STRESS DETECTION USING PHYSIOLOGICAL SENSOR DATA

Bonafide record of work done by

HARSHA VARDHAN V M	(20Z217)
KABILAN K K	(20Z223)
PRANAV P	(20Z237)
PREDNYA RAMESH	(20Z239)
SUVAN SATHYENDIRA B	(20Z256)

19Z720 – PROJECT WORK 1

GUIDE: Dr.G.R.Karpagam

Dissertation submitted in the partial fulfillment of
the requirements for the degree of

BACHELOR OF ENGINEERING

BRANCH: COMPUTER SCIENCE AND ENGINEERING

of Anna University

November 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PSG COLLEGE OF TECHNOLOGY
(Autonomous Institution)

COIMBATORE – 641 004

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641 004

Bonafide record of work done by

HARSHA VARDHAN V M	(20Z217)
KABILAN K K	(20Z223)
PRANAV P	(20Z237)
PREDNYA RAMESH	(20Z239)
SUVAN SATHYENDIRA B	(20Z256)

Dissertation submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF

Branch: Computer Science and Engineering

....

Dr.G.R.Karpagam – Faculty guide

Dr. Sudha Sadasivam G. Head of the Department.

Certified that the candidate was examined in the viva-voce examination held on ....

....

(Internal Examiner)

Certificate

This is to certify that Mr. Harsha Vardhan V M (20Z217), Mr. Kabilan K K (20Z223), Mr. Pranav P (20Z237), Ms. Prednya Ramesh (20Z239) and Mr. Suvan Sathyendira B (20Z256) of BE (CSE) semester 7 has worked on a project entitled Stress Detection Using Physiological Sensor Data under Dr.G.R.Karpagam during June - November 2023 as a part of the course on 19Z720 - Project Work 1.

Investigator

____

Dept. of CSE
PSG College of Technology
Coimbatore

ACKNOWLEDGEMENT

We would like to extend sincere thanks to our Principal, Dr. K. Prakasan, for providing us with this opportunity to develop and complete our project in our field of study.

We would like to express our sincere thanks to our Head of Department, Computer Science and Engineering, Dr. G. Sudha Sadasivam, for encouraging and allowing us to present the project at our department premises for the partial fulfillment of the requirements leading to the award of Bachelor of Engineering degree.

We take immense pleasure in conveying our thanks and a deep sense of gratitude to the Programme Coordinator, Dr N. Arul Anand, Department of Computer Science and Engineering, for providing the necessary resources, facilities, and opportunities to pursue our undergraduate project.

We would like to thank our Tutor, Mrs T. Anusha, Department of Computer Science and Engineering, for her valuable input, guidance, and encouragement throughout our project.

We extend our heartfelt thanks and a deep sense of gratitude to the Faculty Guide, Dr. G. R. Karpagam, Department of Computer Science and Engineering for her valuable insights, suggestions, and inputs that have helped us enhance the quality of our project. Her expertise and knowledge have been instrumental in making this project a success. We would like to thank Ms. Vani K, Committee Member for her valuable suggestions and motivation throughout the project.

SYNOPSIS

The Stress Detection Using Physiological Sensor Data project aims to delve into the relationship between physiological responses and stress levels in individuals. By harnessing various sensor technologies to gather data such as acceleration, skin conductance, and temperature, this initiative seeks to capture the nuanced physical changes that accompany stress.

In order to achieve this, the project will employ a comprehensive array of data collection techniques, including the deployment of wearable sensors in real-world scenarios. Participants will be monitored during activities designed to induce varying levels of stress, ensuring a robust dataset reflective of genuine stress responses.

Machine learning models, especially those adept at handling time-series data, will be utilized to sift through the collected information and identify indicative stress signatures. The efficacy of these models will be meticulously assessed through rigorous validation processes to confirm their predictive accuracy.

The culmination of this project will involve a thorough analysis of the machine-learned correlations and their implications for stress detection. By interpreting these findings, the project aspires to forge innovative pathways for early stress identification and management. The potential impacts of this research are substantial, offering insights that could influence the development of new health monitoring applications and interventions across both clinical and everyday settings.

TABLE OF CONTENTS **PAGE**

CHAPTER 1
INTRODUCTION

Chapter 1 deals with the introduction of the project. It describes the motivation, Problem Statement, Project Objectives and block diagram depicting the overall work of the title of the project work.

1.1 PROBLEM STATEMENT:

Identify a person's stress level as either under stress, in a non-stress state, or experiencing amusement by utilizing a wearable device designed to classify these emotional states.

1.2 MOTTO:

With an abundance of physiological data now accessible online, the traditional stress detection methods, which relied on individual factors such as blood pressure, heart rate, and additional parameters, have advanced into sophisticated devices capable of comprehensively recording and categorizing these factors as indicators of stress, eliminating the need for a doctor's diagnosis. This innovation has far-reaching applications in various fields, such as healthcare, psychology, and wellness, offering improved stress management strategies, early intervention, and personalized treatment plans.

1.3 ORGANIZATION OF REPORT:

Chapter 1: deals with introduction and problem statement of the project.
Chapter 2: deals with survey of the existing works in the scope of project work.
Chapter 3: deals with the designed system’s workflow, modules and dataset.
Chapter 4: utilized for model construction and deep insight into the system.
Chapter 5: deals with the hardware and software specifications of the system.
Chapter 6: explains how the system is implemented with the help of models.
Chapter 7: provides an insight on how the project can be extended for future work.

CHAPTER 2
LITERATURE SURVEY

Chapter 2 deals with the survey of the existing works in the scope of the project work. It presents the key takeaways from different research papers and articles.

2.1 INTRODUCTION:

The advent of wearable technologies combined with the analytical prowess of machine learning models has catalyzed significant advancements in the realm of mental stress detection. These methodologies are meticulously chronicled across various studies, each shedding light on the nuanced processes from data preprocessing to sophisticated classification. Preprocessing techniques, pivotal in refining physiological signals, span the gamut from noise filtration and normalization to the more intricate practices of feature extraction and dimensionality reduction. In the classification terrain, a tapestry of models emerges, from the classical elegance of Support Vector Machines and Decision Trees to the intricate depths of Convolutional Neural Networks and ensemble strategies. Results aggregated from these explorations articulate the precision with which these systems discern stress, marking a commendable stride in mental health monitoring's journey from conceptualization to practical application in diverse and dynamic environments.

The series of steps involved in this process are (however some of the substeps could also be optional):

Pre-Processing:

- Noise Filtering
- Normalization
- Feature Extraction
- Artifact Handling
- Signal Segmentation
- Dimensionality Reduction
- Feature Selection
- Time Series Epoching

Feature Extraction:

- Heart Rate Variability (HRV)
- Galvanic Skin Response (GSR)
- Skin Temperature
- Blood Volume Pulse (BVP)
- Respiration Rate
- Accelerometer Data
- Photoplethysmography (PPG) Signals
- Feature Engineering for Signal Patterns

Classification Models:

- Support Vector Machines (SVMs)
- Decision Trees
- k-Nearest Neighbors (k-NN)
- Convolutional Neural Networks (CNNs)
- Deep Learning Architectures

Results and Discussion:

- Interpretation and analysis of the results to understand the correlation between physiological indicators and stress levels using wearable sensor data and machine learning techniques.

2.2 RELATED WORKS:

1. A Review on Mental Stress Detection Using Wearable Sensors and Machine Learning Techniques

This paper provides a comprehensive survey of the methodologies that employ wearable sensors for mental stress detection. It discusses the evolution of machine learning techniques in interpreting the physiological data captured by these sensors and evaluates their effectiveness in identifying stress. It encompasses a broad spectrum of machine learning techniques, ranging from classical approaches like Support Vector Machines (SVMs) and random forest to cutting-edge deep learning methods, providing a meta-analysis of their effectiveness in stress detection through wearable tech.

2. Stress Detection Using Physiological Sensors

This study focuses on the use of physiological sensors to detect stress, exploring various biometric signals that can serve as stress indicators. The paper details the relationship between these physiological markers, such as heart rate and skin conductance, and the occurrence of stress. Primarily concentrate on the sensor technologies and the physiological signatures of stress, potentially touching upon the standard machine learning classifiers like SVM, KNN, Random Forest that are applied to such physiological data for stress identification.

3. Stress Detection Using Wearable Physiological Sensors

The emphasis is on the wearable nature of physiological sensors and their application in stress detection. The paper looks into the convenience and continuous monitoring capabilities offered by wearable devices, and how they can be used to gather data for stress analysis in everyday settings. It discusses the use of common machine learning algorithms, emphasizing the adaptability and continuous monitoring advantages of wearable devices, alongside specialized feature extraction techniques designed for sensor data.

4. Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection

This paper likely includes benchmarks or references to various algorithms tested on the WESAD dataset, which could range from traditional machine learning to deep learning models.It offers a foundational dataset that is pivotal for benchmarking and improving stress detection models. While not focused on specific machine learning techniques, it supports a variety of algorithms tested against its multimodal data.

5. Detecting Work Stress in Offices by Combining Unobtrusive Sensors

The study explores stress detection in office environments by utilizing unobtrusive sensors. It investigates the viability of detecting work-induced stress discreetly, without disrupting the natural workflow of employees, thereby offering insights into stress management in professional settings. This entails the use of unsupervised learning techniques for anomaly detection (to identify stress patterns) or supervised learning if specific stress indicators are being monitored.

6. A Machine learning approach for stress detection using a wireless physical activity

This research examines the application of machine learning algorithms in detecting stress through wireless monitoring of physical activity. It suggests that certain patterns of physical activity, as captured by wireless sensors, can be indicative of stress, which machine learning models can successfully identify. It focuses on machine learning algorithms capable of processing time-series data from wireless sensors, possibly including LSTM networks or other sequence-processing deep learning models.

2.3 OBSERVATIONS:

2.3.1 RANDOM FOREST:

Random Forest is an ensemble learning method for classification, regression, and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Size - 3.45 MB
Accuracy - 0.7512

Random forests correct for decision trees' habit of overfitting to their training set by combining the predictions of multiple decision trees, each trained on a random subset of the training data. This process of averaging or "voting" among predictions stabilizes and improves accuracy, making Random Forest a powerful and versatile machine learning technique.

CHAPTER 3
PROPOSED SYSTEM

Chapter 3 deals with the designed system's workflow along with its description, list of modules, and the dataset used for training purposes.

3.1 WORKFLOW:

1. **Data Acquisition and Preparation:** Utilizes the WESAD dataset, with a focus on data from wrist sensors, followed by preprocessing to refine the data for analysis
2. **Feature Engineering:** Involves extracting key features like ACC, EDA, and TEMP from the wrist sensor data that are relevant for identifying stress levels
3. **Model Development:** Splits the dataset into training and testing sets, with the training set used to train the machine learning model using Random Forest to recognize stress indicators
4. **Stress Prediction and Evaluation:** Employs the trained model to predict Stress vs Non-Stress vs Amusement and evaluates its accuracy
5. **Deployment and Feedback:** Integrates the model with a web interface, providing real-time stress level feedback to the user based on the analysis of wristband data

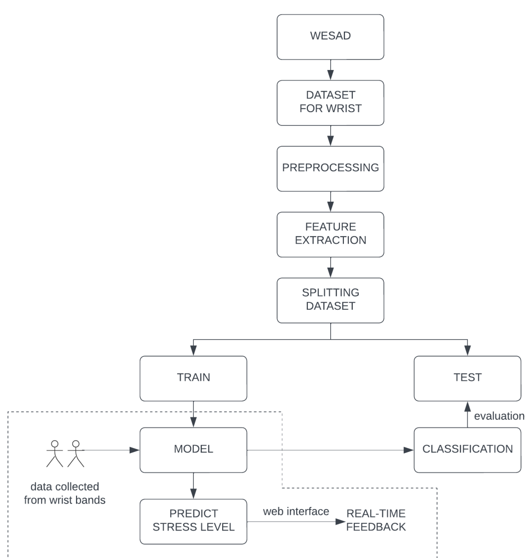


Figure 1: Workflow of the project

3.2 MODULES:

3.2.1 Input Preprocessing:

Every subject's (S2-S16) pickle file that contains the sensor data and labels is converted into a CSV file, essential columns are retrieved and normalized (z-score).

3.2.2 Classification Model:

1. Random Forest: Random Forest is an ensemble machine learning model that is widely used for classification tasks. Its architecture is based on the combination of multiple decision trees to make more accurate predictions. Each decision tree in the ensemble is constructed using a random subset of the training data and a random subset of features. This randomness helps reduce overfitting and makes the model more robust.

The architecture of a Random Forest model consists of three key components:

Decision Trees: These are the building blocks of the Random Forest. Multiple decision trees are created, each trained on a different subset of the training data. Decision trees partition the feature space into regions, making binary decisions at each node based on specific features. This allows the model to capture complex relationships in the data.

Random Subsampling: Random Forest introduces randomness by selecting a random subset of data points from the training dataset for each tree. This process is known as bootstrapping. Additionally, at each node of a decision tree, only a random subset of features is considered for splitting. This randomness reduces the correlation between trees, making the model less prone to overfitting.

Voting or Averaging: During prediction, each tree in the Random Forest provides a class prediction. For classification tasks, the final prediction is determined through a majority vote among the predictions of all the individual trees. For regression tasks, it's done through averaging.

The architecture of a Random Forest model allows it to handle high-dimensional data, capture complex relationships, and maintain good predictive performance. It is robust, resistant to overfitting, and suitable for various classification tasks.

3.3 TRAINING DATASET:

The dataset utilized in this study comprises data from wristbands, specifically the Empatica E4, and chest bands known as RespiBAN. These devices collected various physiological measurements, including ECG, EMG, EDA, TEMP, and Respiration (RespiBAN), as well as ACC, BVP, EDA, and TEMP (Empatica E4). These measurements were stored in a structured format within a pickle file, with each entry containing essential information. The 'subject' field indicated the subject's unique ID, while the 'signal' field encompassed raw data categorized into 'chest' (RespiBAN data) and 'wrist' (Empatica E4 data). The 'label' field provided crucial classification values, with 0 representing 'transient,' 1 for 'baseline,' 2 indicating 'stress,' 3 representing 'amusement,' 4 signifying 'meditation,' and 5/6/7 suggesting data points to be disregarded within this dataset.

Data collection for this study involved exposing participants to specific situations engineered to reliably induce stress. The primary stress-inducing task employed was the "Trier Social Stress Test (TSST)," a well-established stress elicitation method. The TSST involved two main components: a public speaking task and a mental arithmetic task. During the TSST, participants were tasked with delivering a five-minute speech in a public setting, which introduced a social evaluative component. Additionally, participants were required to perform a mental arithmetic task under time constraints and continuous evaluation, with task difficulty adjusted dynamically based on individual performance.

To gain a comprehensive understanding of participants' emotional and mental states, various questionnaires were administered, including the "STAI-Y anxiety questionnaire." These questionnaires aimed to assess anxiety levels and mood states among participants. The questionnaire data complemented the physiological signals collected, enabling researchers to correlate subjective experiences of stress with objective physiological responses.

CHAPTER 4
SYSTEM DESIGN

Chapter 4 gives deep insight into the design of the system through use case diagram and sequence diagram and the architecture utilized for model construction using graphics that provide a full explanation of the layers existing deep within the model.

4.1 DESIGN:

4.1.1 Use Case Diagram:

The functionality of our system is explained through the following use-case diagram:

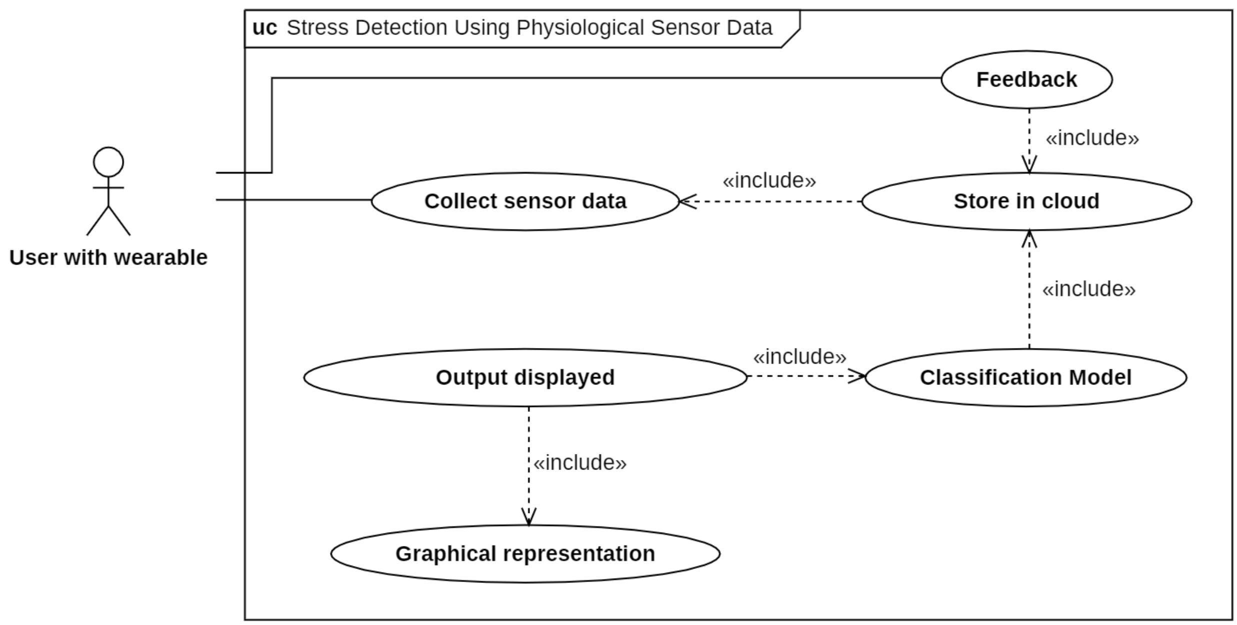


Figure 2: Use Case diagram

4.1.2 Sequence Diagram:

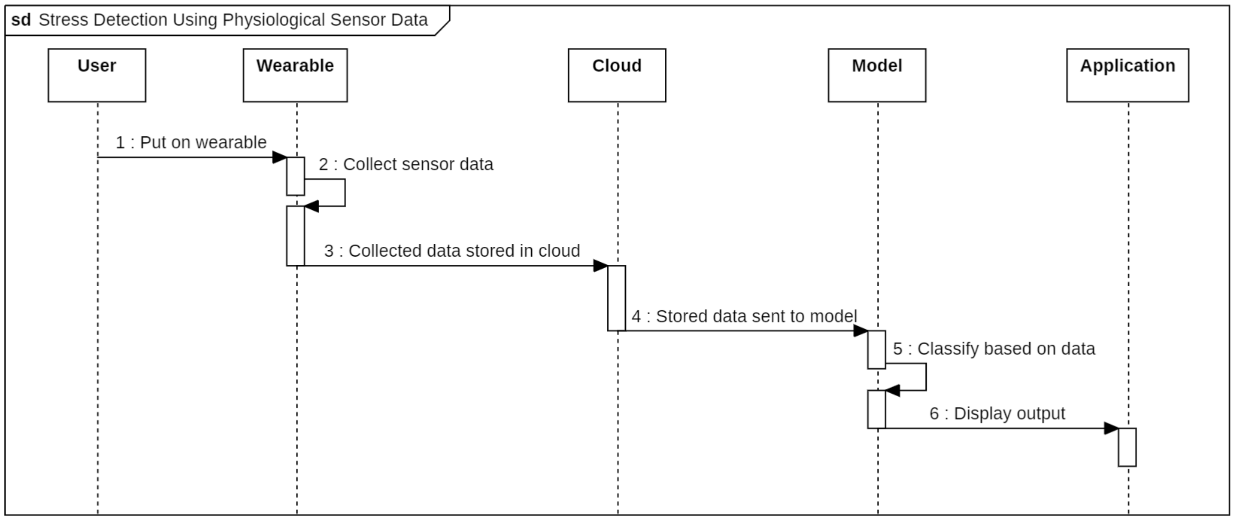


Figure 3: Sequence diagram

4.2 ARCHITECTURE:

4.2.1 Random Forest:

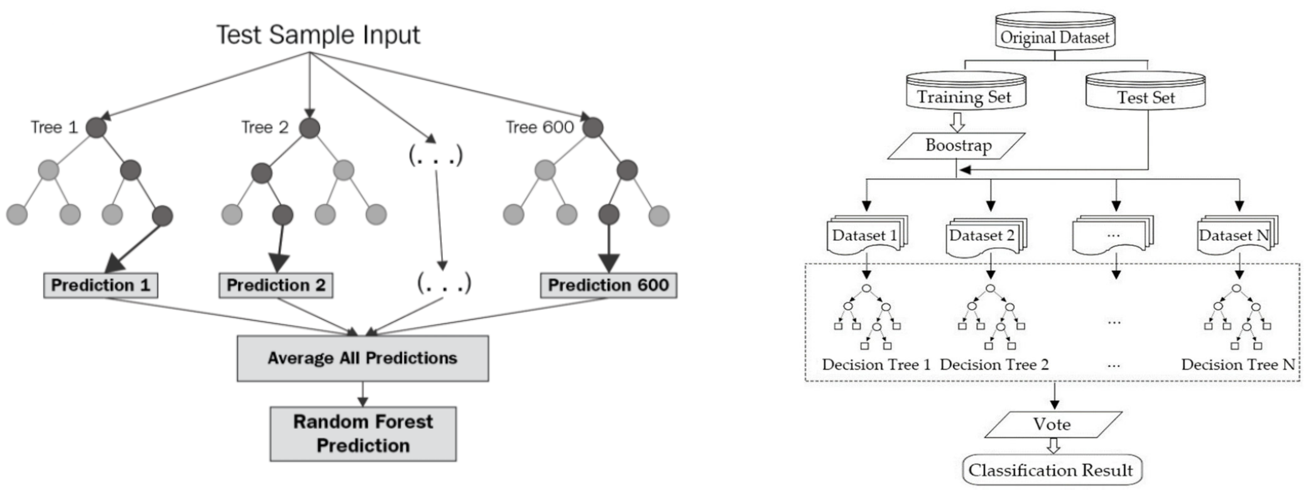


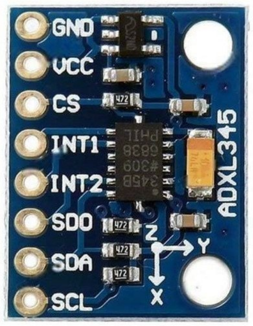
Figure 4: Random Forest for Classification Model

CHAPTER 5
SYSTEM SPECIFICATIONS

Chapter 5 will focus on discussing the specifications of both hardware and software of the system.

5.1 HARDWARE:

Components	Description
 **NodeMCU (ESP8266)**	NodeMCU is chosen primarily for its built-in Wi-Fi capabilities. It's a versatile microcontroller that is used to collect data from sensors and transmit it wirelessly to a central system or cloud for further analysis and monitoring.
 **GSR Sensor (EDA - Electrodermal Activity)**	GSR sensors can accurately capture changes in skin conductance, which is a strong indicator of stress and emotional arousal. Its non-invasive design ensures participant comfort, while its heightened sensitivity to emotional responses, like stress or excitement, makes it ideal for assessing psychological arousal. GSR sensors are user-friendly, necessitating minimal setup, and provide real-time data, facilitating immediate reaction monitoring.

<table>
<tr>
<td>


ADXL345 Sensor (Acceleration)
</td>
<td>The ADXL345 is an accelerometer that measures acceleration in multiple axes. It's useful for detecting motion, orientation, and vibrations. It has a 13-bit resolution and adjustable sensitivity, allowing precise measurement of movement and postural changes, which contribute to stress assessment.</td>
</tr>
<tr>
<td>


DS18B20 Sensor (Temperature)
</td>
<td>DS18B20 is a digital temperature sensor known for its accuracy and reliability. It has great simplicity, affordability, and suitability for measuring ambient temperature changes, which can provide insights into the body's response to stress.This combination provides various advantages such as ease of programming, cost-effective solution.</td>
</tr>
</table>

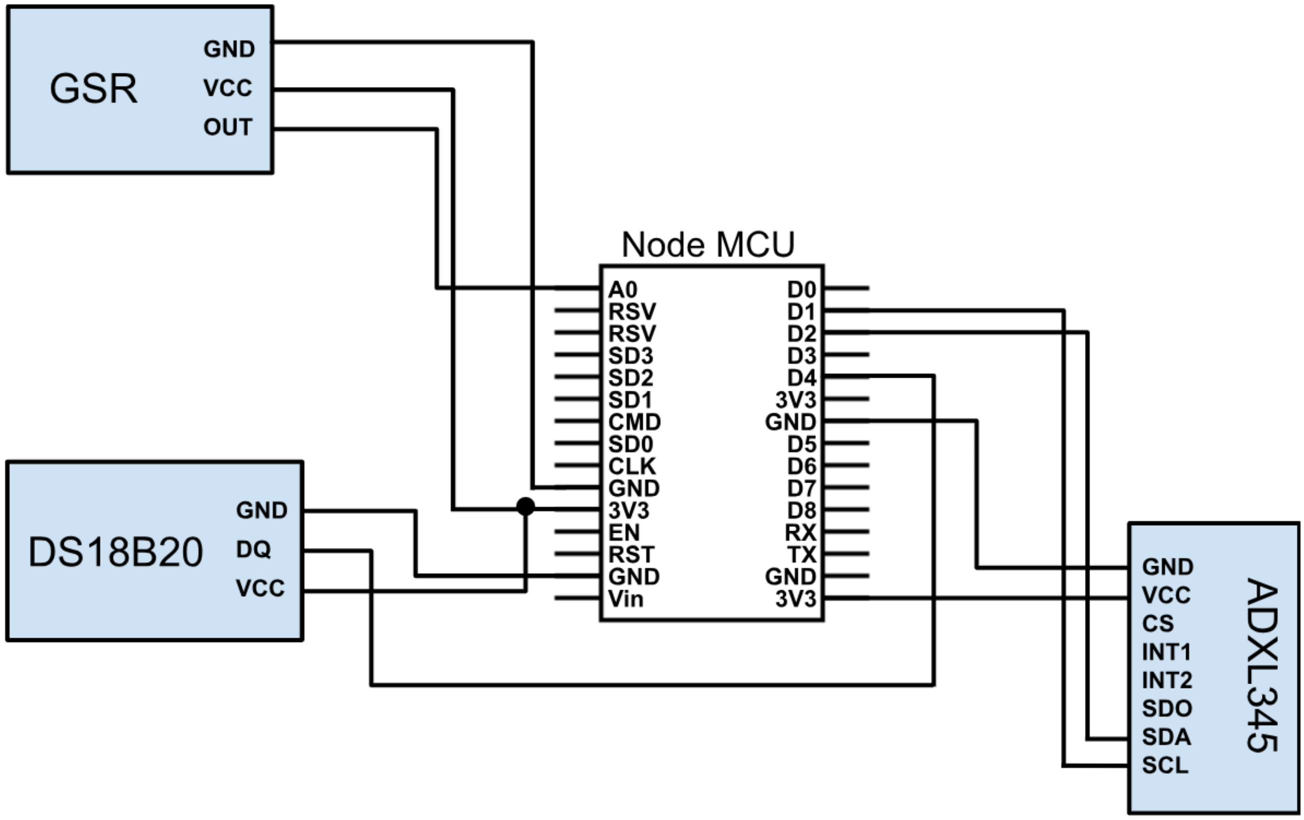


Figure 5: Connection Diagram

Below are the connection details for each sensor with specific pins on a NodeMCU board:

GSR Sensor (EDA):

GSR Sensor VCC to NodeMCU 3.3V
GSR Sensor GND to NodeMCU GND
GSR Sensor Signal (Analog Output) to NodeMCU A0 (Analog Pin 0)

ADXL345 Sensor (Acceleration):

ADXL345 Sensor VCC to NodeMCU 3.3V
ADXL345 Sensor GND to NodeMCU GND
ADXL345 Sensor SDA to NodeMCU D2 (GPIO4) - I2C Data Line
ADXL345 Sensor SCL to NodeMCU D1 (GPIO5) - I2C Clock Line

DS18B20 Sensor (Temperature):

DS18B20 Sensor VCC to NodeMCU 3.3V
DS18B20 Sensor GND to NodeMCU GND
DS18B20 Sensor Data Pin to NodeMCU D4 (GPIO2) - Data pin for One-Wire communication

5.2 SOFTWARE:

- Using the programming language as Python.
- Tools for data pre-processing and cleaning, such as NumPy and Pandas.
- Machine learning libraries such as scikit-learn.
- Visual Studio IDE, Arduino IDE
- HTML, CSS and JavaScript for the website.

CHAPTER 6
IMPLEMENTATION

Chapter 6 explains how the system is implemented with the help of modules.

Windows platform with command prompt or Linux with terminal or any suitable IDE for python development or Google Colab is used.

6.1 DATA TRANSFORMATION, VISUALIZATION AND PREPROCESSING:

6.1.1 Data Transformation:

During the data transformation phase, individual records in the WESAD dataset are represented by pickle dictionaries containing arrays of parameters for each subject. These arrays vary in length corresponding to different wrist parameters. To achieve uniformity and synchronize the data, all parameter arrays were downsampled to a frequency of 4Hz. This downsampling standardizes the array lengths across all parameters, ensuring that each has the same temporal resolution and duration for analysis.

```
{'s2_data':
    {'signal':
        {'chest':
            {'ACC': "<class 'numpy.ndarray'>",
            'ECG': "<class 'numpy.ndarray'>",
            'EMG': "<class 'numpy.ndarray'>",
            'EDA': "<class 'numpy.ndarray'>",
            'Temp': "<class 'numpy.ndarray'>",
            'Resp': "<class 'numpy.ndarray'>"
            },
        'wrist':
            {'ACC': "<class 'numpy.ndarray'>",
            'BVP': "<class 'numpy.ndarray'>",
            'EDA': "<class 'numpy.ndarray'>",
            'TEMP': "<class 'numpy.ndarray'>"
            }
        },
    'label': "<class 'numpy.ndarray'>",
    'subject': "<class 'str'>"
    }
}
```

```
{'s2_data':
    {'signal':
        {'chest':
            {'ACC': 4255300,
            'ECG': 4255300,
            'EMG': 4255300,
            'EDA': 4255300,
            'Temp': 4255300,
            'Resp': 4255300
            },
        'wrist':
            {'ACC': 194528,
            'BVP': 389056,
            'EDA': 24316,
            'TEMP': 24316
            }
        },
    'label': 4255300,
    'subject': 1
    }
}
```

```
downsampling_factors = {
    'ACC': 32 // 4,
    'EDA': 4 // 4,
    'TEMP': 4 // 4,
    'label': 700 // 4
}
```

Figure 6: S2 pkl dictionary, their length of parameters and sampling rate

6.1.2 Data Visualization:

The following table presents the distribution of different labels within our dataset, where each label corresponds to a specific emotional or activity state: Label 0 represents "transient," Label 1 is associated with "baseline," Label 2 indicates "stress," Label 3 signifies "amusement," and Label 4 corresponds to "meditation." Labels 5, 6, and 7 are designated as non-relevant and are disregarded in this particular dataset.

Label	Composition	Percentage (%)
0	0.454793	45.48%
1	0.202733	20.27%
4	0.135907	13.59%
2	0.114726	11.47%
3	0.064178	6.42%
7	0.009486	0.95%
6	0.009094	0.91%
5	0.009083	0.91%

The image displayed illustrates the wristband signals for Subject 2. It provides a visual representation of various parameters, along with their corresponding timestamps and associated values.

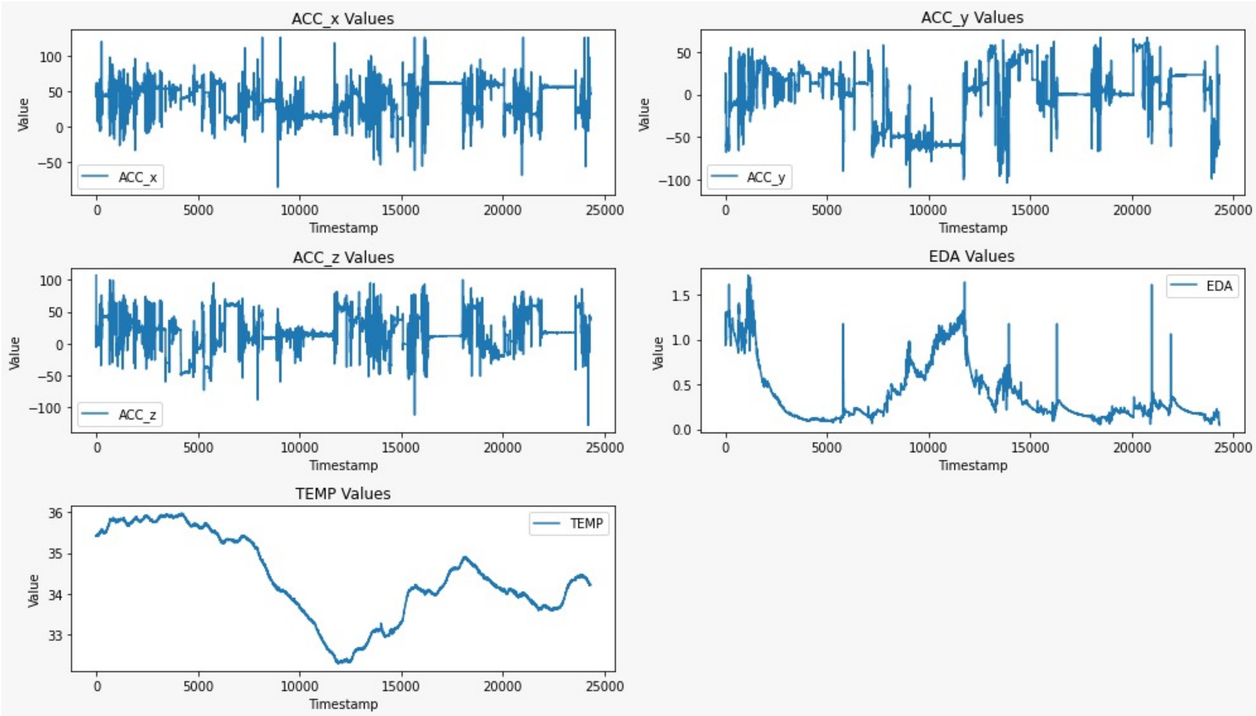


Figure 7: Subject 2 data plotting (valued vs timestamp)

The image below graphically displays the signals for each parameter, with the x-axis representing timestamps and the y-axis depicting the parameter values. The below shows how Subject 2 was exposed to various situations in order to understand the variations of their signals.

Label	Colour
0	
1	
2	
3	
4	

A legend or color key has been included within the graph, offering a clear reference for interpreting the labels associated with various signals.

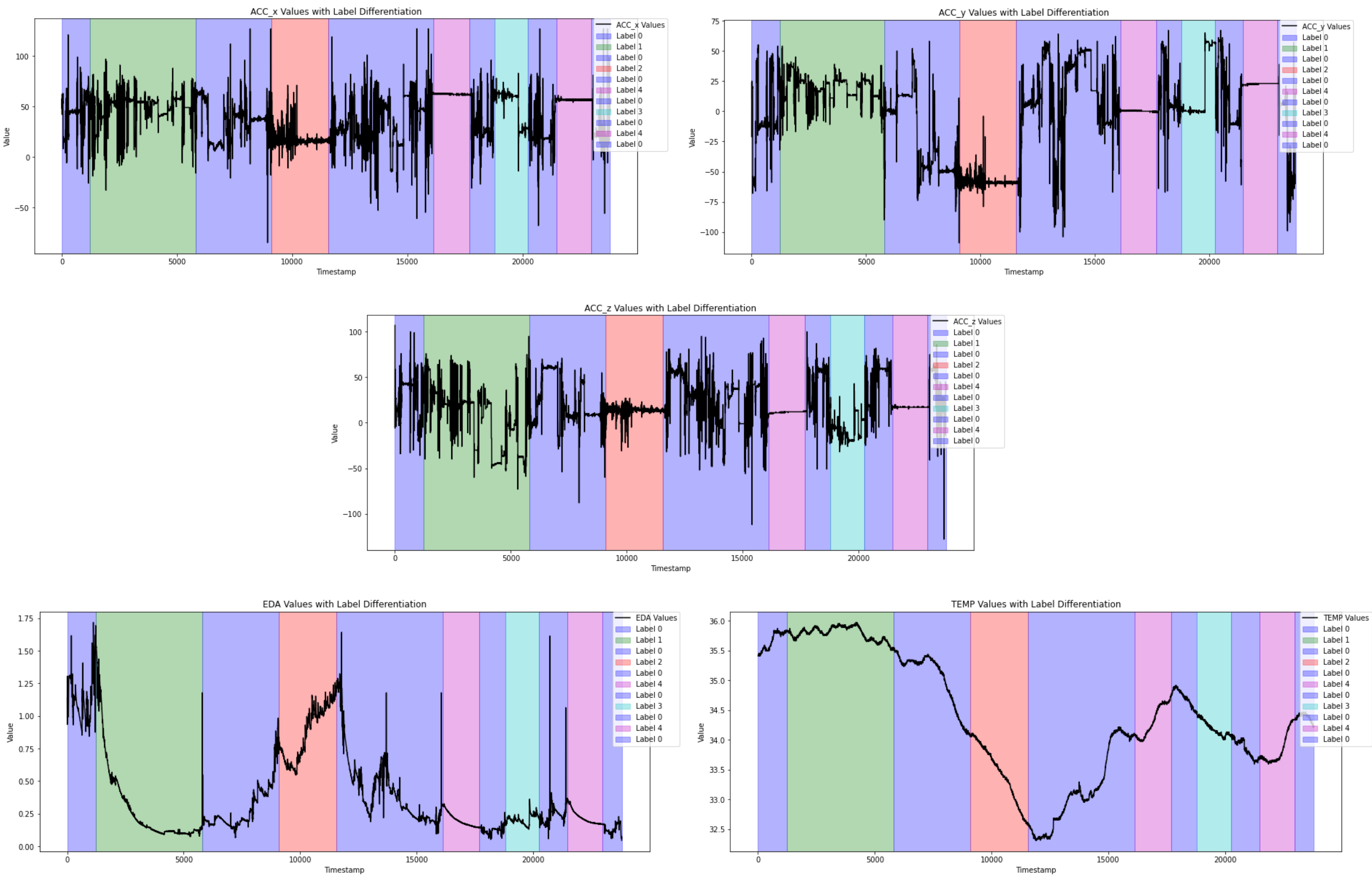


Figure 8: 0 - transition, 1 - baseline, 2 - stress, 3 - amusement, 4 - meditation

6.1.3 Data Preprocessing:

The dataset used in this context was meticulously prepared to ensure the quality and reliability of the machine learning model's predictions. Initially, a thorough examination of the data revealed no missing values ('null values'), which implies that the dataset was complete and did not require imputation, a process often necessary to fill in missing or corrupt data. This completeness of data is critical as it allows for more consistent and accurate modeling without the need to estimate missing information, which could potentially introduce bias or error.

Subsequently, to further refine the dataset, outliers were addressed using the Interquartile Range (IQR) method. Outliers can skew the results of a data analysis and can be indicative of measurement error or distribution tails; thus, removing them improves the robustness of the model. Moreover, feature normalization was conducted across every training feature. Normalization is a technique applied to level the playing field for all features, ensuring that the scale of measurements does not distort the results of the analysis, allowing the model to converge more quickly during training.

Furthermore, the dataset exhibited a class imbalance, which is a common problem in classification tasks where some classes are underrepresented. To counter this, Synthetic Minority Over-sampling Technique (SMOTE) was employed, which generates synthetic samples for the minority class, in this case, the low-count stress classes. Before SMOTE was applied, the class distribution was heavily skewed with 120,256 instances in the non-stress class (Label 0) and only 22,015 in the stress class (Label 1). After applying SMOTE, both classes were balanced, with the stress class now having an equal number of instances (120,256) to the non-stress class. Balancing the dataset in this way is crucial for developing a model that does not bias towards the majority class and can generalize well to new, unseen data.

	ACC_x	ACC_y	ACC_z	EDA	TEMP	label
0	1.074802	0.773372	0.288853	0.226233	1.983414	0
1	0.861713	-1.121982	1.051250	0.221820	1.983414	0
2	0.733860	-0.648144	0.587183	0.208582	1.983414	0
3	0.925640	-0.346610	0.520887	0.169752	1.983414	0
4	0.563389	-0.734296	0.918659	0.153867	1.983414	0

Figure 9: Before Preprocessing

	ACC_x	ACC_y	ACC_z	EDA	TEMP	label
0	62.0	-21.0	107.0	1.138257	35.41	0
1	51.0	16.0	35.0	1.125444	35.41	0
2	53.0	21.0	-6.0	1.011405	35.41	0
3	55.0	17.0	34.0	1.033188	35.41	0
4	48.0	24.0	15.0	0.935807	35.41	0

Figure 10: After Preprocessing

6.2 CLASSIFICATION MODEL:

6.2.1 Random Forest:

Random Forest, a versatile machine learning model, excels in stress detection and classification using sensor data. In this context, various sensors, including accelerometers, EDA sensors, and temperature sensors, gather data. The model extracts key features from this raw sensor data, which could encompass statistical measures and data patterns.
Random Forest's robustness in handling noisy data and adaptability to diverse sensor inputs make it a valuable tool for real-world stress monitoring. It can be deployed for continuous, real-time analysis of sensor data, issuing alerts when elevated stress levels are detected.

6.3 EVALUATION METRICS:

During the evaluation phase, we ensured the reliability of our models by using a 10-fold cross-validation method. This approach allowed us to thoroughly test our models on different subsets of the data, reducing the risk of biased results.

Additionally, we observed that our models performed similarly on both the training and testing datasets, indicating that they didn't overfit the training data.

This consistency gave us confidence that our models could generalize well to new, unseen data, making them robust and reliable for practical applications.

Stress vs No Stress:

Model	Accuracy	Precision	Recall	F1- Score
Random Forest	0.9987582587	0.9987580080	0.9987582587	0.9987569937
Logistic Regression	0.667974055	0.66876758	0.667974055	0.667622974

Stress vs Amusement vs No Stress:

Model	Accuracy	Precision	Recall	F1- Score
Random Forest	0.998946385	0.998947031	0.9989463851	0.9989464422
Logistic Regression	0.4399113660	0.427891068	0.4399113660	0.4304166034

6.4 CLOUD ARCHITECTURE AND DEPLOYMENT:

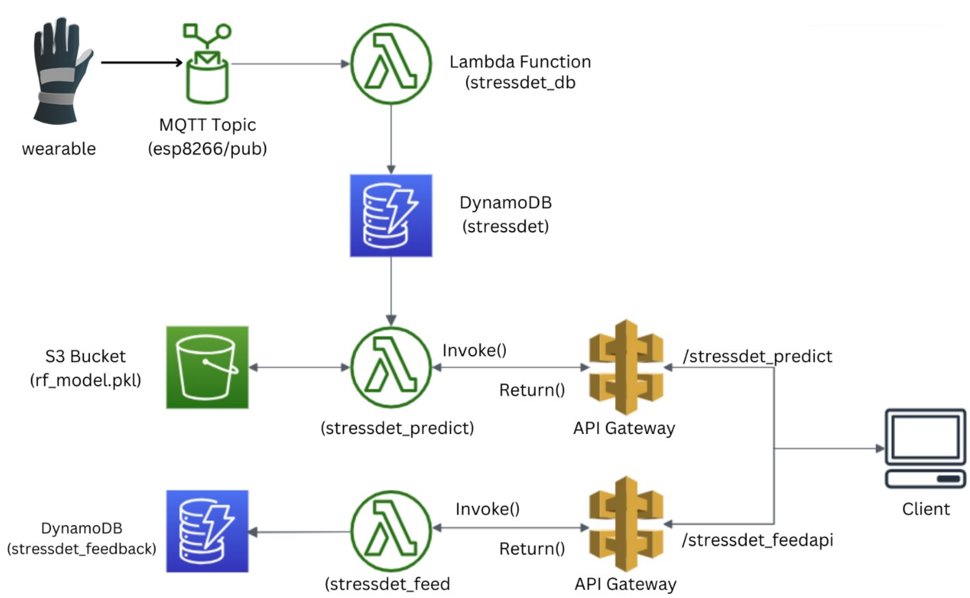


Figure 11: Cloud Deployment and architecture

This architecture represents a cloud-based solution designed to gather data from a wearable device, perform stress detection analysis, and provide feedback mechanisms for the users. The system employs various AWS services, including IoT Core, Lambda, DynamoDB, S3, and API Gateway, to create a seamless flow of data from the device to the end-user application.

6.4.1 Components of the Architecture:

1. Wearable Device:

- The starting point of the data collection process.
- It captures various physiological signals and transmits them to the cloud.

2. MQTT Topic (esp8266/pub):

- Utilizes AWS IoT Core as the MQTT broker.
- The wearable device publishes the collected data to this MQTT topic.

3. Lambda Function (stressdet_db):

- Triggered by the incoming MQTT messages on the topic.
- Processes and stores the received data in DynamoDB (stressdet table).

4. DynamoDB (stressdet):

- A NoSQL database service used to store the data received from the wearable.
- Stores time-series data from the wearable sensors.

5. S3 Bucket (rf_model.pkl):

- Hosts the machine learning model (Random Forest in this case, hence the `.pkl` file extension).
- This model is used to predict stress levels based on the incoming sensor data.

6. Lambda Function (stressdet_predict):

- Invoked by API Gateway or another event.
- Retrieves the machine learning model from the S3 bucket.
- Performs prediction on the data stored in the DynamoDB (stressdet table).
- Sends the prediction results back to the client via the API Gateway.

7. API Gateway (/stressdet_predict):
- Manages the API endpoint for predicting stress levels.
- Exposes a RESTful API that the client application can call to retrieve prediction results.

8. Lambda Function (stressdet_feed):
- Handles the feedback provided by users via the client application.
- Stores feedback in another DynamoDB table (stressdet_feedback).

9. DynamoDB (stressdet_feedback):
- A separate NoSQL database table to store user feedback data.

10. API Gateway (/stressdet_feedapi):
- Exposes the API endpoint for submitting feedback.
- Accepts feedback data from the client application and triggers the Lambda function (stressdet_feed) to process it.

11. Client Application:
- The interface used by end-users.
- It communicates with the API Gateway to submit feedback and retrieve stress predictions.

6.4.2 Data Flow:

- The wearable device captures the sensor data and publishes it to the MQTT topic.
- The Lambda function (stressdet_db) is triggered, processes the incoming data, and stores it in the DynamoDB (stressdet).
- When a prediction is requested, the API Gateway invokes the Lambda function (stressdet_predict), which retrieves the ML model from S3 and applies it to the data in DynamoDB to generate a stress prediction.
- The prediction result is sent back to the client application through the API Gateway.
- The client application allows users to submit feedback, which is routed through the API Gateway to the Lambda function (stressdet_feed), and stored in DynamoDB (stressdet_feedback) for further analysis or model improvement.

6.4.3 Security and Access Control:

- Each AWS service is configured with appropriate permissions using IAM roles and policies to ensure secure access and operations.
- The data flow between the client and the AWS backend is secured using HTTPS, provided by the API Gateway.

6.4.4 Scalability and Availability:

- The use of serverless components like Lambda and DynamoDB ensures scalability as user demand increases.
- The application is deployed in a highly available configuration within AWS to ensure continuity of service.

This architecture provides a robust, scalable, and secure solution for real-time stress detection using wearable technology and cloud-based machine learning capabilities.

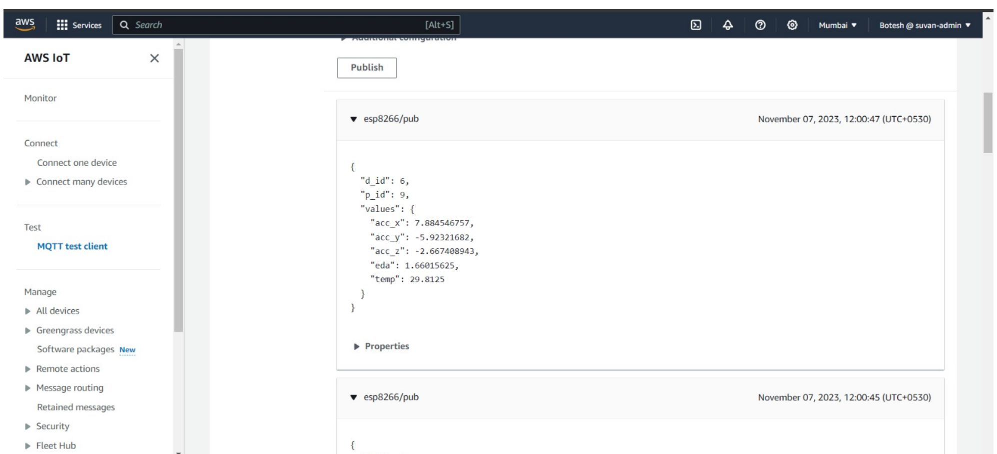


Figure 12: AWS-MQTT Test Client

6.5 WEB APPLICATION:

The website is a Stress Detection Dashboard built using HTML, CSS, and JavaScript. It presents graphical representations of stress-related metrics, aiding users in stress analysis. Users can provide feedback to enhance the dataset for better predictions. The website leverages technologies like Chart.js, Fetch, and a Web API Gateway endpoint. It's hosted on GitHub Pages for convenient access and sharing.

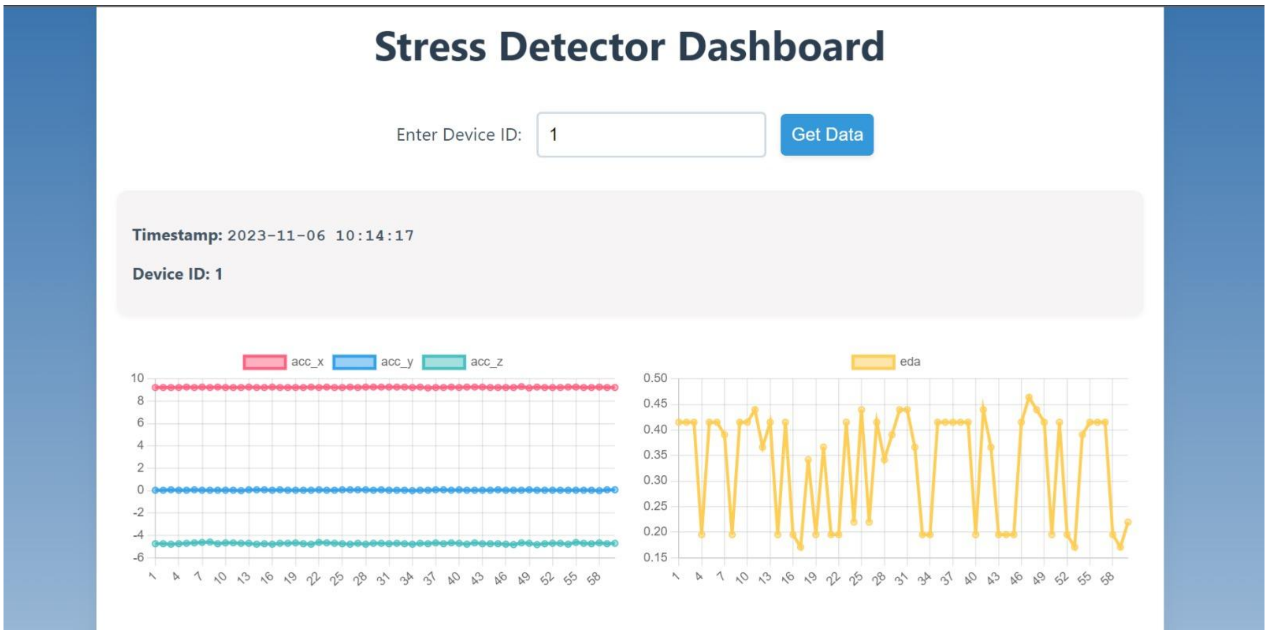


Figure 13: Web Application - ACC and EDA graphs

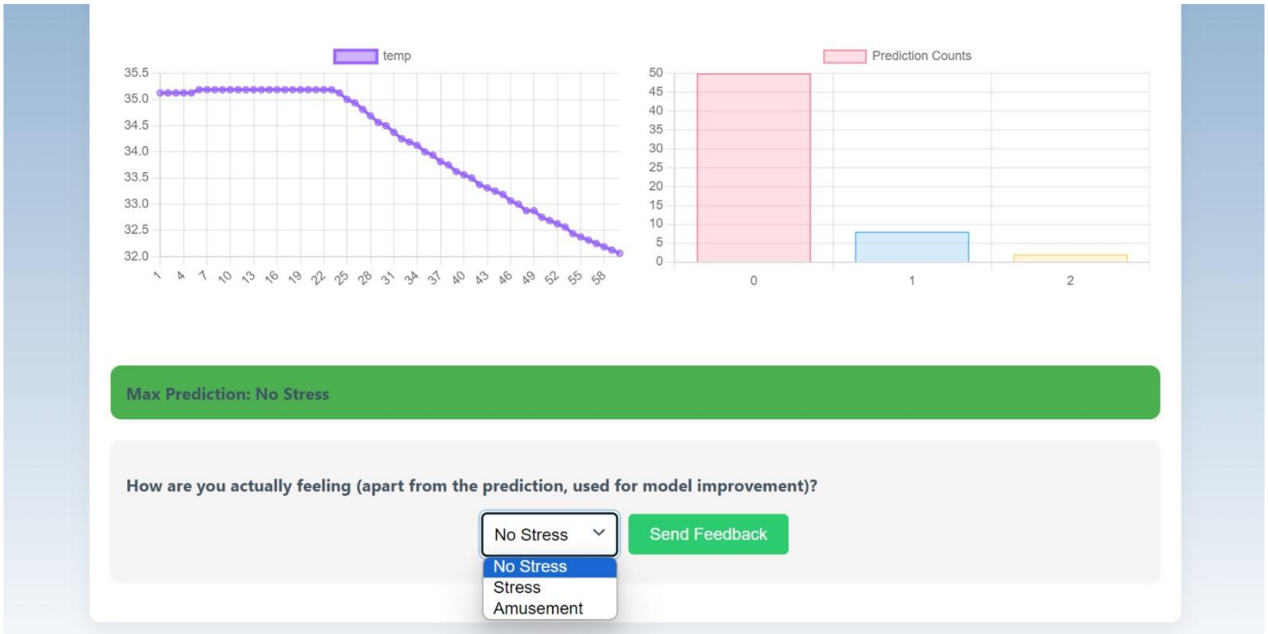


Figure 14: Web Application - Temp and Prediction graphs

6.6 ACTUAL VS PREDICTED:

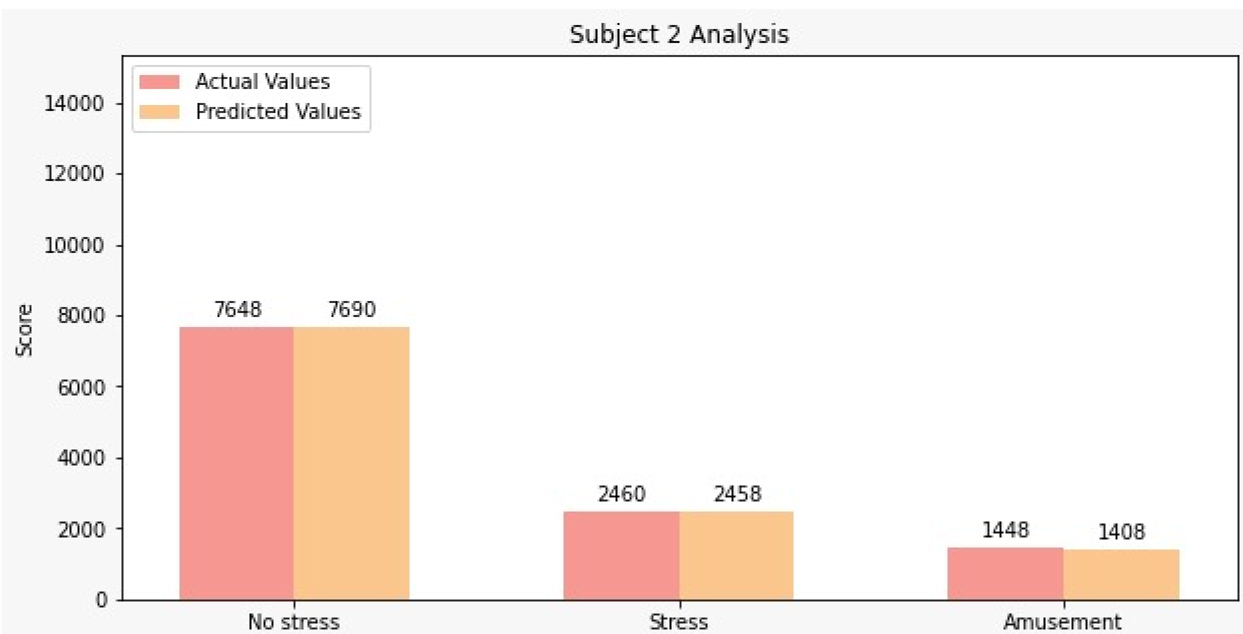
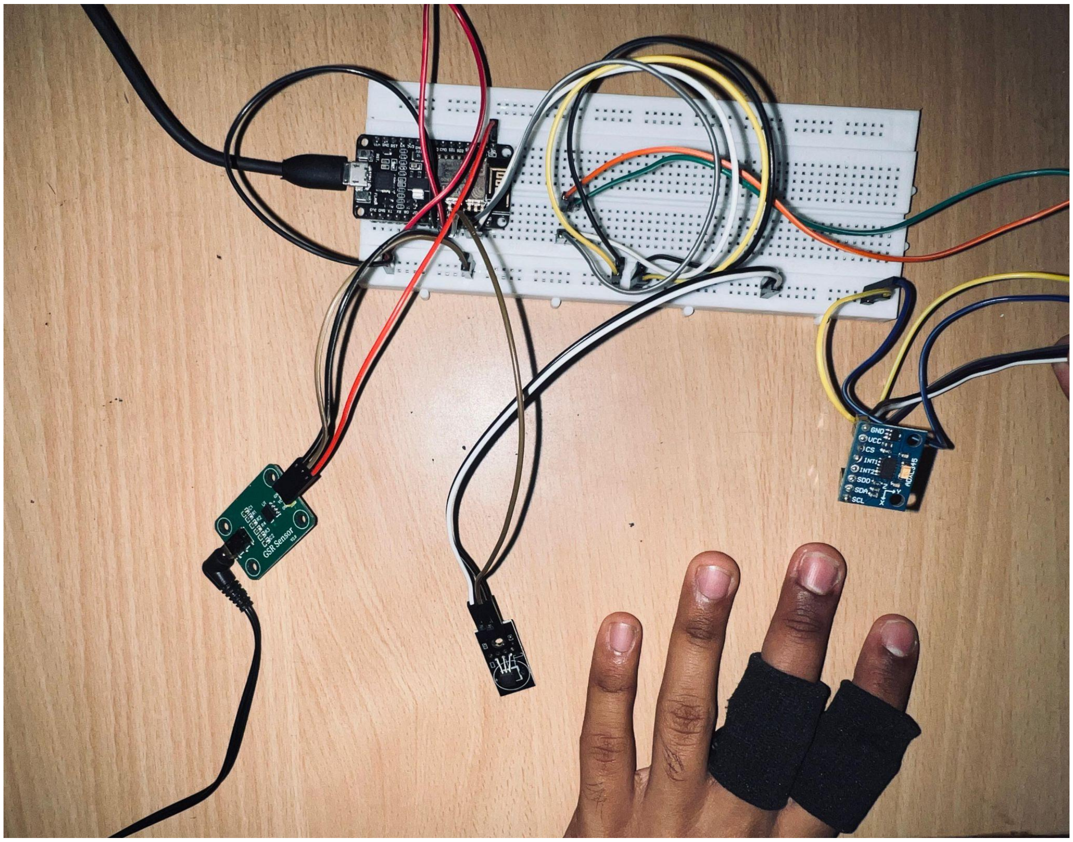


Figure 15: Actual vs Predicted

6.7 PROTOTYPE:



CHAPTER 7
CONCLUSION

Chapter 7 concludes the model built and provides an insight on how the model can be extended for future work.

EXTENDING THE PROJECT AND FUTURE WORK:

The project can be enhanced by broadening the scope and diversity of the collected data to refine the prediction of stress levels. Future endeavors could concentrate on bolstering the model's precision, incorporating a richer array of sensor data, optimizing the underlying algorithms, and delving into novel methodologies for comprehensive multimodal sensor analysis. Moreover, validating the model across a more heterogeneous demographic could confirm its applicability and robustness in various real-world scenarios, ensuring that the tool is versatile and reliable for different populations and stress-inducing conditions.

BIBLIOGRAPHY

1. Gedam, S., & Paul, S. (2021). A review on mental stress detection using wearable sensors and machine learning techniques. IEEE Access, 9, 84045-84066.

2. Sioni, R., & Chittaro, L. (2015). Stress detection using physiological sensors. Computer, 48(10), 26-33.

3. Sandulescu, V., Andrews, S., Ellis, D., Bellotto, N., & Mozos, O. M. (2015). Stress detection using wearable physiological sensors. In Artificial Computation in Biology and Medicine: International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2015, Elche, Spain, June 1-5, 2015, Proceedings, Part I 6 (pp. 526-532). Springer International Publishing.

4. Schmidt, P., Reiss, A., Duerichen, R., Marberger, C., & Van Laerhoven, K. (2018, October). Introducing WESAD, a multimodal dataset for wearable stress and affect detection. In Proceedings of the 20th ACM international conference on multimodal interaction (pp. 400-408).

5. Koldijk, S., Neerincx, M. A., & Kraaij, W. (2016). Detecting work stress in offices by combining unobtrusive sensors. IEEE Transactions on affective computing, 9(2), 227-239.

6. Padmaja, B., Prasad, V. R., & Sunitha, K. V. N. (2018). A machine learning approach for stress detection using a wireless physical activity tracker. International Journal of Machine Learning and Computing, 8(1), 33-38.

7. https://www.who.int/news-room/questions-and-answers/item/stress

8. https://www.mind.org.uk/information-support/types-of-mental-health-problems/stress/what-is-stress

PLAGIARISM REPORT

turnitin

Similarity Report ID: oid:22828:46221988

PAPER NAME	AUTHOR
PW1 final review (1).pdf	**KABILAN 20z223**

WORD COUNT	CHARACTER COUNT
5491 Words	**31588 Characters**
PAGE COUNT	FILE SIZE
36 Pages	**4.7MB**
SUBMISSION DATE	REPORT DATE
Nov 7, 2023 7:41 PM GMT+5:30	**Nov 7, 2023 7:41 PM GMT+5:30**

13% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 10% Internet database
- 4% Publications database
- Crossref database
- Crossref Posted Content database
- 11% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 10 words)