COPPE
UFRJ
Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia

INTELLIGENT TUTORING APPLIED TO VIDEOLECTURES: A SUPPORTING TOOL FOR EXPERIMENTATION AND DATA COLLECTION

Thothadri Rajesh Vidyasankar

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientadores: Rosa Maria Meri Leão
Edmundo Albuquerque de Souza e Silva

Rio de Janeiro
Novembro de 2013

INTELLIGENT TUTORING APPLIED TO VIDEOLECTURES: A SUPPORTING TOOL FOR EXPERIMENTATION AND DATA COLLECTION

Thothadri Rajesh Vidyasankar

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Rosa Maria Meri Leão, Dr.

Dr. Fernando Jorge Silveira Filho, Dr.

Prof. Adilson Elias Xavier, Dr.

RIO DE JANEIRO, RJ – BRASIL
NOVEMBRO DE 2013

Vidyasankar, Thothadri Rajesh

Intelligent Tutoring applied to Videolectures: a supporting tool for experimentation and data collection/Thothadri Rajesh Vidyasankar. – Rio de Janeiro: UFRJ/COPPE, 2013.

XI, 65 p.: il.; $29,7$cm.

Orientadores: Rosa Maria Meri Leão
Edmundo Albuquerque de Souza e Silva

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2013.

Bibliography: p. 61 – 65.

1. Intelligent Tutoring System. 2. Cognitive-Emotional States. 3. Video lectures. 4. Data visualization and analysis. I. Leão, Rosa Maria Meri *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Be the change that you wish to see in the world - M.K Gandhi

Agradecimentos

Primeiramente, gostaria de agradecer a minha família pelo incentivo ao meu contínuo aperfeiçoamento profissional e pessoal. Também por compreenderem minha ausência e se fazerem presentes mesmo estando do outro lado do planeta. Agradeço a minha namorada Amanda que tem um papel muito importante na minha vida e que me apoiou durante a realização do mestrado, sendo paciente com meus dias de tensão e noites em claro e sendo companheira em finais de semana de estudo.

Agradeço aos professores Edmundo e Rosa pela grande ajuda e incentivo na realização deste trabalho. Em especial, ao professor Edmundo por aceitar ser meu orientador quando esteve na India. Durante todo o período que passei no LAND aprendi muito com ambos. Não posso deixar de mencionar o Fernando. Não seria nenhum exagero dizer que, sem a ajuda dele, esta dissertação não existiria na forma ou no conteúdo que tem hoje.

Agradeço imensamente a minha amiga Alejandra que foi crucial para minha adaptação no início do mestrado. Eu não falava português, não conhecia a cidade e foi ela quem, com muita paciência e sem esperar nada em troca, me ajudou. Serei eternamente grato.

Agradeço também ao Gaspare que também fez contribuições valiosas, tanto na implementação dos experimentos realizados, quanto para a construção do conhecimento através de discussões sobre o tema de pesquisa. Finalmente, agradeço a todos os amigos do LAND, por formarem uma equipe tão unida e maravilhosa, sempre dispostos a ajudar. Especialmente a Carol por me fazer sentir ”em casa”.

Este trabalho não poderia ser realizado sem o suporte financeiro (bolsa) da CAPES. Meus agradecimentos vão também para o CEDERJ por criar um ambiente adequado ao desenvolvimento e a pesquisa de novas técnicas computacionais voltadas para educação a distância. Minha experiência como tutor do curso de Tecnologia em Sistemas de Computação do CEDERJ foi de grande valia para o desenvolvimento desta dissertação. Agredeço também a GOOGLE-Brasil pelo financiamento do projeto onde meu trabalho está inserido.

Finalmente, agradeço a Academia Brasileira de Ciências que, através do projeto de cooperação com a India, abriu a possibilidade de estudar no Brasil, e a COPPE, que propociona um ambiente altamente qualificado à pesquisa.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

SISTEMA DE TUTORIA INTELIGENTE APLICADO A VIDEOAULAS: UMA FERRAMENTA DE AUXÍLIO À EXPERIMENTAÇÃO E COLETA DE DADOS

Thothadri Rajesh Vidyasankar

Novembro/2013

Orientadores: Rosa Maria Meri Leão
Edmundo Albuquerque de Souza e Silva

Programa: Engenharia de Sistemas e Computação

Esta dissertação teve como objetivo construir uma ferramenta que possibilitasse não somente a visualização, mas também a análise de informações de múltiplas fontes em uma interface de modo a identificar e remover possíveis ruídos nos dados, estabelecer a correlação entre dados de diferentes fontes, detecção de eventos relevantes presentes nesses dados. O objetivo dessa ferramenta é auxiliar os sistemas de tutoriais online como o Centro de Educação a Distância do Estado do Rio de Janeiro (CEDERJ) a construir um modelo preditivo que utilize emoções identificadas, através de sinais psicológicos, tornando possível adaptar o sistema a partir dessas emoções. Como parte dessa pesquisa, foram conduzidos experimentos com usuários utilizando sensores tais como: eletroencefalograma, electro-dermal activity meter, webcam comum e webcam com infra-vermelho, enquanto os usuários assistiam as video-aulas do CEDERJ. Demostrou-se de que maneira essa ferramenta de visualização pode ser utilizada para visualização e análise dos dados coletados através dos experimentos realizados afim de obter insights adicionais para os dados do sensor.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

INTELLIGENT TUTORING APPLIED TO VIDEOLECTURES: A SUPPORTING TOOL FOR EXPERIMENTATION AND DATA COLLECTION

Thothadri Rajesh Vidyasankar

November/2013

Advisors: Rosa Maria Meri Leão
Edmundo Albuquerque de Souza e Silva

Department: Systems Engineering and Computer Science

This dissertation is part of an ongoing project on Intelligent Tutoring Systems. We built a data visualization tool to help analysing data from multiple sources in order to identify important trends in the data, find possible correlations among data from different sources, detect and remove noise from the data and detect important events. Our tool should be viewed as a first step towards building an automated tutoring systems based on video lectures and, more specifically, target at the CEDERJ videolectures. The long term is to build a predictive model of cognitive states [eg:attention] from data produced by different sensors. As a part of this research, we conducted experiments that employ sensors such as Electroencephalogram, Electrodermal Activity Meter (EDA), normal webcam and infra-red webcam attached to a student while watching a video lecture. We exemplify how our data visualization tool could be used to analyse the data collected from the experiments to gain insights that can be useful for constructing an intelligent tutoring systems and to give feedback to the lecturer to improve his videolecture.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

In the recent years, distance learning technologies have played a significant role in bringing quality education to students of various economic backgrounds. With the help of recent technological developments, the geographical and political constraints with respect to education are slowly diminishing. Distance learning technologies have enabled students across the globe to gain access to lot of free online materials of various renowned universities like MIT and Stanford with an Internet connection and a PC [9]. In Brazil the CEDERJ [11] consortium in the state of Rio de Janeiro pioneered online education using videolectures and formal courses in computer science have been offered for more than ten years [10].

Distance learning can be either synchronous or asynchronous [40]. The synchronous methods are dynamic and allows for real time interaction between the students and teachers through video conferencing, telephone and instant messaging. A clear drawback is that all parts involved are required to be available during a pre-established schedule. The asynchronous method is designed for students and teachers from different time zones, working on different time schedules, where students can access study materials on flexible schedules at their own pace [6]. The interaction is mainly through web portals, email, pre-recorded video lectures and recorded audio instructions.

The popularity of distance learning technologies is significantly increasing due to its benefits. COURSERA, started in 2012, has more than 1.3 million enrolments around the world for its free online courses [50]. In Brazil the computer system course offered by CEDERJ [10], for instance, has more than 1500 students registered in the state of Rio de Janeiro. Presently, the videolectures are made freely available via the National Research and Educational Network (RNP) via the service ”videoaula@RNP”. In the past few months the videolectures have has approximately

6,000 accesses per day.

Although distance learning has been evolving over the recent years, there is still a lot of room for improvements with the advent of new technologies. Many efforts are under study to increase the efficiency and quality of distance learning in order to improve the students learning experience. For instance, in current systems, it is hard to ensure that the concepts conveyed through pre-recorded video lectures are properly understood by the audience. Video lecture systems lack an efficient interactive-feedback real-time channel for student to ask questions and student-lecturer contact in real time that would allow the teacher to infer if the student is properly absorbing the material being taught. In addition, there could be situations where a student may find it difficult to assess their level of understanding and may require assistance, even if she/he is not aware that assistance is needed.

In traditional educational approach, where there is room for interaction between the students and the professors, it is easier for the professor by virtue of his/her experience and skills to assess the student's learning progress. The student learning progress can be evaluated either by identifying their cognitive-emotional states [16, 36] or by evaluating their response to questions [49]. This immediate feedback allows the professor to modify his lecture on the fly, which may have a positive impact in improving students comprehension [36]. Overall, students have the opportunity to enhance their understanding by directly interacting with the professor and their peers.

To incorporate some of these features in distance learning, a real-time multimedia video conferencing between the students and the professors may be necessary. However, such solution requires all students and the professor to log in at the same time, which makes scalability a serious issue. Also, the flexibility of self paced learning with respect to each student would have to be compromised.

An interesting approach to address this issue would be to build a automated tutor tool which could partially emulate the role of the professor, assess the student learning progress during a video lecture and dynamically modify the lecture accordingly. For example, during a lecture, if the student is confused with an equation, the computer tutor could recommend some web links to the user, where the explanation is detailed or more examples are provided.

To assess the student learning progress, the automated tutor tool should mimic some of the capabilities of skilled professors. Most importantly, it should have the ability to recognize students cognitive and emotional states which are the major components of learning [3]. One possibility is for the tutoring tool to ask the students to self report their emotions. This is very intrusive and might distract the student learning process. In addition, self-reported emotions might not be completely reliable, as students may not be aware of their emotional state or may not choose to reveal their emotion [46].

To assess student's cognitive-emotional states in an non-intrusive manner, sensors such as facial expressions, EEG, skin conductance meter etc could be used. Woolf, D'Mello and Picard in their intelligent tutoring systems [20, 21], use data from the mouse pressure, a facial expression system, a posture analysis seat and skin conductance sensor to detect cognitive-emotional states. Multiple sensors are used in order to identify cognitive-emotional states precisely . EEG based technology is an emerging field of research in this arena.

To develop a computer-based tutor with aforesaid sensing devices is very complex and challenging. These sensors produce huge amount of data in various formats. In addition, the data from sensors contains noise from different sources. This makes it difficult to access all the data and analyze them. But, in order to identify the cognitive-emotion states accurately, it is important to study and extract a set of important features related to cognitive-emotion states from the data. Once extracted, a proper machine learning type of model could be employed to infer the cognitive-emotion states accurately.

To visualize and analyze such sensor data, exists various tools such as ACUME, EEGLAB, ELAN [2, 18, 39] etc. ACUME visualizes and analyzes facial expressions, while EEGLAB and ELAN visualizes and analyzes data from EEG sensors. For Intelligent Tutoring Systems which uses multiple sensors, detecting cognitive-emotional states to study the sensor data using these tools might not be an efficient solution as they can visualize and analyse data only from one type of sensor. It could be a time consuming process to study the sensors individually. In addition, treating the data individually could result in the loss of valuable information due to the expected correlation among the different time series obtained from the sensing devices.

A tool which can visualize and analyse data from multiple sensors together can save lot of time and importantly help in identifying relationship between data from same/different sensors. This could help analyst draw important conclusions regarding pattern, trend and correlation which is important for studying the data and extracting relevant features of his requirements. But research software which can handle these types of data together are not available and our dissertation aims in creating a software which can visualize and analyse data from multiple sensors.

1.2 Emotion and Cognition related to learning

According to Darwin, emotions are products of natural selection. Ekman et al [43] substantiated Darwin's theory by demonstrating emotions such as: happiness, sadness, anger, fear, surprise and disgust can be universally recognized from the face in a number of cultures. These emotions are called as basic emotions. However, these basic emotions doesn't occur very often in day to day interactions [33] and have minimal relevance to learning [36].

The term cognition is a group of mental process which includes attention, memory, problem solving, perception, decision making and thinking etc. These cognitive states have high relevance to learning. According to Gagne [25], the eight phases of learning includes attention, expectancy, retrieval of relevant information to Working Memory, selective perception, encoding: entry of information into Long-Term Memory, responding, feedback and cueing retrieval. Hence, by identifying the cognitive states of the user, it might be possible to understand if a student has learned the topic or not.

1.3 Goal and Contributions

The long term goal of this project is to build an intelligent tutoring system for videolectures, in particular for the computer system course offered by CEDERJ. It should be capable of predicting user's cognitive states when they watch a video lecture and adapt the lecture accordingly in-order to enhance their learning experience. As a first step towards this, we focus on identifying the user's attention level since it is the first step in the eight phases of learning [25]. In addition, a recommendation system based on the observed emotional and cognitive states of the user is planned to be developed. It could be used by the professors to modify their pedagogical strategies in the parts of a lecture in which the automated tutor indicates a potential difficulty encountered by a large fraction of the students. With this, we hope to improve the quality of video lectures and enhance student's learning experience. The tool developed in this work is part of this overall effort.

The data visualization tool has the following objectives. Visualizing data from multiple sources, identifying the important trends in the data, statistical and signal analysis of the data, removing noise from the data, identifying possible correlation among data from different sources and extracting relevant features from the sensors that could contribute in identifying cognitive-emotion states accurately.

We use four sensors in our study. They are electroencephalogram (EEG), galvanic skin response (GSR), infrared camera and a video camera. Infrared camera and video camera are used for observing eye pupil movements and facial expressions. In addition to this, we also video record the tasks performed by the student.

A graphical user interface was built, were data from different inputs can be visualized simultaneously. All the data are visually synchronized in time and recorded videos can be analyzed frame by frame for better assessment of events. In addition to visualization, data can be analyzed using statistical and signal processing techniques. The noise in data could be removed by visual inspection. The tool allows the user to choose, visualize and analyze any desired portion of the data. In addition, we also developed calibration tests and implemented a few existed games to be used in the experiments we performed.

We use the CEDERJ video lectures and games such as nback [24] and tower of hanoi to exemplify how our tool can be used. Games such as nback and tower of hanoi are used for experimental purposes only. Since the duration of the game is less than two or three minutes and the performance of the user is evident from the result of the game, these games are helpful to provide an initial assessment of the sensor data and to search for possible correlation between sensor data and the performance of the user. Two types of experiments were conducted. Participants were asked either to play both nback and tower of hanoi or watch the video-lecture with the sensors attached to them. All collected data was recorded by a software built as a part of the tool.

1.4 Overview of Dissertation

The dissertation is organized as follows. In Section 2 we present an overview of existing Intelligent Tutoring Systems, existing data visualization tools and the recent sensing technologies and uses related to our work. Section 3 introduces our data visualization tool. In section 4, we explain the methodologies used to conduct the experiments and their results. Finally Section 5 presents our conclusion and point to future work.

Chapter 2

Background and Related Work

This chapter is devoted to explain important concepts for the evolution of our work and to survey relevant works in the literature related to this dissertation. We also define some terms which will be used in the remaining chapters as well.

In section 2.1, we discuss in detail the videolecture system used by CEDERJ that was the motivation for our work. In the subsequent section, we survey an existing intelligent tutoring system. Section 2.3 describes recent sensing technologies that could be used to enhance the performance of intelligent tutoring systems. In section 2.4 we discuss the existing data visualization tools and their limitations.

2.1 Videolecture System used by the Computer Science Course of the CEDERJ Consortium

In this section we outline the Distance learning system utilized by CEDERJ that was the main motivation of our work. CEDERJ is a consortium of public universities of the state of Rio de Janeiro and provide undergraduate courses using distance learning methodologies. Its objective is to provide students primarily living in areas that lack access to higher education and who cannot afford to join a public university and attend the regular classes offered, with a viable alternative to getting a college degree. The consortium was officially established in the year 2000 [10] and currently provides courses in areas such as administration, physics, mathematics, computer systems etc.

In particular, the computer systems (CS) course has currently more than 1800 students registered. This course is a partnership between two federal universities: UFRJ and UFF. The CS course is offered in 19 sites in the interior of Rio de Janeiro and at these sites , registered students have access to a laboratory, a small library and live tutoring classes. Tutors are a key element of the system. The local tutors support the students at each site. There are also remote tutors, who are graduate students located at each university and are on call via video conferencing or telephone at scheduled hours, or via email.

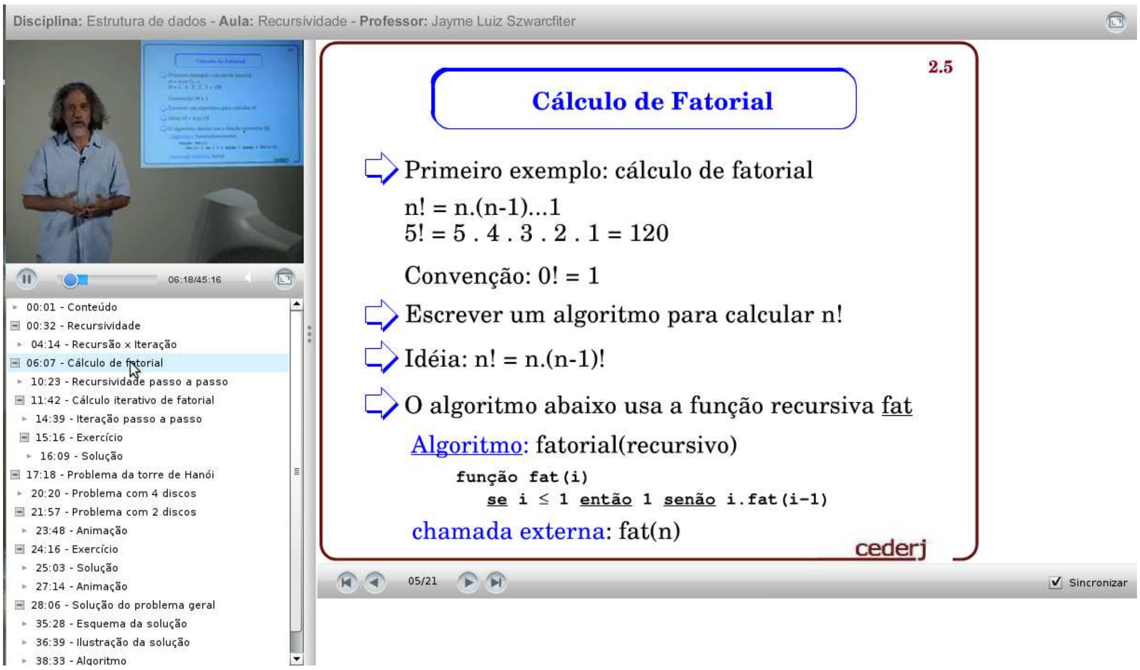
The computer systems course in CEDERJ uses videolectures that are based on the Rio multimedia server [17] and made available by the videoaula@RNP service offered by the National Education and Research Network [45]. Users can access the video lectures through the Internet or at the CEDERJ sites. Each video lecture consists of a pre-recorded video of classes, an index with the topics presented in the video and animated slides. All the information is synchronized. To enhance the learning process, the system also enables the user to interact with it by pausing the video, moving forward or backward and choosing the topic of interest. In addition users can interact with slides.

The RIO multimedia system [17] is an universal multimedia storage system with random allocation and block replication. RIO supports various types of media such as video, audio, text and image. The RIO was originally conceived by UCLA [47]. In 1999, through a joint CNPq/NSF research project, the development of the RIO system continued and latter was improved by the LAND Laboratory at COPPE.

Currently the video lectures of the Computer Science course are available through the service videoaula@rnp offered by Brazilian National Education and Research Network (RNP). The architecture of the service videoaula@rnp comprises several clusters of the RIO system. Each cluster consists of a web server that manages the client requests, a set of storage servers where the contents are stored and a management server implements the interface between the user web front-end and the storage servers.

To visualize the video lectures, a client was developed by LAND. Currently two clients are available , an HTML5 client and a Flash Client. The client interface is shown in Figure 2.1. The client displays the video, the index of the topics and slides which are all synchronized in time. In addition, the client records all the user's interactions such as pause, seek etc and sends the information to the server which maintains a log. Figure 2.1 shows the client interface as seen by the user.

The client of the RIO multimedia system does not adapt the videolecture automatically, in accordance with the student's learning experience. For example, when a student watches a video lecture and does not understand the professor's explanation about a particular topic, the only option the student has available is to rewind and watch the topic again. If the student fails to understand the material after repeatedly watching it, he can become frustrated and this could affect his attention for the rest of the session.



Disciplina: Estrutura de dados - **Aula:** Recursividade - **Professor:** Jayme Luiz Szwarcfiter

06:18/45:16

- 00:01 - Conteúdo
- 00:32 - Recursividade
  - 04:14 - Recursão x Iteração
- 06:07 - Cálculo de fatorial
  - 10:23 - Recursividade passo a passo
  - 11:42 - Cálculo iterativo de fatorial
    - 14:39 - Iteração passo a passo
    - 15:16 - Exercício
      - 16:09 - Solução
- 17:18 - Problema da torre de Hanói
  - 20:20 - Problema com 4 discos
  - 21:57 - Problema com 2 discos
    - 23:48 - Animação
  - 24:16 - Exercício
    - 25:03 - Solução
    - 27:14 - Animação
  - 28:06 - Solução do problema geral
    - 35:28 - Esquema da solução
    - 36:39 - Ilustração da solução
    - 38:33 - Algoritmo

2.5

Cálculo de Fatorial

⇨ Primeiro exemplo: cálculo de fatorial

n! = n.(n-1)...1
5! = 5 . 4 . 3 . 2 . 1 = 120

Convenção: 0! = 1

⇨ Escrever um algoritmo para calcular n!

⇨ Idéia: n! = n.(n-1)!

⇨ O algoritmo abaixo usa a função recursiva fat

Algoritmo: fatorial(recursivo)

```
função fat(i)
   se i ≤ 1 então 1 senão i.fat(i-1)
```

chamada externa: fat(n)

cederj

05/21

☑ Sincronizar

Figure 2.1: Client Interface of RIO Multimedia System

In order to enhance the student learning experience, it is essential for distance learning systems to intelligently tailor the lecture to the student needs. Non-invasive sensors could be used with distance learning systems to identify emotions experienced by the user and could be used as a feedback to adapt the system.

2.2 An Intelligent tutoring system: Wayang Outpost

Wayang Outpost is a multimedia Intelligent Tutoring Systems (ITS) targeted for geometry teaching that help users to improve their math skills and prepare them for math tests such as Scholastic Aptitude Test (SAT) and Massachusetts Comprehensive Assessment System (MCAS). It provides instruction via website which enables easy access to students at school or home. Students begin their session after logging into the site and they are guided through different topics. They receive multiple choice questions that are chosen adaptively based on the their ongoing performance in the current section. Figure 2.1 shows the web interface of Wayang Outpost

Wayang Outpost includes one or more multimedia hints consisting of text messages, audio and animation for all the multiple choice questions. It also includes a gendered learning companion who encourages the student to use the hint while

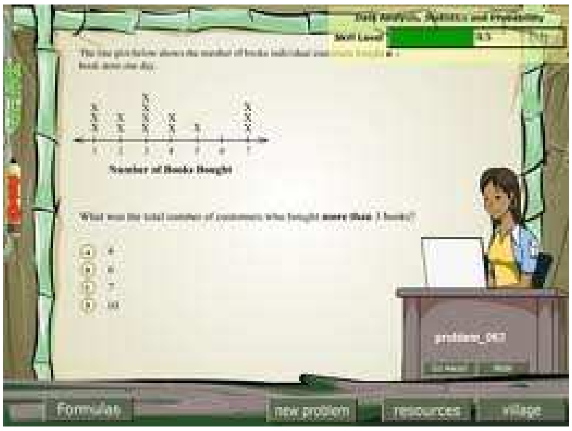


Figure 2.2: Wayang Outpost

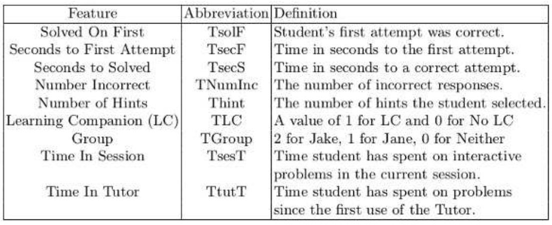
answering a question. Gendered companions are capable of expressing emotions and it emphasizes the importance of effort and perseverance. Wayang Outpost queries the user about their emotions every five minutes and after they have solved the problem.

A recent research by Cooper et al [12] with Wayang Outpost used four different sensors to detect the user emotional states such as frustration, interest, confidence, and excitement. His research goal was to enhance the performance of ITS using the feedback from sensors. Sensors used in their research were skin conductance bracelet, pressure sensitive mouse , pressure sensitive chair and a mental state camera developed in [33].

An experiment was conducted by Cooper et al [12] with 35 high school students and 58 undergraduate students. Students used Wayang Outpost with the sensors attached to them. At interval of 5 minutes, Wayang queried students about a randomly selected emotion (for example, how interested/confused/excited they are feeling now?) on a 5 point scale where three corresponds to neutral and five corresponds to high value of emotion. A total of 588 queries were collected from all the students and these results were used for training and testing their classification model.

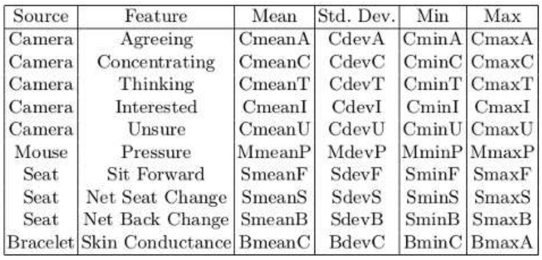
In order to train the classification model, a feature-set from sensors and tutor were extracted. They are shown in the Figure 2.3 and 2.4. Feature-set in this case means transforming the input sensor and tutor data into a reduced set of features so that the desired analysis could be performed using this reduced representation instead of the full input size. Feature set of sensor data represents a summary of sensor values over the interval of emotional query. To predict emotions , best feature set from sensors and tutors were extracted using a step-wise linear regression model with emotions as dependent variables, and tutor and sensor features as independent variables. For each emotion different set of best features were obtained.

The selected features were given as an input to the linear classifier model which



Feature	Abbreviation	Definition
Solved On First	TsolF	Student's first attempt was correct.
Seconds to First Attempt	TsecF	Time in seconds to the first attempt.
Seconds to Solved	TsecS	Time in seconds to a correct attempt.
Number Incorrect	TNumInc	The number of incorrect responses.
Number of Hints	Thint	The number of hints the student selected.
Learning Companion (LC)	TLC	A value of 1 for LC and 0 for No LC
Group	TGroup	2 for Jake, 1 for Jane, 0 for Neither
Time In Session	TsesT	Time student has spent on interactive problems in the current session.
Time In Tutor	TtutT	Time student has spent on problems since the first use of the Tutor.

Figure 2.3: Set of Features from Tutor



Source	Feature	Mean	Std. Dev.	Min	Max
Camera	Agreeing	CmeanA	CdevA	CminA	CmaxA
Camera	Concentrating	CmeanC	CdevC	CminC	CmaxC
Camera	Thinking	CmeanT	CdevT	CminT	CmaxT
Camera	Interested	CmeanI	CdevI	CminI	CmaxI
Camera	Unsure	CmeanU	CdevU	CminU	CmaxU
Mouse	Pressure	MmeanP	MdevP	MminP	MmaxP
Seat	Sit Forward	SmeanF	SdevF	SminF	SmaxF
Seat	Net Seat Change	SmeanS	SdevS	SminS	SmaxS
Seat	Net Back Change	SmeanB	SdevB	SminB	SmaxB
Bracelet	Skin Conductance	BmeanC	BdevC	BminC	BmaxA

Figure 2.4: Set of Features from Sensors

is built for each emotion. Leave-one-student-out cross validation was performed for each model and accuracy ranging from 78 percentage to 87.5 percentage was obtained for each emotion.

Even though they obtained the accuracy in the range of 78 percentage to 87.5 percentage, improvements can be made in a number of places. To train the classifier and stepwise regression model, they depend on the user's self-report emotions. But the self-report emotion from the user is not very reliable because it is not easy for some one to distinguish his emotional state accurately or may even choose not to reveal his accurate emotion [46]. In order to mitigate this problem, a judge could be used. The judge would visualize the user's recorded facial expressions and confirm the self-report.

Another major place for improvement is in the selection of feature sets from sensors. In the current scenario, they just consider a summary of information provided by all the sensor values. But when the information is averaged, important events in the sensor data, like sudden peaks, could be lost. However, this could be a possible indicator of any cognitive-emotional states. Also correlation among the sensor data, which might contain some information about emotions, is not studied.

In order to address these issues, it is essential to understand and study the sensor and tutor data. For that purpose, a specialized tool to visualize and analyze all the data from different sensor would be very helpful. In this way, researchers could choose to study sensors individually or jointly and try to understand if there are any important characteristics in the data which are indicators of cognitive-emotional states and use them as feature set.

2.3 Sensing Technologies

Sensing technologies may play an important role in understanding how learning occurs. Different sensors have been used in various researches to detect cognitive-emotion states of the user and these states are essential components of learning [48]. A theoretical model which explore the connection between these states and learning is discussed below.

The model proposed by Kort, Reilly and Picard [36] is depicted in the Figure 2.5. It is a four quadrant model which relates phases of learning to emotions. Emotions with respect to each quadrant are mentioned in Figure 2.5. Horizontal axis is an emotion axis. The more pleasant emotions are on the right of the axis and more unpleasant emotions on the left. The vertical axis is a learning axis where knowledge construction represents the top of the axis and unlearning represents the bottom.

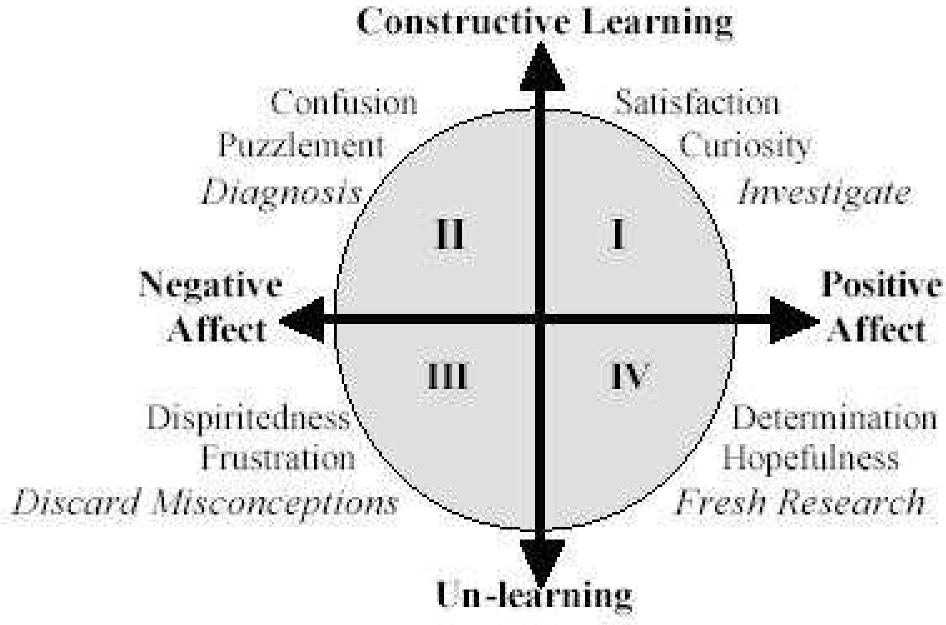


Figure 2.5: Proposed model relating phases of learning to emotions [36]

According to this model, learners can move around the quadrants in many different patterns. Ideally learners begins in quadrant one or two where they are curious to learn a new topic or they want to reduce the confusion about the topic. If learners encounter any difficulties or if their idea fails, they move to quadrant three where they are frustrated or have discarded any misconceptions. Then, they might move to quadrant four with hope and/or determination to solve the problem and the way to understand a topic. In case the learner obtains a fresh idea they return back to quadrant one. Learners may experience multiple cycles around the phase plane until they finally learn a topic. This model was validated by the author using a set of experiments.

Intelligent Tutoring Systems could investigate the learning process using the model described above or other models[19] which relates emotions and learning. This might enable ITS to improve the performance by adapting according to the student learning performance which in turn could enhance student's learning experience.

To detect the cognitive-emotion states of the user, one of the best approaches is to use sensors like electroencephalogram (EEG), skin conductance meter or facial expression camera [12, 19, 20]. In order to determine the possible correlations among the data from sensors and the user states and to train ITSs, users could be asked to self report their emotions depending on the experiment. But this may not be completely reliable, as students would not be able to distinguish their emotional state accurately or may even choose not to reveal their emotion [46]. Another approach is to record the facial expressions of the student and to use a trained expert to judge the emotion. However, this approach is not scalable since it requires human intervention. So a scalable solution could be to use sensors to automatically detect a cognitive-emotion user state. In the remaining section we explain in detail about these sensors and their past uses.

2.3.1 Facial Expressions

The human face provides one of the most versatile and natural means of communicating cognitive and emotional states [33]. Humans can detect and recognize these states from the facial expressions with little effort. Many studies have tried to recognize automatically these states using the facial video input [13, 33].

To automatically detect emotional states from facial expressions one of the approach is to extract facial feature points from each video frames. Facial feature points represents the shape and location of facial components such as eyes,nose etc. Then, using the facial features, different facial action units are identified. A sample of action units is also shown in the Figure 2.6. Facial action unit represents the facial muscular movements that produces facial appearance changes. The Facial action code system (FACS), implemented by Ekman and Freisen [43], comprises of 46 facial action units which can be combined to describe all facial expressions. Some of the facial action units are inner brow raiser, upper lip raiser, mouth stretch, etc. Then using the facial action units as input, appropriate mathematical models are created to predict emotions.

Rana et al [33], in her mind reading system, used Hidden Markov Model (HMM) to map facial feature input to action units. Then, these action units were used as an input for dynamic bayesian networks to predict cognitive mental states such as thinking, deciding, confused, interested, attention and unsure. Cowie et al [13], in his intelligent system for facial emotion recognition, use a slightly different approach. Initially, they extract facial features and use them to calculate a new parameter called facial animation parameter(FAP). For six universal emotions (joy, surprise, fear, anger, disgust, sadness) a set of rules were created by analyzing the FAP. These rules are calculated for all the facial expressions in the Ekman and Freisen dataset. Using these rules, emotions were predicted for any facial inputs.

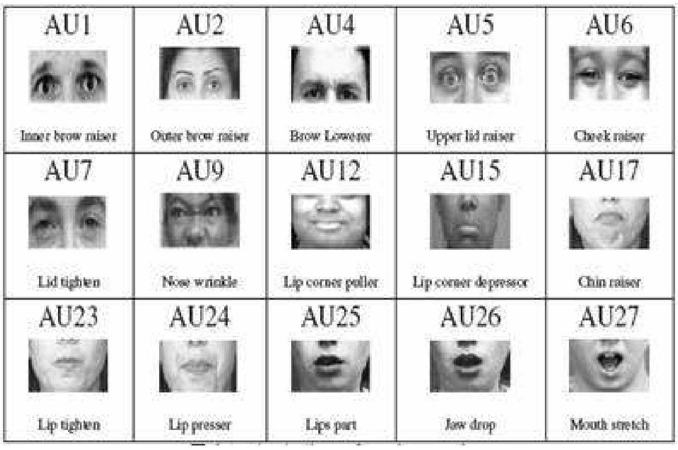


Figure 2.6: Facial Action Units

Studies [30, 38] also have tried to identify cognitive state using the eye metrics. The increase in the size of pupil is correlated to user paying more attention. Also, using the information of pupil size and point of gaze, cognitive states such as relaxed and engaged in the context of problem solving, focused and distracted attention in the context of driving and the states of alert and fatigued in the context of visual search were detected.

2.3.2 Electrodermal Activity Sensor

Electro dermal activity (EDA) refers to the electrical changes measured at the surface of the skin when activated by various brain circuits. Electrical changes are caused by the variation of hydration in the sweat glands. For example, brain sends signals to the skin to increase the level of sweating when the user experiences arousal of strong emotions or increased cognitive working load.

EDA can be measured in many different ways which includes skin conductance, skin resistance, skin impedance etc. Skin conductance is the most preferred way for reporting EDA. It is calculated by applying an electric potential between the two electrodes in contact with the skin and measuring the resulting current flow between them.

EDA measurement is mainly characterized by two main components [8]. One component is the general tonic level EDA which relates to slower acting components, for example, slow climbing and slow declinations over time. This slow-changing level is generally referred to as Skin Conductance Level (SCL). Relevant changes in the skin conductance level typically occur in a period of tens of seconds to minutes.

The other component is the phasic component which refers to the faster changing elements of the signal. They usually show up as peaks in the skin conductance. These peaks are generally referred to as Skin Conductance Responses (SCRs). They normally occur due to the presence of external stimuli while in tonic skin conductance, there are no such external stimuli.

EDA for cognitive-emotive states evaluation

EDA has the potential to recognize cognitive-emotive states. Skin conductance usually increases when a person is more aroused. For example, when he is engaged, stressed, or excited. Skin conductance tends to stay low or drop when a person is less aroused. For example, when he is disengaged, bored, or calm [1].

Picard at el [44] performed a study which shows that EDA sensor is sensitive to emotional and cognitive changes . The participants were asked to perform three different tasks using EDA sensor. The tasks were physical test, cognitive test and emotional test. Participants were asked to sit idle before and after each tasks for establishing a baseline.

In all the three tasks, skin conductance gradually decreased during the baseline period as the participants were relaxed. Once they started performing the physical test, cognitive test or emotional test, a steep increase in skin conductance level was observed. In cognitive and emotional tests, multiple skin conductance responses (peaks) were also observed.

2.3.3 Electroencephalography

EEG sensor provides recorded electrical activity of the brain. There are millions of neurons in brain and each generates small electric voltage. The aggregation of this electrical voltage provides electrical readings which electrodes on the scalp can detect [41]. Positioning of the electrodes on the scalps is based on the international 10/20 system which is shown in the figure 2.7. Since EEG is a non-invasive passive measuring device it is safe for extended and repeated use.

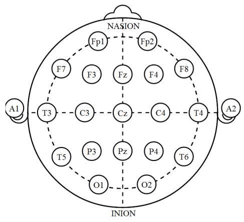
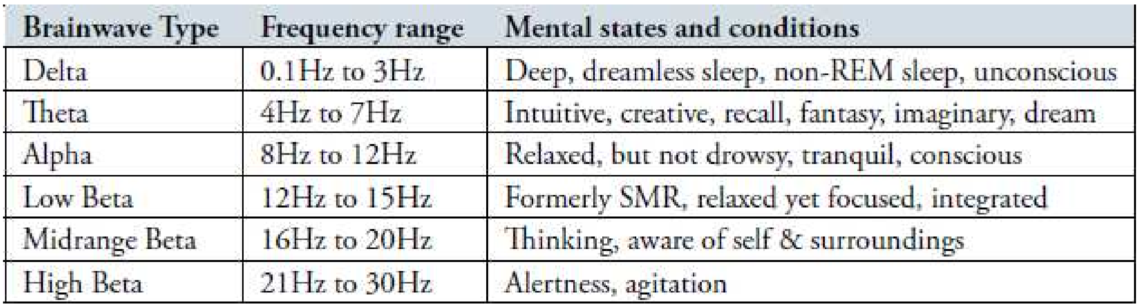


Figure 2.7: Scalp sites in International 10/20 System

Rhythmic Activity

EEG signal is described in terms of rhythmic activity. This signal can be divided into several frequency bands and are defined as delta, theta, alpha, beta and gamma. Their frequency range and functions are shown in the Figure 2.8 [41].



Brainwave Type	Frequency range	Mental states and conditions
Delta	0.1Hz to 3Hz	Deep, dreamless sleep, non-REM sleep, unconscious
Theta	4Hz to 7Hz	Intuitive, creative, recall, fantasy, imaginary, dream
Alpha	8Hz to 12Hz	Relaxed, but not drowsy, tranquil, conscious
Low Beta	12Hz to 15Hz	Formerly SMR, relaxed yet focused, integrated
Midrange Beta	16Hz to 20Hz	Thinking, aware of self & surroundings
High Beta	21Hz to 30Hz	Alertness, agitation

Figure 2.8: Frequency bands of EEG and their functions

Studies [4, 27, 34] have shown that alpha and theta frequency bands are significantly used in the studies related to cognitive states. A study [4] which involves a subject closing and opening the eye indicates that the power of alpha rhythm increases when the subject closes his eye. In another study [34] where the subject performs a mental task indicates that the alpha activity is sensitive to the changes in mental effort. A decrease in the amplitude of the alpha rhythm was observed during mental arithmetic tasks. The study in [28] which is related to working memory, it was observed that the power of theta observed in the frontal regions of the brain increases in strength as tasks require more focused attention while the alpha rhythm observed in the posterior regions of the brain tend to decrease in strength as task becomes more difficult. In the study by Klimesch [35] where the subject is presented with a series of words and their memory performance was performed shows that the power of theta rhythm increases in response to the memory demand.

EEG Artifacts

Eye movements, eye blinks, muscle noise, heart signals, and line noise produce distracting artifacts in EEG recordings[32]. These artifacts generate potential that can be of magnitude much larger than brain-generated electrical potentials. This creates serious problems when EEG is analyzed. They can bias the measures computed from the data and as a result significantly alter the results of the experiments.

It is important to remove the artifacts from EEG before analyzing its signals. Rejecting the EEG amplitudes greater than a threshold value is the most easiest method used in the research settings. However, if there are many artifacts, by rejecting such values, a large quantity of experiment data is lost. To avoid such issues, techniques like independent component analysis are used to remove the artifacts [32, 37].

2.4 Existing Data Visualization Tools

Data visualization tools are built for visualizing and analyzing huge quantity of data. It should enable the user to understand the data clearly and effectively which can help them in making right decisions and discover interesting ideas. It is a powerful way to simplify the complexity in the data. The best data visualization tools can expose something new about the data and also expose relationships contained within the data. Such tool should not be sophisticated to use as it might leave the user frustrated. Ideally a tool should be simple and easy to use.

When sensors are used with ITS to identify cognitive and emotional states, a huge quantity of data is obtained from the sensors. It is important to visualize and analyze all the data clearly and effectively for selecting relevant feature that can help to identify the emotion. It is also important for a tool to provide the user with an ambient for research, where he can ask new questions, expose something new and also find interesting solutions. To the best of our knowledge, we don't know any ITS which is supported by a specialized data visualization tool. This might be due to the complexity of multiplexing all the data together and communicating the information clearly and effectively to the user.

There are tools which displays and analyze data specific to a single sensor. EEGLAB [18] and ELAN [2] are the most commonly used tools for visualizing and analyzing EEG. ACUME [39] is used for understanding facial expression and gesture data. These tools allows the user to understand the data well and also provides an environment for users to perform research on the data.

EEGLAB is a toolbox and graphical user interface used for studying EEG data of any number of channels. EEGLAB development is now centered at the Swartz Center for Computational Neuroscience of the Institute for Neural Computation at the University of California San Diego. ELAN was developed in Lyon, France. Including EEG , it can analyze Magnetoencephelography (MEG) and local field potentials (LFP) signals. Most of the functions provided by EEGLAB and ELAN are similar.

EEGLAB and ELAN can display the raw EEG data of all the channels in one window for a desired timescale. Analysts can visualize the data and use it for rejecting artifacts manually or using a threshold value. Analysts can apply and evaluate the results of independent component analysis of EEG data. Independent component analysis can be used to detect and remove the noise automatically without any loss of information from data.

In both EEGLAB and ELAN tools, EEG data (modified or unmodified) can be filtered using high pass or low pass filters for example, Wavelet analysis and Fourier analysis can be applied on raw EEG data. ELAN provides inbuilt statistical tests like Wilcoxon paired texts, Kruskal Wallis tests which can be applied to get information on across trail statistics. In addition, EEGLAB and ELAN can also plot and analyze Event Related Potentials.

ACUME [39] is a visualization tool for understanding facial expression and gesture data across large population. ACUME uses Action Unit Spectogram (AUSpec) which contains frequency of each action unit (indicated by color intensity) displayed every second across all viewers for different stimuli (two different movie clips for example) . So if the color intensity at a particular point is high, it means that particular action unit occurred in large number in that particular second across the group. This feature allows to compare the group responses at a particular point in time. It is also possible to compare group responses to different stimuli. For example, occurrence of particular action unit was very high in 1st movie clip when compared to second. ACUME also provides a way to access an individual response by using Grid View. Grid view allows researcher to view spectrogram of specific participant side by side and compare their response .This allows the researcher to identify participants whose responses were much different from the others.

All these data visualization tools provide a very detail study about a single sensor. These tools are not created to handle data from different sensors. For an ITS application where multiple sensors may be employed, it is desirable to analyze jointly the output from different sensors to possibly detect correlations among the sensors. Such a tool would necessarily handle different types of sensors and would have to automatically synchronize all the input data, for analysis. The following chapter will describe the tool we implemented.

Chapter 3

System Design

In this Chapter, we first present overview of the system, then we detail the system's architecture and its components.

3.1 Overview of the System

In this work, we built a data visualization tool where the user can visualize and analyze time-series from different sensors, event-related data and video, all in the same interface. All data is synchronized and can be visualized and analyzed either collectively or individually.

The objective of this work is to provide an environment for the researcher to study the data from multiple sources. For example, an analyst can use this tool to search for any particular trend in the data, to find possible correlations among the data from the same source or multiple sources, to identify important characteristics of the data and to detect noise in the data.

The main objective of the tool is to assist on going work in our laboratory on Intelligent Tutoring Systems (ITS). In this way, signals from multiple non-intrusive sensors must be analyzed in order to obtain insights which would be helpful to build a predictive model of user emotions. To illustrate how this tool can be of help to ITS, we provide an example.

A student session monitored by an ITS is captured by reading facial expressions and the output of an EDA (Electrical Dermal Activity). Once the session is over, the analyst can visualize all the data collectively in our data visualization tool. He can watch the facial expression video and plot the EDA graph in the same graphical interface. When the user plays the recorded session, the video is shown synchronized with a time line (black vertical line) which moves over the plotted data. As a result, the analyst can visually correlate facial expressions with the EDA time series and judge if there are changes in the time series associated with the facial expressions of the student. In addition, statistical and signal processing techniques can be applied on the data which could enhance the understanding of the sensor data. These features may help the analyst to understand which are the relevant features in the data. For example, if a peak is noticeable in the EDA data the analyst may check for student's expression at the time in order to see if the expression indicates that the user is concentrated or distracted. The analyst may infer a relation between a student emotion and EDA data. The Figure 3.1 shows an overview of our software interface.

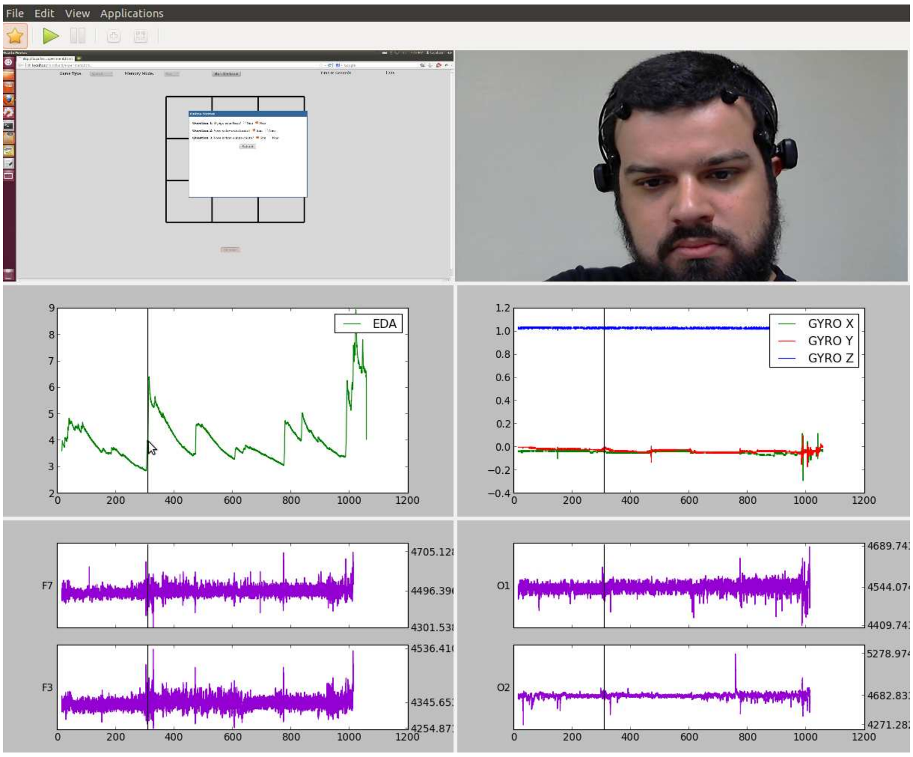


Figure 3.1: All the data jointly displayed

3.2 System's Architecture

Our data visualization tool is written in python. We use GTK+ toolkit for creating our graphical user interface. It is open-source, very stable, light weight, fast and can run on cross platforms such as Windows, MAC and Linux. Two important libraries that we use in our graphical user interface are opencv (for decoding video) and matplotlib (for plotting charts).

The architecture of our system is shown in the Figure 3.2 . It consists of an input module, a module for video synchronization , a module for data selection, a signal processing and statistical module, a video display module and an animated chart display module. Input module, for example, reads data from sensors like EEG, EDA, normal video camera and infra red camera. The Video synchronizer module synchronizes the video inputs and the videos are displayed using the video display module. The data to be plotted from the sensors such as EEG and EDA are selected using the data selection module and displayed using the animated chart display module. The system contains a system clock attached to a time line. When triggered, it produces a value that is consistently incremented at fixed intervals. Based on the values generated by the clock, the video synchronizer module extracts the corresponding video frames. The animated chart display module plots a black vertical line(timeline) in the graph which corresponds to the position in time of the video. As a result, the analyst can relate the sensor readings, for example EEG or EDA, with respect to the video. Analyst can also apply the statistical and signal processing techniques by the tool (processing and statistical module) to the data generated by the sensors. The system allows the analyst to pause the clock, advance the clock forward and backward.

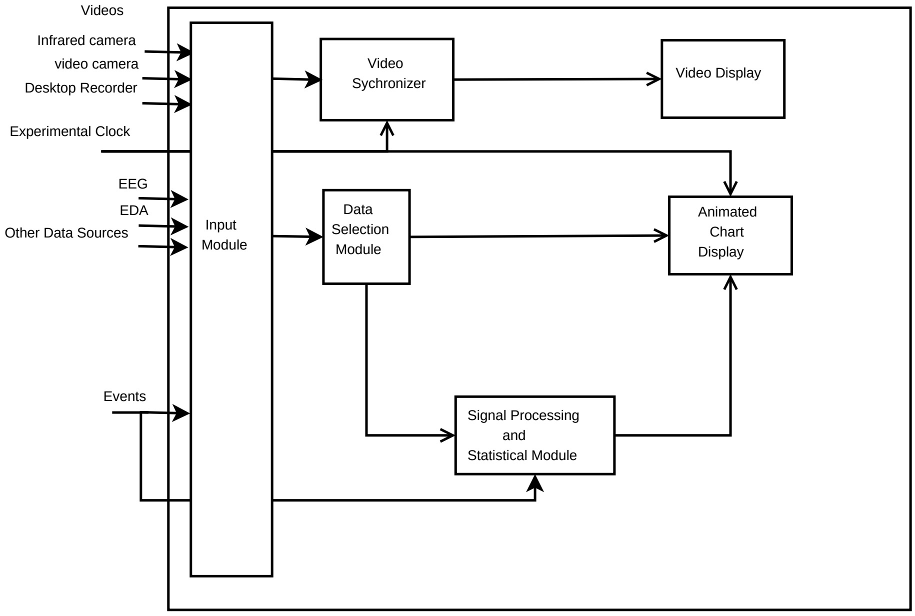
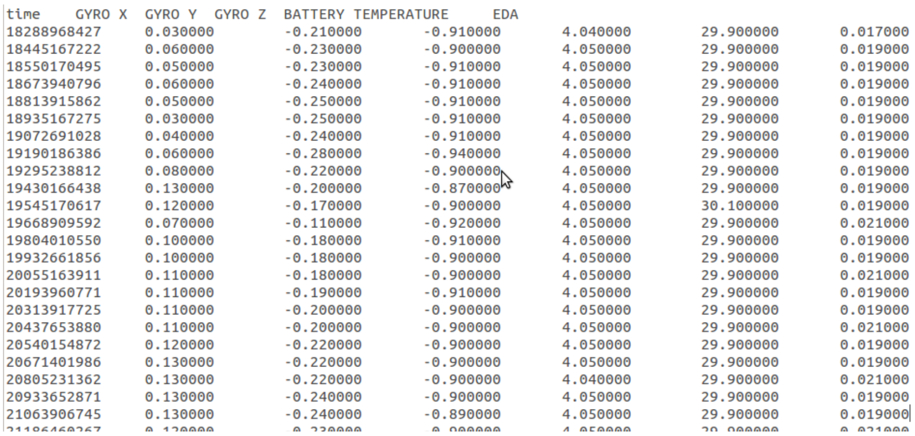


Figure 3.2: Architecture of the Data Visualization Software

3.2.1 Input Module

The input module can handle three types of data streams: data from sensors(time series), video and event-related data. Time-series data should be stored in a text file in which the first column represents the time and the rest of the columns represent the information provided by a sensor such as EEG, EDA etc. The first row of the file contains the type of the data provided by the sensor. Figure 3.3 shows an example of a time-series input file. The input module accepts data from any number of sensors.

In our tool, each information provided by the sensor is stored by the input module in different instances of the class ”data”. The main members of the class are the name of the information , name of the sensor, the data value and the corresponding



```
time    GYRO X  GYRO Y  GYRO Z  BATTERY TEMPERATURE     EDA
18288968427     0.030000        -0.210000       -0.910000       4.040000        29.900000       0.017000
18445167222     0.060000        -0.230000       -0.900000       4.050000        29.900000       0.019000
18550170495     0.050000        -0.230000       -0.910000       4.050000        29.900000       0.019000
18673940796     0.060000        -0.240000       -0.910000       4.050000        29.900000       0.019000
18813915862     0.050000        -0.250000       -0.910000       4.050000        29.900000       0.019000
18935167275     0.030000        -0.250000       -0.910000       4.050000        29.900000       0.019000
19072691028     0.040000        -0.240000       -0.910000       4.050000        29.900000       0.019000
19190186386     0.060000        -0.280000       -0.940000       4.050000        29.900000       0.019000
19295238812     0.080000        -0.220000       -0.900000       4.050000        29.900000       0.019000
19430166438     0.130000        -0.200000       -0.870000       4.050000        29.900000       0.019000
19545170617     0.120000        -0.170000       -0.900000       4.050000        30.100000       0.019000
19668909592     0.070000        -0.110000       -0.920000       4.050000        29.900000       0.021000
19804010550     0.100000        -0.180000       -0.910000       4.050000        29.900000       0.019000
19932661856     0.100000        -0.180000       -0.900000       4.050000        29.900000       0.019000
20055163911     0.110000        -0.180000       -0.900000       4.050000        29.900000       0.021000
20193960771     0.110000        -0.190000       -0.910000       4.050000        29.900000       0.019000
20313917725     0.110000        -0.200000       -0.900000       4.050000        29.900000       0.019000
20437653880     0.110000        -0.200000       -0.900000       4.050000        29.900000       0.021000
20540154872     0.120000        -0.220000       -0.900000       4.050000        29.900000       0.019000
20671401986     0.130000        -0.220000       -0.900000       4.050000        29.900000       0.019000
20805231362     0.130000        -0.220000       -0.900000       4.040000        29.900000       0.021000
20933652871     0.130000        -0.240000       -0.900000       4.050000        29.900000       0.019000
21063906745     0.130000        -0.240000       -0.890000       4.050000        29.900000       0.019000
21186460267     0.120000        -0.230000       -0.900000       4.050000        29.900000       0.021000
```

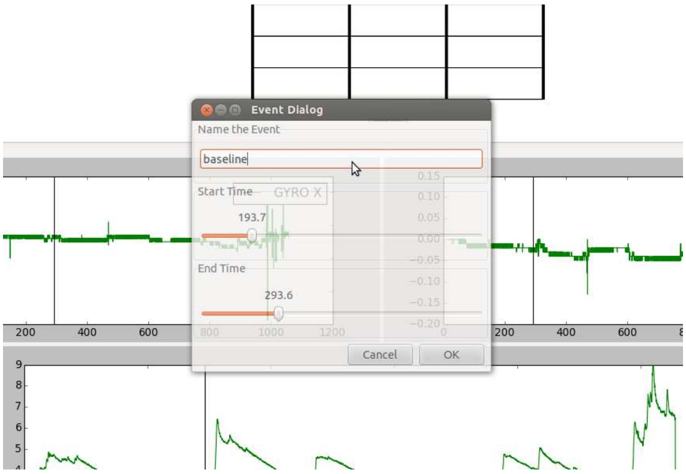
Figure 3.3: Time-series data representation

time stamp. The list of such instances is used in all the other modules such as data selection module, animated chart display module and statistical and signal processing module for processing the data.

In the case of video input, the input module only accepts video encoded using x264 library. We note that any video can be represented in this format using ffmpeg. More than one video can be selected using this module. Apart from the video file, the input module also requires a file which contains the position of each frame of the video with respect to time. In case this file is not available, the video synchronizer can construct this information.

Other than time-series and video data, the input module handles all the events generated by the graphical user interface (GUI). They include events like pausing the clock, advancing the clock forward and backward and module selection events. Input module provides an interface, as shown in Figure 3.4, to receive event-related data. This is a very important feature in our tool. The analyst may input a name for the event and choose the initial and final time for it. The Statistical and signal processing module use such events for analyzing the sensor data between their starting and ending points in time. The scroll bars shown in figure 3.4 is used to choose the start and end time and, as the scroll bar is moved, the position of the video and position of the black line in the graph changes simultaneously. This enables the analyst to create event that may important during the analysis. The interface is flexible, so that the analyst can visualize the video and the plots during event creation.

Figure 3.5 shows how data from sensor may be added to our tool. A user can select multiple sensors at the same time. Similar process is used for adding the video. This module also gives the user the option to remove any sensor, video or event-related data from the system. This allows the user to keep only the data which he considers useful for analysis . Since the input module stores all the information from sensors in different instances of the class "data" , we also provide an option for



Event Dialog

Name the Event

baseline

Start Time

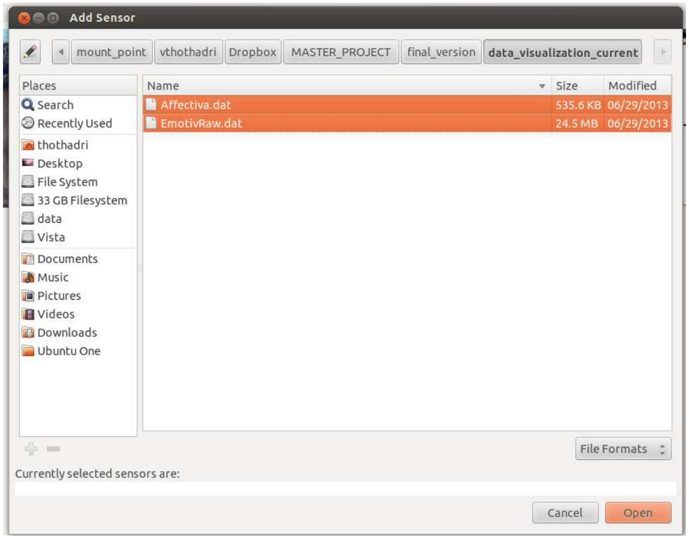
193.7

End Time

293.6

Figure 3.4: Event Interface

the user to delete any of these instances as shown in the Figure 3.6. These instances are identified by concatenating the name of parent directory where data sensor is stored, name of the sensor and name of the information provided by the sensor.



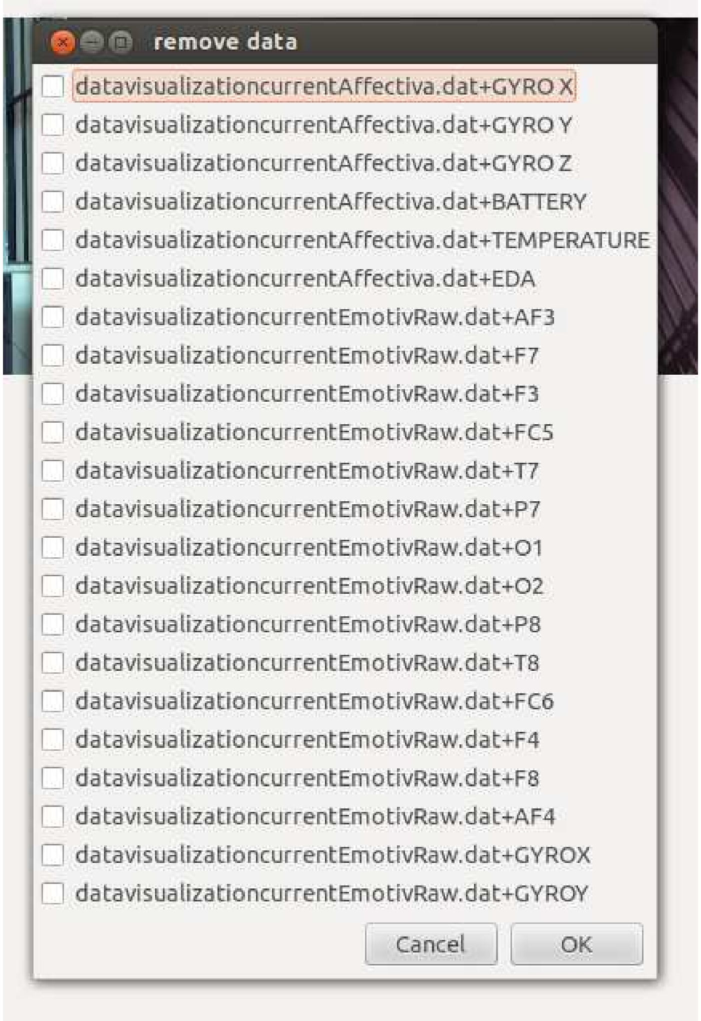
Add Sensor

mount_point / vthothadri / Dropbox / MASTER_PROJECT / final_version / data_visualization_current

Name	Size	Modified
Affectiva.dat	535.6 KB	06/29/2013
EmotivRaw.dat	24.5 MB	06/29/2013

Currently selected sensors are:

Figure 3.5: Adding Sensor data to our system



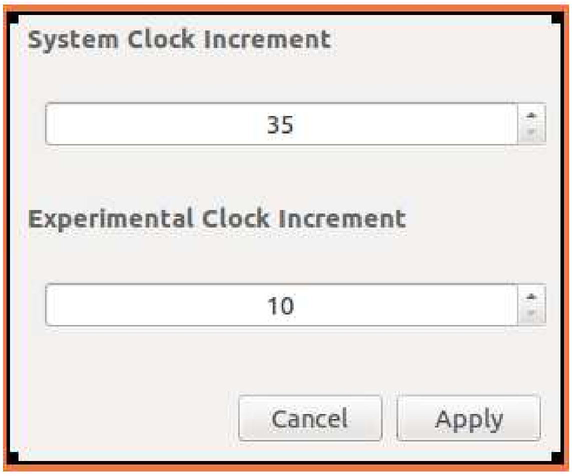
□ datavisualizationcurrentAffectiva.dat+GYRO X
□ datavisualizationcurrentAffectiva.dat+GYRO Y
□ datavisualizationcurrentAffectiva.dat+GYRO Z
□ datavisualizationcurrentAffectiva.dat+BATTERY
□ datavisualizationcurrentAffectiva.dat+TEMPERATURE
□ datavisualizationcurrentAffectiva.dat+EDA
□ datavisualizationcurrentEmotivRaw.dat+AF3
□ datavisualizationcurrentEmotivRaw.dat+F7
□ datavisualizationcurrentEmotivRaw.dat+F3
□ datavisualizationcurrentEmotivRaw.dat+FC5
□ datavisualizationcurrentEmotivRaw.dat+T7
□ datavisualizationcurrentEmotivRaw.dat+P7
□ datavisualizationcurrentEmotivRaw.dat+O1
□ datavisualizationcurrentEmotivRaw.dat+O2
□ datavisualizationcurrentEmotivRaw.dat+P8
□ datavisualizationcurrentEmotivRaw.dat+T8
□ datavisualizationcurrentEmotivRaw.dat+FC6
□ datavisualizationcurrentEmotivRaw.dat+F4
□ datavisualizationcurrentEmotivRaw.dat+F8
□ datavisualizationcurrentEmotivRaw.dat+AF4
□ datavisualizationcurrentEmotivRaw.dat+GYROX
□ datavisualizationcurrentEmotivRaw.dat+GYROY

Figure 3.6: Removing data from our system

Clock

Our system contains two different clocks. We call them Experimental Clock and System Clock. The System Clock is triggered by clicking the play button in the graphical interface. When triggered, the System Clock starts a timer which repeatedly executes a task after a fixed time interval. The time interval can be configured by the analyst and the minimum value is set to 35 milliseconds which is the minimum time required to execute a task in our test bed with intel i5 quad core processor and 16 GB RAM. The purpose of the task is to set and increment the value of the Experimental Clock and make a call to the video synchronizer, the animate chart display and the video display modules. The video synchronizer module extracts the frame with the timestamp value closest to the Experimental Clock and the video display module displays those frames. The animate chart display module plots a red vertical line in all data plots determined by the Experimental clock value.

The increment value of the Experimental Clock set by the System Clock can be configured by the analyst. For example, if the analyst wants to briefly overview all the data synchronized, the increment value can be set high (around 1000 milliseconds). This implies that every time the timer set by System Clock executes a task, the value of the Experimental Clock is incremented by 1000 and video synchronizer module and animate chart display module use this value to determine the position of time. On the other hand if the analyst wants to perform a detailed analysis of all the data, the increment value can be set around 10-30 milliseconds. Figure 3.7 shows the interface where analyst can configure these values and Figure3.8 shows our GUI with respect to the value of the Experiment Clock and the value of the System Clock .


System Clock Increment: 35

Experimental Clock Increment: 10

Figure 3.7: Interface for configuring clock values

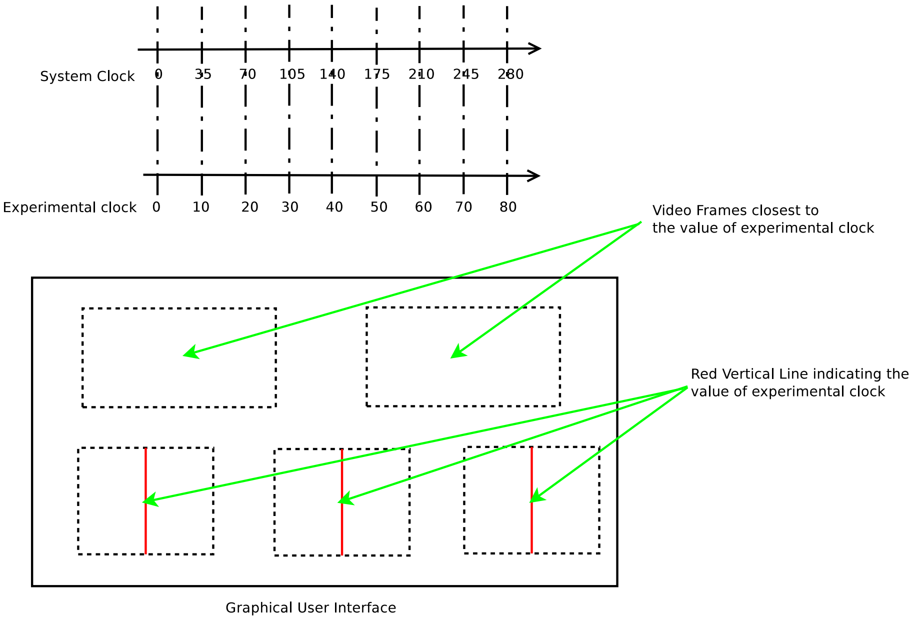

System Clock
0
35
70
105
140
175
210
245
280
Experimental clock
0
10
20
30
40
50
60
70
80
Video Frames closest to the value of experimental clock
Red Vertical Line indicating the value of experimental clock
Graphical User Interface

Figure 3.8: GUI

3.2.2 Video synchronizer Module and Video Display Module

In this module we synchronize the videos that are given in the input module. Suppose that the analyst has two videos where one displays facial expressions of the user while the other displays the screen shot of what the user was doing. The video synchronizer module synchronizes these two videos, so that the analyst can visualize the facial expression of the user corresponding to the task being performed. Our video synchronizer is capable of synchronizing any number of videos. Figures 3.9 show three from different sources synchronized.

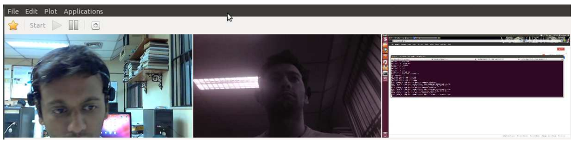


Figure 3.9: Videos from three different sources synchronized

The video synchronizer module accepts videos which are encoded using x264 library. We use opencv for reading the frames from the videos. The Mind Reader, system designed by a colleague who is a part of our project to capture the data from sensors, creates a separate file with the time-stamp captured for each frame of a video which is included automatically by the input module when the video is selected. In case the time-stamp for any video is not available, the video-synchronizer creates the file with the assumption that video frame starts from the initial time obtained from the other video or from the other graph (minimum value is selected). Then, all the frames are equally divided based on the frame rate of the video. In this case, the time-stamp of each frame in the video wont match the actual time stamp and there will be a small error.

The algorithm of video synchronization is as follows. Video frames are synchronized based on the time-stamp value of each frame. When the System Clock is triggered, with respect to the time-stamp value provided by the Experimental Clock, the video synchronizer module identifies the frames with the closest time stamp with respect to the value of Experimental Clock and passes them to the video display module. When the video display module receives the frames from all the videos, it displays them to the analyst.

Video Seeking

One of the main restrictions in opencv is that it does not have the feature of video seeking to any desired frame. Video can only be advanced to particular frame numbers which are determined by the position of key frames. The position of the key frames can be determined when you encode the video using x264. To seek the video to any desired frame number, we created an algorithm which seeks to the closest key frame before the desired frame number and sequentially reads the frame till the desired frame number is reached.

When the video is encoded, the interval between the key frames should be set to a small value so that the distance between the desired frame and the key frame is small. In case this distance is large, every time the analyst seeks the video, the computer has to sequentially read all the frames between the key frame and the desired frame which can take a considerable amount of time. Hence the video seek will be slow. To avoid this issue, the key frame interval has to be set to a value small when encoding the video.

Video seeking plays a major role in our visualization tool. For example, when the analyst identifies a peak in the sensor data at time x in the animated chart display module, he does not have to wait till time x. By clicking directly at the peak, the video is advanced to the frame number with respect to time x.

Video Zooming

Zooming the video is done by pausing the video and drawing a rectangle over the desired area. The coordinates of the rectangle with respect to the video are determined by our user interface and uses a opencv library for rescaling the desired area. The analyst can select any desired area of the video he wants to visualize in detail such as the pupil of the eye, mouth etc. For example, he can compare the size of the pupil with the data he has plotted and see if there is are correlation between the two inputs.

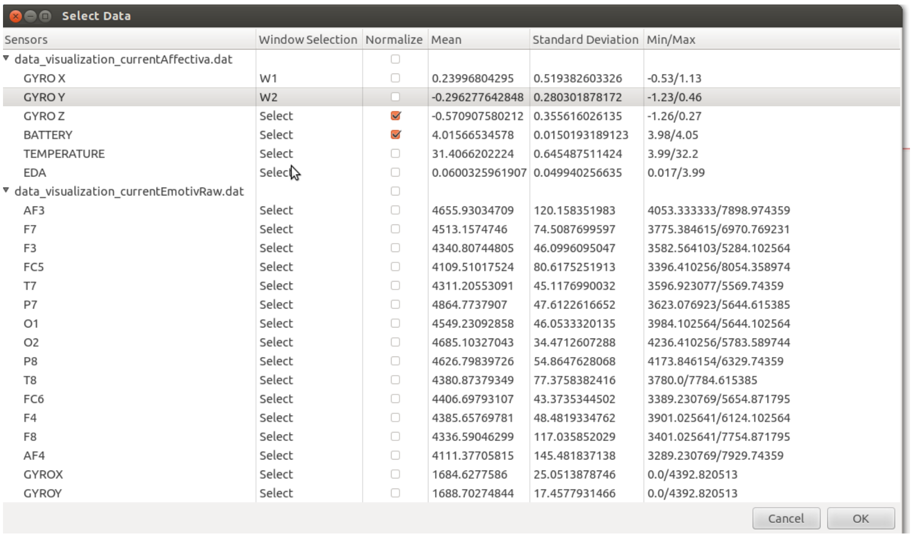
Video Display Module

The video display module is serially connected to the video synchronizer module. The frames synchronized in the video synchronizer module are automatically passed to the video display module that handles data display to the analyst. The area of the video display is determined by the number of videos the analyst wants to visualize. This size is computed to fit the available space in the user interface. The videos are rendered to the analyst using the GTK drawing area.

3.2.3 Data Selection Module

Data Selection module uses the list of instances of class ”data” created in the input module. It contains the time series data along with information such as its name and the sensor it belongs. This module allows the analyst to select which information he wants to plot and analyze. For example, if the analyst wants to plot the raw EEG from 3 channels from 19 channels available, he selects the desired three of them using this module. He can visualize and analyze the data of his choice. Also, the data can be plotted in three different areas in our interface which we refer to as windows. This module allows the user to choose the window he wants to plot the data. The Figure 3.10 shows our data selection module. It contains two sensors EEG(Emotiv) and EDA. Each sensor has multiple information as shown in the Figure 3.10.

The analyst has the option of normalizing the data before plotting, and in addition, the module also provides basic statistical metrics such as maximum, mean, minimum and standard deviation for each of the informations provided by the sensor.



Sensors	Window Selection	Normalize	Mean	Standard Deviation	Min/Max
▾ data_visualization_currentAffectiva.dat		□			
GYRO X	W1	□	0.23996804295	0.519382603326	-0.53/1.13
GYRO Y	W2	□	-0.296277642848	0.280301878172	-1.23/0.46
GYRO Z	Select	☑	-0.570907580212	0.355616026135	-1.26/0.27
BATTERY	Select	☑	4.01566534578	0.0150193189123	3.98/4.05
TEMPERATURE	Select	□	31.4066202224	0.645487511424	3.99/32.2
EDA	Select	□	0.0600325961907	0.049940256635	0.017/3.99
▾ data_visualization_currentEmotivRaw.dat		□			
AF3	Select	□	4655.93034709	120.158351983	4053.333333/7898.974359
F7	Select	□	4513.1574746	74.5087699597	3775.384615/6970.769231
F3	Select	□	4340.80744805	46.0996095047	3582.564103/5284.102564
FC5	Select	□	4109.51017524	80.6175251913	3396.410256/8054.358974
T7	Select	□	4311.20553091	45.1176990032	3596.923077/5569.74359
P7	Select	□	4864.7737907	47.6122616652	3623.076923/5644.615385
O1	Select	□	4549.23092858	46.0533320135	3984.102564/5644.102564
O2	Select	□	4685.10327043	34.4712607288	4236.410256/5783.589744
P8	Select	□	4626.79839726	54.8647628068	4173.846154/6329.74359
T8	Select	□	4380.87379349	77.3758382416	3780.0/7784.615385
FC6	Select	□	4406.69793107	43.3735344502	3389.230769/5654.871795
F4	Select	□	4385.65769781	48.4819334762	3901.025641/6124.102564
F8	Select	□	4336.59046299	117.035852029	3401.025641/7754.871795
AF4	Select	□	4111.37705815	145.481837138	3289.230769/7929.74359
GYROX	Select	□	1684.6277586	25.0513878746	0.0/4392.820513
GYROY	Select	□	1688.70274844	17.4577931466	0.0/4392.820513

Figure 3.10: Data Selection Module

3.2.4 Animated Chart Display Module

The Animated Chart Display module contains three windows where the analyst can plot the data. We use matplotlib to plot the data which provides an object-oriented API for embedding plots into applications using GTK. This module plots the data selected by the user in data selection module.

The most important feature of this module is the animated black vertical line which we call the ”time indicator”. The module plots the time indicator line in all the windows and its position is determined by the value of the Experimental Clock. When the value of the Experimental Clock changes, the position of the time indicator changes as shown in Figure 3.11. The time indicator is an important feature because it shows the data value of a sensor at time t with respect to the image of a video at that time. The analyst, for example, can monitor what was the facial expression of the user during the peak in a raw EEG data. So if the user was blinking his eyes, the analyst may infer that peak might be the result of blinking and not due to the cognitive-state of the student bring monitored.

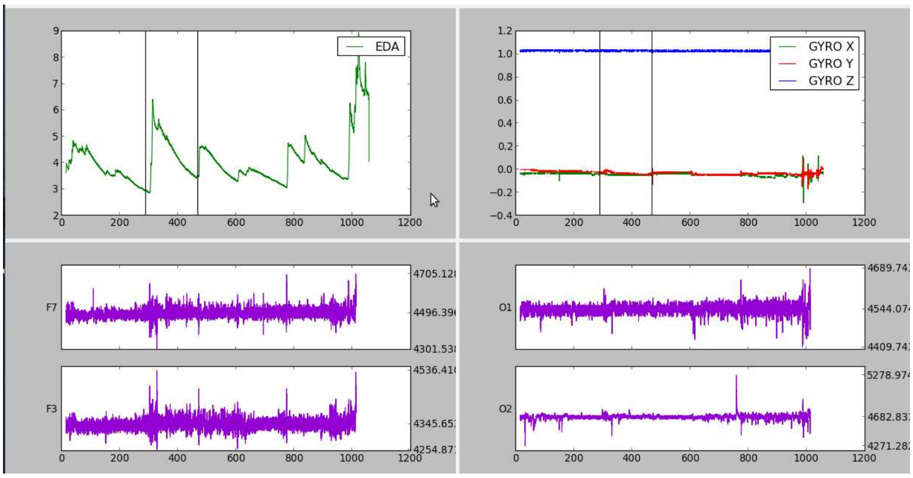


Figure 3.11: Time indicator is highlighted by the red vertical line

Zooming of the graph can be done by clicking the zoom button in the interface and drawing a rectangle over the desired area. The coordinates of the rectangle are determined by our tool and and the graph is redrawn based on the coordinate values of the rectangle. If the analyst wants to return back to original plot, he just have to click the zoomfit button in the GUI.

The Animated Chart Display module consists two types of charts. The charts at the top of our data visualization tool plots all the data in the same graph while the charts at the bottom of our tool, plot all the data separately as shown in the figure 3.11.

3.3 Statistics and Signal Processing Techniques

Our data visualization tool is capable of performing several signal processing techniques such as Fourier transform, Short time Fourier transform and Wavelet transform . These transformations are applied to the data captured from the sensor to obtain further information which cannot be seen in the time-domain format. In this section, we explain briefly the theoretical background of these transforms and explain how our tool could be utilized to apply these transforms.

3.3.1 Fourier Transform

The Fourier transform describes a continuous signal x(t) as a linear combination of sinusoidal basis functions in which each of these basis functions is a complex exponential of a different frequency. The Fourier transform of a continuous signal x(t) is defined by

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{-i2\pi ft} dt \tag{3.1}$$

where

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{i2\pi ft} dt \tag{3.2}$$

is a complex exponential of a frequency f. Equation 3.2 is the continuous Fourier transform of the signal x(t).

The square of the absolute value of X(f) is given by

$$|X(f)|^2 = \Re[X(f)]^2 + \Im[X(f)]^2 \tag{3.3}$$

and it corresponds to the power of the frequency f. The larger the value of $|X(f)|^2$ the greater is the importance of the corresponding frequency to the signal x(t).

In continuous Fourier Transform, the input signal x(t) and output signal X(f) are real-valued functions of t and f and so their values must be discretized. The discretization of the input signal in time domain is usually performed at constant intervals Δ as shown in equation 3.4 and applying the Discrete-Time Fourier transform.

$$x(n) = \{x_0, x_1, \ldots, x_{N-1}\} = \{x_i\}, x_i \text{ is the sample taken at time } t_i = t_0 + i\Delta \tag{3.4}$$

The highest frequency that can be detected for the sampling period Δ is given by the Nyquist frequency $F_n = \frac{1}{2\Delta}$.

The Discrete-Time Fourier transform (DTFT) of the signal x(n) is

$$X(e^{i\omega}) = \sum_{n=-\infty}^{\infty} x(n) e^{-i\omega n}, \tag{3.5}$$

where ω denotes the digital frequency and its inverse is

$$x(n) = \frac{1}{\pi} \int_{2\pi} X(\omega) e^{i\omega n} \tag{3.6}$$

In equation 3.5, it can be noted that the DTFT $X(e^{i\omega})$ is a periodic function with period 2π radians. Dividing the digital frequency in the interval $(0, 2\pi)$ into N points discretizes the digital frequency and yields the Discrete Fourier Transform (DFT). DFT of x(n) is given by

$$X(\omega)|_{\omega=2\pi\frac{k}{N}} = X(k) = \sum_{n=0}^{N-1} x(n) e^{-i2\pi k\frac{n}{N}} \quad k = 0, \ldots, N-1 \tag{3.7}$$

where

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{i2\pi k\frac{n}{N}} \quad n = 0, \ldots, N-1 \tag{3.8}$$

To obtain the value of the frequency from DTFT, we use the fundamental relation between the digital frequency and analog frequency which is given by

$$\begin{aligned} \omega = \Omega\Delta &= 2\pi f\Delta, \text{ where } \Delta \text{ is the sampling period.} &(3.9)\\ f &= \frac{\omega}{2\pi\Delta} &(3.10)\end{aligned}$$

In case of DFT where ω is discrete in nature, the frequencies are given by

$$\begin{aligned} \omega &= 2\pi f_k\Delta, \quad k = 0, \ldots, N-1 \\ &= 2\pi\frac{k}{N} = 2\pi f_k\Delta \rightarrow f_k = \frac{k}{N\Delta}, \quad k = 0, \ldots, N-1 &(3.11)\end{aligned}$$

In our data visualization tool, the Fourier Transform module was built using the fft module provided by numpy [42] to compute the DFT efficiently. The output of the Fourier transform module returns the square of the absolute value of $X(\omega)|_{\omega=2\pi\frac{k}{N}}$ which corresponds to the power of the frequency at $\omega = 2\pi\frac{k}{N}$ in the equation 3.7. To convert $\omega = 2\pi\frac{k}{N}$ into frequencies f_k, the Fourier Transform module uses equation 3.11.

The graphical interface of our Fourier Transform Module shown in Figure 3.12 contains the following fields.

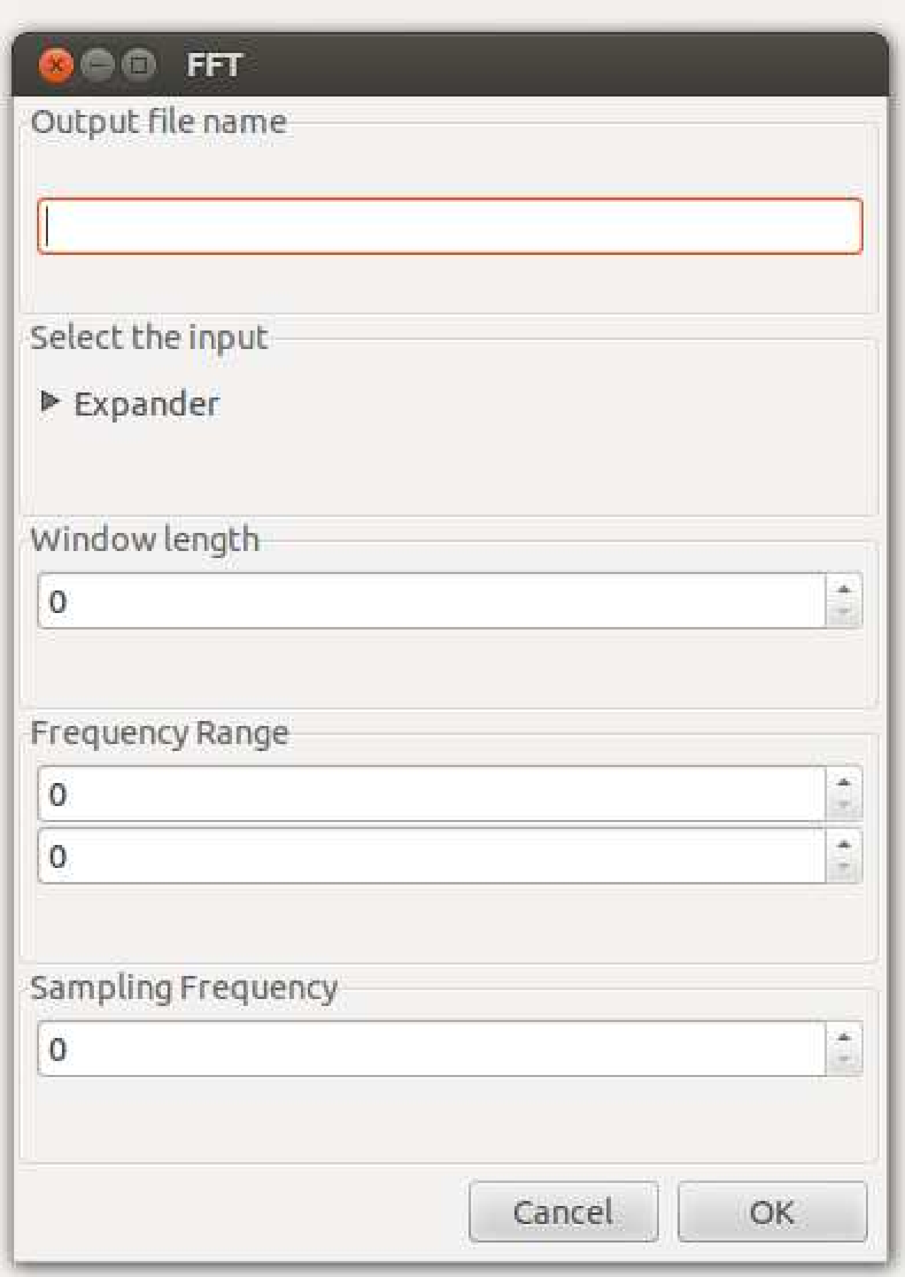
Input signal : The input signal for which the Fourier Transform should be computed. More than one input signal can be selected.

Output file name : The name that should be assigned to the output returned by the Fourier Transform Module. In our data visualization tool, the output file name is the concatenation of the name of the input signal and the name assigned to the "output file name" chosen.

Window Length : This field is used for Short Time Fourier Transform which will be explained in the next section. For computing Discrete Fourier transform 0 should be assigned to this field.

Frequency Range : The value of lower frequency and upper frequency should be assigned to this field. The Fourier Transform module passes frequencies within this range and rejects frequencies outside the range. When zero is assigned to lower frequency and upper frequency, the Fourier Transform module outputs all the frequencies and thus no frequency is rejected.

Sampling Frequency : Sampling frequency is the number of samples in the input signal per second. This is equal to $\frac{1}{\Delta}$, where Δ is the constant interval in which the input signal is discretized.



FFT

Output file name

Select the input

▶ Expander

Window length

0

Frequency Range

0

0

Sampling Frequency

0

Figure 3.12: The Fourier Transform Module

3.3.2 Short Time Fourier Transform

One of the main drawbacks of the Fourier transform is that it is not possible to determine whether a signal includes a particular frequency at a particular point in time. For example, Figure 3.13 contains 4 graphs where the plots on the left represent the time domain signal and those on the right represent the spectrum of the signal. The plot on the top left side represents f(t) where

$$f(t) = \sin(2\pi 5t) + \sin(2\pi 20t) + \sin(2\pi 40t), \text{ for } t = 0, 0.01, \ldots, 10$$

and the plot on the bottom left side represents $f_1(t)$ where

$$\begin{aligned} f_1(t) &= \sin(2\pi 5t), \text{ for } t = 0, 0.01, \ldots, 3 \\ f_1(t) &= \sin(2\pi 20t), \text{ for } t = 3, 3.01, \ldots, 7 \\ f_1(t) &= \sin(2\pi 40t), \text{ for } t = 7, 7.01, \ldots, 10 \end{aligned}$$

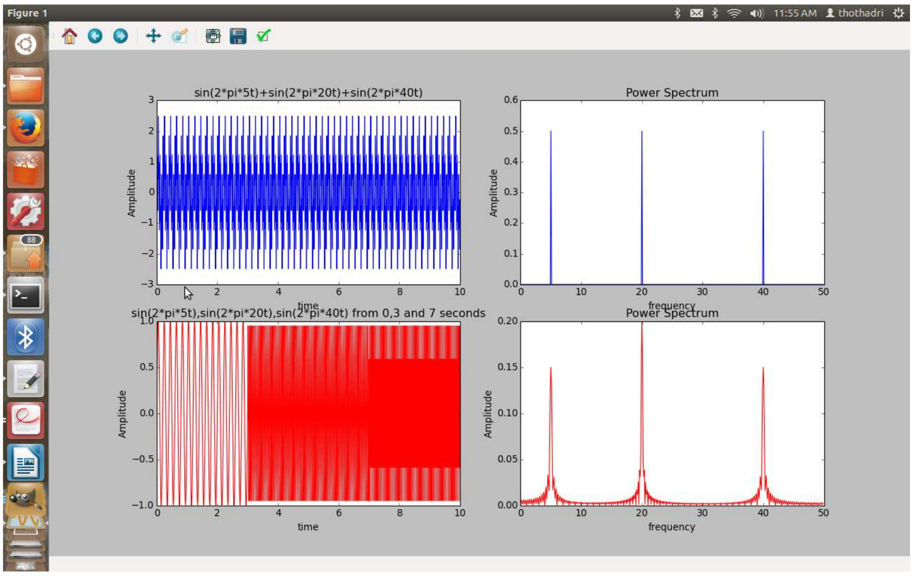


Figure 3.13: f(x) and $f_1(x)$ and dft of both

Notice that the frequencies 5,20 and 40 exists for the whole interval of f(x) while in the case of $f_1(x)$, frequencies 5,20 and 40 exists only for certain time intervals. But when we consider the spectrum of both $f(x)$ and $f_1(x)$ which are shown at the right side of the figure 3.13 , they are very similar and it is not possible to determine which frequencies occur for each time interval.

The Short Time Fourier transform (STFT) method divides the time domain signal into small intervals of time, which are usually called as windows, and applies Discrete Fourier Transform over each of these intervals. If the window length is infinite, then the short time Fourier transform becomes Fourier transform.

The graphical interface to compute Short Time Fourier Transform is the same as that used for the Fourier Transform Module (see figure 3.12). In this case, the user has to assign a value for the window length. This window is the size of the intervals in which the time domain signal is divided and the DFT is computed for each of these intervals.

The STFT has also its disadvantages such as if the window is of finite length then no longer perfect frequency resolution can be achieved. The narrower the window, the better the time resolution and poorer the frequency resolution. The wider the window, the better the frequency resolution and the poorer the time resolution. So, short pulses in the input signal can be easily localized in time if the window is narrow but the resolution of frequency would be poor. But when the window is wider, short pulses cannot be localized in time, but the frequency resolution would be better.

3.3.3 Wavelet Transform

The wavelet analysis overcomes the shortcomings of Short Time Fourier Transform. In continuous wavelet transform, a scalable window is shifted along the signal and for each position of the window the wavelet transform is computed. The wider the window the better is the frequency resolution and the narrower is the window the better is the time resolution. The Fourier transform represents input signal in terms of sine and cosines while the wavelet transform represents the input signal in terms of shifted and scaled versions of the so called mother wavelet. Figure 3.14 shows different kind of mother wavelets. A mother wavelet is selected based on the function of characteristics of the input signal to be processed [26].

Formally, the continuous wavelet transform is defined as

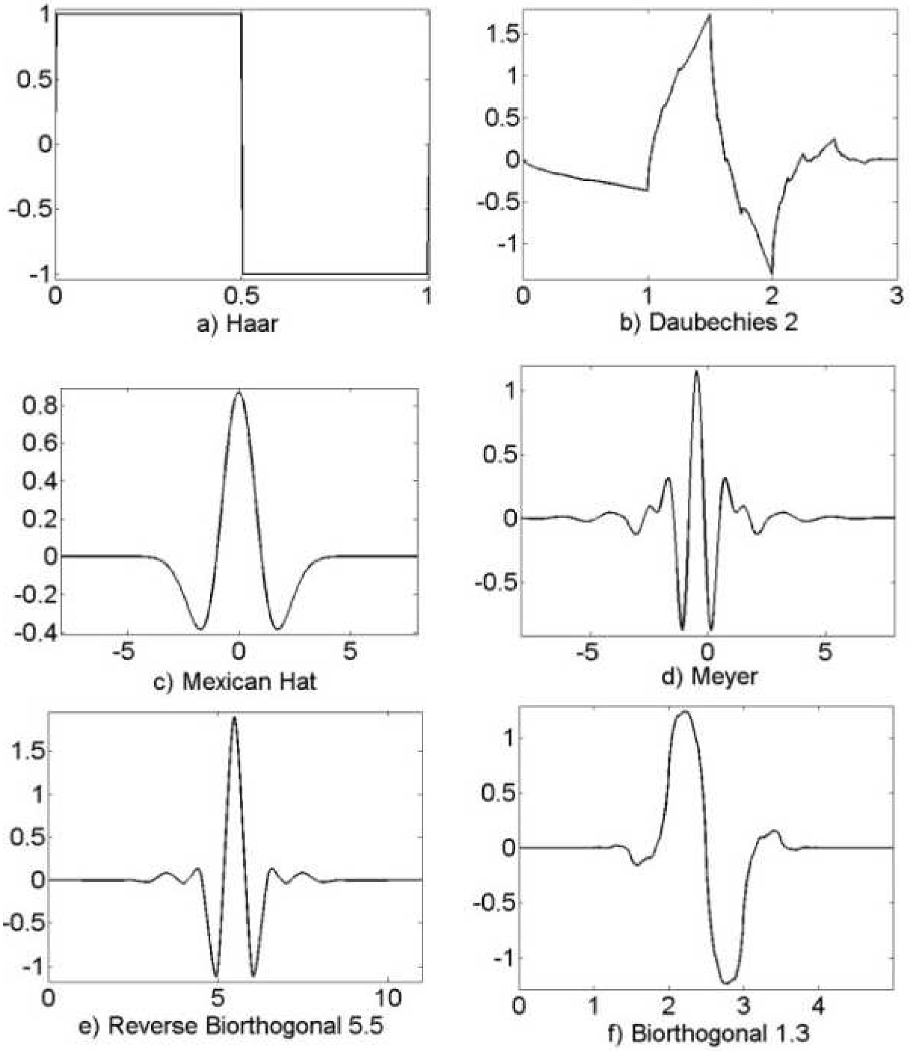
$$CWT(\tau, s) = \frac{1}{\sqrt{s}} \int x(t)\psi^*(\frac{t-\tau}{s})dt \tag{3.12}$$

where τ and s are the translation and scaling parameters. $\psi(t)$ represents the mother wavelet , $\psi(t-\tau)$ represents the mother wavelet shifted by τ units and $\psi(\frac{t-\tau}{s})$ represents the mother wavelet scaled by a factor of s and shifted by τ units.

The translation parameter τ of CWT represents the amount of shift in the wavelet and the scale parameter s represents the factor with which the wavelet is scaled. The wavelet dilates for values $s > 1$ and compresses for values $s < 1$. The higher the value of s, the more dilated is the wavelet. A dilated wavelet allows the detection of slow signal changes, that is, low frequencies . On the other hand, a compressed wavelet allows the detection of fast signal changes, that is, higher frequencies. Figure 3.15 shows the shifting and scaling operation of CWT for the input signal Y(t).

It is important to note that the CWT cannot be practically computed for all values of τ and s. So CWT is computed for only certain values of s and τ which we call discretization of CWT. However, there is another method to compute the wavelet transform with a significant reduction in the computing time and it is called discrete wavelet transform (DWT).

In DWT, the input signal is passed through a series of half band high pass filter to



a) Haar

b) Daubechies 2

c) Mexican Hat

d) Meyer

e) Reverse Biorthogonal 5.5

f) Biorthogonal 1.3

Figure 3.14: wavelets

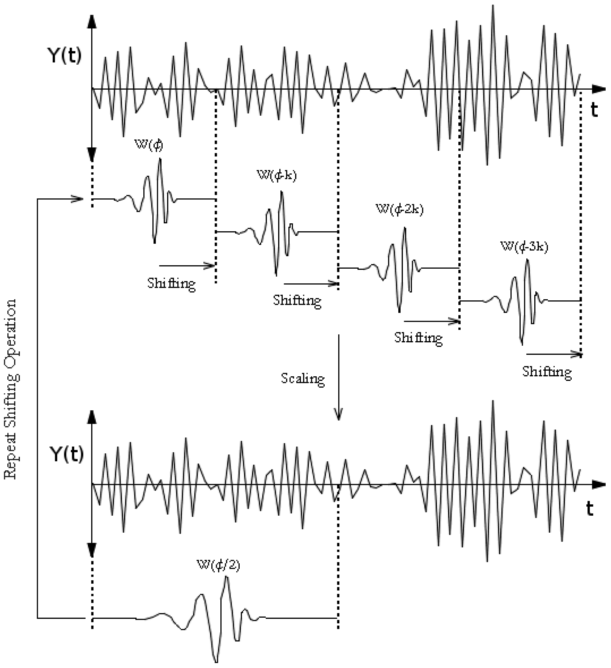


Figure 3.15: Scaling and Shifting operation of Continous Wavelet Transform

analyze the high frequency components which are called details and passed through a series of half band low pass filters to analyze the low frequency components which are called approximations.

In Figure 3.16, an input signal of 1000 samples of bandwidth (0,F_s), where F_s is the sampling frequency, is passed through a half band low pass filter and half band high pass filter. The half band low pass filter generates an output signal which spans half the frequency band (0,$\frac{F_s}{2}$). The half band high pass filter generates output signal which spans half the frequency band ($\frac{F_s}{2}$,F_s). Since both the output signals only have a bandwidth of $\frac{F_s}{2}$, the sampling rate could be reduced to half without any loss of information according to the Nyquist theorem. This could be done by removing every other sample of the output signal and this is called the down sampling operation. The down sampling operation reduces the time resolution as the output signal is represented by half the number of samples and doubles the frequency resolution since the bandwidth $\frac{F_s}{2}$ is now represented in half the number of samples. The down sampling operation determines the scale parameter in the Discrete Wavelet Transform and down sampling by 2 doubles the scale. cA in the figure represents approximation coefficient and cD represents the detail coefficient. This whole process is called decomposition process.

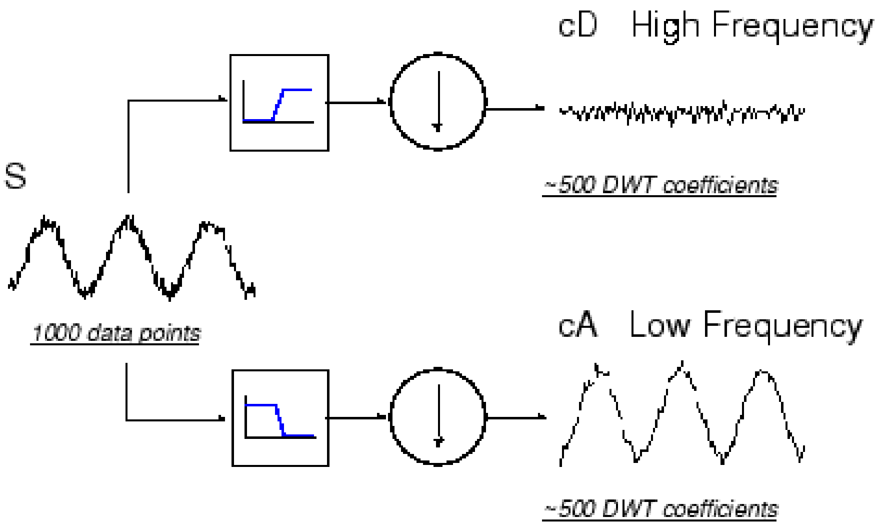


Figure 3.16: One stage filtering

The input signal can be broken down into many lower resolution components by iterating the decomposition process, with successive approximations being decomposed in turn. This is called the wavelet decomposition tree. Figure 3.17 shows a three-level wavelet decomposition tree and g(n) represents half band low pass filter and h(n) represents half band high pass filter.

The decomposition can proceed only until the signal consist of a single sample. For performing DWT, the input signal should be of length which is a multiple of 2^n

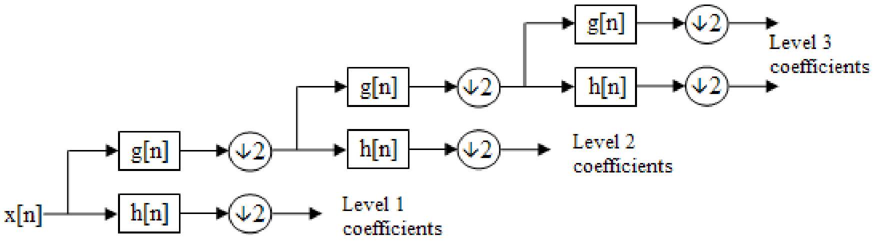


Figure 3.17: 3-level wavelet decomposition

where n is the number of levels. To make sure that input signal is a multiple of 2^n, the input signal could be padded with zeros.

In our data visualization tool, the Discrete Wavelet Transform module was built using the pywavlet module [51]. The output of Discrete Wavelet transform module returns the detail coefficients for all the levels until n and approximation coefficients for the level n. For example, in the case of 3-level decomposition wavelet tree as shown in the Figure 3.17, the output contains level 1, level 2 and level 3 coefficients. Level 1 and level 2 contains only detail coefficients while level 3 contain both detail and approximation coefficients.

The graphical interface of our Wavelet Transform Module is shown in Figure 3.18 and contains the following fields.

Input signal : The input signal for which the Wavelet Transform should be computed. More than one input signal can be selected.

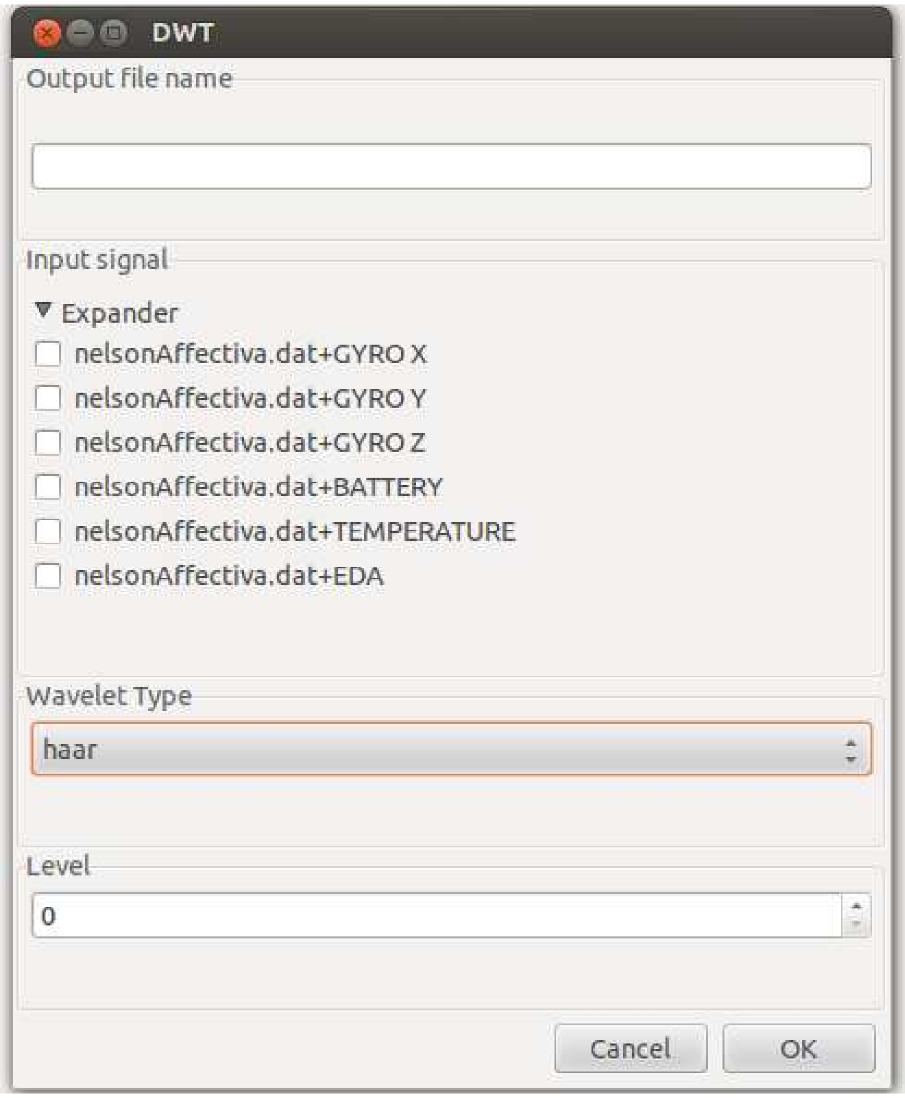
Output file name : The name that should be assigned to the output returned by the Wavelet Transform Module. In our data visualization tool, the output file name is the concatenation of the name of the input signal, the name assigned to the output file name, the decomposition level and ”a” if the output signal is an approximation or ”d” if the output signal is a detail .

Wavelet type : This field lists all the types of mother wavelets available in the tool. Only one type of mother wavelet can be selected at a time.

Level : The maximum desired level of wavelet decomposition.

3.3.4 Statistical Module

In this module, basic statistics details such as mean, standard deviation can be calculated for any desired interval of the sensor data. The analyst can create events using the input module and compute statistics of the sensor data during these events by choosing the option ”Event Selection” in the data selection module. Figure 3.19 shows the mean and standard deviation computed during the peak event.



DWT

Output file name

Input signal

▼ Expander

□ nelsonAffectiva.dat+GYRO X

□ nelsonAffectiva.dat+GYRO Y

□ nelsonAffectiva.dat+GYRO Z

□ nelsonAffectiva.dat+BATTERY

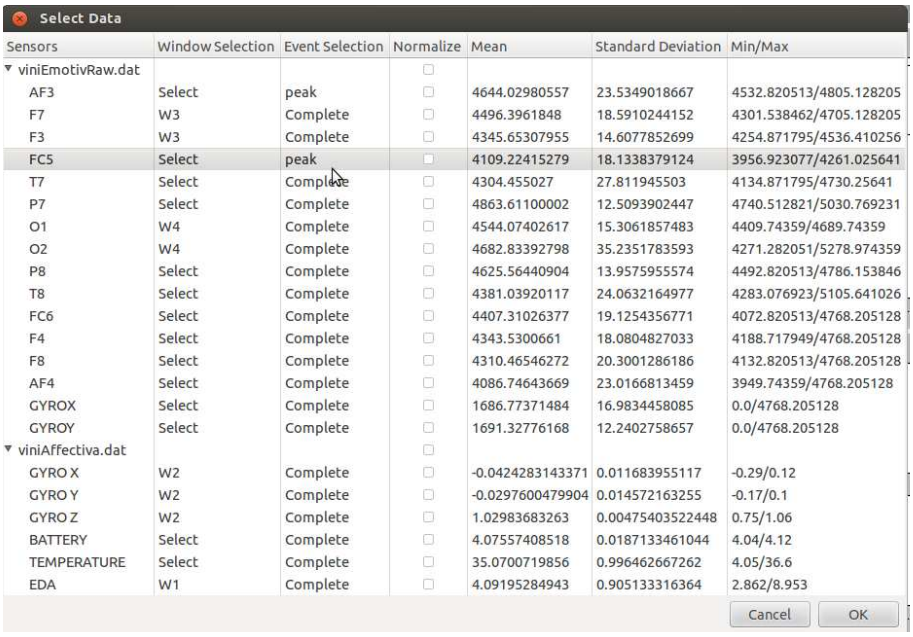
□ nelsonAffectiva.dat+TEMPERATURE

□ nelsonAffectiva.dat+EDA

Wavelet Type: haar

Level: 0

Figure 3.18: DWT Module



Sensors	Window Selection	Event Selection	Normalize	Mean	Standard Deviation	Min/Max
viniEmotivRaw.dat			□			
AF3	Select	peak	□	4644.02980557	23.5349018667	4532.820513/4805.128205
F7	W3	Complete	□	4496.3961848	18.5910244152	4301.538462/4705.128205
F3	W3	Complete	□	4345.65307955	14.6077852699	4254.871795/4536.410256
FC5	Select	peak	□	4109.22415279	18.1338379124	3956.923077/4261.025641
T7	Select	Complete	□	4304.455027	27.811945503	4134.871795/4730.25641
P7	Select	Complete	□	4863.61100002	12.5093902447	4740.512821/5030.769231
O1	W4	Complete	□	4544.07402617	15.3061857483	4409.74359/4689.74359
O2	W4	Complete	□	4682.83392798	35.2351783593	4271.282051/5278.974359
P8	Select	Complete	□	4625.56440904	13.9575955574	4492.820513/4786.153846
T8	Select	Complete	□	4381.03920117	24.0632164977	4283.076923/5105.641026
FC6	Select	Complete	□	4407.31026377	19.1254356771	4072.820513/4768.205128
F4	Select	Complete	□	4343.5300661	18.0804827033	4188.717949/4768.205128
F8	Select	Complete	□	4310.46546272	20.3001286186	4132.820513/4768.205128
AF4	Select	Complete	□	4086.74643669	23.0166813459	3949.74359/4768.205128
GYROX	Select	Complete	□	1686.77371484	16.9834458085	0.0/4768.205128
GYROY	Select	Complete	□	1691.32776168	12.2402758657	0.0/4768.205128
viniAffectiva.dat			□			
GYRO X	W2	Complete	□	-0.0424283143371	0.011683955117	-0.29/0.12
GYRO Y	W2	Complete	□	-0.0297600479904	0.014572163255	-0.17/0.1
GYRO Z	W2	Complete	□	1.02983683263	0.00475403522448	0.75/1.06
BATTERY	Select	Complete	□	4.07557408518	0.0187133461044	4.04/4.12
TEMPERATURE	Select	Complete	□	35.0700719856	0.996462667262	4.05/36.6
EDA	W1	Complete	□	4.09195284943	0.905133316364	2.862/8.953

Figure 3.19: Statistical Details computed for the Peak event

Chapter 4

Experimentation: Methodology and Results

In this chapter, we explain the set of experiments conducted to exemplify how our data visualization tool can be used . Section 4.1 explains the tasks the participants of the experiments were asked to do and, in section 4.2, the experimental setup, which includes the sensors utilized, the environment, the interface and experiment methodology are presented. Finally in section 4.3 , we show the results.

4.1 Task Selection

In the experiment, participants were asked to perform three tasks which include playing the "games" nback and tower of hanoi and watching a section of the video lectures of CEDERJ. Studies [15, 29, 31] have shown that the tasks, nback and tower of hanoi are used by researcher's to classify the cognitive states of the user using the Electroencephalogram (EEG) sensor.

nback

In the nback task, participants are presented a random sequence of stimuli in which each stimuli is shown one at a time. The n in nback refers to the number of previous stimuli the participant should remember to perform the matching task. For example, in the case n=1, participants should respond whether or not the current stimuli is the same as the previous one and in the case of n=2, participant should respond whether or not the current stimuli is the same as what they saw two presentations before. In order to perform the matching task, participants should always keep a sequence of stimuli in their "working memory and update it when the new stimuli is presented. "

The following sequence of characters, A B R **B** J C F **C** H K **H** **K** R, demonstrates the nback stimuli's. In the case of n=2, the alphabets in bold indicate the matching stimuli, that is, it is exactly equal to the stimuli presented 2 steps before. Stimuli could be alphabets, numbers or spatial position. In the case of spatial position , instead of memorizing images, the participant should remember the position of the stimuli and match it with the nth stimuli presented before.

Many studies have used the nback task to measure the user "working memory". The term "working memory" refers to the temporary storage and processing of necessary information inside the brain for cognitive tasks such as learning and reasoning [5]. Working memory involves storage of new information as well as processing of new information. This makes working memory one of the major cognitive component for learning, because when one learns, he has to store new information as well as process them [4]. The works of [29, 31] used the nback test with electroencephalogram to classify the user working memory.

We intend to use the nback test to exemplify that it is possible to extract evidences from the sensors on working memory load using our data visualization tool. It is evident from studies [29, 31] that user's put more effort to do a matching task in 3-back when compared to 2-back and 1-back.

Tower of Hanoi

In tower of hanoi, three rods are presented to the user in which the first rod contains n discs arranged in descending order of decreasing disc size, smallest one at the top and biggest one at the bottom. The objective of the user is to move this entire stack to the third rod with the use of the second rod. However, user can move only one disc at a time and no disk can be placed over a smaller disc. Figures 4.1 and 4.2 show the starting and ending configuration of the game.

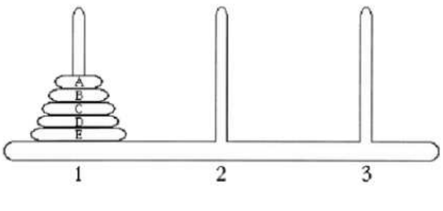


Figure 4.1: Staring Configuration

The tower of hanoi can be solved using a specific algorithm. Minimum of $2^n - 1$ steps is required to move the entire n disks from the first rod to the third rod. For people who do not know the algorithm, it is not easy to solve the puzzle and in case of higher number of discs, it would seem even impossible to solve the puzzle. In [15], participants played tower of hanoi for different values of n with an EEG sensor attached to them. Their stress level was estimated for each value of n and it was

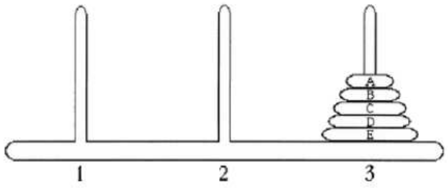


Figure 4.2: Ending Configuration

noted that stress level were high for higher value of n when compared to the lower value of n.

Video Lecture

In this task, a subject is presented through a video lecture from the CEDERJ computer system course [17]. The CEDERJ computer system course is an undergraduate distance learning program and covers a wide range of areas. The objective of the user is to listen and understand the video lecture.

Each video lecture contains a pre-recorded video and associated slides synchronized with the video. The lecture contains a list of topics, so that each student could jump to any topic at any time. The student also has the option to pause or jump forward the video lecture at any time. The average duration of a video lecture is 40 minutes.

4.2 Experimental Setup

4.2.1 Sensors Utilized

For conducting experiments we use four sensors. They are Emotiv's neuroheadset, Affectiva's Qsensor, a normal video camera and an infrared camera. Neurosky's Mindband and Emotiv's neuroheadset measures the brain activity of the participant and Affectiva's Qsensor measure the skin conductance of the user. Normal video camera captures the facial video input of the user and infrared camera captures the facial video input and tracks the user's eye.

Q sensor

We use Affectiva's Qsensor to record the EDA data. It contains 2 electrodes and is attached around the wrist of the user as shown in the Figure 4.3. The electrodes could also be placed either in the palm or fingers as shown in the Figure 4.4.

Qsensor measures skin conductance, skin temperature and hand acceleration and transmits it via Bluetooth technology. The hand acceleration is measured by

Figure 4.3: Q Sensor

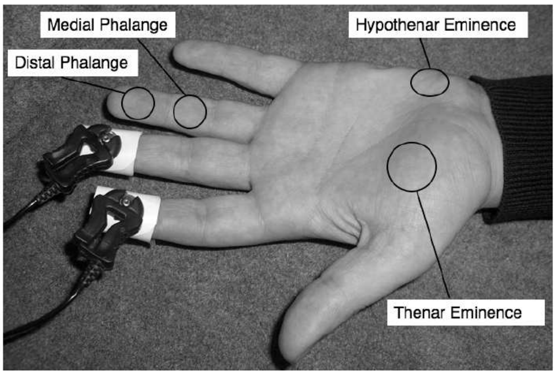


Figure 4.4: Alternate Electrode Placement Locations

accelerometers that registers motion changes in three directions x, y and z . The Qsensor can be configured to record data samples from 2 to 32 times per second.

In our experiment, electrodes from Qsensor were placed in the distal phalange of the index and middle finger as shown in the Figure 4.4 instead of palm and wrist. According to [7] reading EDA from distal phalanges is more sensitive to habituation and has higher amplitudes when compared to other locations.

Emotiv

Emotiv's Neuroheadset measures the brain activity, EEG, of the user and it comes with 14 electrodes and 2 reference electrodes. The positioning point of these electrodes are AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4 and they are shown geographically in the Figure 4.5. The headset transmits raw EEG data sampled at 128Hz from all the electrodes via Bluetooth technology to the computer.

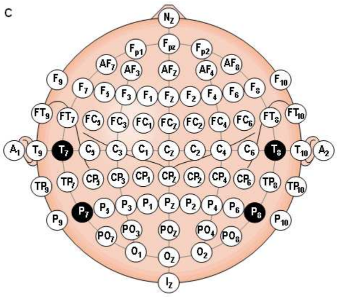


Figure 4.5: Electrode placement positions

Emotiv neuroheadset also provides additional information interpreted from the EEG data which includes emotional states and cognitive-states such as engagement, boredom, excitement, frustration and meditation. Figure 4.6 shows the Emotiv's neuroheadset.

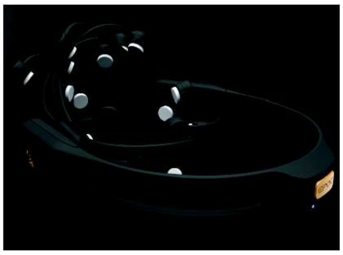


Figure 4.6: Emotiv Neuroheadset

Normal video camera and Infrared video camera

A normal video camera, Logitech 920C, of resolution 1280x720 is used to record the user's facial video. With an infrared camera and IR LEDS, we create dark pupil tracking illumination setup to visualize the pupil size and pupil movements of the user. Infrared camera is developed from a normal webcam, Logitech 920C, by placing a filter which stops visible light but allows infrared.

In the dark pupil tracking illumination setup, the eye is illuminated by IR light at an angle from the IR LEDS located very close to infrared camera. The eye reflects the illumination but the pupil absorbs IR light, so pupil appears to be much darker than the iris in the video input from the infrared camera. Figure 4.7 shows the dark pupil tracking illumination setup. This experimental set-up was built with a help of the PhD student participant of this project.

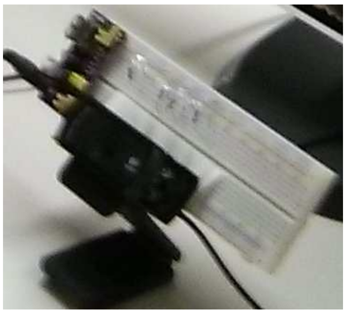


Figure 4.7: Dark pupil tracking illumination setup

4.2.2 Experimental Environment

The experiments were performed in a conference room with controlled illumination and all the experiments were realized using a LED monitor of resolution 1920*1280.

The participants were positioned approximately 1 meter away from the monitor. The participant's were attached with Emotiv's Neuroheadset and Affectiva's Qsensor. Normal video camera was positioned above the LED monitor and infrared camera was placed near the base of the monitor pointing towards the upward direction. The participant's finger was cleaned well before placing the Qsensor electrodes and all electrodes in the Emotiv's Neuroheadset were hydrated using saline solution to achieve good contact.

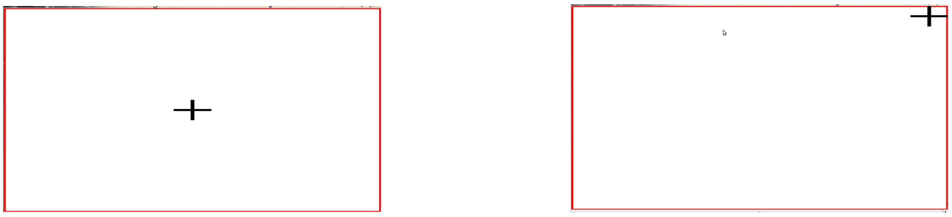
The experiments were conducted on a dedicated PC(i7 Quad core and 16GB RAM) and all the data from the sensors were recorded in the same PC. The software Mindland Reader, developed by my colleague who is a part of this project, was used to callibrate all the sensors before the experiment and once the experiment was started, the software records all the data from the sensor in the computer disk.

The Mindland Reader enables the experimenter to visualize the video data from normal and infrared web camera before the experiment. This enables the experimenter to position the normal and infrared webcam in such a way that maximum area of facial feed input is captured. In the case of Emotiv Neuroheadset, the Mindland Reader shows all the electrode locations in the screen and signals green, yellow, orange, red or black on each of those electrode locations. Green signal indicates that high quality of signal is captured from the electrode. Yellow and orange indicate good and fair quality while red and back indicate bad and very poor quality of signal is captured. This enables the experimenter to position the user's headset in such a way that Mindreader signals are green in all the electrode locations. In the case of Qsensor, Mindland Reader signals green if it receives the data from Qsensor.

The Mindland Reader records all the data from the sensor with a time stamp attached to it. Video frames are encoded using libx264 format and a timestamp for each frame of the video is provided in a different file. The data from Emotiv's neuroheadset and Affectiva's Qsensor are stored in binary format. In addition, Mindland Reader provides an option of continuously recording the screenshot of the desktop when the user starts performing the experiment. After the experiment, our data visualization tool is used to show all the data synchronized from the sensors.

4.2.3 The Interface

Two websites were built and hosted in the server `http://land.ufrj.br` where the participants performed all the tests. The websites were designed using HTML5, javascipt, svg and css. The websites consist of tasks such as nback, tower of hanoi, cederj video lecture and a calibration test. The websites were designed in such a way that it provides instructions to the participant on how to perform a task and collects feedback about their level of difficulty and interest after the task is finished.



(a) Center

(b) Topmost right corner

Figure 4.8: Baseline recording test

In addition, the website also records all the user interactions with the website.

There are two types of experiments called A and B and participants performs either experiment A or experiment B or both the experiments. A participant performs the experiment of type A in the website `http://land.ufrj.br/~thothadri/mindland.html`. Experiment A consists of calibration test, nback test and tower of hanoi test . The calibration and the nback test were implemented by us while the tower of hanoi implementation was an adaptation of the source code in the site [22].

A participant performs the experiment of type B in the website `http://land.ufrj.br/~thothadri/cederj.html`.In the experiment B, the task consists of a calibration test and a video lecture. The calibration test was implemented by us while the video lecture was took from the CEDERJ website and placed in our website.

Participants perform the Experiment A in the following order. A calibration test, play nback, calibration test and play tower of hanoi . In the website, the change of one type of task to another type is completely automatic and the participant is alerted before every change. The website also instructs the participant the details of how to perform the task before the start of the experiment.

In the calibration task, a black cross is displayed in the center of the screen. The black cross remains in the center for one minute and moves to all the corners and stay in each corner for 10 seconds as shown in the Figure 4.8.

In the nback task, the stimuli is a green colored box randomly displayed in one of the eight white colored boxes around the center box as shown in the Figure 4.9. The nback task contains a start game and Click me button.The start game button should be clicked to start the game and the clickme button should be clicked to perform the matching task. The stimuli is matched when the current stimuli presented is the same as the nth stimuli before.

The nback task contains 30 stimulis which are selected at random and displayed one at a time. The interval between two consecutive stimulis is 4 seconds. The green colored box appears during one second and is replaced by a white color box for the next 3 seconds. Then, a new green colored box appears again during one second

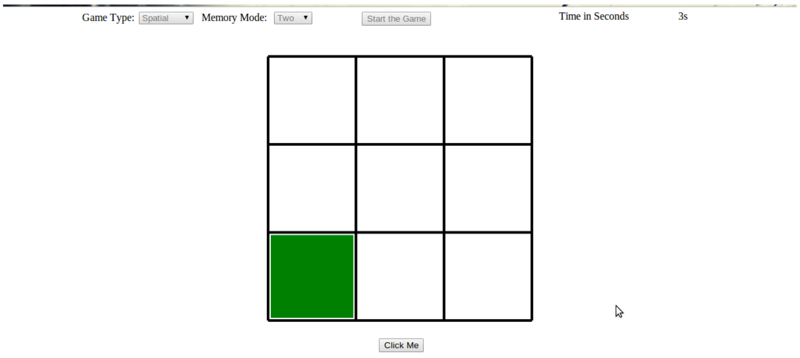

Game Type: Spatial Memory Mode: Two Time in Seconds 3s

Figure 4.9: nback test

and the process starts again.

Since the duration between two consecutive stimulis is 4 seconds, the total time required for this task is 2 minutes. We designed the task based on the work done in [29]. For any value of n in nback we made sure that out of 30 random stimulis , 8 or 9 stimulis matched.

Figure 4.10 shows an example of the nback task where n=2 and contains a sequence of 7 stimulis. For n=2, the participant has to see if the present stimuli matches with the stimuli displayed two presentations before. In this example, the matched stimulis are the third and seventh stimuli which are highlighted by a red border. So the participant always has to remember the previous two stimulis.

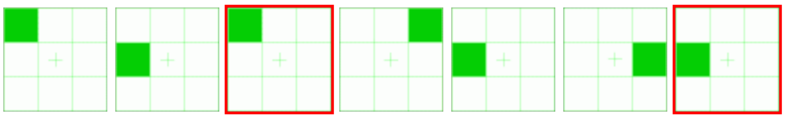
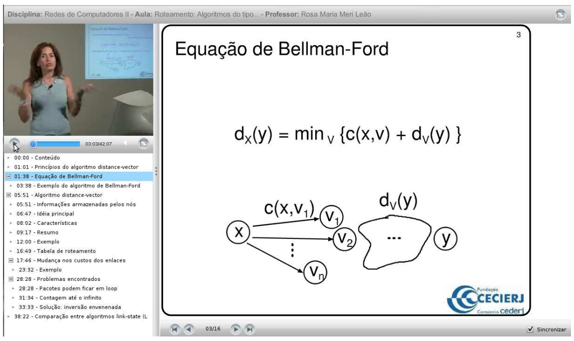


Figure 4.10: nback Sequences

In the Experiment B, the participant first make the calibration test which is the same as that in Experiment A. Then, participants watch a video lecture from the CEDERJ computer science course on Distance Vector Routing Protocol instructed by Professor Rosa Maria Meri Leão. The duration of the lecture is 42 minutes, but the participant is asked to watch only part of the lecture until the topic ”Example of Distance Vector Algorithm ”. This part of the lecture has a duration of 17 minutes. Figure 4.11 presents the screenshot of the video lecture.

4.2.4 Methodology

Ten male students aged between 20 and 30 participated in the Experiment A and three in Experiment B. All the participants were from the Federal University of Rio



Disciplina: Redes de Computadores II - **Aula:** Roteamento: Algoritmos do tipo... - **Professor:** Rosa Maria Meri Leão

Equação de Bellman-Ford

$$d_X(y) = \min_V \{c(x,v) + d_V(y)\}$$

- 00:00 - Conteúdo
- 01:01 - Princípios do algoritmo distance-vector
- 01:38 - Equação de Bellman-Ford
  - 03:38 - Exemplo do algoritmo de Bellman-Ford
- 05:51 - Algoritmo distance-vector
  - 05:51 - Informações armazenadas pelos nós
  - 06:47 - Idéia principal
  - 08:02 - Características
  - 09:17 - Resumo
  - 12:00 - Exemplo
  - 16:49 - Tabela de roteamento
  - 17:46 - Mudança nos custos dos enlaces
    - 23:32 - Exemplo
  - 28:28 - Problemas encontrados
    - 28:28 - Pacotes podem ficar em loop
    - 31:34 - Contagem até o infinito
    - 33:33 - Solução: inversão envenenada
- 38:22 - Comparação entre algoritmos link-state (L

Figure 4.11: Video Lecture

de Janeiro, Brazil. Before they started the experiment they were informed about the details of the tasks and its duration. After the participant informed that the instructions were clear, he is taken to the experiment room and the sensors were placed on him. Using Mindland Reader software, all the data from the sensor were tested and made sure that high quality data is received in the PC. Figure 4.12 shows a participant immediately before the experiment.

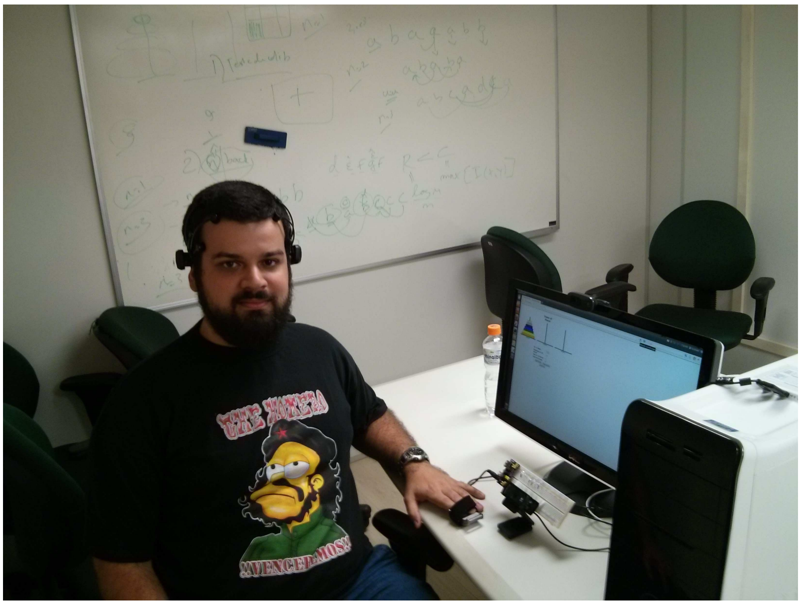


Figure 4.12: Participant immediately before the experiment

The participants were advised to start when they felt they were ready and comfortable. Once the participant has started the experiment , Experimenter leaves the room and makes sure that no one else enters the room to avoid any distractions. As explained in the previous section, all the tasks were performed in the website [23]. The participants were advised to read all the instructions provided in the website carefully.

In the case of Experiment A, the duration of the whole experiment is around 17-20 minutes. The experiment started with a two minute baseline recording where the participant was instructed to remain relaxed and follow the black cross displayed in the screen as shown in the Figure 4.8. During the first one minute, the black cross remains stationary in the middle of the screen and in the next one minute the cross starts moving towards all the corners and stays in each corner for 10 seconds as shown in the Figure 4.8.

After finishing the baseline test, the participant takes the nback task for three different values of n (1,2 and 3) and the duration of each task is two minutes leading to a total of 6 minutes. The user response time for every matching task is recorded. The feedback given by each participant on the test's difficulty level and interesting level is collected after every test.

Once the participant completes the nback session, he again does the baseline test for 30 seconds in which he looks at the stationary black cross in the middle of the screen and relaxes. After the second baseline test, the participant plays the game, tower of hanoi, for the next 8 minutes. The participant starts from 4 discs and on successful completion, the participant's feedback on the test's difficulty level and his level of interesting is collected and they move to next level, n=n+1, where n is the number of discs.

In Experiment B, the maximum duration of the whole experiment is 30 minutes. The calibration phase is the same as that of Experiment A. Once the calibration phase is over, the participant starts watching the video lecture on Distance Vector Routing Protocol till the topic Example of distance vector algorithm. The participant is instructed that he is allowed to pause, advance forward or backward during the lecture. After 30 minutes, Experiment B is automatically terminated and the participant is requested to submit a feedback on the level of difficulty, understanding and attention experienced during the video lecture.

After the completion of Experiment A and Experiment B, the participant is alerted by the web page that the experiment is over and he leaves the room and informs that the experiment is over. Then, the experimenter enters the room, stops the Mindland Reader and removes all the sensors from the participant.

4.3 Results

In this section, we exemplify how our data visualization tool could be used to help analyzing the data collected from the experiments in order to gain insights that can help building an automated tutoring system. We chose data from two experiments of type A and one experiment of type B for this purpose.

All the experiment data was collected using the software Mindland reader developed in our Laboratory. Mindland reader record the videos in .h264 format, time stamps of the video frames in a column based text file and data from EEG and EDA in the binary format. A script was created to convert this data into a format in accordance with our data visualization tool. Videos are converted into .mp4 file and the key frame interval in the video is set to 10 frames. Sensor data from EEG and EDA are converted into a column based text file where the first column represent the time stamp and the other columns represents the information provided by the sensor with respect to the time stamp.

Once the data sets are converted into the desired format, the tool allows the analyst to visualize the data of his choice. The analyst can plot any data from the sensors EEG and EDA and display any number of recorded videos. The data visualization tool synchronizes both the video and the data. A vertical line displayed in each data plot marks the specific time instant wiht respect to the video shown. The data visualization tool can also process the data from the sensors using signal processing techniques.

4.3.1 EEG analysis using the data visualization tool

The recorded data from the Emotiv's Neuroheadset includes raw EEG data from the electrodes AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4 and the speed of head movements in x and y axis. Any of these data can be chosen and plotted in the tool as shown in the Figure 4.13. As shown in the figure, the tool contains 4 windows (window1, window2, window3, window4) where the analyst can plot the data.

In windows 1 and 2, the data selected by the user is plotted in the same graph while in windows 3 and 4, the data selected is plotted in separate graphs. Hence, in window 1 or 2, the data plots can be easily contrasted, while in window 3 or 4 the plots can be individually studied.

In Figure 4.13, the raw EEG data from the electrodes P_8 and P_7 are plotted in window 1, the head acceleration parameters Gyro x and Gyro Y are plotted in window 2, the raw EEG data from the electrodes O_1 and O_2 are plotted in window 3, and in window 4 the raw EEG data from the electrodes F_7, F_3 and F_4 are plotted.

Once the data is plotted, the tool can help the analyst to search (visually) for any related patterns from the raw EEG data, facial expression and the task the studet was performing. This is done by selecting and playing the recorded video of the subject's facial expression and the recorded video of the screen shot of the task synchronized with the black vertical line over the plotted data as shown in Figure 4.13. In all the experiments conducted in our work, only simple expressions like eye blinking and head movements on EEG data were studied.

Eye blinking and head movements produce short pulses of high amplitude in the raw EEG data. Due to eye blink, a short pulse consistently appeared in the raw EEG data for the channel F_3, while for the other channels the appearance of

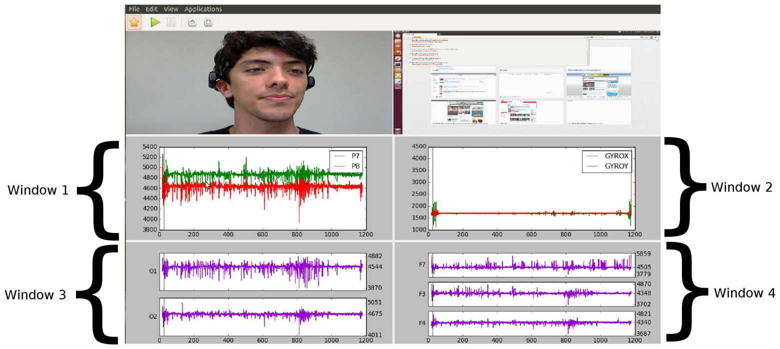


Figure 4.13: EEG plot

short pulse was not consistent. Figure 4.14 shows exactly the instant in the video where the the student blinks his eye and the corresponding peak in EEG data plot of channel F_3.

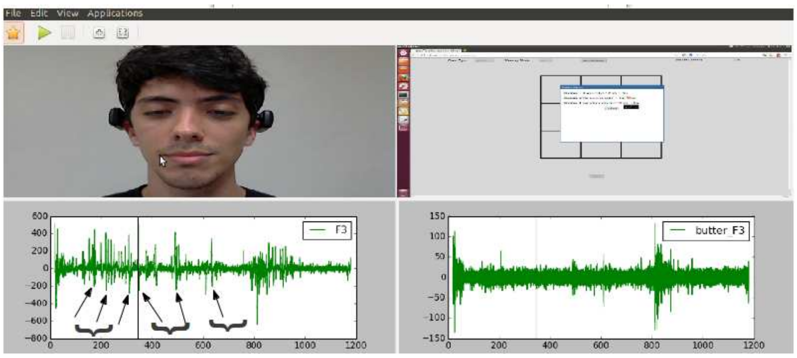


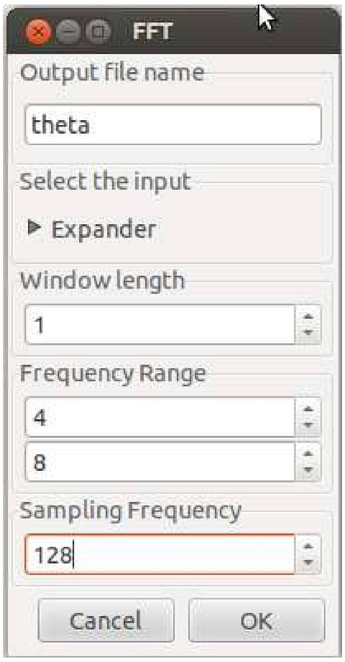
Figure 4.14: Eye Blink and EEG Correlation

The short pulses in the raw EEG data due to eye blinks are artifacts and are indicated using arrows in Figure 4.14. They distort the real EEG raw signal and contaminate the raw EEG data. In order to reduce the effects of this artifact a *butter worth filter* provided in the tool can be used. A butter worth filter is a band pass filter which passes only a band of frequencies specified by the analyst. To remove the effect of artifacts, a butter filter of order four which passes frequencies in the range (2Hz,30Hz) is used. The plot labeled ”butter_F3” shown in the Figure 4.14 shows that the specified butter worth filter has partially eliminated the blink contamination in the raw EEG data. Most of the peaks that appear in plot labeled ”F3” do not appear in the filtered data.

Several studies [4, 27, 34] have shown that the sub-band frequencies in the EEG data are related to cognitive states and emotional states. Recognizing and analyzing visually these cognitive states is a difficult task. Our data visualization tool provides an application to decompose EEG data into frequency sub-bands by applying Fourier

Transform, Short Time Fourier Transform or Discrete Wavelet Transform.

In Figure 4.15, we show the interface to specify the parameters to generate frequency sub-bands from the raw signal. The window length is set to 1 and sampling frequency is set to 128 because Emotiv samples EEG signal every ${\frac{1}{128}}^{th}$ second. Short Time Fourier Transform is computed with these parameters. To obtain the theta rhythm (4-8Hz) from the raw EEG signal the frequency range is set between 4 and 8. Figure 4.16 shows both alpha and theta computed for different window lengths 1 and 2. The time axis is the horizontal axis and the y axis is the average of the sum of power of frequencies between the allocated frequency range. After plotting, varaitions of alpha and theta and some facial expressions in the video can be observed and compared.


FFT

Output file name: theta

Select the input

▶ Expander

Window length: 1

Frequency Range: 4, 8

Sampling Frequency: 128

Figure 4.15: Interface to compute the Short Time Fourier Transfrom for Theta

4.3.2 EDA analysis using the data visualization tool

The recorded data from the QSensor includes EDA (skin conductance), battery strength of the device , the temperature of the ambient and the hand movement in x,y and z axis. All the data are sampled at 8Hz. Figure 4.17 shows a plot of all the data recorded from the Qsensor.

The EDA data generally consists of two types of components. One component is the general tonic level EDA which relates to slower acting components and the other component is the phasic component which refers to the faster changing elements of the signal. The phasic component occurs in EDA when the subject is physiologically aroused due to external or internal stimuli [8].

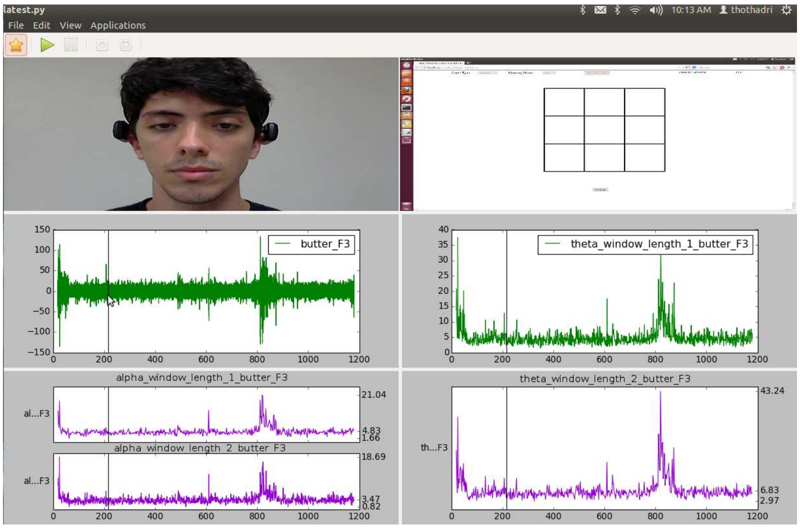


Figure 4.16: Short Time Fourier Transform of EEG signals

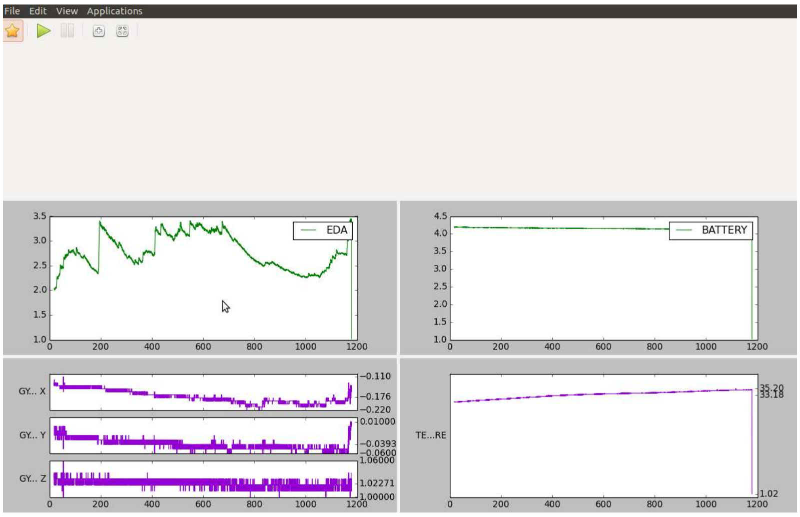


Figure 4.17: Data from Qsensor

Figure 4.18 displays the video of the recorded face, the video of the screenshot of the experiment, the plot of the skin conductance(EDA), the plot of discrete wavelet trasform of skin conductance and the plot describing the hand acceleration in the 3 axis. Using the data visualization tool, the initial point of the most of the phasic components of the EDA signal could be detected by using the wavelet daubechies 8 and applying discrete wavelet transform of level 5. Some of the phasic components are indicated by an arrow in the window 1. The data from the detail coefficient of level 5 of discrete wavelet transform has pulses at the point where the phasic component exactly starts as shown by an arrow in window 3. The discrete wavelet transform of EDA data could be used as an indicator to automatically identify the starting point of the phasic component of the EDA data. After saving this data to the disk, a threshold can be applied to understand where exactly the phasic component starts since the duration of the pulses are very short. The same method cannot be applied directly to the EDA data because estimating a threshold value of EDA is not easy and could result into many false positives or many peaks would be missed.

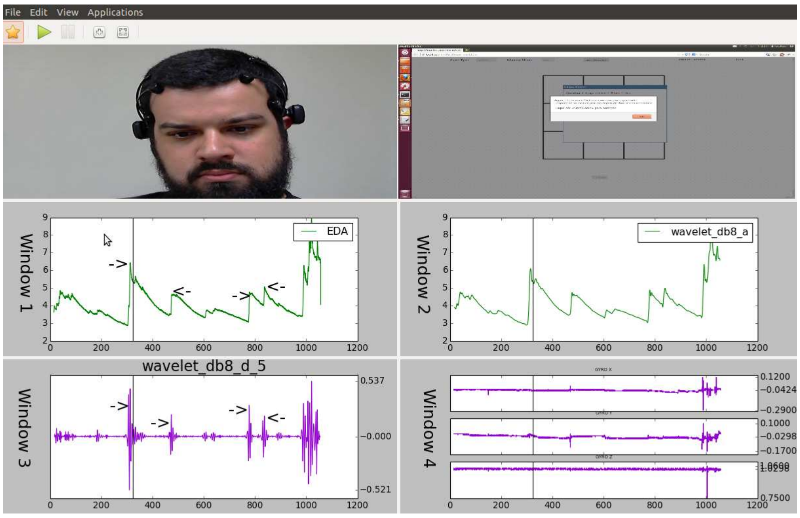


Figure 4.18: EDA analysis

By displaying the video of the screen shot data synchronized with the EDA data, it was observed that the majority of phasic components occurred when there was a change in the task. In the experiment of type A, experimenter does baseline test, play nback level 1, play nback level 2, play nback level 3, repeat the baseline test and play tower of hanoi.

Figure 4.19 indicates the starting point of each task and the associated phasic component. It is natural to suppose that a phasic component occurred due to the hand movement. However, by observing the plot of hand acceleration in window 4 we could visualize that the phasic components were not a result of hand movement.

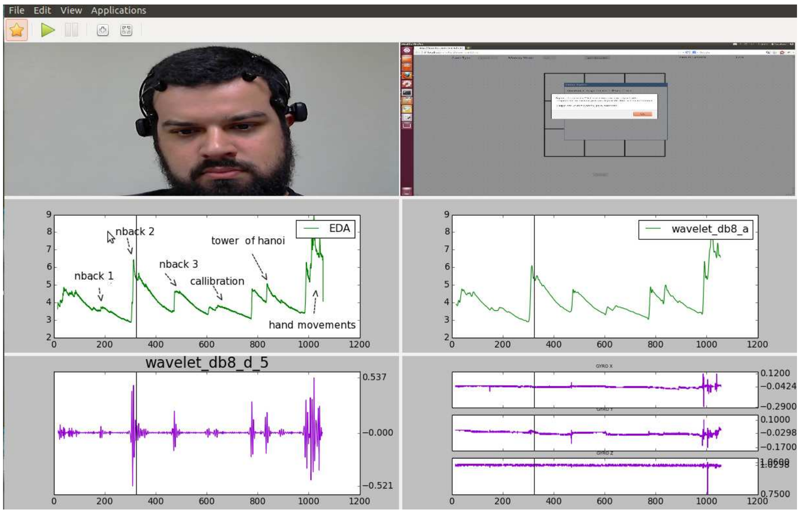
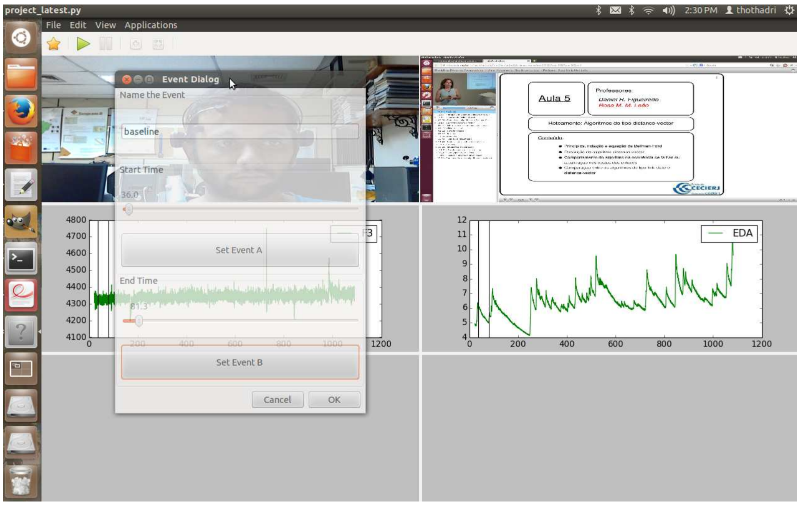


Figure 4.19: The phasic components due to the result in the change of task

4.3.3 Event Based Analysis

Our data visualization tool enables the analyst to create ”events”(selected points in time) to observe the data collected from the experiment during selected time intervals. For example, in the experiment of type B where the subject performs a baseline test and watch a video lecture, event based analysis can be performed to understand changes in the data patterns of the EEG and EDA sensor between the baseline test and the video lecture. Figure 4.20 shows the graphical user interface for creating an event.


Name the Event: baseline

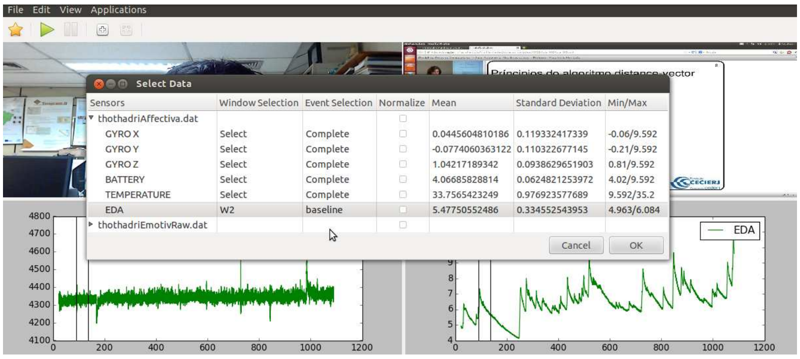
Start Time: 36.0

End Time: 81.3

Figure 4.20: GUI for creating an event

Events are created by moving the slide bar below the start time and the end time as shown in the Figure 4.20. By moving these slide bars, with respect to its position, the displayed videos are seeked and a black vertical line is plotted on all the graphs . In experiment B, in order to create an event which spans only the baseline test, the video which displays the screen shot of the task performed is required. By seeking the video using the slide bar, the starting point and the ending point of the baseline event is identified.

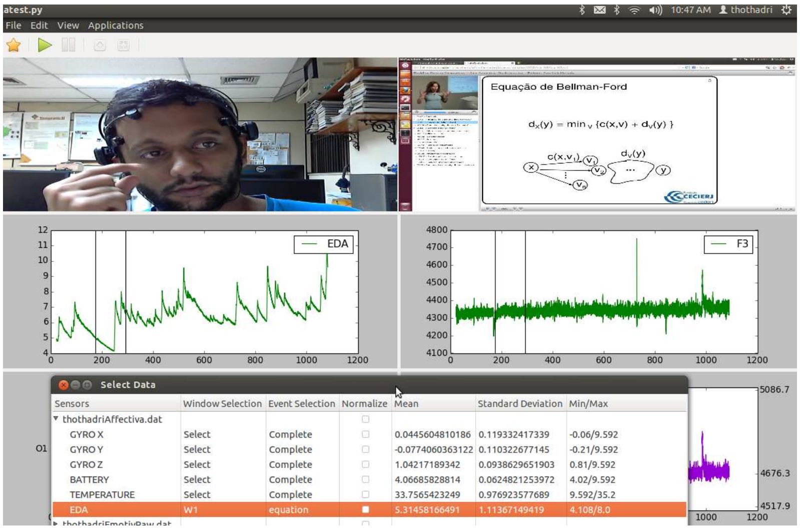
The next step is to compute the basic statistics for some selected data from the sensors. Figure 4.21 shows the interface to select the data from the sensors. The *Event Selection* field in the figure is a drop down button and when clicked displays all the events created. By default, an event named ”complete” is inbuilt which spans the whole time interval. By selecting one of the events for a particular data, the basic statistics of the data are computed and presented to the user. In this figure, for the baseline event of EDA data, values of its mean, standard deviation , maximum and minimum are presented to the analyst.



Sensors	Window Selection	Event Selection	Normalize	Mean	Standard Deviation	Min/Max
▾ thothadriAffectiva.dat			□			
GYRO X	Select	Complete	□	0.0445604810186	0.119332417339	-0.06/9.592
GYRO Y	Select	Complete	□	-0.0774060363122	0.110322677145	-0.21/9.592
GYRO Z	Select	Complete	□	1.04217189342	0.0938629651903	0.81/9.592
BATTERY	Select	Complete	□	4.06685828814	0.0624821253972	4.02/9.592
TEMPERATURE	Select	Complete	□	33.7565423249	0.976923577689	9.592/35.2
EDA	W2	baseline	□	5.47750552486	0.334552543953	4.963/6.084
▸ thothadriEmotivRaw.dat			□			

Figure 4.21: Baseline event of the experiment type B

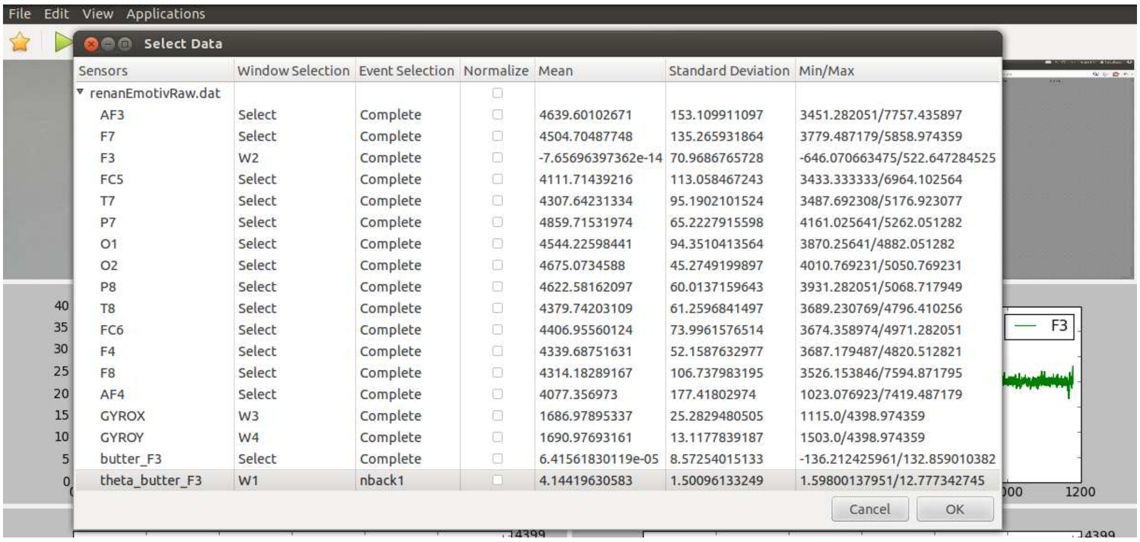
Similarly, an event can be defined to analyze the data for a given slide containing, for example, an equation. Figure 4.22 shows the basic statistical details presented for the event equation in slide 3, where the lecturer presents the Bellman-Ford equation.



Sensors	Window Selection	Event Selection	Normalize	Mean	Standard Deviation	Min/Max
▾ thothadriAffectiva.dat			□			
GYRO X	Select	Complete	□	0.0445604810186	0.119332417339	-0.06/9.592
GYRO Y	Select	Complete	□	-0.0774060363122	0.110322677145	-0.21/9.592
GYRO Z	Select	Complete	□	1.04217189342	0.0938629651903	0.81/9.592
BATTERY	Select	Complete	□	4.06685828814	0.0624821253972	4.02/9.592
TEMPERATURE	Select	Complete	□	33.7565423249	0.976923577689	9.592/35.2
EDA	W1	equation	■	5.31458166491	1.11367149419	4.108/8.0
▸ thothadriEmotivRaw.dat			□			

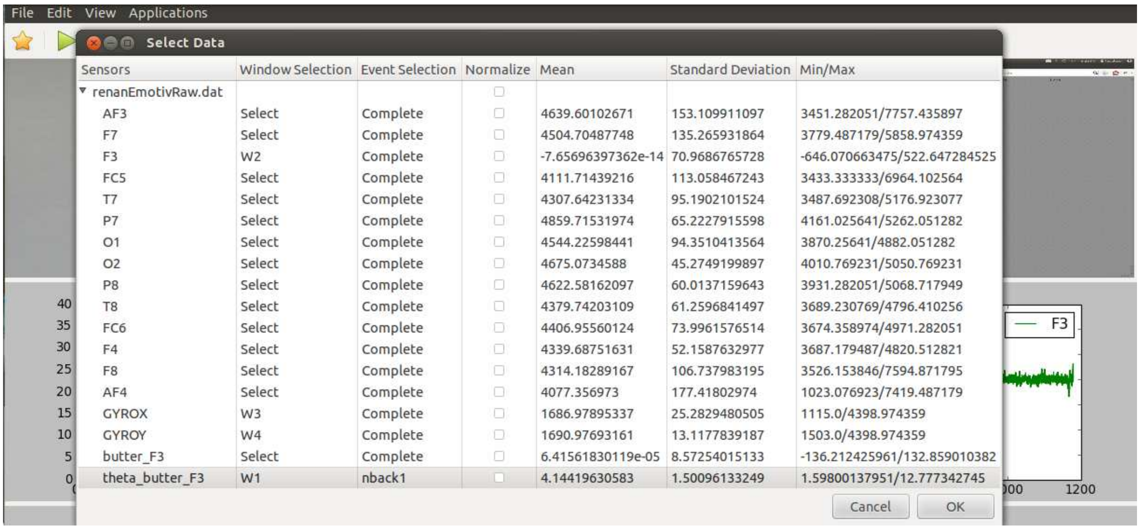
Figure 4.22: EDA

To analyze the experiment of type A, 3 events were created where each event includes the subject playing nback level 1 , level 2 and level 3. The raw EEG data of channel F_3 was decomposed into 4-8 Hz theta signal using the Fourier Transform Module. The average, standard deviation, minimum and maximum values of theta were observed for each of the events and for most of the experiments the average value of theta increased with the level of nback. Figures 4.23, 4.24, and 4.25 show the data selection module highlighting the basic statistics computed for theta for nback 1, 2 and 3. It is noted that for most of the conducted experiments as the level of nback increases the value of theta increases in frontal lobe F3 and this is in accordance with the papers published by Antonenko and David Grimes [4, 29]. However, in this work we are just trying to show that such kind of analysis could be done.



Sensors	Window Selection	Event Selection	Normalize	Mean	Standard Deviation	Min/Max
▾ renanEmotivRaw.dat			□			
AF3	Select	Complete	□	4639.60102671	153.109911097	3451.282051/7757.435897
F7	Select	Complete	□	4504.70487748	135.265931864	3779.487179/5858.974359
F3	W2	Complete	□	-7.65696397362e-14	70.9686765728	-646.070663475/522.647284525
FC5	Select	Complete	□	4111.71439216	113.058467243	3433.333333/6964.102564
T7	Select	Complete	□	4307.64231334	95.1902101524	3487.692308/5176.923077
P7	Select	Complete	□	4859.71531974	65.2227915598	4161.025641/5262.051282
O1	Select	Complete	□	4544.22598441	94.3510413564	3870.25641/4882.051282
O2	Select	Complete	□	4675.0734588	45.2749199897	4010.769231/5050.769231
P8	Select	Complete	□	4622.58162097	60.0137159643	3931.282051/5068.717949
T8	Select	Complete	□	4379.74203109	61.2596841497	3689.230769/4796.410256
FC6	Select	Complete	□	4406.95560124	73.9961576514	3674.358974/4971.282051
F4	Select	Complete	□	4339.68751631	52.1587632977	3687.179487/4820.512821
F8	Select	Complete	□	4314.18289167	106.737983195	3526.153846/7594.871795
AF4	Select	Complete	□	4077.356973	177.41802974	1023.076923/7419.487179
GYROX	W3	Complete	□	1686.97895337	25.2829480505	1115.0/4398.974359
GYROY	W4	Complete	□	1690.97693161	13.1177839187	1503.0/4398.974359
butter_F3	Select	Complete	□	6.41561830119e-05	8.57254015133	-136.212425961/132.859010382
theta_butter_F3	W1	nback1	□	4.14419630583	1.50096133249	1.59800137951/12.777342745

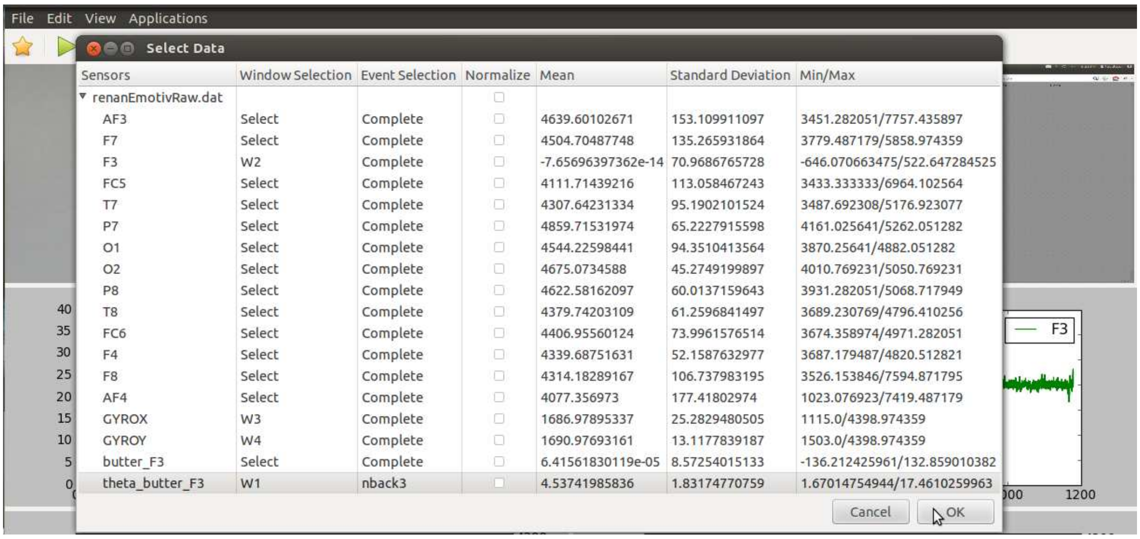
Figure 4.23: Theta statistics for nback 1



Sensors	Window Selection	Event Selection	Normalize	Mean	Standard Deviation	Min/Max
▾ renanEmotivRaw.dat			□			
AF3	Select	Complete	□	4639.60102671	153.109911097	3451.282051/7757.435897
F7	Select	Complete	□	4504.70487748	135.265931864	3779.487179/5858.974359
F3	W2	Complete	□	-7.65696397362e-14	70.9686765728	-646.070663475/522.647284525
FC5	Select	Complete	□	4111.71439216	113.058467243	3433.333333/6964.102564
T7	Select	Complete	□	4307.64231334	95.1902101524	3487.692308/5176.923077
P7	Select	Complete	□	4859.71531974	65.2227915598	4161.025641/5262.051282
O1	Select	Complete	□	4544.22598441	94.3510413564	3870.25641/4882.051282
O2	Select	Complete	□	4675.0734588	45.2749199897	4010.769231/5050.769231
P8	Select	Complete	□	4622.58162097	60.0137159643	3931.282051/5068.717949
T8	Select	Complete	□	4379.74203109	61.2596841497	3689.230769/4796.410256
FC6	Select	Complete	□	4406.95560124	73.9961576514	3674.358974/4971.282051
F4	Select	Complete	□	4339.68751631	52.1587632977	3687.179487/4820.512821
F8	Select	Complete	□	4314.18289167	106.737983195	3526.153846/7594.871795
AF4	Select	Complete	□	4077.356973	177.41802974	1023.076923/7419.487179
GYROX	W3	Complete	□	1686.97895337	25.2829480505	1115.0/4398.974359
GYROY	W4	Complete	□	1690.97693161	13.1177839187	1503.0/4398.974359
butter_F3	Select	Complete	□	6.41561830119e-05	8.57254015133	-136.212425961/132.859010382
theta_butter_F3	W1	nback1	□	4.14419630583	1.50096133249	1.59800137951/12.777342745

Figure 4.24: Theta statistics for nback 2

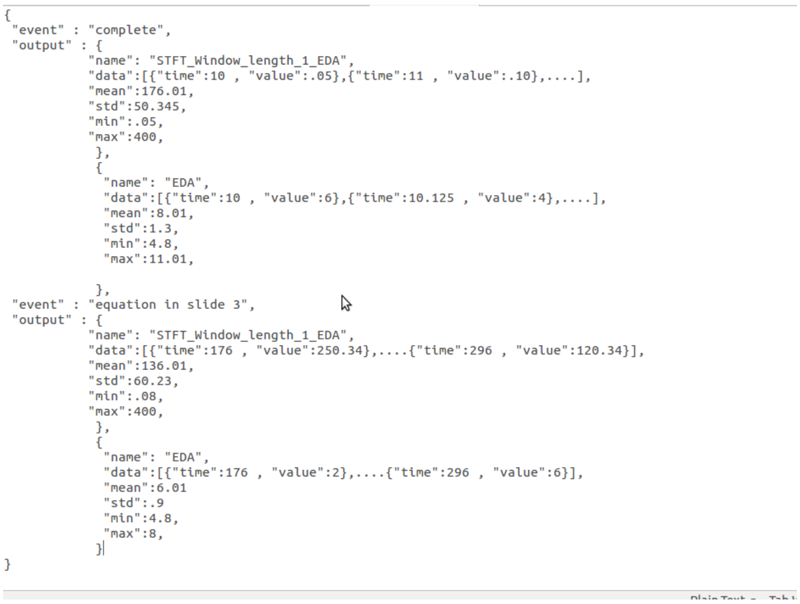
Once the data visualization tool is signaled to close, all the data in the data selection module is saved in a file output.dat in json format [14] where data is stored



Sensors	Window Selection	Event Selection	Normalize	Mean	Standard Deviation	Min/Max
renanEmotivRaw.dat			□			
AF3	Select	Complete	□	4639.60102671	153.109911097	3451.282051/7757.435897
F7	Select	Complete	□	4504.70487748	135.265931864	3779.487179/5858.974359
F3	W2	Complete	□	-7.65696397362e-14	70.9686765728	-646.070663475/522.647284525
FC5	Select	Complete	□	4111.71439216	113.058467243	3433.333333/6964.102564
T7	Select	Complete	□	4307.64231334	95.1902101524	3487.692308/5176.923077
P7	Select	Complete	□	4859.71531974	65.2227915598	4161.025641/5262.051282
O1	Select	Complete	□	4544.22598441	94.3510413564	3870.25641/4882.051282
O2	Select	Complete	□	4675.0734588	45.2749199897	4010.769231/5050.769231
P8	Select	Complete	□	4622.58162097	60.0137159643	3931.282051/5068.717949
T8	Select	Complete	□	4379.74203109	61.2596841497	3689.230769/4796.410256
FC6	Select	Complete	□	4406.95560124	73.9961576514	3674.358974/4971.282051
F4	Select	Complete	□	4339.68751631	52.1587632977	3687.179487/4820.512821
F8	Select	Complete	□	4314.18289167	106.737983195	3526.153846/7594.871795
AF4	Select	Complete	□	4077.356973	177.41802974	1023.076923/7419.487179
GYROX	W3	Complete	□	1686.97895337	25.2829480505	1115.0/4398.974359
GYROY	W4	Complete	□	1690.97693161	13.1177839187	1503.0/4398.974359
butter_F3	Select	Complete	□	6.41561830119e-05	8.57254015133	-136.212425961/132.859010382
theta_butter_F3	W1	nback3	□	4.53741985836	1.83174770759	1.67014754944/17.4610259963

Figure 4.25: Theta statistics for nback 3

in an unordered, comma-separated collection of key:value pairs enclosed in curly braces, with the ':' character separating the key and the value. Figure 4.26 shows an example of the output file for the EDA data during the events complete and equation in slide 3.



```
{
 "event" : "complete",
 "output" : {
           "name": "STFT_Window_length_1_EDA",
           "data":[{"time":10 , "value":.05},{"time":11 , "value":.10},....],
           "mean":176.01,
           "std":50.345,
           "min":.05,
           "max":400,
            },
            {
             "name": "EDA",
             "data":[{"time":10 , "value":6},{"time":10.125 , "value":4},....],
             "mean":8.01,
             "std":1.3,
             "min":4.8,
             "max":11.01,

            },
 "event" : "equation in slide 3",
 "output" : {
           "name": "STFT_Window_length_1_EDA",
           "data":[{"time":176 , "value":250.34},....{"time":296 , "value":120.34}],
           "mean":136.01,
           "std":60.23,
           "min":.08,
           "max":400,
            },
            {
             "name": "EDA",
             "data":[{"time":176 , "value":2},....{"time":296 , "value":6}],
             "mean":6.01
             "std":.9
             "min":4.8,
             "max":8,
            }
}
```

Figure 4.26: Output data

Chapter 5

Conclusion and Future Work

In this work, our objective was to build a tool to assist the on going work in our Laboratory on Intelligent Tutoring Systems to build a predictive model for user emotions as they watch the online tutoring systems. Non-intrusive sensors such as Electroencephalogram (EEG), Electro-dermal Activity Meter (EDA), normal webcam and infra-red webcam were used as inputs to detect user emotions in the future. The tool built visualizes and analyzes data from multiple sources of type video, time series data and event-related data. The tool provides the analyst the flexibility to choose from any desired data collected from multiple sources, synchronized and displayed simultaneously in the same interface.

A web interface was designed in which the participants either watch the CEDERJ video lecture or play games such as nback and tower of hanoi with the sensors (EEG, EDA, normal webcam and infra-red webcam) attached to them. The games nback and tower of hanoi were implemented in order to provide an initial assessment of the sensor data. Mindland reader was used to record the data from all the sensors when the subject is performing an experiment. Mindland reader also captures the video screenshot when the subject performs the experiment.

After the experiment, the data recorded from the sensors is visualized and analyzed in our tool. The output from either the EEG or the EDA sensor is displayed as a time series and the output from a regular webcam, the infra-red webcam and screenshots are displayed as videos. The tool synchronizes all the videos frame by frame and a black vertical line slides through the plotted graph indicating the position of the video with respect to the plotted data.

In addition to visualization, the tool can analyze the data from the sensors using signal processing techniques such as Fourier transform, Short Time Fourier Transform, Butter worth filter and Wavelet Transform . The resulting data can be plotted again and be visualized synchronized with any of other sensor data. The tool also enables the analyst to create events to observe the data collected from the experiment during selected time intervals. Basic statistics such as mean, variance, minimum and maximum values of the sensor data are presented to the user for any desired intervals. These features provide an environment for the analyst to study all the data together or individually in one interface and to help identifying important trends in the data, removing the noise from the data and identifying possible correlation among data from different sources.

In the future, the tool will be used as a first aid for data analysis from sensors to help the analyst in the task to build reliable models to predict user's emotional and cognitive states accurately. Our tool should support further development to achieve the ultimate goal of our Lab's project, supported by GOOGLE, in which an intelligent distance learning system will be built for the computer system course offered by CEDERJ. As such the video lectures should be able to adapt according to the student cognitive state and enhance the student's learning experience. Our tool should support continuing enhancements, for instance to incorporate other signal processing techniques, statistical techniques and machine learning algorithms in the data visualization tool.

Bibliography

[1] Affectiva. Affectiva q solutions white paper. Technical report, Affectiva.

[2] Pierre-Emmanuel Aguera, Karim Jerbi, Anne Caclin, and Olivier Bertrand. Elan: a software package for analysis and visualization of meg, eeg, and lfp signals. *Computational intelligence and neuroscience*, 2011:5, 2011.

[3] Sidney K. D'Mello ands Scotty D. Craig, Jeremiah Sullins, and Arthur C. Graesser. Predicting affective states expressed through an emote-aloud procedure from autotutor's mixed-initiative dialogue. *I. J. Artificial Intelligence in Education*, 16(1):3–28, 2006.

[4] Pavlo Antonenko and Grabner. Using electroencephalography to measure cognitive load. In *Education Psychology Review*, volume 22, pages 425–438, 2010.

[5] Alan Baddeley. Working memory. *Science*, 255(5044):556–559, 1992.

[6] Yan BODAIN and Jean-Marc ROBERT. Investigating distance learning on the internet. In *Inet*, 2000.

[7] Wolfram Boucsein. *Electrodermal activity*. Springer, 2012.

[8] Jason Braithwaite, Derrick Watson, Robert Jones, and Mickey Rowe. A guide for analysing electrodermal activity (eda) and skin conductance responses (scrs) for psychological experiments. Technical report, SAAL, University of Birmingham, 2013.

[9] Tom Caswell, Shelley Henson, Marion Jensen, and David Wiley. Open educational resources: Enabling universal education. 9:207–282, 2008.

[10] CEDERJ. Cederj videolectures for computer science course. `http://www.cederj.edu.br/videoaulas`.

[11] CEDERJ. www.cederj.edu.br. website.

[12] David G. Cooper, Ivon Arroyo, Beverly Park Woolf, Kasia Muldner, Winslow Burleson, and Robert Christopherson. Sensors model student self concept in the classroom. In *Proceedings of the 17th International Conference on UserModeling, Adaptation, and Personalization: formerly UM and AH*, pages 30–41, 2009.

[13] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J.G. Taylor. Emotion recognition in human-computer interaction. *Signal Processing Magazine, IEEE*, 18(1):32–80, 2001.

[14] Douglas Crockford. The application/json media type for javascript object notation (json). 2006.

[15] Katie Crowley, Aidan Sliney, Ian Pitt, and Dave Murphy. Evaluating a brain-computer interface to categorise human emotional response. In *Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on*, pages 276–278. Ieee, 2010.

[16] Goleman D. *Emotional Intelligence*. Bantam Books, 1995.

[17] Edmundo de Souza, Rosa MM Leao, Anna D Santos, Jorge Allyson Azevedo, and Bernardo C Machado Netto. Multimedia supporting tools for the cederj distance learning initiative applied to the computer systems course. In *Proc. 22nd ICDE World Conference on Distance Education*, pages 01–11, 2006.

[18] Arnaud Delorme and Scott Makeig. Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis. *Journal of neuroscience methods*, 134(1):9–21, 2004.

[19] Sidney D'Mello and Art Graesser. Modeling cognitive-affective dynamics with hidden markov models. 2010.

[20] Sidney K. D'Mello, Rosalind W. Picard, and Arthur C. Graesser. Toward an affect-sensitive autotutor. *IEEE Intelligent Systems*, 22(4):53–61, 2007.

[21] Toby Dragon, Ivon Arroyo, Beverly Park Woolf, Winslow Burleson, Rana El Kaliouby, and Hoda Eydgahi. Viewing student affect and learning through classroom observation and physical sensors. In *Intelligent Tutoring Systems*, pages 29–39, 2008.

[22] dynamicdrive. http://www.dynamicdrive.com/dynamicindex12/hanoi.htm. website.

[23] Thothadri Rajesh e Gaspare Bruno. www.land.ufrj.br/ thothadri/mindland.html. website.

[24] Cognitive fun. http://cognitivefun.net/test/4. website.

[25] Robert M Gagne. The conditions of learning. 2012.

[26] AW Galli, GT Heydt, and PF Ribeiro. Exploring the power of wavelet analysis. *Computer Applications in Power, IEEE*, 9(4):37–41, 1996.

[27] Alan Gevins and Michael E Smith. Neurophysiological measures of working memory and individual differences in cognitive ability and cognitive style. *Cerebral Cortex*, 10(9):829–839, 2000.

[28] Alan Gevins, Michael E Smith, Linda McEvoy, and Daphne Yu. High-resolution eeg mapping of cortical activation related to working memory: effects of task difficulty, type of processing, and practice. *Cerebral Cortex*, 7(4):374–385, 1997.

[29] David Grimes, Desney S Tan, Scott E Hudson, Pradeep Shenoy, and Rajesh PN Rao. Feasibility and pragmatics of classifying working memory load with an electroencephalograph. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 835–844. ACM, 2008.

[30] BERT HOEKS and WILLEM J. M. LEVELT. Pupillary dilation as a measure of attention:a quantitative system analysis. *Behavior Research Methods, Instruments, and Computers*, 25(1):16–26, 1993.

[31] Claudio Imperatori, Benedetto Farina, Riccardo Brunetti, Valentina Gnoni, Elisa Testani, Maria I Quintiliani, Claudia Del Gatto, Allegra Indraccolo, Anna Contardi, Anna M Speranza, et al. Modifications of eeg power spectra in mesial temporal lobe during n-back tasks of increasing difficulty. a sloreta study. *Frontiers in human neuroscience*, 7, 2013.

[32] Tzyy-Ping Jung, Scott Makeig, Colin Humphries, Te-Won Lee, Martin J. McKeown, Vicente Iragui, and Terrence J. Sejnowski. Removing electroencephalographic artifacts by blind source separation. *Psychophysiology*, 37(2):163–178, 2000.

[33] Rana El Kaliouby and Peter Robinson. Mind reading machines: automated inference of cognitive mental states from video. In *SMC (1)*, pages 682–688, 2004.

[34] W. Klimesch, H. Schimke, and G. Pfurtscheller. Alpha frequency, cognitive load and memory performance. *Brain Topography*, 5(3):241–251, 1993.

[35] Wolfgang Klimesch. Eeg alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. *Brain research reviews*, 29(2):169–195, 1999.

[36] Barry Kort, Rob Reilly, and Rosalind W. Picard. An affective model of interplay between emotions and learning: Reengineering educational pedagogy - building a learning companion. In *ICALT*, pages 43–48, 2001.

[37] Yandong Li, Zhongwei Ma, Wenkai Lu, and Yanda Li. Automatic removal of the eye blink artifact from eeg using an ica-based template matching approach. *Physiological measurement*, 27(4):425, 2006.

[38] Marshall and Sandra P. Identifying cognitive state from eye metrics. In *Aerospace Medical Association*, pages 165–175, 2007.

[39] Daniel McDuff, Rana El Kaliouby, Karim Kassam, and Rosalind Picard. Acume: A new visualization tool for understanding facial expression and gesture data. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 591–596. IEEE, 2011.

[40] Scott F. Midkiff and Luiz A. DaSilva. Leveraging the web for synchronous versus asynchronous distance learning. In *ICEE*, 2000.

[41] Neurosky. Brain wave signal of neurosky. Technical report, Neurosky, December 2009.

[42] Travis E Oliphant. *A Guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.

[43] Ekman P and W Friesen. The facial action coding system: A technique for the measurment of facial movement. press, 1978.

[44] Ming-Zher Poh, Nicholas C. Swenson, and Rosalind W. Picard. A wearable sensor for unobtrusive, long-term assessment of electrodermal activity. *IEEE Trans. Biomed. Engineering*, 57(5):1243–1252, 2010.

[45] RNP. Rede nacional de ensino e pesquisa. `https://www.rnp.br/en/`.

[46] Jennifer Robinson, Scott McQuiggan, and Jean-Marc ROBERT. Evaluating the consequences of affective feedback in intelligent tutoring systems. 2009.

[47] Jose Renato Santos and Richard Muntz. Performance analysis of the rio multimedia storage system with heterogeneous disk configurations. In *Proceedings of the sixth ACM international conference on Multimedia*, pages 303–308. ACM, 1998.

[48] Richard Snow, Corno L, and Jackson D. *Individual differences in affective and conative functions.* D. C. Berliner and R. C. Calfee, Handbook of educational psychology, 1996.

[49] Marshal D Sundberg. Assesing student learning. In *American Society for Cell Biology*, 2002.

[50] techcrunch. your online ivy coursera now hosts 200 courses from 33 schools and reaches 1-3m students. website. Coursera,2012.

[51] Filip Wasilewski. http://pywavelets.readthedocs.org. website.