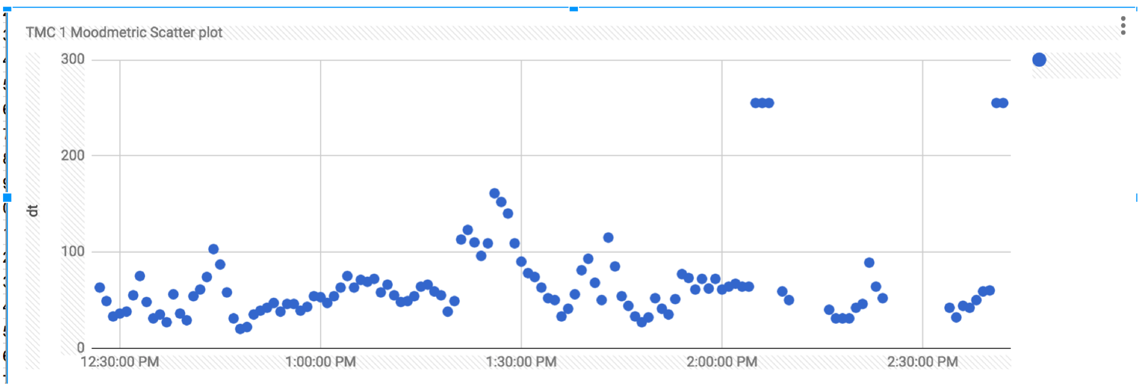


Figure 5.1 *Moodmetric data of participant 1 represented in scatter plot*

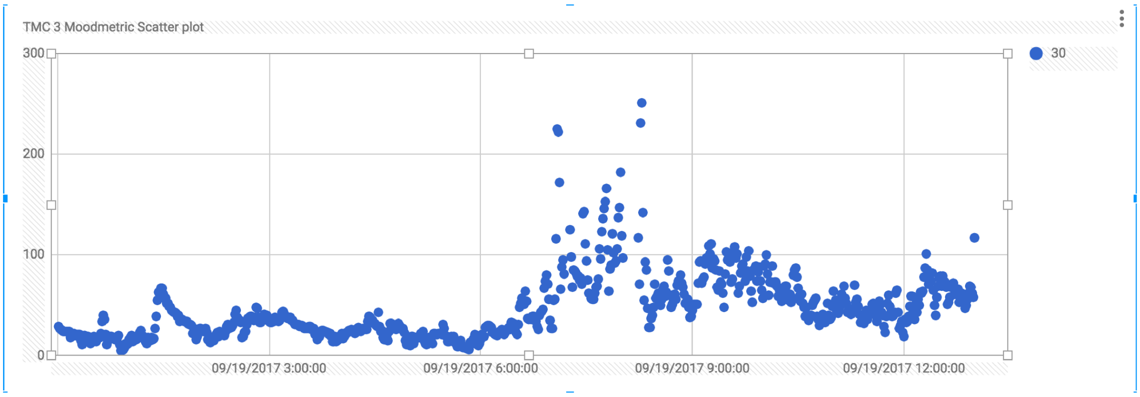


Figure 5.2 *Moodmetric data of participant 3 represented in scatter plot*

As seen in above figures, y-axis represents the mood level where the value of mood is ranged between 0 to 300 whereas the x-axis represents the time of the day. All participants were asked to wear the ring for one day before the experiment session. This was mentioned in Moodmetric catalog instructions that the Moodmetric ring can give more accurate results only after wearing for couple of hours. However, as seen in Figure 5.1, the participant's data of experiment between 2PM to 3:30 PM has several gaps which conclude that the measurement failed in between those gaps. Similarly, in Figure 5.2, the participant started the experiment session at 13:20, however, the data after 13:20 does not occur. This made Moodmetric data collection difficult for the research.

5.3 Analysis of stress from survey data with timing parameters

Since the collection of data from Moodmetric failed, the only other option to analyze the perceived stress depended on using the stress survey data filled up by each participant during the experiment session. The participants were asked to fill up stress related survey questions after completion of each question which would pop up in inside Netbeans IDE supported by TMC plugin.

Figure 5.3 shows the trend line fitted based on the plots of stress data obtained from the survey as shown in column I of Figure 4.4 of page 42. The majority of the trend line shows the increasing pattern as shown in the figure above. The trend line uses logarithmic scales that help to compare the two value's ratios and draw conclusion easily based on the clear visual graph generated on time series data.

However, the result obtained from the graph does not totally guarantee that the stress is rational to the timing effect. As from the survey answers, most of the participants voluntarily participated without real motivation to achieve some goals

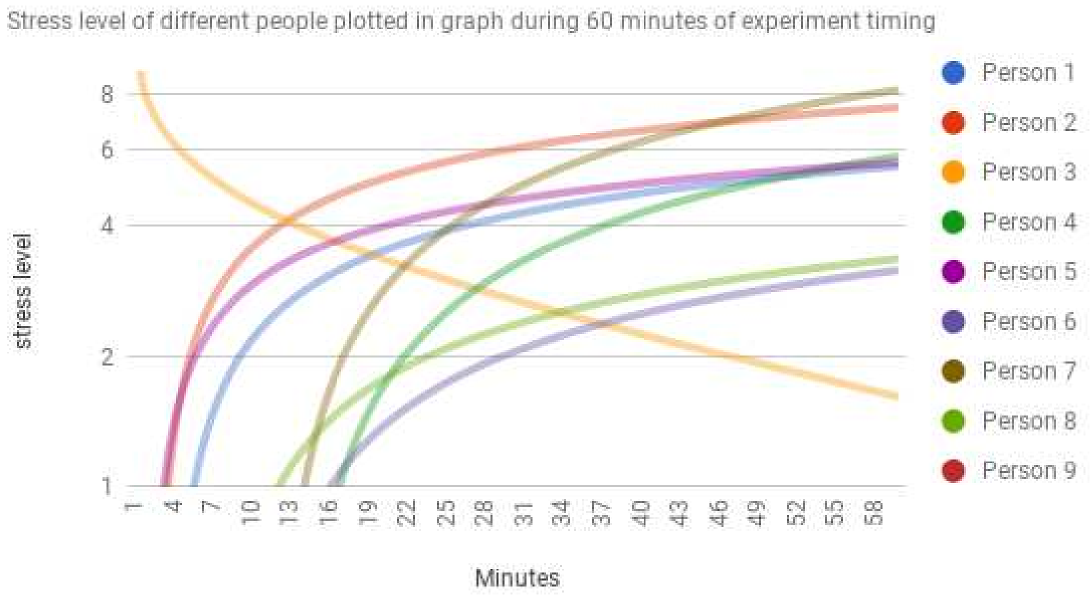


Figure 5.3 *Trend line of plotted stress data during 60 minutes experiment timing*

which may vary if such experiments are conducted with real participants with real goals like students doing exercises to pass, interviewee doing timed coding tests etc.

5.4 RQ3. Analysis of keystroke and mouse dynamics parameters

This section describes the result obtained from the examination of Pearson correlation with mouse and keyboard dynamics data as mentioned in RQ3 in chapter 1, along with studying the data pattern of other parameters like difficulty level, idleness etc.

The stress level of each participant is examined and compared with respect to their data collected from survey against other parameters like total key errors, total pauses, total mouse clicks, total characters typed etc. Since each participant solved different number of tasks and the data obtained is very low, the data would not be feasible for group-based statistical analysis. Therefore, data analysis will also be done by grouping the data of each participant based on high task solving vs low task solving category. Hence, to separate different groups we use the below notation -

- Group H - Participants solving more than 3 tasks.
- Group L - Participants solving first 2 easy tasks and having difficulty on difficult questions.

5.4.1 Group H - keystroke data analysis

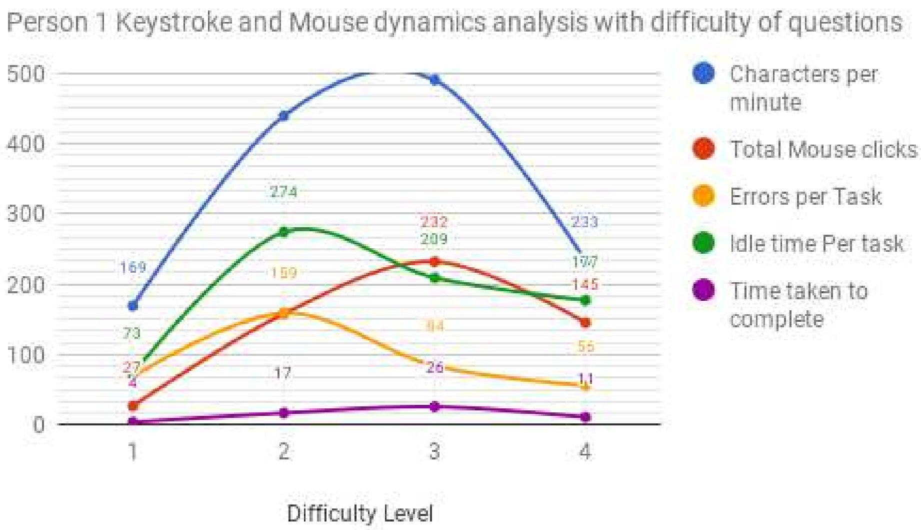
As mentioned in Section 5.4, Group H involves participants who solved more than 3 tasks out of 7 tasks. In other words, they did not have problem with easier tasks as mentioned in Chapter 3.2 but also solved difficult questions. Out of 9 participants, 4 participants were involved in this category. However, one participant had non-usable data for analysis, therefore the data was eliminated.

As shown in Figures 5.4, 5.5, 5.6 each figure contains two subfigures with one plotted in smooth curved graph and the the other in a straight trend line graph. The smooth line graph shows the data plots whereas the trend-line graphs show the pattern how data is changing over another parameter. The x-axis represents the difficulty level whereas other straight lines represents the parameters as labeled on the right side with different colors.

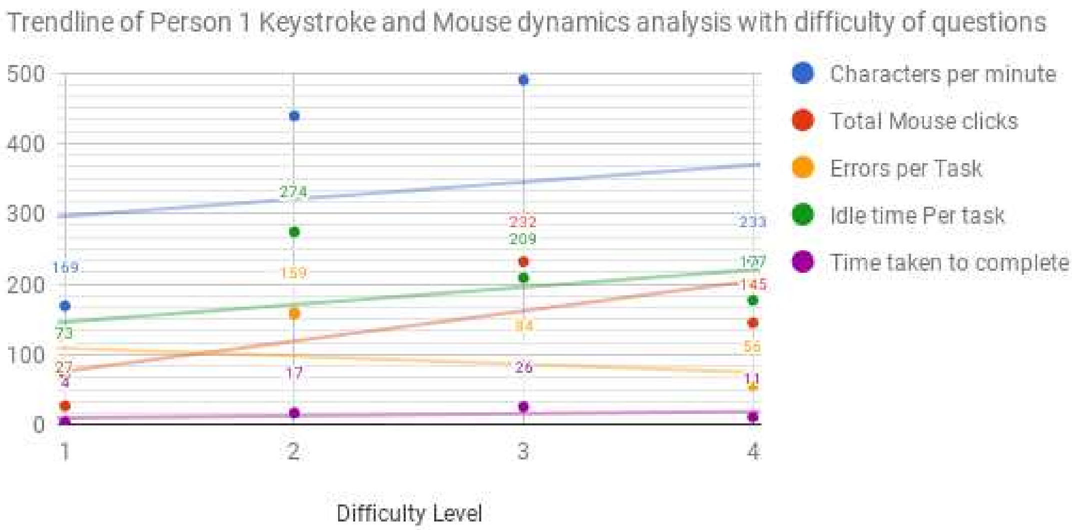
In each figure, the last data plot can be ignored as 8 participants out of 9 did not solve the last task due to end of experiment session time or they did not want to do more tasks.

As seen in the plots of group H, each participants data has different curves. For Participants 1, 3 and 10 the time taken to complete the task, the total errors per task, idleness per task, mouse clicks and the number of characters typed per task is directly proportional to the difficulty of task.

In aggregate, the most common result from all participants suggest that features like mouse clicks, the time taken for completion of task, the number of characters typed increase when there is increased difficulty in task. Later in coming section, these features will be studied along with stress data to compare the correlation.

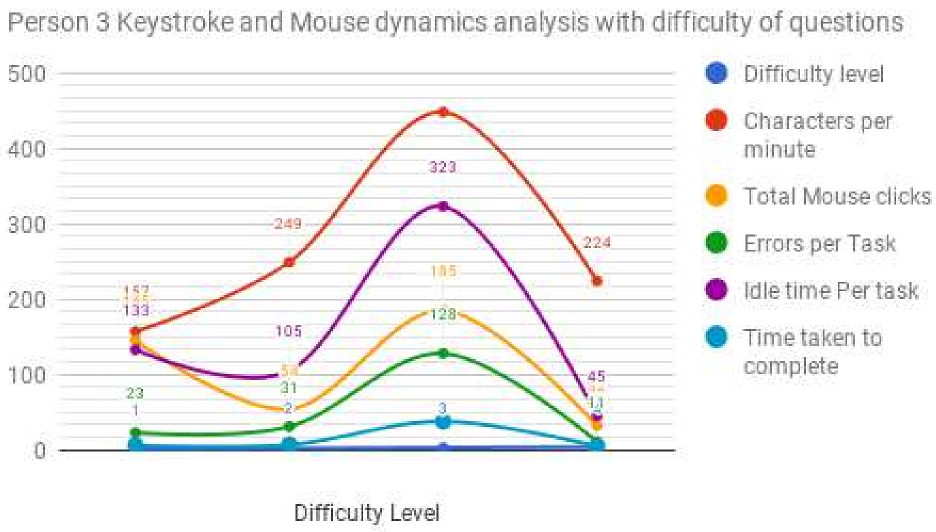


(a) Participant 1 keystroke features comparison with task difficulty

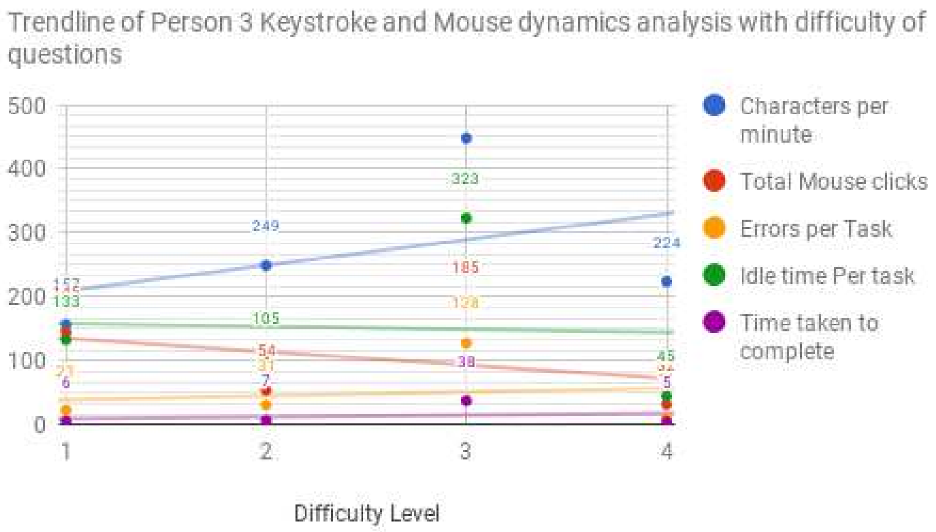


(b) Trendline drawn based on keystroke features of participant 1 data

Figure 5.4 *Participant 1 - Keystroke parameters analysis*

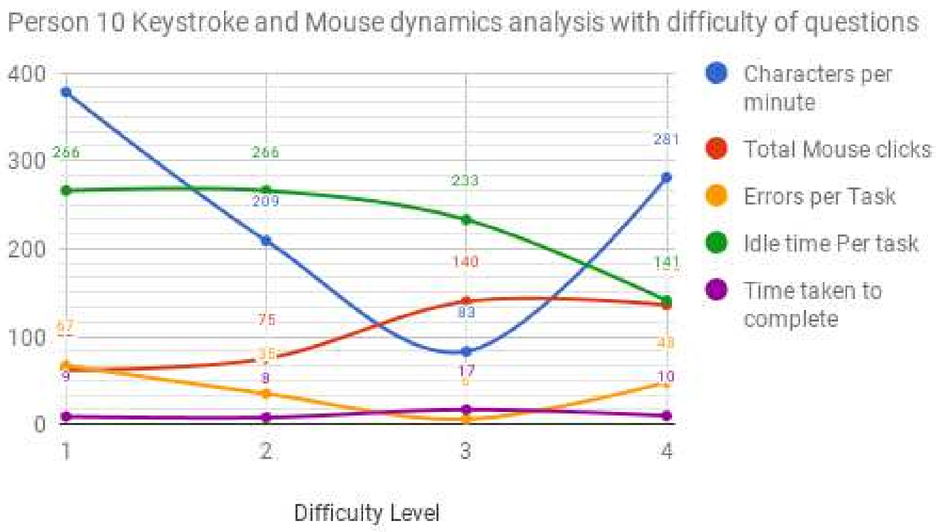


(a) Participant 3 - keystroke features comparison with task difficulty

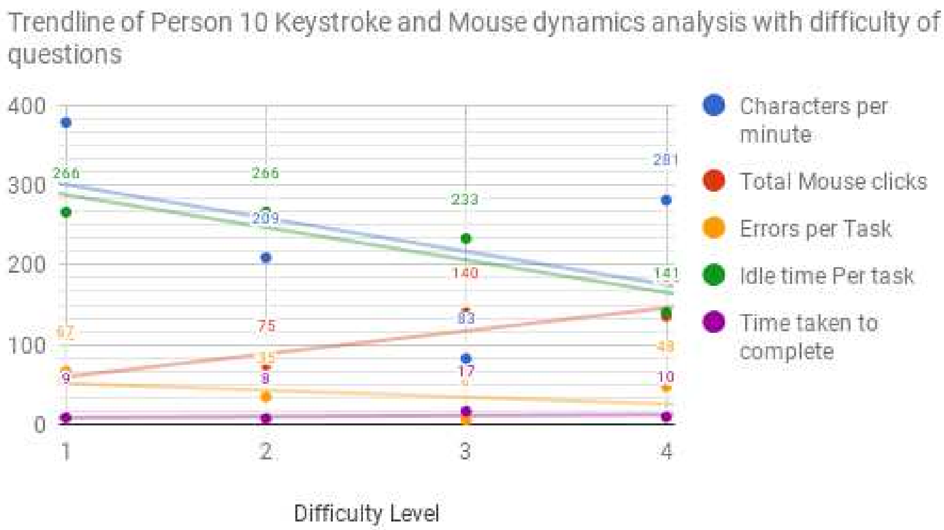


(b) Trendline drawn based on keystroke features of participant 3 data

Figure 5.5 *Participant 3- Keystroke parameters analysis*



(a) Participant 10 - keystroke features comparison with task difficulty



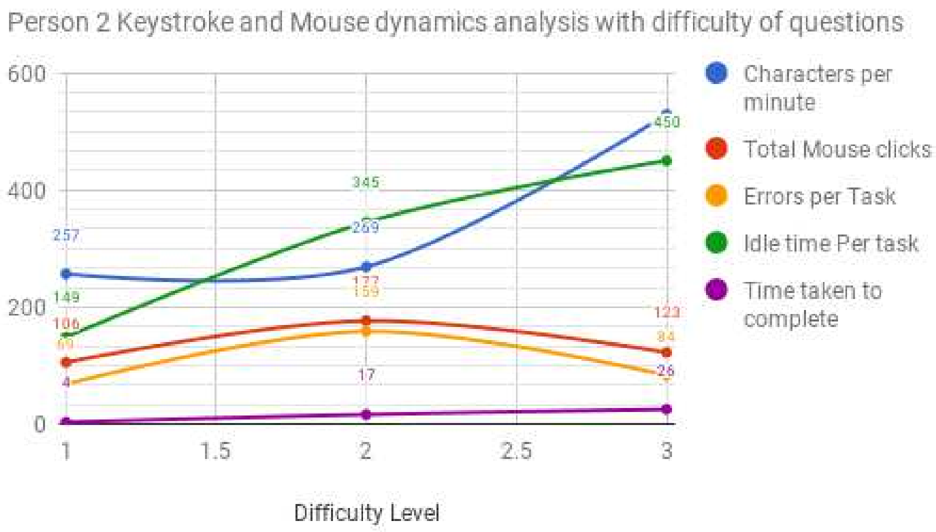
(b) Trendline drawn based on keystroke features of participant 10 data

Figure 5.6 *Participant 10 - Keystroke parameters analysis*

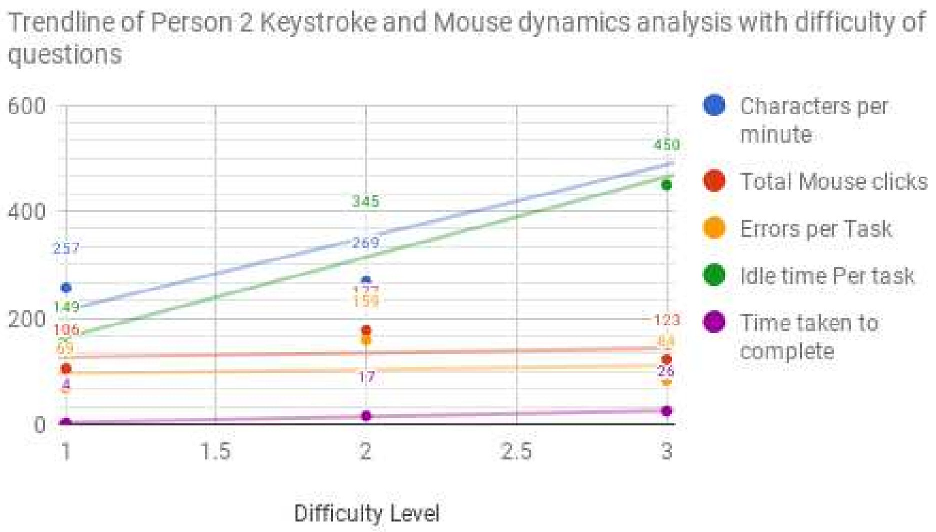
5.4.2 Group L - keystroke data analysis

Figures 5.7, 5.8, 5.9 represents the graph of 2nd, 4th and 11th participant who solved less than 3 tasks. Rest of participant's data were eliminated due to invalidation in data.

As seen in graphs, most of the parameters scale is increasing with the increase in task difficulty. This result suggests that in common, the participants solving lesser tasks have almost all features directly proportional to the level of difficulty.

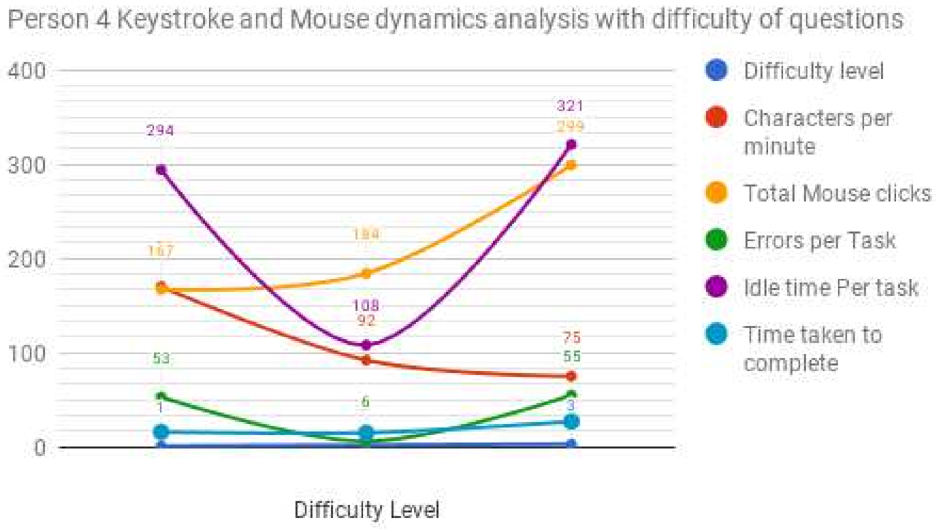


(a) Participant 2 - keystroke features comparison with task difficulty

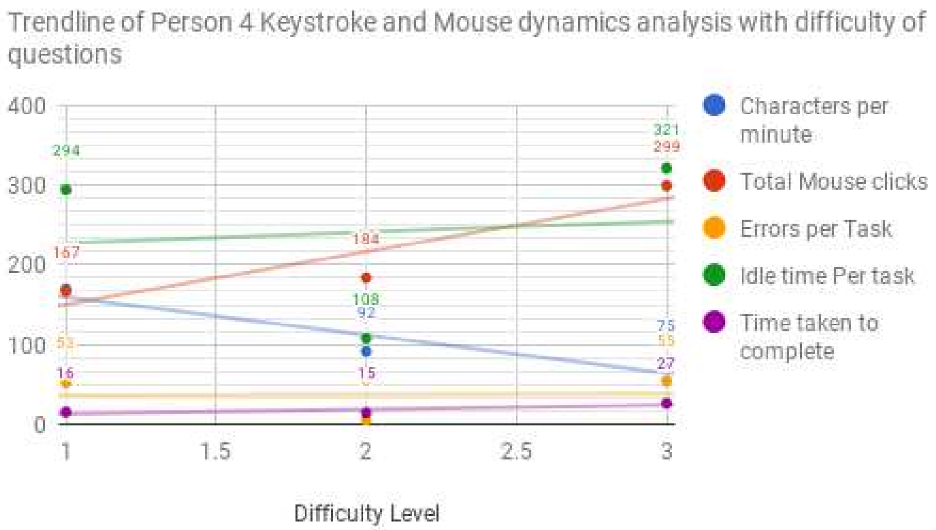


(b) Trendline drawn based on keystroke features of participant 2 data

Figure 5.7 *Participant 2 - Keystroke parameters analysis*

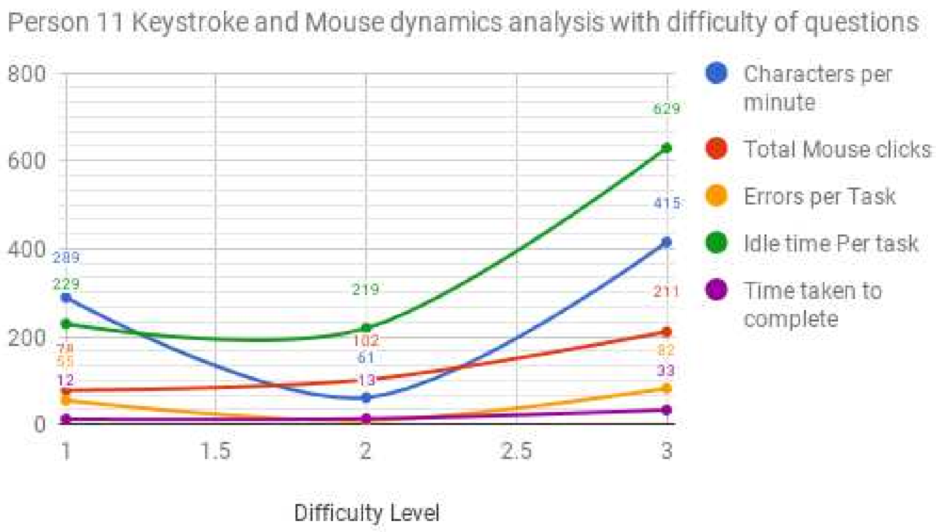


(a) Person4 keystroke features comparison with task difficulty

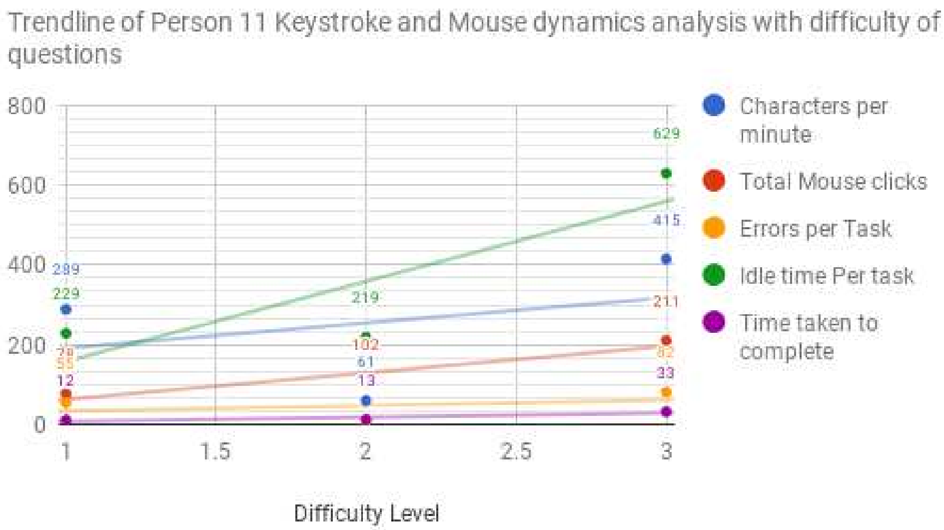


(b) Trendline drawn based on keystroke features of Person 4

Figure 5.8 *Trendline of keystroke pattern of person 4*



(a) Person1 keystroke features comparison with task difficulty



(b) Trendline drawn based on keystroke features of Person 11

Figure 5.9 *Trendline of keystroke pattern of person 11*

5.5 Pearson correlation analysis of stress data with keystroke and mouse dynamics

In this section, we discuss the main important analysis of physiological data and stress using Pearson correlation method. For this analysis, IBM SPSS software is used which supports the Pearson correlation graph and Matrix figure generation that helps to visualize the association of parameters.

5.5.1 Group H - Pearson correlation of stress and other parameters

In this section, the Pearson correlation is examined with the group of participants completing more than 3 tasks successfully. The Pearson correlation for participant 1, 3, 10 is shown in Figures 5.10, 5.11 and 5.12.

The full correlation matrix of various parameters is plotted by SPSS tool by default. As in images, each correlation appears twice. The diagonal columns with Pearson correlation value 1 passing through the mid of table in images represent the correlation to itself. In each column, each column contains three values:

- Pearson correlation - Measurement of Pearson correlation value.
- Significance level - The two-tailed Pearson correlation significance level calculates two-tailed probability. The parameters have significant correlation if their significance level is less than 0.05 otherwise correlation does not hold a significant relationship. significance level closer to 0 means low significance level and closer to 0.05 represents a high significance level.
- Number of sample(N) - Number of samples taken to calculate Pearson correlation.

In this research, main interest is on understanding the correlation of stress parameter with other physiological parameters. As seen in correlation matrix Figures 5.10, 5.11 and 5.12, the 8th column represents the correlation of stress parameter with other 8 rows represented in first column.

In Figure 5.10 the stress has a strong correlation with difficulty level, errors during typing in Netbeans IDE, total mouse clicks etc. where the Pearson value is greater than 0.5. The correlation values in the figure can be seen as 0.99, 0.531 and 0.751 for the task difficulty, total errors generated and total mouse clicks. These values are highlighted in green color. However, these values do not tell the actual relation ship. So we use the two-tailed significance to calculate the significant relationship.

For participant 1 as shown in Figure 5.10, the Pearson correlation significance value is 0.05, however, only the correlation of stress and difficulty parameter shows the significant strong relationship as value is less than significant level 0.05.

In Figure 5.11 for participant 3, the stress parameter has strong relationship with total amount of mouse clicks with value 0.769. There is no significant relationship between stress and other parameters as all two-tailed significance value is greater than 0.05.

Finally, Figure 5.12 shows the Pearson correlation matrix of a 10th participant who solved more than 3 tasks. The Pearson correlation value is stronger with task difficulty, total characters typed and total errors generated during programming whereas negatively correlated with task completeness with value -0.333. There is no significant relationship between stress and other parameters as all correlation have a value greater than the significant level.

Correlations

		difficulty	Completed	CharactersTyped	ErrorsNetBeans	TotalClicks	TotalIdleTime	Stressed	Successfull
difficulty	Pearson Correlation	1	-.775	.201	.407	.651	.380	.990*	-.868
	Sig. (2-tailed)		.225	.799	.593	.349	.620	.010	.132
	N	4	4	4	4	4	4	4	4
Completed	Pearson Correlation	-.775	1	.427	.253	-.035	.050	-.676	.700
	Sig. (2-tailed)	.225		.573	.747	.965	.950	.324	.300
	N	4	4	4	4	4	4	4	4
CharactersTyped	Pearson Correlation	.201	.427	1	.855	.868	.826	.331	.030
	Sig. (2-tailed)	.799	.573		.145	.132	.174	.669	.970
	N	4	4	4	4	4	4	4	4
ErrorsNetBeans	Pearson Correlation	.407	.253	.855	1	.906	.520	.531	-.395
	Sig. (2-tailed)	.593	.747	.145		.094	.480	.469	.605
	N	4	4	4	4	4	4	4	4
TotalClicks	Pearson Correlation	.651	-.035	.868	.906	1	.770	.751	-.454
	Sig. (2-tailed)	.349	.965	.132	.094		.230	.249	.546
	N	4	4	4	4	4	4	4	4
TotalIdleTime	Pearson Correlation	.380	.050	.826	.520	.770	1	.454	.064
	Sig. (2-tailed)	.620	.950	.174	.480	.230		.546	.936
	N	4	4	4	4	4	4	4	4
Stressed	Pearson Correlation	.990*	-.676	.331	.531	.751	.454	1	-.853
	Sig. (2-tailed)	.010	.324	.669	.469	.249	.546		.147
	N	4	4	4	4	4	4	4	4
Successfull	Pearson Correlation	-.868	.700	.030	-.395	-.454	.064	-.853	1
	Sig. (2-tailed)	.132	.300	.970	.605	.546	.936	.147	
	N	4	4	4	4	4	4	4	4

*. Correlation is significant at the 0.05 level (2-tailed).

Figure 5.10 *Person 1 - Pearson correlation of stress and other parameters*

Correlations

		difficulty	Completed	CharactersTyped	ErrorsNetBeans	TotalClicks	TotalIdleTime	Stressed	Successfull
difficulty	Pearson Correlation	1	.[a]	.413	.158	-.369	-.049	-.878	-.529
	Sig. (2-tailed)		.	.587	.842	.631	.951	.122	.471
	N	4	4	4	4	4	4	4	4
Completed	Pearson Correlation	.[a]	.[a]	.[a]	.[a]	.[a]	.[a]	.[a]	.[a]
	Sig. (2-tailed)	.	.	.	.	.	.	.	.
	N	4	4	4	4	4	4	4	4
CharactersTyped	Pearson Correlation	.413	.[a]	1	.937	.517	.854	-.024	-.796
	Sig. (2-tailed)	.587	.		.063	.483	.146	.976	.204
	N	4	4	4	4	4	4	4	4
ErrorsNetBeans	Pearson Correlation	.158	.[a]	.937	1	.784	.978*	.291	-.828
	Sig. (2-tailed)	.842	.	.063		.216	.022	.709	.172
	N	4	4	4	4	4	4	4	4
TotalClicks	Pearson Correlation	-.369	.[a]	.517	.784	1	.881	.769	-.592
	Sig. (2-tailed)	.631	.	.483	.216		.119	.231	.408
	N	4	4	4	4	4	4	4	4
TotalIdleTime	Pearson Correlation	-.049	.[a]	.854	.978*	.881	1	.484	-.739
	Sig. (2-tailed)	.951	.	.146	.022	.119		.516	.261
	N	4	4	4	4	4	4	4	4
Stressed	Pearson Correlation	-.878	.[a]	-.024	.291	.769	.484	1	.059
	Sig. (2-tailed)	.122	.	.976	.709	.231	.516		.941
	N	4	4	4	4	4	4	4	4
Successfull	Pearson Correlation	-.529	.[a]	-.796	-.828	-.592	-.739	.059	1
	Sig. (2-tailed)	.471	.	.204	.172	.408	.261	.941	
	N	4	4	4	4	4	4	4	4

*. Correlation is significant at the 0.05 level (2-tailed).

a. Cannot be computed because at least one of the variables is constant.

Figure 5.11 *Person 3- Pearson correlation of stress and other parameters*

Correlations

		difficulty	Completed	CharactersTyped	ErrorsNetBeans	TotalClicks	TotalIdleTime	Stressed	Successfull
difficulty	Pearson Correlation	1	-.775	-.109	-.111	-.365	-.208	.775	.775
	Sig. (2-tailed)		.225	.891	.889	.635	.792	.225	.225
	N	4	4	4	4	4	4	4	4
Completed	Pearson Correlation	-.775	1	.514	.514	.655	.544	-.333	-1.000**
	Sig. (2-tailed)	.225		.486	.486	.345	.456	.667	.000
	N	4	4	4	4	4	4	4	4
CharactersTyped	Pearson Correlation	-.109	.514	1	1.000**	.963*	.991**	.544	-.514
	Sig. (2-tailed)	.891	.486		.000	.037	.009	.456	.486
	N	4	4	4	4	4	4	4	4
ErrorsNetBeans	Pearson Correlation	-.111	.514	1.000**	1	.964*	.992**	.543	-.514
	Sig. (2-tailed)	.889	.486	.000		.036	.008	.457	.486
	N	4	4	4	4	4	4	4	4
TotalClicks	Pearson Correlation	-.365	.655	.963*	.964*	1	.986*	.306	-.655
	Sig. (2-tailed)	.635	.345	.037	.036		.014	.694	.345
	N	4	4	4	4	4	4	4	4
TotalIdleTime	Pearson Correlation	-.208	.544	.991**	.992**	.986*	1	.456	-.544
	Sig. (2-tailed)	.792	.456	.009	.008	.014		.544	.456
	N	4	4	4	4	4	4	4	4
Stressed	Pearson Correlation	.775	-.333	.544	.543	.306	.456	1	.333
	Sig. (2-tailed)	.225	.667	.456	.457	.694	.544		.667
	N	4	4	4	4	4	4	4	4
Successfull	Pearson Correlation	.775	-1.000**	-.514	-.514	-.655	-.544	.333	1
	Sig. (2-tailed)	.225	.000	.486	.486	.345	.456	.667	
	N	4	4	4	4	4	4	4	4

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

Figure 5.12 *Person 10- Pearson correlation of stress and other parameters*

5.5.2 Group L - Pearson correlation of stress and other parameters

In this section, the Pearson correlation of participants solving less than 3 tasks is analyzed. The Pearson correlation matrices of participants of group L is shown in Figures 5.13, 5.14 and 5.15.

For the participant 2, Pearson matrix represented in Figure 5.13 which shows higher Pearson value on stress parameter with an increase in the difficulty level of questions and total characters typed during solving the task. The total idleness time during programming can be seen as 0.924, 0.996 and 0.845 respectively. The Pearson value between stress and task completeness shows strong negative relationship with the value 0.991. The two-tailed significant correlation shows only valid significance level for stress and success parameter with value of 0.000. However, this concludes the zero correlation which means the increase or decrease on one variable does not affect others.

Similarly, for the participant 4, the Pearson correlation is higher with the increase in the difficulty level, total mouse clicks during programming, idleness time during task and success factor. Pearson value is negative for task completeness and errors occurrence during programming. There is no significant two-tailed value to suggest the significance of correlation.

For participant 11, the Pearson correlation value for stress is strongly correlated for characters typed, error occurrence during programming and amount of idleness time during programming with values 0.936, 0.981 and 0.518. Pearson correlation value with stress and task completeness, success on task submission is strongly negatively related but significance values suggest no significant relationships between those parameters.

Correlations

		difficulty	Completed	CharactersTyped	ErrorsNetBeans	TotalClicks	TotalIdleTime	Stressed	Successfull
difficulty	Pearson Correlation	1	-.866	.885	.554	.229	.985	.924	-.924
	Sig. (2-tailed)		.333	.308	.626	.853	.110	.249	.249
	N	3	3	3	3	3	3	3	3
Completed	Pearson Correlation	-.866	1	-.999*	-.064	.288	-.767	-.991	.991
	Sig. (2-tailed)	.333		.025	.959	.814	.443	.084	.084
	N	3	3	3	3	3	3	3	3
CharactersTyped	Pearson Correlation	.885	-.999*	1	.103	-.251	.792	.996	-.996
	Sig. (2-tailed)	.308	.025		.934	.839	.418	.059	.059
	N	3	3	3	3	3	3	3	3
ErrorsNetBeans	Pearson Correlation	.554	-.064	.103	1	.937	.689	.195	-.195
	Sig. (2-tailed)	.626	.959	.934		.227	.516	.875	.875
	N	3	3	3	3	3	3	3	3
TotalClicks	Pearson Correlation	.229	.288	-.251	.937	1	.393	-.159	.159
	Sig. (2-tailed)	.853	.814	.839	.227		.743	.898	.898
	N	3	3	3	3	3	3	3	3
TotalIdleTime	Pearson Correlation	.985	-.767	.792	.689	.393	1	.845	-.845
	Sig. (2-tailed)	.110	.443	.418	.516	.743		.359	.359
	N	3	3	3	3	3	3	3	3
Stressed	Pearson Correlation	.924	-.991	.996	.195	-.159	.845	1	-1.000**
	Sig. (2-tailed)	.249	.084	.059	.875	.898	.359		.000
	N	3	3	3	3	3	3	3	3
Successfull	Pearson Correlation	-.924	.991	-.996	-.195	.159	-.845	-1.000**	1
	Sig. (2-tailed)	.249	.084	.059	.875	.898	.359	.000	
	N	3	3	3	3	3	3	3	3

*. Correlation is significant at the 0.05 level (2-tailed).

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 5.13 *Person 2 - Pearson correlation of stress and other parameters*

Correlations

		difficulty	Completed	CharactersTyped	ErrorsNetBeans	TotalClicks	TotalIdleTime	Stressed	Successfull
difficulty	Pearson Correlation	1	-.866	-.938	-.671	.919	.116	.866	.982
	Sig. (2-tailed)		.333	.226	.532	.258	.926	.333	.121
	N	3	3	3	3	3	3	3	3
Completed	Pearson Correlation	-.866	1	.638	.210	-.993	-.597	-1.000**	-.945
	Sig. (2-tailed)	.333		.559	.865	.076	.592	.000	.212
	N	3	3	3	3	3	3	3	3
CharactersTyped	Pearson Correlation	-.938	.638	1	.887	-.725	.236	-.638	-.855
	Sig. (2-tailed)	.226	.559		.306	.484	.848	.559	.347
	N	3	3	3	3	3	3	3	3
ErrorsNetBeans	Pearson Correlation	-.671	.210	.887	1	-.325	.658	-.210	-.519
	Sig. (2-tailed)	.532	.865	.306		.790	.542	.865	.653
	N	3	3	3	3	3	3	3	3
TotalClicks	Pearson Correlation	.919	-.993	-.725	-.325	1	.498	.993	.977
	Sig. (2-tailed)	.258	.076	.484	.790		.668	.076	.137
	N	3	3	3	3	3	3	3	3
TotalIdleTime	Pearson Correlation	.116	-.597	.236	.658	.498	1	.597	.302
	Sig. (2-tailed)	.926	.592	.848	.542	.668		.592	.805
	N	3	3	3	3	3	3	3	3
Stressed	Pearson Correlation	.866	-1.000**	-.638	-.210	.993	.597	1	.945
	Sig. (2-tailed)	.333	.000	.559	.865	.076	.592		.212
	N	3	3	3	3	3	3	3	3
Successfull	Pearson Correlation	.982	-.945	-.855	-.519	.977	.302	.945	1
	Sig. (2-tailed)	.121	.212	.347	.653	.137	.805	.212	
	N	3	3	3	3	3	3	3	3

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 5.14 *Person 4- Pearson correlation of stress and other parameters*

Correlations

		difficulty	Completed	CharactersTyped	ErrorsNetBeans	TotalClicks	TotalIdleTime	Stressed	Successfull
difficulty	Pearson Correlation	1	-.866	.351	.192	.938	.855	.000	-.866
	Sig. (2-tailed)		.333	.772	.877	.225	.347	1.000	.333
	N	3	3	3	3	3	3	3	3
Completed	Pearson Correlation	-.866	1	-.772	-.657	-.986	-1.000*	-.500	1.000**
	Sig. (2-tailed)	.333		.438	.543	.108	.014	.667	.000
	N	3	3	3	3	3	3	3	3
CharactersTyped	Pearson Correlation	.351	-.772	1	.986	.654	.786	.936	-.772
	Sig. (2-tailed)	.772	.438		.105	.547	.425	.228	.438
	N	3	3	3	3	3	3	3	3
ErrorsNetBeans	Pearson Correlation	.192	-.657	.986	1	.520	.673	.981	-.657
	Sig. (2-tailed)	.877	.543	.105		.652	.530	.123	.543
	N	3	3	3	3	3	3	3	3
TotalClicks	Pearson Correlation	.938	-.986	.654	.520	1	.982	.346	-.986
	Sig. (2-tailed)	.225	.108	.547	.652		.122	.775	.108
	N	3	3	3	3	3	3	3	3
TotalIdleTime	Pearson Correlation	.855	-1.000*	.786	.673	.982	1	.518	-1.000*
	Sig. (2-tailed)	.347	.014	.425	.530	.122		.653	.014
	N	3	3	3	3	3	3	3	3
Stressed	Pearson Correlation	.000	-.500	.936	.981	.346	.518	1	-.500
	Sig. (2-tailed)	1.000	.667	.228	.123	.775	.653		.667
	N	3	3	3	3	3	3	3	3
Successfull	Pearson Correlation	-.866	1.000**	-.772	-.657	-.986	-1.000*	-.500	1
	Sig. (2-tailed)	.333	.000	.438	.543	.108	.014	.667	
	N	3	3	3	3	3	3	3	3

*. Correlation is significant at the 0.05 level (2-tailed).

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 5.15 *Person 11- Pearson correlation of stress and other parameters*

5.6 Interpretation of results from both groups

In sections 5.4 and 5.5, the data analysis was performed with two groups of participants- H and L. Group H represents the group with participants solving more than 3 tasks and group L represents the participant solving less than 3 tasks. The division of participants into two groups was performed to study the differences in performance and physiological data of successful and struggling participants. The SPSS tool generates the full matrix correlation with significance level. The significance level describes the statical significance correlation between two variables. Fewer data will generate moderate correlation which misleadingly does not reach the significance value while on the other hand the high number of data will generate small correlation that turns out to be out of the significance value but it is also good to report the significance level. [69, 63]. Therefore, Pearson r value will be used to summarize the findings of SPSS Correlation from the tables above as the data set in this research is small in size but also some significance level based result will be discussed.

The following points summarize the findings from the trendline graph and Pearson correlation.

- Positive correlation of stress level with the rate of error, time to complete tasks and mouse clicks: The trend line graph depicts the increase in parameters like error generation, the time taken per task, total clicks as difficulty increases. The same increasing positive correlation can be seen on the Pearson result table highlighted with green colors. The main common positive correlation of physiological data and stress level is seen in error rates, amount of characters typing and idleness on time for the majority of participants in both groups. It can be seen that most of the participants try various methods to solve the task when the difficulty is increased and thus raises parameters like typing rates and seeking help in browsers that increases mouse click and idleness etc. Analysis on time parameters also shows that most of participants takes more time when they struggle to solve the question.

- Weak or negative correlation to task completion and success rate on solving task: As seen in the most of the above Pearson correlation tables, most participants have negative r value in with stress and completion rate plus stress and task success level. This seems obvious that when they are in a stressful situation, this reflects the difficulty in task yielding the negative correlation.

- An irregular pattern in stress level, typing behavior and idleness time for high task solving participants:- The result from typing rate and idleness shows an irregular pattern for high task solving participants. The high number of task solving participant showed less keystrokes for some difficult questions due to the fact they were already familiar with such tasks which required less typing. Participant in L group showed fewer typing behavior when they faced the difficult questions as well as high typing behavior some participant tried solving the difficult task in various possible methods. Most participants were also seen seeking more help from online resources as difficulty increases which increases less typing and more clicks.

- Higher tasks solvers have different stress levels regardless of tasks difficulty:- Most of the participants in group H and L showed an increasing trend when task difficulty is increased in questions. However, the variations were common in stress level of participants solving high number of tasks. The stress level did not show any positive relation to familiarity with such task but fairly correlated positively to the time taken to solve the task. Figure 4.4 illustrates the stress level in number for each participant.

- Unpredictable correlation based on personality and stress perception: The analysis of stress level also explains that stress level rating depends on personality like the familiarity with the task, familiarity with programming techniques, personal behavior etc. In Figure 4.4. the participant, TMC8, did not show any changes in stress level on the attempted tasks which certainly generates the invalid Pearson correlation values. Therefore this value data was not considered during the study. In such cases, sensor based data reading like Heartbeat, mood levels would have been a helpful measurement. This also concludes that the stress perception is based on the personal profile.

- Significance level is inefficient to compute statistical significance:- As seen in all the Pearson co-matrix tables in section 5.5, the amount of data for every participant to compute significant correlation is very low. Therefore repeated session with a large number of data can be used for significance level in order to consider the significance level.

6. CONCLUSIONS

The goal of this thesis is to review the correlation of physiological data and stress measured during programming session. In this thesis, the Pearson correlation method is used to examine the correlation of physiological data and perform analysis to examine the existence of a relationship with stress and physiological data variables.

During the research, only physiological data were collected through existing computer peripherals like keyboard and mouse along with data logged during software usage behavior, webcam video recording etc. As a novel approach to research, the Moodmetric ring was initially planned to be used for collecting the mood of user, however, due to technical problem the Moodmetric ring was eliminated.

After the experiment with 10 samples, a huge number of data was collected which included more than 530,000 rows. However, in case of typing data, there were many noises in data created by modifier keys like ctrl, shift etc. So data filtering was necessary to perform before data analysis. Main features used in data analysis were total errors generated, the idle time of participant without any activity, total keys typed, stress level measured through survey after every task, total mouse clicks, total mouse movements etc.

During the analysis, the most common feature found in participant was increase in stress level along with increase in mistakes per task, time taken to complete the task and idleness in the time taken for task. On the other hand, the stress level and other variables like success and task completion rates have negative relation which is obvious that when stress is perceived the more mistakes are generated and there is less chance to complete the task successfully. It was also clear from most participants data that stress and other parameters like success rate on task completion were negatively correlated, as increase in stress and difficulty would cause less chance in completing task successfully.

The difficulty level is another main factor that mapped the positive correlation with stress level. The higher the difficulty level is increased, the higher the stress increases. As seen in most participant, we can conclude that stress increases with increase in difficulty level.

However, the stress level and Pearson correlation examined in this thesis also concludes that the correlation is biased as there is fewer data collected from non repeated experiment. Especially in case of the participant who solved the higher number of the task a have different pattern of physiological data variables which is irregular despite the increase or decrease in difficulty level. This suggests that stress level also depends on the personality. Likewise, the participant named TMC8 who rated non affected stress level despite the difficulty level increased or decreased and achieved success or either failed in completing task.

In overall conclusion, it can be concluded that some physiological parameters like total typing of characters, total errors generation, time to complete the task can be used with a combination of difficulty level to measure correlation with stress level. Also, this research did not include a significant way to measure the stress and relied only on survey data, therefore there is a biasness to conclude if those stress level are based on participants perception or just intuitive ratings. The other important finding in this research is that every person has stress level based on their profile like how they response and take stress in easy or difficult circumstances. Therefore, different patterns of stress exists on different profiles.

As this research had a limited number of participants, in future it can be improved with a larger number of participants. Experimenting in a repeated session can be performed in order to confirm the validity of data analysis to draw a valid concrete conclusion. Also, repeated data can be used to make a base profile of participant about when they get stressed and how they react in a stressful situation using machine learning algorithms. Additionally, sensors can be used to measure stress level which can prevent the invalid data. Programming languages can be set with multiple compiler options which will let participants use their most familiar language. The keyboard layout also caused a bit issue for some participants which can be in future research can be made eliminated by facilitating multiple keyboard layouts as well as language setting in computer settings.

BIBLIOGRAPHY

[1] A. Alberdi, A. Aztiria, and A. Basarab, "Towards an automatic early stress recognition system for office environments based on multimodal measurements: A review," *Journal of Biomedical Informatics*, vol. 59, no. Supplement C, pp. 49 – 75, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1532046415002750

[2] 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 *Proceedings of the 2Nd ACM SIGHIT International Health Informatics Symposium*, ser. IHI '12. New York, NY, USA: ACM, 2012, pp. 673–678. [Online]. Available: http://doi.acm.org/10.1145/2110363.2110439

[3] B. A. Becker, G. Glanville, R. Iwashima, C. McDonnell, K. Goslin, and C. Mooney, "Effective compiler error message enhancement for novice programming students," *Computer Science Education*, vol. 26, no. 2-3, pp. 148–175, 2016. [Online]. Available: https://doi.org/10.1080/08993408.2016.1225464

[4] R. Bixler and S. D'Mello, "Detecting boredom and engagement during writing with keystroke analysis, task appraisals, and stable traits," in *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, ser. IUI '13. New York, NY, USA: ACM, 2013, pp. 225–234. [Online]. Available: http://doi.acm.org/10.1145/2449396.2449426

[5] N. V. Boulgouris, K. N. Plataniotis, and E. Micheli-Tzanakou, *Multimodal Physiological Biometrics Authentication*. Wiley-IEEE Press, 2010, pp. 461–482. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5396661

[6] C. Busch, "Facing the future of biometrics: Demand for safety and security in the public and private sectors is driving research in this rapidly growing field," *EMBO Rep*, vol. 7, no. Spec No, pp. S23–S25, Jul 2006, 16819444[pmid]. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1490310/

[7] D. Carneiro, P. Novais, J. C. Augusto, and N. Payne, "New methods for stress assessment and monitoring at the workplace," *IEEE Transactions on Affective Computing*, vol. PP, no. 99, pp. 1–1, 2017.

[8] D. Carneiro, P. Novais, J. M. Pêgo, N. Sousa, and J. Neves, *Using Mouse Dynamics to Assess Stress During Online Exams*. Cham: Springer International Publishing, 2015, pp. 345–356. [Online]. Available: https://doi.org/10.1007/978-3-319-19644-2_29

[9] R. Chellappa, J. Phillips, and D. Reynolds, "Special issue on biometrics: Algorithms and applications," *Proceedings of the IEEE*, vol. 94, no. 11, pp. 1912–1914, Nov 2006.

[10] S. Cohen, D. Janicki-Deverts, W. J. Doyle, G. E. Miller, E. Frank, B. S. Rabin, and R. B. Turner, "Chronic stress, glucocorticoid receptor resistance, inflammation, and disease risk," *Proc Natl Acad Sci U S A*, vol. 109, no. 16, pp. 5995–5999, Apr 2012, 201118355[PII]. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3341031/

[11] B. U. Cowley and J. Torniainen, "A short review and primer on electrodermal activity in human computer interaction applications," *CoRR*, vol. abs/1608.06986, 2016. [Online]. Available: http://arxiv.org/abs/1608.06986

[12] C. Epp, M. Lippold, and R. L. Mandryk, "Identifying emotional states using keystroke dynamics," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 715–724. [Online]. Available: http://doi.acm.org/10.1145/1978942.1979046

[13] G. S. Everly and J. M. Lating, *The Anatomy and Physiology of the Human Stress Response*. New York, NY: Springer New York, 2013, pp. 17–51. [Online]. Available: https://doi.org/10.1007/978-1-4614-5538-7_2

[14] D. Fareri and N. Tottenham, "Effects of early life stress on amygdala and striatal development," vol. 19, 04 2016.

[15] D. D. Fehrenbacher and S. Smith, "Behavioural affect and cognitive effects of time-pressure and justification requirement in software acquisition: Evidence from an eye-tracking experiment," in *20th Americas Conference on Information Systems, AMCIS 2014, Savannah, Georgia, USA, August 7-9, 2014*, 2014. [Online]. Available: http://aisel.aisnet.org/amcis2014/DataQuality/GeneralPresentations/1

[16] D. Ford and C. Parnin, "Exploring causes of frustration for software developers," in *Proceedings of the Eighth International Workshop on Cooperative and Human Aspects of Software Engineering*, ser. CHASE '15. Piscataway, NJ, USA: IEEE Press, 2015, pp. 115–116. [Online]. Available: http://dl.acm.org/citation.cfm?id=2819321.2819346

[17] J. B. Freeman and N. Ambady, "Mousetracker: Software for studying real-time mental processing using a computer mouse-tracking method," *Behavior Research Methods*, vol. 42, no. 1, pp. 226–241, Feb 2010. [Online]. Available: https://doi.org/10.3758/BRM.42.1.226

[18] T. Fritz, A. Begel, S. C. Müller, S. Yigit-Elliott, and M. Züger, "Using psycho-physiological measures to assess task difficulty in software development," in *Proceedings of the 36th International Conference on Software Engineering*, ser. ICSE 2014. New York, NY, USA: ACM, 2014, pp. 402–413. [Online]. Available: http://doi.acm.org/10.1145/2568225.2568266

[19] R. Giot, M. El-Abed, and C. Rosenberger, "Keystroke dynamics authentication for collaborative systems," *CoRR*, vol. abs/0911.3304, 2009. [Online]. Available: http://arxiv.org/abs/0911.3304

[20] D. Graziotin, X. Wang, and P. Abrahamsson, "Software developers, moods, emotions, and performance," *IEEE Software*, vol. 31, no. 4, pp. 24–27, July 2014.

[21] S. D. W. Gunawardhane, P. M. D. Silva, D. S. B. Kulathunga, and S. M. K. D. Arunatileka, "Non invasive human stress detection using key stroke dynamics and pattern variations," in *2013 International Conference on Advances in ICT for Emerging Regions (ICTer)*, Dec 2013, pp. 240–247.

[22] S. G. Hart, "Nasa-task load index (nasa-tlx); 20 years later," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 50, no. 9, pp. 904–908, 2006. [Online]. Available: https://doi.org/10.1177/154193120605000909

[23] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Human Mental Workload*, ser. Advances in Psychology, P. A. Hancock and N. Meshkati, Eds. North-Holland, 1988, vol. 52, no. Supplement C, pp. 139 – 183. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166411508623869

[24] E. Hehman, R. M. Stolier, and J. B. Freeman, "Advanced mouse-tracking analytic techniques for enhancing psychological science," *Group Processes & Intergroup Relations*, vol. 18, no. 3, pp. 384–401, 2015. [Online]. Available: http://dx.doi.org/10.1177/1368430214538325

[25] J. Hernandez, P. Paredes, A. Roseway, and M. Czerwinski, "Under pressure: Sensing stress of computer users," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI

'14. New York, NY, USA: ACM, 2014, pp. 51–60. [Online]. Available: http://doi.acm.org/10.1145/2556288.2557165

[26] A. K. Jain, S. S. Arora, K. Cao, L. Best-Rowden, and A. Bhatnagar, "Fingerprint recognition of young children," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 7, pp. 1501–1514, July 2017.

[27] k. P. khushk and A. A. Iqbal, "An overview of leading biometrics technologies used for human identity," in *2005 Student Conference on Engineering Sciences and Technology*, Aug 2005, pp. 1–4.

[28] S. A. Kaplan, V. P. Madden, T. Mijanovich, and E. Purcaro, "The perception of stress and its impact on health in poor communities," *Journal of Community Health*, vol. 38, no. 1, pp. 142–149, Feb 2013. [Online]. Available: https://doi.org/10.1007/s10900-012-9593-5

[29] M. Karnan, M. Akila, and N. Krishnaraj, "Biometric personal authentication using keystroke dynamics: A review," *Applied Soft Computing*, vol. 11, no. 2, pp. 1565 – 1573, 2011, the Impact of Soft Computing for the Progress of Artificial Intelligence. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S156849461000205X

[30] J. Kerstholt, "The effect of time pressure on decision-making behaviour in a dynamic task environment," *Acta Psychologica*, vol. 86, no. 1, pp. 89 – 104, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0001691894900132

[31] K. S. Killourhy, "A scientific understanding of keystroke dynamics," Ph.D. dissertation, Pittsburgh, PA, USA, 2012, aAI3519874.

[32] E. J. Kindt, *An Introduction into the Use of Biometric Technology.* Dordrecht: Springer Netherlands, 2013, pp. 15–85. [Online]. Available: https://doi.org/10.1007/978-94-007-7522-0_2

[33] R. B. Knapp, J. Kim, and E. André, *Physiological Signals and Their Use in Augmenting Emotion Recognition for Human–Machine Interaction.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 133–159. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15184-2_9

[34] E. A. Kochegurova, E. S. Gorokhova, and A. I. Mozgaleva, "Development of the keystroke dynamics recognition system," *Journal of Physics: Conference Series*, vol. 803, no. 1, p. 012073, 2017. [Online]. Available: http://stacks.iop.org/1742-6596/803/i=1/a=012073

[35] C.-S. Koong, T.-I. Yang, and C.-C. Tseng, "A user authentication scheme using physiological and behavioral biometrics for multitouch devices," *ScientificWorldJournal*, vol. 2014, p. 781234, Jul 2014, 25147864[pmid]. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4131464/

[36] A. Koakowska, "A review of emotion recognition methods based on keystroke dynamics and mouse movements," in *2013 6th International Conference on Human System Interactions (HSI)*, June 2013, pp. 548–555.

[37] M. Kuutila, M. V. Mäntylä, M. Claes, and M. Elovainio, "Reviewing literature on time pressure in software engineering and related professions - computer assisted interdisciplinary literature review," *CoRR*, vol. abs/1703.04372, 2017. [Online]. Available: http://arxiv.org/abs/1703.04372

[38] P.-M. Lee, W.-H. Tsui, and T.-C. Hsiao, "The influence of emotion on keyboard typing: an experimental study using visual stimuli," *BioMedical Engineering OnLine*, vol. 13, no. 1, p. 81, Jun 2014. [Online]. Available: http://dx.doi.org/10.1186/1475-925X-13-81

[39] Y. M. Lim, A. Ayesh, and M. Stacey, "Detecting emotional stress during typing task with time pressure," in *2014 Science and Information Conference*, Aug 2014, pp. 329–338.

[40] ——, *The Effects of Typing Demand on Emotional Stress, Mouse and Keystroke Behaviours*. Cham: Springer International Publishing, 2015, pp. 209–225. [Online]. Available: https://doi.org/10.1007/978-3-319-14654-6_13

[41] A. Luneski and P. D. Bamidis, "Towards an emotion specification method: Representing emotional physiological signals," in *Twentieth IEEE International Symposium on Computer-Based Medical Systems (CBMS'07)*, June 2007, pp. 363–370.

[42] M. V. Mäntylä, K. Petersen, T. O. A. Lehtinen, and C. Lassenius, "Time pressure: A controlled experiment of test case development and requirements review," in *Proceedings of the 36th International Conference on Software Engineering*, ser. ICSE 2014. New York, NY, USA: ACM, 2014, pp. 83–94. [Online]. Available: http://doi.acm.org/10.1145/2568225.2568245

[43] D. J. McDuff, J. Hernandez, S. Gontarek, and R. W. Picard, "Cogcam: Contact-free measurement of cognitive stress during computer tasks with a digital camera," pp. 4000–4004, 2016. [Online]. Available: http://doi.acm.org/10.1145/2858036.2858247

[44] B. S. MCEWEN, "Protection and damage from acute and chronic stress: Allostasis and allostatic overload and relevance to the pathophysiology of psychiatric disorders," *Annals of the New York Academy of Sciences*, vol. 1032, no. 1, pp. 1–7, 2004. [Online]. Available: http://dx.doi.org/10.1196/annals.1314.001

[45] F. N. Melanda, H. G. dos Santos, M. R. Urbano, W. O. de Carvalho, A. D. Gonzlez, A. E. Mesas, and S. M. de Andrade, "Poor relationships and physical violence at school are associated with more forms of psychological violence among brazilian teachers: A cross-sectional study," *Journal of Interpersonal Violence*, vol. 0, no. 0, p. 0886260517696857, 0. [Online]. Available: https://doi.org/10.1177/0886260517696857

[46] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Generation Computer Systems*, vol. 16, no. 4, pp. 351 – 359, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X9900059X

[47] S. C. Müller and T. Fritz, "Using (bio)metrics to predict code quality online," in *Proceedings of the 38th International Conference on Software Engineering*, ser. ICSE '16. New York, NY, USA: ACM, 2016, pp. 452–463. [Online]. Available: http://doi.acm.org/10.1145/2884781.2884803

[48] Y. Nakashima, J. Kim, S. Flutura, A. Seiderer, and E. André, *Stress Recognition in Daily Work*. Cham: Springer International Publishing, 2016, pp. 23–33. [Online]. Available: https://doi.org/10.1007/978-3-319-32270-4_3

[49] V. Petreanu, R. Iordache, and M. Seracin, "Assessment of work stress influence on work productivity in romanian companies," *Procedia - Social and Behavioral Sciences*, vol. 92, no. Supplement C, pp. 420 – 425, 2013, logos Universality Mentality Education Novelty (LUMEN 2013), Iasi, Romania, 10-13 April 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877042813028267

[50] P. H. Pisani and A. C. Lorena, "A systematic review on keystroke dynamics," *Journal of the Brazilian Computer Society*, vol. 19, no. 4, pp. 573–587, Nov 2013. [Online]. Available: https://doi.org/10.1007/s13173-013-0117-7

[51] M. Pleva, P. Bours, S. Ondáš, and J. Juhár, "Improving static audio keystroke analysis by score fusion of acoustic and timing data," *Multimedia Tools and Applications*, Mar 2017. [Online]. Available: https://doi.org/10.1007/s11042-017-4571-7

[52] M. Rabe, S. Giacomuzzi, and M. Nübling, "Psychosocial workload and stress in the workers' representative," *BMC Public Health*, vol. 12, no. 1, p. 909, Oct 2012. [Online]. Available: https://doi.org/10.1186/1471-2458-12-909

[53] J. Rabkin and E. Struening, "Live events, stress, and illness," *Science*, vol. 194, no. 4269, pp. 1013–1020, 1976. [Online]. Available: http://science.sciencemag.org/content/194/4269/1013

[54] P. Rea, "Chapter 1 - introduction to the nervous system," in *Essential Clinical Anatomy of the Nervous System*, P. Rea, Ed. San Diego: Academic Press, 2015, pp. 1 – 50. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128020302000017

[55] M. Rodrigues, S. Gonçalves, D. Carneiro, P. Novais, and F. Fdez-Riverola, *Keystrokes and Clicks: Measuring Stress on E-learning Students.* Heidelberg: Springer International Publishing, 2013, pp. 119–126. [Online]. Available: https://doi.org/10.1007/978-3-319-00569-0_15

[56] J. Roth, X. Liu, A. Ross, and D. N. Metaxas, "Biometric authentication via keystroke sound," in *ICB*, 2013.

[57] K. Saeed, *Biometrics Principles and Important Concerns.* New York, NY: Springer New York, 2012, pp. 3–20. [Online]. Available: https://doi.org/10.1007/978-1-4614-5608-7_1

[58] W. Sanchez, A. Martinez, and M. Gonzalez, *Towards Job Stress Recognition Based on Behavior and Physiological Features.* Cham: Springer International Publishing, 2017, pp. 311–322. [Online]. Available: https://doi.org/10.1007/978-3-319-67585-5_33

[59] N. Schneiderman, G. Ironson, and S. Siegel, "Stress and health: Psychological, behavioral, and biological determinants," *Annual Review of Clinical Psychology*, vol. 1, pp. 607–628, 12 2005.

[60] N. Schneiderman, G. Ironson, and S. D. Siegel, "Stress and health: Psychological, behavioral, and biological determinants," *Annu Rev Clin Psychol*, vol. 1, pp. 607–628, 2005, 17716101[pmid]. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2568977/

[61] T. K. Setor, "The effect of job stress on job performance amongst it professionals: The moderating role of proactive work behaviours," in *Proceedings of the 52Nd ACM Conference on Computers and People Research*, ser. SIGSIM-CPR '14. New York, NY, USA: ACM, 2014, pp. 17–21. [Online]. Available: http://doi.acm.org/10.1145/2599990.2599993

[62] N. Sharma and T. Gedeon, *Optimal Time Segments for Stress Detection*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 421–433. [Online]. Available: https://doi.org/10.1007/978-3-642-39712-7_32

[63] S. F. students, "Spss help for students - how do i interpret data in spss for pearson's r and scatterplots?" *SPSS*. [Online]. Available: https://www.spss-tutorials.com/spss-correlation-analysis/

[64] D. Sun, P. Paredes, and J. Canny, "Moustress: Detecting stress from mouse motion," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14. New York, NY, USA: ACM, 2014, pp. 61–70. [Online]. Available: http://doi.acm.org/10.1145/2556288.2557243

[65] J. Tearne, M. Robinson, P. Jacoby, K. Allen, N. Cunningham, J. Li, and N. Mclean, "Older maternal age is associated with depression, anxiety, and stress symptoms in young adult female offspring," vol. 125, 11 2015.

[66] P. S. Teh, A. B. J. Teoh, and S. Yue, "A survey of keystroke dynamics biometrics," *ScientificWorldJournal*, vol. 2013, p. 408280, Nov 2013, 24298216[pmid]. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3835878/

[67] D. Toll, T. Olsson, M. Ericsson, and A. Wingkvist, "Fine-grained recording of student programming sessions to improve teaching and time estimations," *International Journal of Engineering ,Science and Innovative Technology*, vol. 32, no. 3, pp. 1069–1077, 2016.

[68] J. Torniainen, B. Cowley, A. Henelius, K. Lukander, and S. Pakarinen, "Feasibility of an electrodermal activity ring prototype as a research tool," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Aug 2015, pp. 6433–6436.

[69] S. Tutorials, "Spss correlation analyis tutorial," *SPSS*. [Online]. Available: https://www.spss-tutorials.com/spss-correlation-analysis/

[70] M. Ulinskas, M. Woźniak, and R. Damaševičius, *Analysis of Keystroke Dynamics for Fatigue Recognition*. Cham: Springer International Publishing, 2017, pp. 235–247. [Online]. Available: https://doi.org/10.1007/978-3-319-62404-4_18

[71] L. M. Vizer, L. Zhou, and A. Sears, "Automated stress detection using keystroke and linguistic features: An exploratory study," *International Journal of Human-Computer Studies*, vol. 67, no. 10, pp. 870 – 886, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1071581909000937

[72] P. Wasnik and A. Jeyakumar, “Monitoring stress level parameters of frequent computer users,” in *2016 International Conference on Communication and Signal Processing (ICCSP)*, April 2016, pp. 1753–1757.

[73] C. Wood and R. Raj, “Keyloggers in cybersecurity education,” in *Security and Management*, 2010.

[74] S. Xefteris, V. Andronikou, K. Tserpes, and T. Varvarigou, “Case-based approach using behavioural biometrics aimed at assisted living,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 2, no. 2, pp. 73–80, Jun 2011. [Online]. Available: https://doi.org/10.1007/s12652-010-0029-8

[75] Y. Xu and S. Cao, “Design and implementation of a multi video transcoding queue based on mysql and ffmpeg,” in *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Sept 2015, pp. 629–632.

[76] Y. Zhong and Y. Deng, “A survey on keystroke dynamics biometrics: Approaches, advances, and evaluations,” in *Recent Advances in User Authentication Using Keystroke Dynamics Biometrics*. Science Gate Publishing, 2015, vol. 2, pp. 1–22.