A PROJECT REPORT ON

Task Performance Measures for Cognitive Workload Evaluation Using EEG Signals

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY , PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

BACHELOR OF ENGINEERING
(Computer Engineering)

BY

Swarali Belsare	B150314216
Maitreyi Kale	B150314276
Priya Ghayal	B150314258
Aishwarya Gogate	B150314261

Under the guidance of

Prof. Dr. Mrs. S. A. Itkar

DEPARTMENT OF COMPUTER ENGINEERING

PES MODERN COLLEGE OF ENGINEERING
SHIVAJINAGAR,PUNE 411005

SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
2020- 21

CERTIFICATE

This is to certify that the Project Entitled

Task Performance Measures for Cognitive Workload Evaluation Using EEG Signals

Submitted by

Swarali Belsare	B150314216
Maitreyi Kale	B150314276
Priya Ghayal	B150314258
Aishwarya Gogate	B150314 261

is a bonafide work carried out by them under the supervision of Prof. Dr. Mrs. S. A. Itkar and it is approved for the partial fulfillment of the requirement of Savtribai Phule Pune university, Pune for the award of the degree of Bachelor of Engineering (Computer Engineering).

Prof. Dr. Mrs. S. A. ITKAR
Guide
Department of Computer Engineering

Prof. Dr. Mrs. S. A. ITKAR
Head
Department of Computer Engineering

Signature of Internal Examiner

Signature of External Examiner

PROJECT APPROVAL SHEET

A Project Report Titled as

Task Performance Measures for Cognitive Workload Evaluation Using EEG Signals

is successfully completed by

Swarali Belsare	B150314216
Maitreyi Kale	B150314276
Priya Ghayal	B150314258
Aishwarya Gogate	B150314261

at

DEPARTMENT OF COMPUTER ENGINEERING

PES MODERN COLLEGE OF ENGINEERING

SAVITRIBAI PHULE PUNE UNIVERSITY,PUNE

ACADEMIC YEAR 2020-2021

Prof.Dr. Mrs. S. A. ITKAR
Guide
Department of Computer Engineering

Prof.Dr. Mrs. S. A. ITKAR
Head
Department of Computer Engineering

Acknowledgement

It gives us pleasure in presenting the project report on **'Task Performance Measures for Cognitive Workload Evaluation using EEG signals'**.

Firstly, we would like to express our indebtedness appreciation to our guide **Prof. Dr. Mrs. S. A. Itkar**. Her constant guidance and advice played very important role in successful completion of the project. She always gave us her suggestions, that were crucial in making this report as flawless as possible.

We would like to express our gratitude towards **Prof. Dr. Mrs. S. A. Itkar** Head of Computer Engineering Department, PES Modern College of Engineering for her kind co-operation and encouragement which helped us during the completion of this report.

Also we wish to thank our Principal, **Prof. Dr. Mrs. K. R. Joshi** and all faculty members for their whole hearted co-operation for completion of this report. We also thank our laboratory assistants for their valuable help in laboratory.

Last but not the least, the backbone of our success and confidence lies solely on blessings of dear parents and lovely friends.

Swarali Belsare
Maitreyi Kale
Priya Ghayal
Aishwarya Gogate

Abstract

Recent advances in the area of Brain Computer Interface (BCI) have opened doors to many interdisciplinary applications like safety, performance evaluation etc. Measuring the level of cognitive workload (CWL) is a crucial aspect in these applications. Measurement of cognitive workload level of pilots,drivers can ensure safety of the passengers travelling. Also, evaluation of students or even employees of an organization based on the levels of workload can be done for performance analysis. The level of mental effort put forth by a human is measurable in terms of response to one or many cognitive tasks. This level is known as mental or cognitive workload. The electroencephalogram (EEG) signal is a recording of the electrical activity of the brain from the scalp and is used to interpret cognitive functionalities. Analysis of EEG signals helps to detect the level of cognitive workload of a person pertaining to a certain task under operation. So we propose a deep learning approach to classify CWL levels based on EEG signals. The input to the system is raw EEG data extracted from the STEW dataset. A three step approach is proposed for obtaining the output. First step includes data preprocessing for removing noise from the data using Artifact Subspace Reconstruction(ASR). This project compares the results of Convolutional Neural Networks(CNN),Gated Recurrent Unit(GRU) and composite model of CNN extracts features from the preprocessed EEG data which are then used by GRU algorithm for classification. Also machine learning models like Random Forest and Decision tree are used to compare the performance. Performance parameters of the proposed system include classification accuracy.

Keywords

EEG - Electroencephalogram
CWL-Cognitive workload
CNN- Convolutional Neural Network
GRU-Gated Recurrent Unit
RFC - Random Forest Classifier

Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Overview

Recent advances in the area of Brain Computer Interface (BCI) have opened doors to many interdisciplinary applications like safety, performance evaluation etc. Measuring the level of cognitive workload (CWL) is a crucial aspect in these applications. Measurement of cognitive workload level of pilots,drivers can ensure safety of the passengers travelling. Also, evaluation of students or even employees of an organization based on the levels of workload can be done for performance analysis. The level of mental effort put forth by a human is measurable in terms of response to one or many cognitive tasks.

1.2 Motivation

EEG signal analysis is one of the most accurate measure for determining the cognitive state of a person while performing any task. Since brainwaves are raw signals , proper preprocessing operations need to be performed in order to extract meaningful information. This information is then used for various classification purposes. Efficient analysis of these signals plays an important role in various domains crucial for the well-being of the society.These include safety of passengers , evaluation of student performance for proper designing of curriculum and many more. Pertaining to such applications , proposed system helps to efficiently analyze EEG signals by considering the task performance of a person to determine cognitive workload levels

1.3 Problem Definition and Objectives

To improve classification accuracy and propose a modified approach of EEG signal analysis for cognitive workload evaluation.
Objectives

1. To analyze and classify users' mental workload while performing cognitive tasks.

2. To apply deep learning techniques on EEG signals for classifying Cognitive Workload.

1.4 Project Scope & Limitations

This project has scope in the safety ,defense and performance evaluation areas. In the listed areas, project output can be used to detect driver fatigue, pilot stress and ensure passenger safety. Also students and employee performance evaluation is possible.
Limitations

1. EEG signals are accurate measures of brain activity but cognitive workload levels differ subjectively.
2. Applying complex deep learning models requires more data samples, so more data might yield better results.

1.5 Methodologies for Problem Solving

1. Define the problem.
2. Create a mathematical model.
3. Develop a computational method for solving the problem.
4. Implement the computational method.
5. Test and assess the solution.

CHAPTER 2

LITERATURE SURVEY

Cognitive workload (CWL) contributes considerably to the outcome or the performance of any task. The concern of human workload increases during a human-machine collaboration task or in a multitasking environment.The workload classification can be further used in human-machine tasks to decide task allocation between the system to achieve optimal performance in a complex critical system. The CWL analysis helps in crucial fields like safety of passengers and crew members in the aviation as well as automation industry and is also effective in performance evaluation of students or employees. Traditional approaches of EEG signal analysis rely on signal processing techniques which are time-consuming. Recent advances in the field of Brain-Computer Interface (BCI) have opened doors to many analysis techniques including deep learning. So, we propose a modified approach to efficiently analyze EEG signals and classify cognitive workload levels using deep learning models.

EEG-based mental workload estimation using deep BLSTM-LSTM network and evolutionary algorithm

In this paper, the authors have proposed a deep BLSTM-LSTM network along with the gray wolf optimization technique to estimate different levels of workload.
Previous drawback : Improper estimation due to feature selection

Proposed Model Model - BLSTM-LSTM + GWO
Dataset - STEW dataset
Algorithm - LSTM, GWO

Overcomings : GWO algorithm for feature selection

EEG based spatio-temporal Convolutional Neural Network for Driver Fatigue Evaluation

In this paper, authors have developed a novel framework of spatio-temporal convolutional Neural Network to detect driver fatigue using EEG signals. The overall performance of the model has been compared with another 8 methods.

Previous drawback Feature extraction process of time-frequency analysis may neglect valuable information of electrode correlation

Proposed Model :
Model - Core block + Dense block

Dataset - Self prepared dataset
Algorithm - ESTCNN

Overcomings :

Core block having advantages of temporal dependencies extraction and a dense block combined with core block for better spatio-temporal information Reduced data dimension in inference procedure giving rise to computational efficiency and reference response

Detecting Fatigue Status of Pilots based on Deep Learning Network using EEG Signals

This work has proposed a new autoencoder network to learn the features of pilots' EEG signals. Sparse learning mechanism of DCSAEN not only eliminated the redundancy of raw features, but also improved computational efficiency of the model. The contractive loss function of DCSAEN significantly improved the identification of micro fatigue. Compared with the-state-of-the-art methods, DCSAEN greatly improved the accuracy of fatigue recognition.

Model - DCSAEN
Dataset - BCI200
Algorithm - stacked contractive sparse autoencoder (CSAE)

Overcomings :

The FIR filter is used to extract four rhythms like theta, delta, alpha, and beta waves. The study presents their relationship with the change of fatigue status.Authors present fatigue indicators related to the power change of rhythms as the basis for fatigue judgment. The results of the study show that the fatigue evaluation index has a high recognition rate for the fatigue state of the pilot.Compared with a single contractive autoencoder, the model adds a sparse penalty term =¿ significant features and greatly improves learning efficiency. Jacobian constraint term =¿ can better learn the perturbation characteristics in all directions around the input points, and greatly improve the efficiency of local feature learning

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Assumptions and Dependencies

Assumptions

1. EEG data is collected in proper environment and under proper guidance of expert.
2. Subjects have performed the experiment following proper guidelines.

Dependencies

1. The system is highly dependent on EEG data availability.
2. The system is also dependent on the functioning of EEG headset.

3.2 Functional Requirements

3.2.1 System Feature 1(Functional Requirement)

1. **EEG data to signal** - EEG data is converted to signal form .
2. **Pre-processing using ASR** - Artifacts such as eye movement , blinking , muscle movement are removed using Artifact Subspace Reconstruction algorithm.
3. **Feature Extraction** - Features are learnt by the convolutional neural network.
4. **Classification** - The Gated Recurrent unit classifies the levels of cognitive workload.

3.2.2 System Feature 2(Functional Requirement)

1. **Evaluation** - Cognitive workload levels as low , moderate or high.
2. **Classification** - Cognitive workload tasks as SIMKAP or No Task.

3.3 External Interface Requirements (if any)

3.3.1 User Interfaces

Firefox Browser.
Chrome Browser.
Web application.

3.3.2 Hardware Interfaces

Emotiv EPOC EEG headset with 14 electrodes.

3.3.3 Software Interfaces

Tensorflow API, Keras Library, Imblearn , Scikit learn , EEGLAB.

3.3.4 Communication Interfaces

Web Application to upload files and display output

3.4 Nonfunctional Requirements

3.4.1 Performance Requirements

1. Classification accuracy should increase with deep learning model.
2. Model must perform better after fine tuning

3.4.2 Safety Requirements

1. Proper environment should be established for subjects in the experiment.
2. Electrode positions on scalp must be according to 10-20 International system.

3.4.3 Security Requirements

EEG data must be handled correctly in order to prevent external dimension or noise addition while analyzing and classification.

3.4.4 Software Quality Attributes

1. Correctness :
   The system should satisfy all its functional requirements and accurately analyze EEG signals based on these requirements.
2. Reliability :
   The system should be reliable. At any point of time there should be no failure in the system.

3. Robustness :
   The server should be scalable so that it can handle more data at the same time.

4. Maintainability :
   The system should be maintainable even after deployment. The different versions of the system should be easy to maintain. New upgrades should be easily added. It should be easy to debug and update from time to time.

5. Testability :
   The system should be divided into multiple modules, for testing purpose. The modules should be easy to test and find defects.

6. Efficiency :
   The server should utilize its processors capacity, memory and disk space efficiently in order to train and test the deep learning model. If the server is using minimal resources then the system may observe an optimal performance.

7. Security : The system should prevent the addition of external noise while preprocessing as well as analyzing data.

3.5 System Requirements

3.5.1 Database Requirements

1. Large amount of data for deep learning models.

2. Proper guidelines on how the data is collected and it's specifications for feature extraction.

3.5.2 Software Requirements

Python 3.0+ , Ubuntu or Windows OS.

3.5.3 Hardware Requirements

Device: Personal Computer Desktop or laptop.
Processor: i5 core processor.
HDD: 500 GB.
RAM: 8GB.

3.6 Analysis Models:SDLC Model

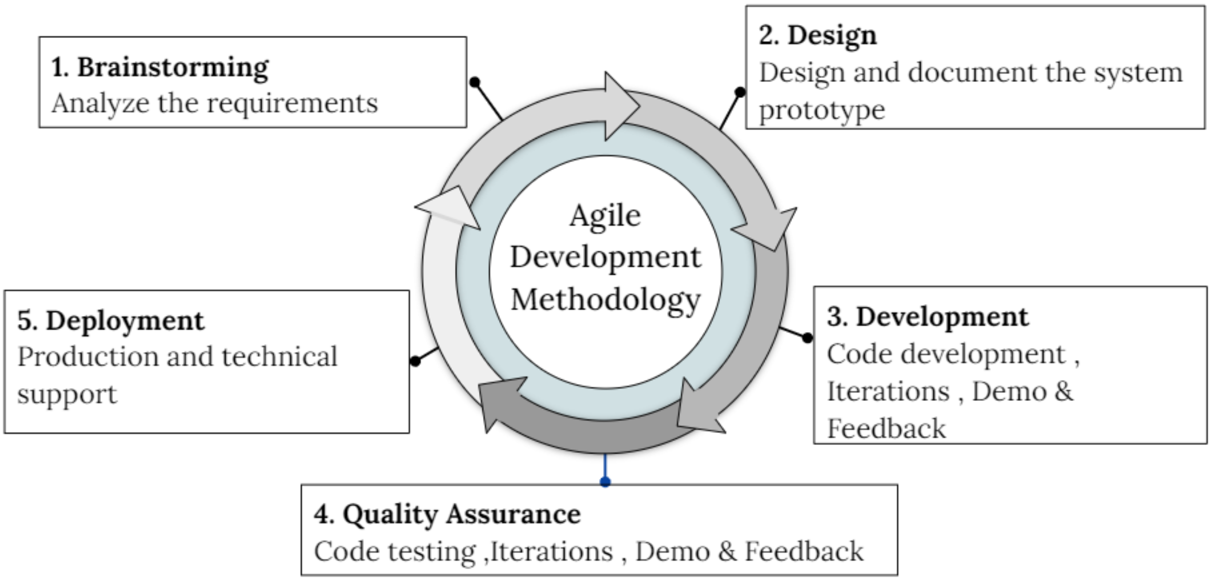

Figure 3.1: SDLC Agile model

Agile model is an iterative development model. Every single iteration in agile involves the whole software development life cycle which is requirement analysis of EEG data, designing suitable architecture for EEG data analysis ,code development for preprocessing , feature extraction , classification, quality assurance or testing by fine tuning the model and deployment. The number of iterations , scope and duration of each iteration is defined in advance. In the agile process model , a short time frame between one to four weeks is considered for each iteration. During the beginning of project , requirement specifications and scope of the project is clearly defined in agile model. For our system , we are using agile model as the whole plan will get divided into small modules which will help us to complete them on time. Dividing entire project to smaller modules will help us to minimize the project risk and will also help us to reduce the delivery time of project.

CHAPTER 4

SYSTEM DESIGN

4.1 System Architecture

The input to the system is the EEG signals extracted from the STEW dataset. The system is divided into following modules

1. Pre-processing raw EEG data
2. Feature Extraction
3. Classification

Artiface subspace reconstruction (ASR) algorithm is used for analysing components acting as artifacts and removing them to obtain clean EEG recording. Convolutional Neural Network (CNN) is used for extracting spatial and temporal features from pre-processed EEG data. Gated Recurrent Unit (GRU) algorithm will classify the EEG data which is a sequential time series data. The final output is in the form of CWL levels which are low , moderate or high.

4.2 Mathematical Model

Convolutional Neural Network (CNN):

- Convolution operation

$$conv(I,K)_{x,y} = \sum_{i=1}^{n_H} \sum_{j=1}^{n_W} \sum_{k=1}^{n_C} K_{i,j,k} I_{x+i-1,y+j-1,k}$$

$$dim(conv(I,K)) = \left(\left\lfloor \frac{n_H+2p-f}{s}+1 \right\rfloor, \left\lfloor \frac{n_W+2p-f}{s}+1 \right\rfloor\right); s>0$$
$$= (n_H+2p-f, n_W+2p-f); s=0$$

$$dim(pooling(image)) = \left(\left\lfloor \frac{n_H+2p-f}{s}+1 \right\rfloor, \left\lfloor \frac{n_W+2p-f}{s}+1 \right\rfloor, \mathbf{n_C}\right); s>0$$
$$= (n_H+2p-f, n_W+2p-f, \mathbf{n_C}); s=0$$

I = Input EEG vector
K= kernel/filter

n_h = size of height
P = padding
f = filter
n_w = size of width
s = stride

- Forward Propagation (computing weights)

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{y}_i^{\theta}, y_i)$$

- Backward propagation using descent

$$\theta =: G(\theta)$$

L = Cost function
y_i = actual output of EEG feature
$\hat{y}_i$ = predicted output of EEG feature
m = number of EEG samples
G = Gradient descent function

Gated Recurrent Unit (GRU):

- Update gate :Gives the amount of past information needed to pass to the future

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1})$$

- Reset gate :Gives the amount of past information that needs to be forgotten

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1})$$

- Current memory content

$$h_t' = \tanh(Wx_t + r_t \odot Uh_{t-1})$$

- Final memory content and current time step

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t'$$

σ = sigmoid function
x_t = input EEG data at the time step t
W = Weights associated with X_t
h_{t-1} = Previous input data of t-1 steps
U = Update information

4.3 Data Flow Diagrams / UML Diagrams

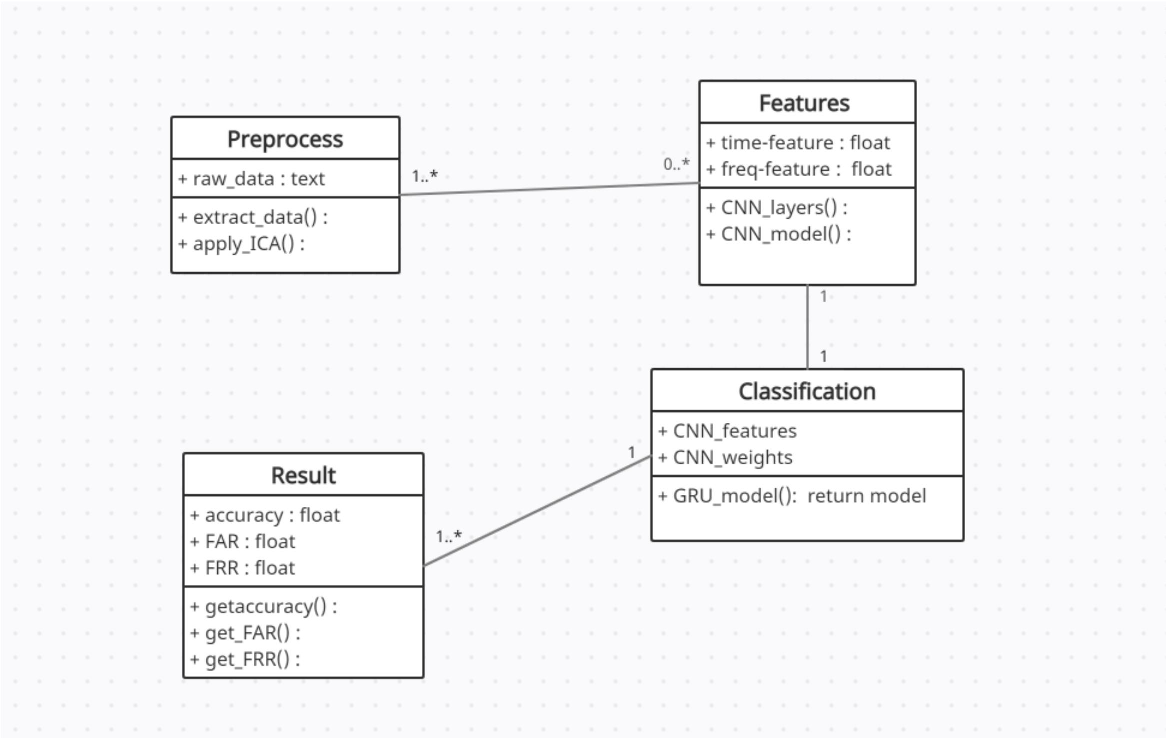

Figure 4.1: Class Diagram

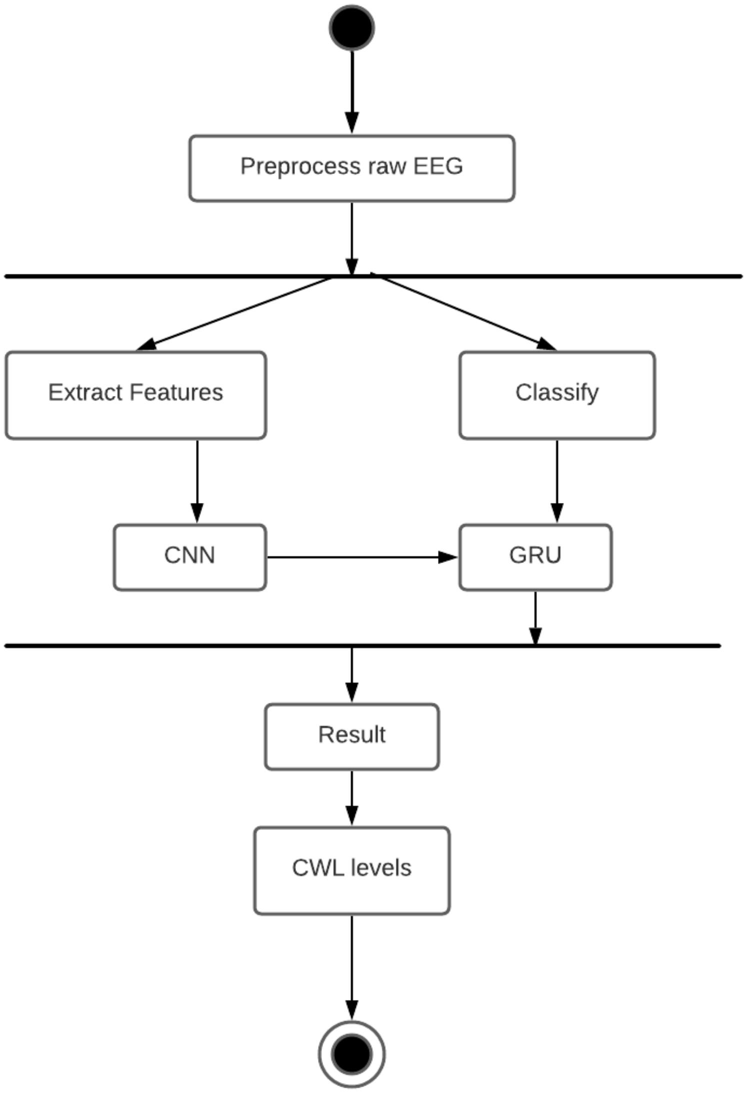

Figure 4.2: Activity Diagram

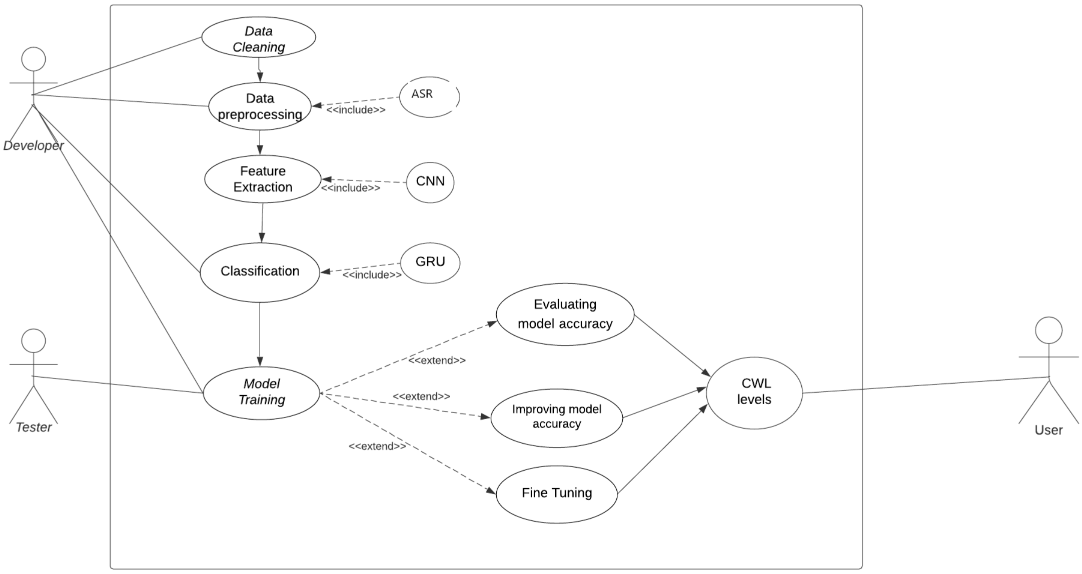

Figure 4.3: Usecase Diagram

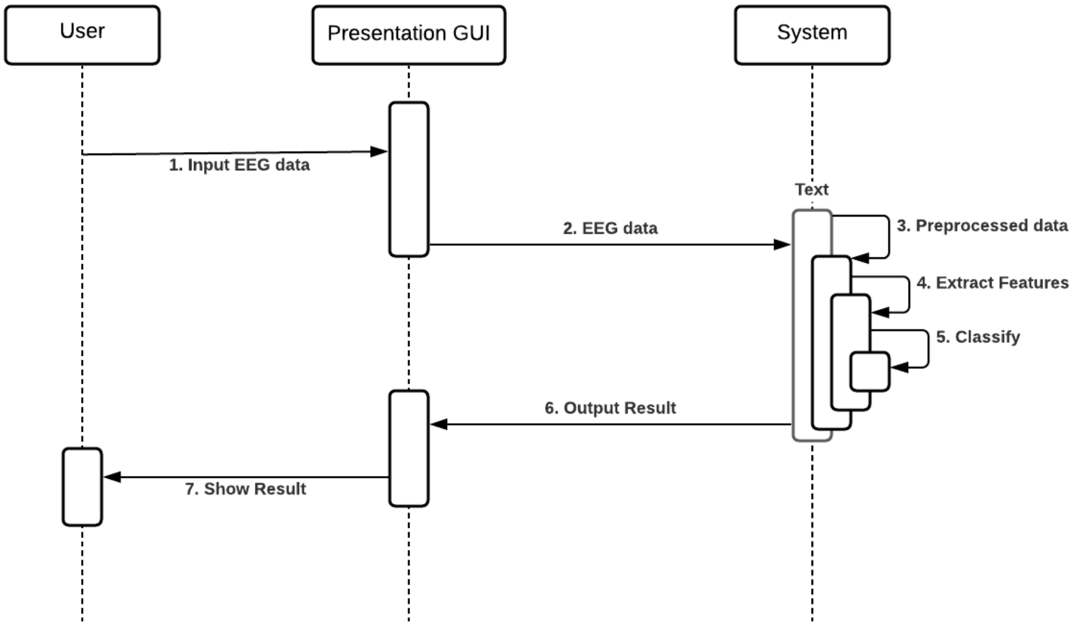

Figure 4.4: Sequence Diagram

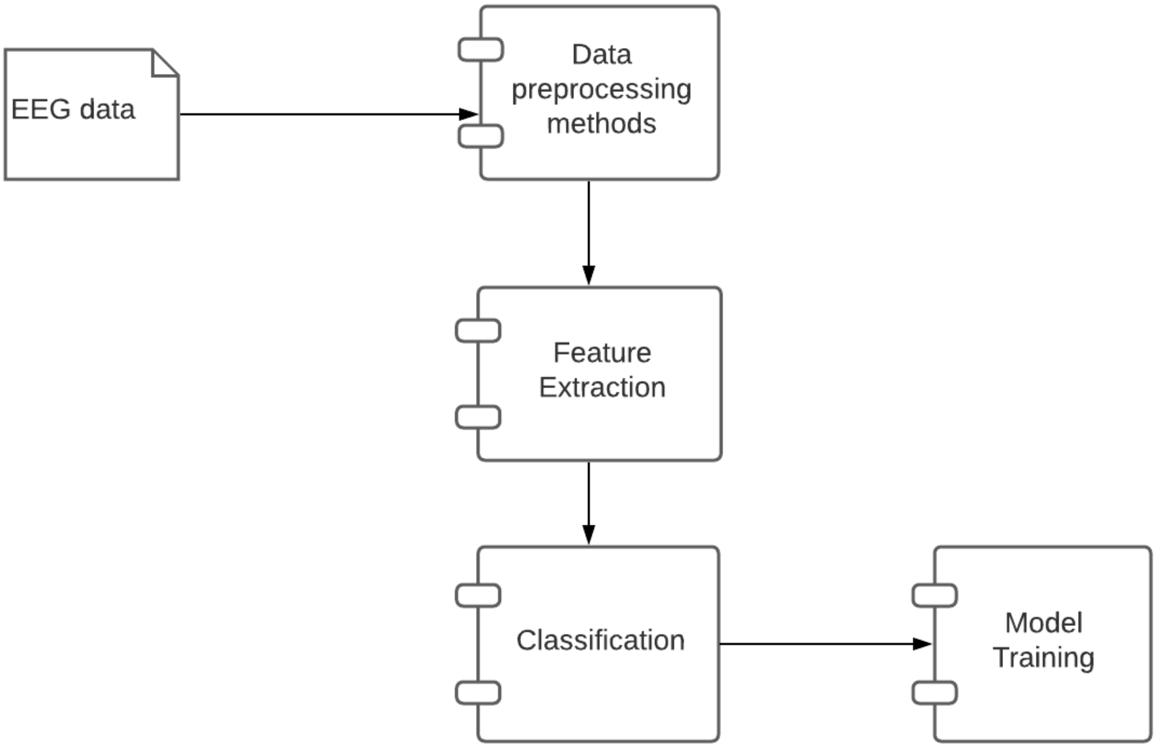

Figure 4.5: Component Diagram

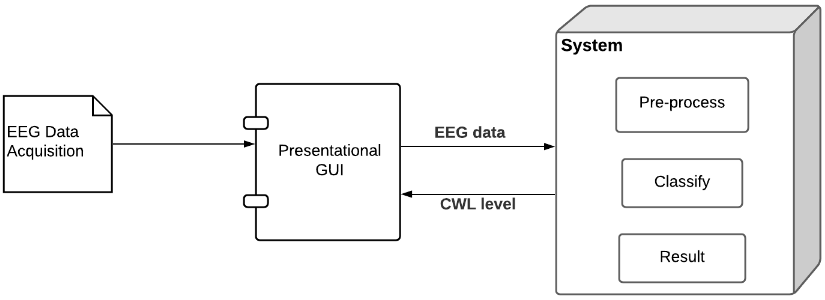

Figure 4.6: Deployment Diagram

CHAPTER 5
PROJECT PLAN

5.1 Project Estimate

We are using the SDLC model with iterations and associated streams for estimation.

5.1.1 Reconciled Estimates

1. **Cost Estimation:**
   This project prototype is developed free of cost as it has been developed using open source softwares and has no special hardware requirements. However, deployment of this project for industrial purposes may include an additional cost of AWS Cloud which is USD 4.5/day approximately.

2. **Time Estimation:**
   The initial time taken for data collection was about 4 weeks. This was followed by actual development of the system, that is, building CNN-GRU model. This took about another 20 days. After back end functionality was successfully completed, front end development was completed in about 10 days. Finally, documentation took another week to be completed. Thus, the total project took roughly 5 months to be completed, depending on the schedule of the developers.

5.1.2 Project Resources

1. A dedicated team of developers.

2. A computer with specifications as mentioned in the hardware requirements.

3. Software with specifications as mentioned earlier in the software requirement.

5.2 Risk Management

5.2.1 Risk Identification

1. Software Failure

2. Software Architecture Incompatibility

3. Restricted access to internal set up of database

5.2.2 Risk Analysis

The risks for the project can be analyzed within the constraints of time and quality.

1. **Software Failure :**
   Any components of the software, like GUI, may collapse any time due to unavoidable reasons like software or hardware specification incompatibility. The entire system may be halted until the component is mended.

2. **System Architecture Incompatibility :**
   The proposed system may not work if it is not compatible with the existing hardware and software specifications. It may be altered to suit available resources.

3. **Database Access :**
   Access to database must be restricted or else unauthorised access may cause inconsistency and other problems in the maintenance of database.

5.2.3 Overview of Risk Mitigation, Monitoring, Management

ID	Risk Description	Probability	Schedule	Quality	Overall
1	Loss of data if database crashes	Medium	Medium	High	High
2	Software Failure	Low	Medium	Medium	Low
3	Delay in project schedule	Medium	High	Medium	Medium
4	System Architecture Incompatibility	Medium	Medium	High	Medium
5	Inexperienced project members	Low	Medium	Medium	Medium
6	Unauthorized access to database	Medium	Medium	Medium	Medium

Figure 5.1: Risk Table

5.3 Project Schedule

Sr.No.	Task	Probable Duration
1	Topic Selection	15 days
2	Literature Review	15 days
3	Preliminary Studies and Background Knowledge	15 days
4	Application Prototype Development	1 month
5	Prototype Testing	1 month
6	Expansion and Future Work	15 days
7	Final Prototype, Results and Conclusion	2 months
8	Secondary Tests	10 days
9	Documentation	15 days

Figure 5.2: Project Schedule

5.4 Timeline Chart

Figure 5.3: Project Timeline

5.5 Team Organization

5.5.1 Team Structure

The project team consists of four people.

- Swarali Belsare, who performed data processing, code implementation and partial documentation.
- Maitreyi Kale, who performed data processing, code implementation and partial documentation.
- Priya Ghayal, who performed data processing, code implementation and partial documentation.
- Aishwarya Gogate, who performed data processing, code implementation and partial documentation.

5.5.2 Management reporting and communication

- Project Guide : Prof. Dr. Mrs. S.A.Itkar
- Project Coordinator : Prof. Mr. S.N.Deshpande

CHAPTER 6

PROJECT IMPLEMENTATION

6.1 Overview of Project Modules

1. **Cognitive Workload Prediction System:**
   The architecture of the system is designed in such a way that, when the user logs in to the software successfully, he can analyze his cognitive workload levels. He can upload csv file of his EEG recording. After uploading the file , he has to specify is if he is simkap or no-task. After specifying this , he can calculate his workload.

2. **Database Connectivity:**
   The user has to register himself first. After registering , he has to fill in some personal details. After logging in, the user can move further with his workload analysis.

6.2 Tools and Technologies Used

1. TensorFlow:
   TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.It is used to build and train ML models easily using intuitive high-level APIs like Keras with eager execution, which makes for immediate model iteration and easy debugging. It is available for download on Windows, Mac OS and Linux based operating systems.

2. Visual Studio:
   Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. It supports software development using multiple programming languages like C, Visual Basic, .NET, React, etc. It can be used across multiple operating systems like Windows, Mac OS, Linux, etc.

3. Pycharm: PyCharm is an integrated development environment used in computer programming, specifically for the Python language.PyCharm is designed to provide all the tools needed for productive Python development.

6.3 Algorithm Details

1. CNN - GRU :

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 1, 128)	18432128
batch_normalization (BatchNo	(None, 1, 128)	512
max_pooling1d (MaxPooling1D)	(None, 1, 128)	0
conv1d_1 (Conv1D)	(None, 1, 64)	8256
batch_normalization_1 (Batch	(None, 1, 64)	256
max_pooling1d_1 (MaxPooling1	(None, 1, 64)	0

Figure 6.1: CNN Model

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
sequential_2 (Sequential)    (None, 1, 64)             18445568
_________________________________________________________________
gru_3 (GRU)                  (None, 1, 64)             24768
_________________________________________________________________
gru_4 (GRU)                  (None, 32)                9312
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 66
=================================================================
Total params: 18,479,714
Trainable params: 18,479,202
Non-trainable params: 512
```

Figure 6.2: GRU Model

2. Steps Of Algorithms :
   (a) START.
   (b) Import the required libraries such as numpy, matlab, scikit.
   (c) Load the EEG recording file from user (csv).
   (d) Perform Data Prepossessing.
   (e) Split the data into training and testing data (Typically 80 percent and 20 percent respectively)
   (f) Create a CNN-GRU model.
   (g) Fit the created model using the training data
   (h) Use the trained model for predicting the cognitive workload levels.
   (i) STOP.

CHAPTER 7

SOFTWARE TESTING

7.1 Types Of Testing :

The various types of testing that may be used are as follows:

1. **Unit Testing:** Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

2. **Integration Testing:** Integration testing is used to check whether the small components accurately interact with each other accordingly to the instruction

3. **System Testing:** System Testing is the testing of a complete and fully integrated software product. Usually, software is only one element of a larger computer based system. Ultimately, software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

4. **White Box Testing:** White Box Testing is defined as the testing of a software solution's internal structure, design, and coding. In this type of testing, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application, improving design and usability, strengthening security. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box testing. It is usually performed by developers.

5. **Black Box Testing:** Black box testing is defined as a testing technique in which functionality of the Application Under Test (AUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on software requirements and specifications.In Black Box Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program.

6. **Regression Testing:** Regression Testing is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features.Regression Testing is nothing but a full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.

7. **Load Testing:** Load testing is a kind of Performance Testing which determines a system's performance under real-life load conditions. This testing helps determine how the application behaves when multiple users access it simultaneously.

8. **Stress Testing:** Stress testing is the process of determining the ability of a computer, network, program or device to maintain a certain level of effectiveness under unfavorable conditions. The process can involve quantitative tests done in a lab, such as measuring the frequency of errors or system crashes.

9. **Smoke Testing:** Smoke testing is also known as 'Build Verification Testing', is a type of software testing that comprises of a non-exhaustive set of tests that aim at ensuring that the most important functions work. The result of this testing is used to decide if a build is stable enough to proceed with further testing.

10. **Validation Testing:** Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfills its intended use when deployed on appropriate environment.

7.2 Test Cases and Test Results

Test Case Id	Test Case	Objectives	Steps	Expected Result	Actual Result	Test Status
1	Analysis	Check that all functionalities are working properly according to plan	Input parameters	All parameters must be input by user	Input accepted	Pass
			Invalid input	If invalid input is given, logs and error are noted.	Display errors	Pass
			Erroneous data	Discard erroneous data	Erroneous data discarded	Pass
2	System	Check that the system performance does not degrade	Performance	System performance should not degrade	System performs optimally	Pass
3	Applica-tion	Check that the application works properly with all system configuration and response time must not degrade	Efficient flow of UI	User must be able to login.	Quick access to system	Pass
			Adding new EEG data	Time taken to upload EEG data to the system must be optimal	Data is uploaded within optimal time period	Pass
			Data displayed on screen	The values displayed after analysis should be correct	The values are displayed correctly	Pass
4	Database	Data upload and extraction	Registration data upload and login credentials	Only authorized user should be able to access the system	Unauthorized access is not allowed	Pass

Figure 7.1: Test Cases

CHAPTER 8

RESULTS

8.1 Outcomes

1. Suitable cognitive workload level prediction with a high accuracy.
2. Stable desktop application successfully implemented to generate exepected results.
3. Serving as a fast and efficient way to help people access cognitive workload levels on a daily basis.

8.2 Screen Shots

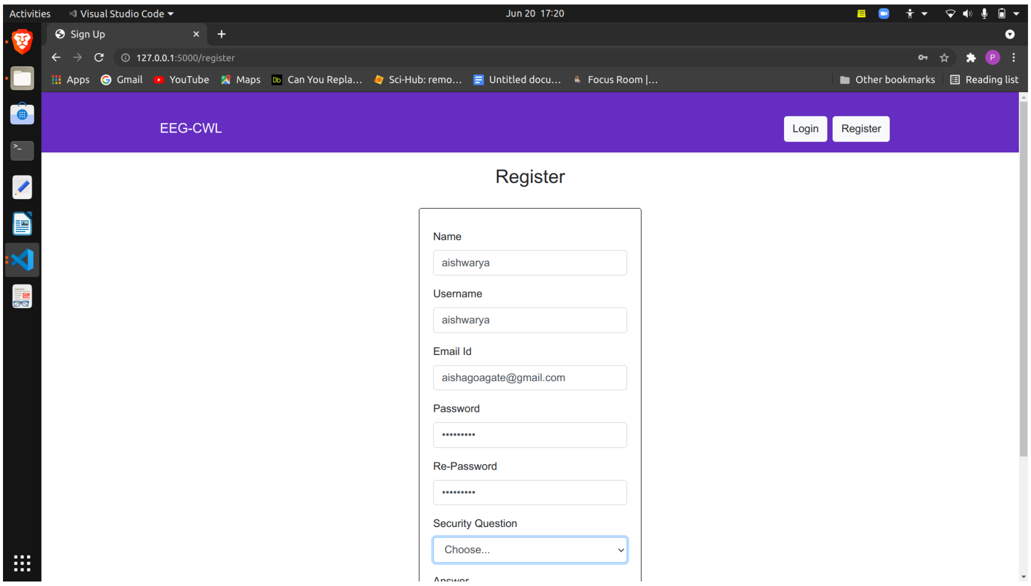

Figure 8.1: Register

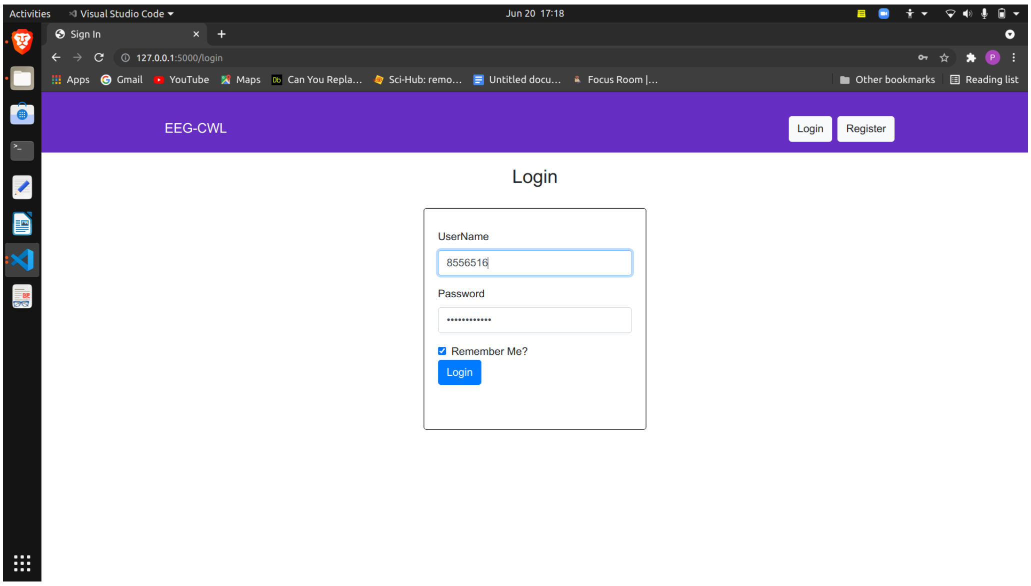

Figure 8.2: Login

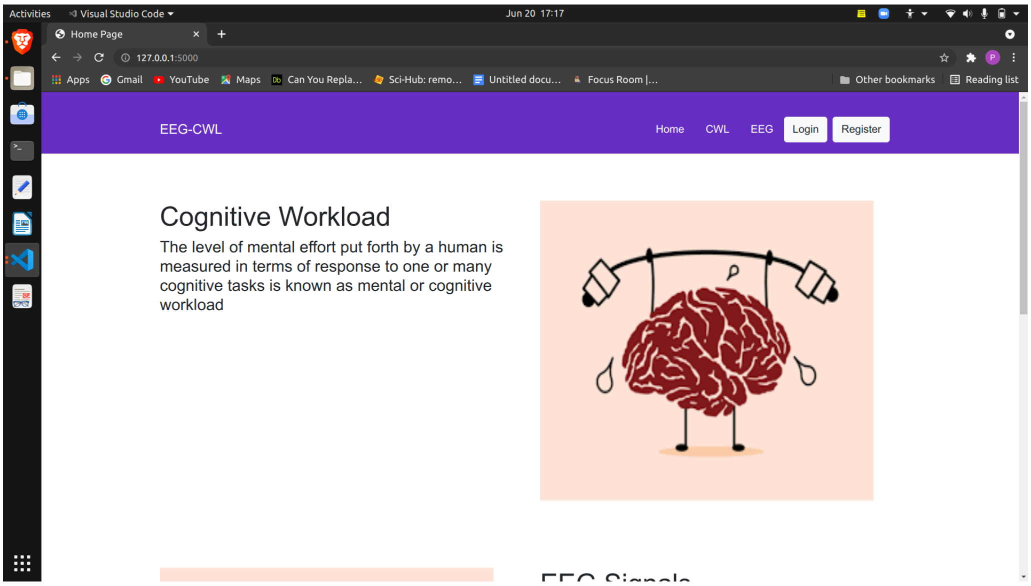

Figure 8.3: Front Page

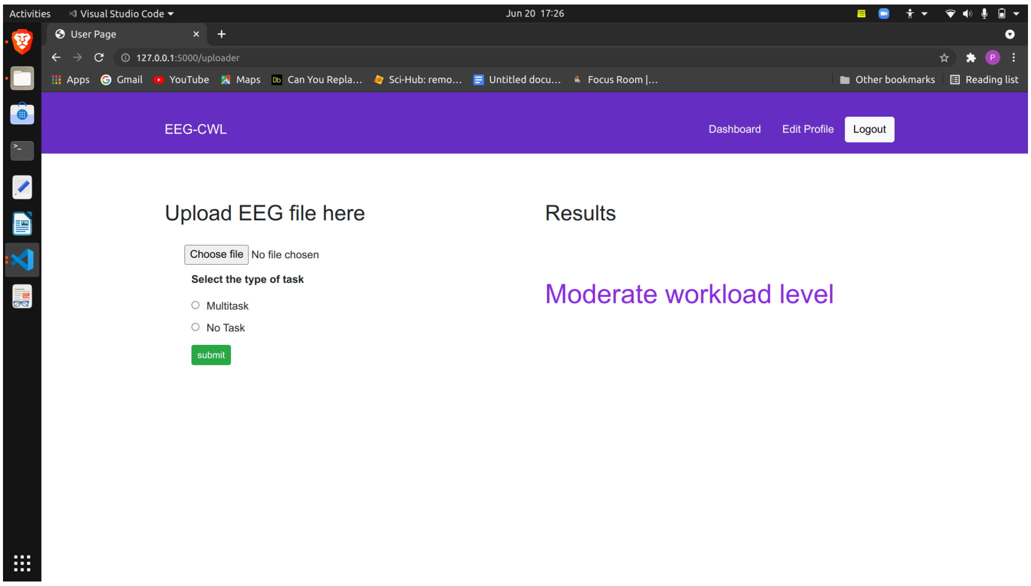

Figure 8.4: Workload evaluation

CHAPTER 9

CONCLUSIONS

9.1 Conclusions

1. Proposed a composite sequential model CNN-GRU for predicting cognitive workload levels.

2. Proposed model gives increased accuracy through oversampled data.

3. The cognitive workload levels is classified as low, moderate and high for both resting as well as multitasking condition shown via web application.

4. Proposed model has the ability to enhance the analysis process of EEG signals for any information extraction.

9.2 Future Work

Task type prediction along with workload level classification will enhance the automation process making it suitable for advancements in BCI applications.

9.3 Applications

1. Used in safety sector to examine pilot stress , driver fatigue.

2. Used as a performance metrics to evaluate student performance during different tasks.

ANNEXURE A

Problem statement feasibility assessment using, satisfiability analysis and NP Hard, NP-Complete or P type using modern algebra and relevant mathematical models.

ANNEXURE B

Details of paper publication

Title

Performance Comparison of Different EEG Analysis Techniques Based on Deep Learning Approaches

Conference

2021 IEEE International Conference on Emerging Smart Computing and Informatics (ESCI)

DOI

https://doi.org/10.1109/ESCI50559.2021.9396856

Published Paper

S. Belsare, M. Kale, P. Ghayal, A. Gogate and S. Itkar, "Performance Comparison of Different EEG Analysis Techniques Based on Deep Learning Approaches," 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), 2021, pp. 490-493.

Project Competition:

Participated in Round 2 of CSI-inApp 2021 Competition

CHAPTER 10

REFERENCES

[1] V. Pandey, D. K. Choudhary, V. Verma, G. Sharma, R. Singh, S. Chandra, "Mental Workload Estimation Using EEG", Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), May 2020 .

[2] Das Chakladar, D., Dey, S., Roy, P. and Dogra, D., "EEG-based mental work load estimation using deep BLSTM-LSTM network and evolutionary algorithm" Biomedical Signal Processing and Control,vol. 60, May 2020.

[3] Gao, Z., Wang, X., Yang, Y., Mu, C., Cai, Q., Dang, W. and Zuo, S., "EEG-Based Spatio–Temporal Convolutional Neural Network for Driver Fatigue Evaluation", IEEE Transactions on Neural Networks and Learning Systems, Volume 30(9), January 2019.

[4] Wu, E., Deng, P., Qu, X., Tang, Z., Zhang, W., Zhu, L., Ren, H., Zhou, G. and Sheng, R., "Detecting Fatigue Status of Pilots Based on Deep Learning Network Using EEG Signals", IEEE Transactions on Cognitive and Developmental Systems, January 2020.

[5] Budak, U., Bajaj, V., Akbulut, Y., Atila, O. and Sengur, A., "An Effective Hybrid Model for EEG-Based Drowsiness Detection", IEEE Sensors Journal, vol. 19(17), September 2019.

[6] Lim, W., Sourina, O. and Wang, L., "STEW: Simultaneous Task EEG Workload Data Set." IEEE Transactions on Neural Systems and Rehabilitation Engineering", vol. 26(11), November 2018.

[7] Saha A., Minz V., Bonela S., Sreeja S.R., Chowdhury R., Samanta D. "Classification of EEG Signals for Cognitive Load Estimation Using Deep Learning Architectures." In: Tiwary U. (eds) Intelligent Human Computer Interaction. IHCI 2018. Lecture Notes in Computer Science, vol 11278. Springer, 2018

[8] P. Nagabushanam, S. Thomas George ,S. Radha ," EEG signal classification using LSTM and improved neural network algorithm," Soft Computing, vol. 24, November 2019

[9] A. Saha, V. Minz, S. Bonela, S. R. Sreeja, R. Chowdhury, D. Samanta, "Classification of EEG Signals for Cognitive Load Estimation Using Deep Learning Architecture", International Conference on Intelligent Human Computer" vol 11278. Springer, November 2018

[10] B. Hosseinifard, M. H. Moradi, R. Rostami, "Classifying depression patients and normal subjects using machine learning techniques and nonlinear features from EEG signal", Computer Methods and Programs in Biomedicine, Vol 103 Issue 3, March 2013

[11] R. Ramirez, Z. Vamvalousis, "Detecting Emotion from EEG Signals using the Emotive Epoch Device", Brain Informatics., vol 7670. Springer, Berlin, Heidelberg., 2012

[12] P. Liping, G. Liang, Z. Jie, W. Xiaoru, Q. Hongquan, W. Xin, "Subject-specific mental workload classification, using EEG and stochastic configuration network(SCN)", Biomedical Signal Processing and Control, Vol 68, July 2021

[13] S. Midha, H. A. Maior, M. L. Wilson, S. Sharples, "Measuring Mental Workload Variations in Office Work Tasks using fNIRS", International Journal of Human-Computer Studies, Vol. 147, March 2021