Sleep Stage Detection Using Machine Learning Algorithm

by

Examination Roll: 242113

A project report submitted to the Institute of Information Technology
in partial fulfilment of the requirements for the degree of
Professional Masters in Information Technology

Supervisor: Dr. Rashed Mazumder

Institute of Information Technology
Jahangirnagar University
Savar, Dhaka-1342
2026

DECLARATION

I hereby declare that the work presented in this project is the 'result of inves- tigation' conducted by me under the supervision of Professor Dr. Rashed Mazumder, Professor, Institute of Information Technology (IIT), Jahangirnagar University, Dhaka. I fur- ther declare that no part of this project has been or is being submitted for the award of any degree elsewhere.

Roll:242113

CERTIFICATE

The project titled "Sleep Stage Detection Using Machine Learning Algorithm" submitted by Md. Rakibul Islam Nahid, ID: 242113 , Session: Summer 2024, has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Professional Masters in Information Technology on the 9th of January 2026.

Dr. Rashed Mazumder
Supervisor

BOARD OF EXAMINERS

Dr. M. Shamim Kaiser Professor, IIT, JU	Coordinator PMIT Coordination Committee
Dr. Risala Tasin Khan Professor, IIT, JU	Member, PMIT Coordination Committee & Director, IIT
Dr. Jesmin Akhter Professor, IIT, JU	Member PMIT Coordination Committee
K M Akkas Ali Professor, IIT, JU	Member PMIT Coordination Committee
Dr. Rashed Mazumder Professor, IIT, JU	Member PMIT Coordination Committee

ACKNOWLEDGEMENTS

We feel pleased to have the opportunity of expressing our heartfelt thanks and gratitude to those who all rendered their cooperation in making this report.

This thesis is performed under the supervision of Dr. Rashed Mazumder, professor, Institute of Information Technology (IIT), Jahangirnagar University, Savar, Dhaka. During the work, he has supplied us a number of books, journals, and materials related to the present investigation. Without his help, kind support and generous time spans he has given, we could not perform the project work successfully in due time. First and foremost, we wish to acknowledge our profound and sincere gratitude to him for his guidance, valuable suggestions, encouragement and cordial cooperation.

We express our utmost gratitude to Dr.Risala Tasin Khan, Director,IIT,Jahangirnagar University,Savar,Dhaka, for his valuable advice that have encouraged us to complete the work within the time frame. Moreover, we would also like to thank the other faculty members of IIT who have helped us directly or indirectly by providing their valuable support in completing this work.

We express our gratitude to all other sources from where we have found help. We are indebted to those who have helped us directly or indirectly in completing this work.

Last but not least, we would like to thank all the staff of IIT, Jahangirnagar University and our friends who have helped us by giving their encouragement and cooperation throughout the work.

ABSTRACT

Classifying sleep states is one of the most important ways to figure out what's wrong with people who have problems sleeping. Since electroencephalography (EEG) can reliably detect neurological changes that take place during sleep, It is an effective method for studying the correlation between different stages of sleep and brain activity. Multiple brain and body systems are altered and play crucial roles throughout the NREM and REM phases of sleep, respectively. This study's overarching goal is to design a dependable system for automatically recognizing and classifying different stages of sleep, such as NREM and REM, without the need for human intervention. That will classify NREM and REM sleep phases that may be used to explain the model's findings and provide a description of REM and NREM sleep phases based on EEG sleep data. Classifying different types of sleep with the use of ensemble classification models like LSTM was the focus of this study. Overall, we got accuracy 84%.

Keywords: Physiological Signals, Random Forests, Sleep Stage Detection, Electroencephalography (EEG) Healthcare, Electrooculography (EOG), Sleep Disorders, Electromyography (EMG), LSTM.

LIST OF ABBREVIATIONS

ML	Machine Learning
Dl	Deep Learning
SVM	Support Vector Ma- chines

LIST OF FIGURES

Figure

TABLE OF CONTENTS

CHAPTER I

Introduction

1.1 Overview

Sleep stage detection is the process of analyzing and classifying the different stages of sleep based on bodily signs or data received during sleep. A typical night of sleep consists of alternating periods of REM sleep and non-REM sleep, among other phases. One of the most basic biological activities, sleep is essential for relieving stress. The brain's basic operations are crucial to the capacities of learning, performance, and physical activity [1]. Research in the field of neuroscience has recently shifted its focus to the study of sleep quality and the diagnosis of sleep disorders. Sleep stage scoring has become the primary method for analyzing sleep. [2]. In order to diagnose and treat sleep disorders, it is helpful to know which phases of sleep are the most significant. Researchers use polysomnographic (PSG) data, the continuous monitoring of multiple electrophysiological signals, to assign ratings to different phases of sleep. The most generally used standard for sleep stage categorization is provided by the American Academy of Sleep Medicine (AASM). Non-Rapid Eye Movement (NREM) sleep, Rapid Eye Movement (REM) sleep, and Waking (W) are the three categories into which PSG recordings fall under this standard. The American Academy of Sleep Medicine (AASM) has issued updated guidelines in this area, which may be found at [3]. Each of the five stages of sleep has its own unique wave pattern, and the AASM guidelines outline those waves in detail

1.2 Problem Statement

Understanding sleep patterns, identifying sleep disorders, and developing effective therapies to enhance sleep quality all benefit greatly from the identification and study of distinct sleep phases. Manual grading of polysomnography (PSG) recordings by specialist sleep technologists is a time-consuming and subjective procedure that has been used for sleep stage identification in the past. The goal of this study is to examine whether or not machine learning algorithms can accurately categorize sleep phases. In this article, we focus on three physiological signals—the electroencephalogram (EEG), the electrooculogram (EOG), and the electromyogram (EMG)—that might be utilized to construct a model that accurately classifies sleep stages. By employing machine learning techniques to automate the sleep stage classification process, we want to increase both the quality and efficiency of sleep research. In this study, we go into the problem of identifying different stages of sleep, and we detail the challenges involved in doing so using just physiological data. We will talk about the disadvantages of using conventional approaches and the advantages of using machine learning algorithms for this job. We will also examine the current methods and research into sleep stage identification to see where the field stands and where it may be improved

1.3 Motivation

The many phases of sleep provide important information about the quality of sleep, which is crucial to one's health and well-being. Manual sleep stage assessment by human experts is not advised due to its drawbacks as a time-consuming, subjective, and variable process. Using machine learning methods to automate sleep stage identification makes the procedure more objective, consistent, and rapid. Continuous and non-intrusive monitoring of sleep patterns is made possible by automated sleep stage detection, which aids in the detection of anomalies, sleep disorders, and changes in sleep architecture over time. Individuals and medical professionals alike may benefit from a deeper understanding of sleep and the ability to better regulate their sleep as a result. By classifying sleep phases automatically, people may learn more about their own sleep habits and pinpoint the causes of poor sleep. Individuals may enhance their health and well-being by using this data to make educated decisions about their lifestyles, their sleeping conditions, and the quality of their sleep. Using machine learning methods to identify sleep phases contributes to expanding our understanding of how we sleep. Researchers may better understand sleep disorders, sleep physiology, and the influence of sleep on many aspects of health by automating the analytic process and handling enormous datasets and doing longitudinal studies.In conclusion, the need for objective, efficient, and individual sleep analysis is met by sleep stage identification utilizing machine learning algorithms. It may completely change the way we keep track of and learn about sleep, leading to more effective sleep management and higher standards of health for everyone

1.4 Objective

The primary objective is to develop machine learning algorithms that can automatically categorize sleep data into awake, REM sleep, and NREM sleep stages. The purpose is to eliminate subjectivity and save time by switching to an automated system for scoring.The algorithms' primary goal should be to reliably and accurately identify the sleep stages of its users. This precision is required for effective sleep analysis and the diagnosis of sleep problems. Because everyone sleeps differently and has their own unique traits, it's important to create models that can appropriately categorize sleep phases for a broad variety of people. The algorithms should enable individual sleep monitoring by detecting the various stages of sleep during which each person typically finds themselves. The goal is to help people become more self-aware of their sleep habits and routines, as well as their specific sleep requirements, so that they may make more educated choices about how to best meet those needs. Connectivity to Personal Wearables: The goal is to create algorithms that can be included into wearable devices like smartwatches or sleep trackers to provide easy and unobtrusive stage identification in one's slumber. The user experience may be improved and the scope of sleep stage identification can be broadened thanks to this connection, which enables smooth data gathering and real-time monitoring. The goal is to improve confidence and facilitate decision-making by allowing users, such as researchers, physicians, and people, to comprehend the aspects and attributes contributing to sleep stage categorization. The ultimate objective is to create accurate and reliable methods for classifying sleep phases in order to further sleep research and enhance clinical practice. Researchers, physicians, and patients might all benefit from these algorithms' potential to enhance sleep disorder diagnosis and treatment monitoring.

1.5 Research Outline

Rest of the report is structured as follows: In **Chapter II** a literature study on related work is given including explanations for the most important terms used in this thesis-basic concept and architecture of IoT Network Model, Attacks on IoT, Concept and architecture of SDN, different model of IDS, Concept of Firewall has been discussed through this chapter. **Chapter III** introduces system model including system architecture, algorithm and flowchart of working procedure of entire system model. **Chapter IV** explains the details of traffic analysis techniques, Feature Extraction and Selection mechanism and tools for this mechanism and reasoning how these mechanisms work for our model. **Chapter V** discusses about the simulation and model performance, to analysis result of the model it describes the basic mechanism of attack detection like Fuzzification, NSL KDD dataset, FIS, Defuzzification,Simulation and confusion matrix. Lastly in **Chapter VI** future work and conclusion is mentioned.

CHAPTER II

Literature Review

2.1 Overview

A novel and efficient technique that can be implemented in a microcontroller device to identify sleep stages can help physicians diagnose and treat related sleep disorders by improving the accuracy of the developed algorithm using a single channel of EEG signals. Butterworth band-pass filters split the EEG signal into delta, theta, alpha, beta, and gamma subbands [1].

This paper provides a comprehensive review of 36 studies, published between March 2013 and August 2020, which employed DL models to analyze overnight polysomnogram (PSG) recordings for the classification of sleep stages. Our analysis shows that more than half of the studies employed convolutional neural networks (CNNs) on electroencephalography (EEG) recordings for sleep stage classification and achieved high performance. Our study also underscores that CNN models, particularly one-dimensional CNN models, are advantageous in yielding higher accuracies for classification. More importantly, we noticed that EEG alone is not sufficient to achieve robust classification results [2].

This research introduces non-invasive deep learning approaches for sleep stage recognition utilizing cardiac, respiratory, and movement information. Signals are measured over time, making the challenge time-series data categorization. Two deep learning approaches, convolutional neural network and long-short term memory network, are used to tackle the issue. Input data is a time-series sequence of signals separated by 30 seconds, a conventional interval for sleep study. The records utilized are from 23 subjects, separated into two groupings. Data from 18 participants were utilized for training and 5 subjects for testing [3].

Big data analytics in healthcare has emerged due to portable sensor devices, cloud computing, and machine learning algorithms. Portable sensor devices can frequently check bodily conditions like ECG signals. The machine learning system would then give a regular health overview contrasted to a doctor's diagnosis, which can only be made after a hospital visit. This study developed a model to characterize sleep phases using HRV data from electrocardiograms. The sleep phases categorization predicts sleep stages percentage. Sleep phases percentage might reveal sleep quality. Feature selection and hidden node count were done using Extreme Learning Machine (ELM) and Particle Swarm Optimization (PSO). Comparisons using SVM and ELM techniques showed lesser results than ELM with PSO [4].

The research proposes an effective sleep stage categorization method combining machine learning algorithms and EEG data processing across 10 s epochs. EEG signals help classify sleep stages automatically. Band-pass filters divide EEG signals into frequency subbands. Decision Tree, Support Vector Machine, and Random Forest techniques extract and train statistical features using varied testing dataset percentages. Random Forest method yields 97.8% accuracy [5].

Sleep studies are essential to recreate sleep loss symptoms and identify psychiatric causes. Most investigators manually categorize polysomnography into vigilance states, which is time-consuming, needs substantial training, and is subject to inter-scorer variability. Many studies have developed automated vigilance state classifiers based on multiple EEG channels, but we aim to create an open-access classifier that can reliably predict vigilance state from a single rodent cortical electroencephalogram (EEG) to avoid the drawbacks of tethering small animals to computer programs. A domain expert classified 427 hours of constantly observed EEG, EMG, and activity out of 571 hours [6].

EEG data is used to construct a unique automated sleep stage categorization method in this article. The suggested technique uses sleep EEG data and modern mathematical tools including synchronization probability and graph theory metrics. Using k-nearest neighbors, support vector machines, and neural networks, the resulting features are fitted. Their accuracy is used to compare their performance. The support vector machine has the highest accuracy, 89.07%, making it acceptable for sleep stage categorization [7].

We retrieved linear and non-linear qualities from the input signal. Next, the ReliefF weight method was used to reduce the retrieved feature vector to its best features. The chosen characteristics were categorized using support vector machine, K-nearest neighbor, decision tree, and random forest. The suggested technique used AASM sleep grading criteria and dual-channel EEG recordings from the ISRUC-Sleep dataset. Comparison of suggested technique performance to comparable approaches.

This study used 10-fold cross validation [8].

We present an MML-DMS distributed multimodal and multilabel decision-making system. Classifier modules like deep convolutional neural networks (CNNs) and shallow perceptron neural networks are coupled. Each module uses distinct data formats and labels. The MML-DMS modules' information flow determines the sleep stage and problem. We demonstrate that the fused multilabel and multimodal technique leads to better diagnostic results than single-label and single-modality methods [9].

Sleep problem categorization is essential for quality of life. Sleep problems like apnoea may harm health. Experts must classify sleep stages, which is difficult and error-prone. Analyzing, monitoring, and diagnosing sleep disturbances is necessary to construct reliable machine learning algorithms (MLAs) for categorization. Deep learning algorithms and traditional MLAs are compared to classify sleep disorders. This paper provides an optimized sleep disorder classification algorithm and evaluates it using the online Sleep Health and Lifestyle Dataset. Using a genetic algorithm, machine learning algorithm parameters were optimized. Evaluation and comparison of the suggested sleep disorder classification method with leading machine learning techniques [10].

Quantitative analysis of polysomnographic data begins with sleep stage categorization. Sleep stage grading is subjective and time-consuming, requiring professional visual pattern detection. Thus, automated categorization is needed. We created random forest (RF) classification based on features and artificial neural networks (ANNs) using features and raw data for sleep categorization. We tried our approaches on healthy people and ill. Most algorithms produced findings on par with human interrater agreement. Our research found that raw data-based deep neural networks (DNNs) outperformed feature-based techniques. We showed that a priori consideration of sleep's local temporal pattern is crucial. Our findings show that neural network designs can classify sleep [11].

CHAPTER III

Methodology

3.1 Proposed Architecture

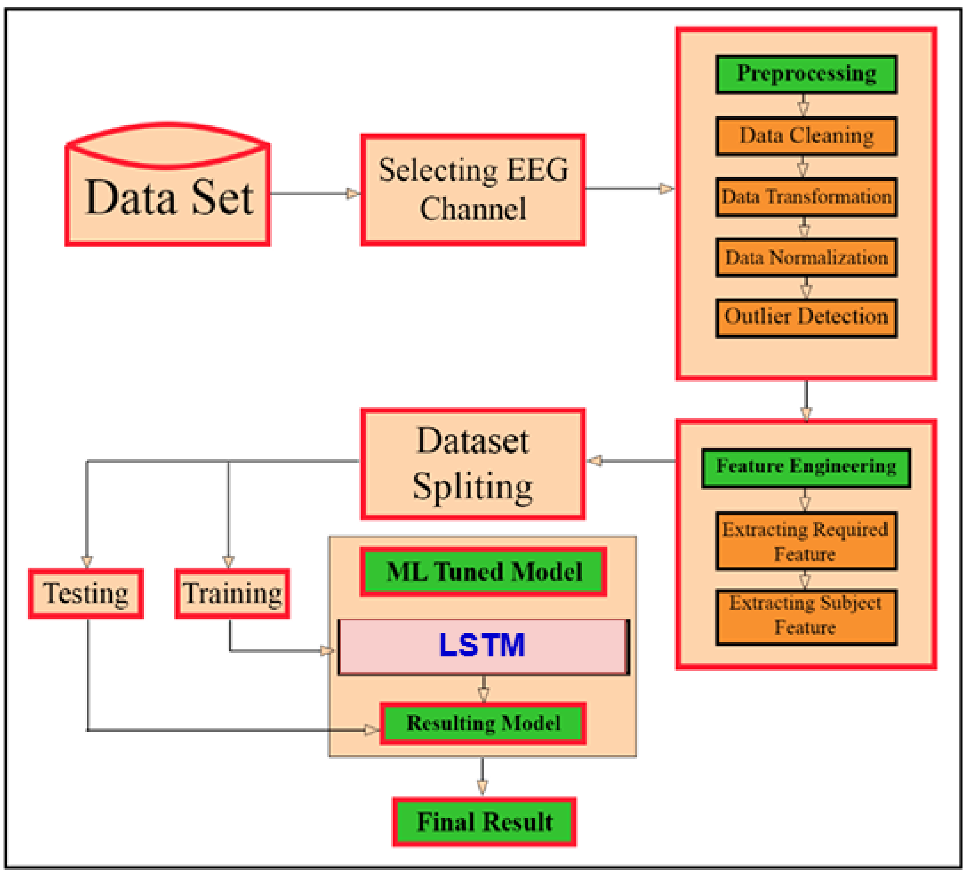

Figure 3.1: Workflow

The diagram represents a workflow for processing EEG data using machine learning. The process starts with the dataset, from which EEG channels are selected for analysis. Preprocessing is the next step, which includes data cleaning, transformation, normalization, and outlier detection to prepare the data. After preprocessing, the dataset is split into two parts: one for training and the other for testing. Feature engineering is performed to extract the required features, including subject-specific features. An ML model is then tuned with the processed features to optimize its performance. The chosen model for analysis is Long Short-Term Memory (LSTM), a type of recurrent neural network. The resulting model is tested to assess its accuracy and performance. Finally, the process concludes with the final result, which indicates the outcome of the model's predictions.

3.2 Data Set:

The dataset is the cornerstone of every machine learning model, particularly in EEG-based research. The acquisition of high-quality data is essential since it influences the model's capacity to learn and provide precise predictions. The dataset must be exhaustive, including the whole spectrum of possibilities the model may face. In EEG signal processing, data often originates from many EEG equipment that may differ in electrode count, sampling rates, and other parameters. A bigger and more varied dataset enhances the model's generalization capabilities. Accurate labeling of the data is essential for the model to correlate patterns with significant results. Upon data collection, it is essential to evaluate its quality to ascertain that it is devoid of mistakes, extraneous information, or noise that may compromise the accuracy of the findings. Consequently, meticulous data management, including appropriate storage, categorization, and annotation, is crucial for developing a dependable model.

3.3 Choosing EEG Channels:

EEG data is often captured via several channels, and the selection of appropriate channels is a crucial step in preprocessing. Each EEG channel relates to distinct regions of the brain, with certain channels being more pertinent for various sorts of cerebral activity or messages. In some instances, a reduced number of channels may be essential to concentrate on the most relevant data, but in other situations, an increased number of channels might be advantageous for capturing intricate signals across many brain areas. The channel selection method is contingent upon the particular purpose of the investigation, such as identifying sleep phases, epilepsy, or other conditions. Appropriate selection guarantees that just the most relevant and high-caliber data is used in model training. Channels that record background noise or extraneous data might result in diminished performance; hence, eliminating superfluous channels is crucial for enhancing model accuracy. The selection of EEG channels may need domain expertise, including knowledge of the brain areas activated during the task in question, or statistical techniques to determine which channels provide the most relevant information.

3.4 Data preprocessing:

Data preprocessing is an essential phase in machine learning workflows that prepares unrefined data for analysis. In EEG signal processing, preparation encompasses many processes, including data cleansing, transformation, normalization, and outlier identification. Data cleaning eliminates noise or erroneous signals that may compromise the final model's performance. Data transformation may use methods like as Fourier transforms to alter the data's domain, so facilitating the extraction of characteristics that are more informative for the model. Normalization guarantees that all data characteristics are uniformly scaled, hence avoiding features with greater scales from overshadowing the learning process. Outlier detection facilitates the identification and elimination of anomalous data items that may not reflect standard circumstances. These preparation techniques improve data quality and make it more appropriate for machine learning models. A more comprehensive preprocessing enhances the model's capacity to discern patterns and provide precise predictions. Preprocessing mitigates the danger of overfitting and guarantees that the model assimilates the most relevant features of the input.

3.5 Dataset Partitioning:

Following data preparation, the subsequent phase involves partitioning the dataset into subsets designated for training and testing purposes. This partition is essential to avoid overfitting, hence guaranteeing the model's ability to generalize well to novel data. The dataset is often divided into two segments: one for training the model and the other for evaluating its performance. The training data constructs the model, while the testing data assesses the model's predictive accuracy on novel, unseen data. Occasionally, further divisions may be implemented to include a validation set, used during training to optimize hyperparameters and mitigate overfitting. A conventional division may allocate 70% for training, 15% for testing, and 15% for validation, however the precise distribution may vary depending on the dataset and project specifications. Appropriate dataset partitioning guarantees that the model does not retain the training data but rather acquires the ability to generalize to novel examples. Cross-validation approaches enhance model resilience by partitioning the dataset into numerous subsets, with each subset used for both training and testing purposes.

3.6 ML Tuned Model:

During this phase, a machine learning model is constructed and optimized for peak performance. Tuning denotes the modification of a model's hyperparameters to enhance its predictive capability. Hyperparameters may include elements such as learning rate, batch size, or the quantity of layers in a neural network. For EEG data, the machine learning model may consist of a neural network, decision tree, or any other technique suitable for time-series or sequential data analysis. Tuning is often accomplished using a blend of human adjustments and automated techniques such as grid search or random search, whereby several hyperparameter combinations are evaluated to identify the optimal configuration. This phase may also include regularization methods to avert the model from overfitting to the training dataset. The objective of this step is to enhance the model's accuracy and efficiency on both the training dataset and previously unexamined testing data. A more finely adjusted model will exhibit greater accuracy and enhanced generalization capability.

3.7 Feature Engineering:

Feature engineering is the process of identifying and choosing the most relevant features from raw data for the machine learning model. In the realm of EEG signals, feature engineering may include methods such as the extraction of statistical attributes (mean, variance, etc.), spectral characteristics (e.g., power spectral density), or temporal features (e.g., time-domain analysis). It may also include domain-specific characteristics, such as certain frequency bands linked to cerebral activity during sleep or jobs. The model's efficacy is significantly contingent upon the quality and pertinence of the characteristics used for training. At this step, superfluous, irrelevant, or strongly correlated characteristics may be eliminated to decrease model complexity and prevent overfitting. Furthermore, information pertinent to certain subjects may be retrieved to customize the model for particular persons or jobs. An expertly designed feature set may markedly improve model performance by supplying more relevant information for the learning process, resulting in superior predictions.

3.8 LSTM (Long Short-Term Memory):

LSTM networks are a kind of recurrent neural networks (RNNs) specifically designed for processing sequential input. LSTMs are especially advantageous for EEG data since they can effectively capture long-range relationships in the signal, making them ideal for tasks such as categorization of temporal or time-series data. LSTMs mitigate the vanishing gradient issue seen by conventional RNNs, enabling them to preserve information over extended durations, which is essential for EEG signals that may include several time steps. LSTM models in EEG analysis may identify temporal patterns of brain activity, facilitating the identification of various states or disorders, including sleep phases and seizure occurrences. LSTMs can capture the temporal dynamics of EEG signals by processing data sequentially. The output from LSTM layers may be transmitted across thick layers or integrated with other models to enhance predictive accuracy. LSTM is a potent instrument for evaluating EEG data, where the timing of events has equal significance to the events themselves.

3.9 Resultant Model:

The resultant model is the output post-training, whereby the machine learning or deep learning model has assimilated knowledge from the training data and calibrated itself for optimum efficacy. This model will be used to predict or categorize novel, unobserved data. The model must now be assessed using the testing data to ascertain its accuracy, precision, recall, and other pertinent metrics. Should the model exhibit satisfactory performance on the testing data, it is deemed prepared for deployment. If not, more modifications may be required, including retraining the model with alternative parameters or supplementary data. The resultant model should optimally generalize to novel data, which is the primary objective of machine learning. In some instances, the resultant model may be included into an application or system that perpetually analyzes fresh data, such as real-time EEG monitoring devices.

3.10 MeanP_Alpha_F4 Spectrogram

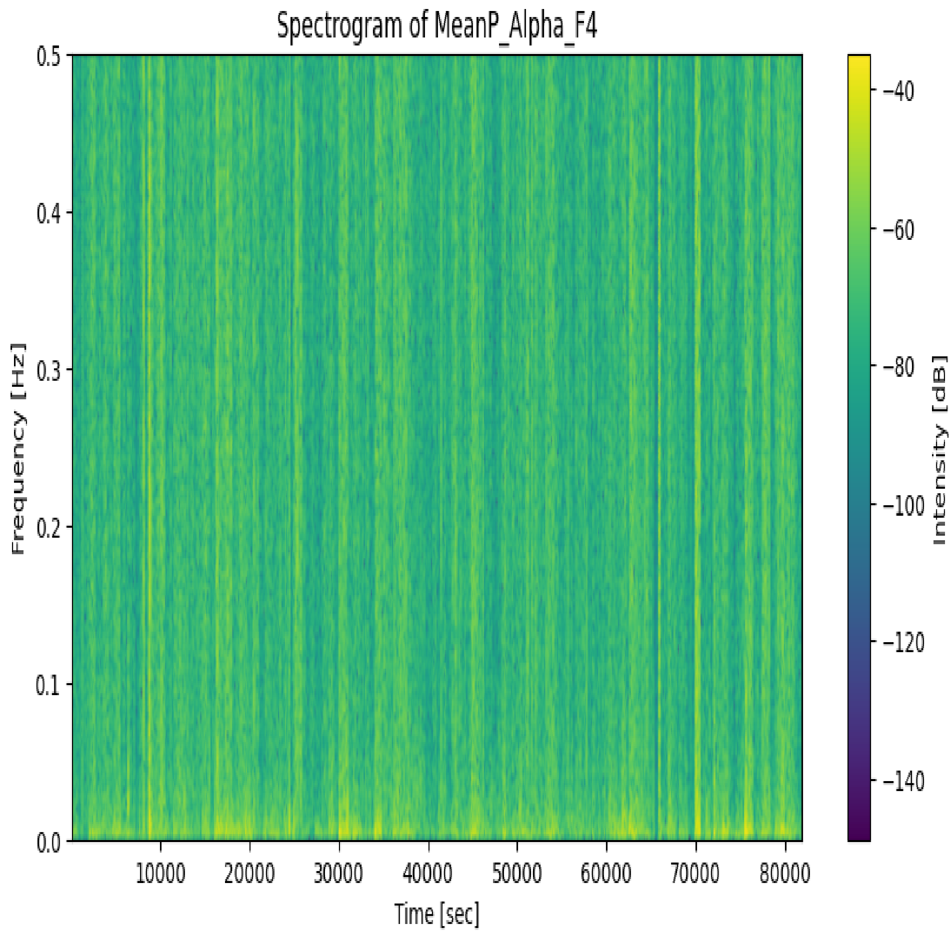

Figure 3.2: MeanP_Alpha_F4 Spectrogram

The spectrogram displays the frequency distribution of the MeanP_Alpha_F4 signal as it evolves over time. Time is shown on the x-axis in seconds, while frequency is shown on the y-axis in Hertz (Hz). The color scale shows the signal strength at each frequency; yellow indicates greater intensity and purple indicates lower intensity. When studying the dynamics of brain activity—whether during cognitive activities or periods of relaxation—the spectrogram is a helpful visual tool since it shows how the frequency content of the Alpha band varies over time.

3.11 Phase Synchronization

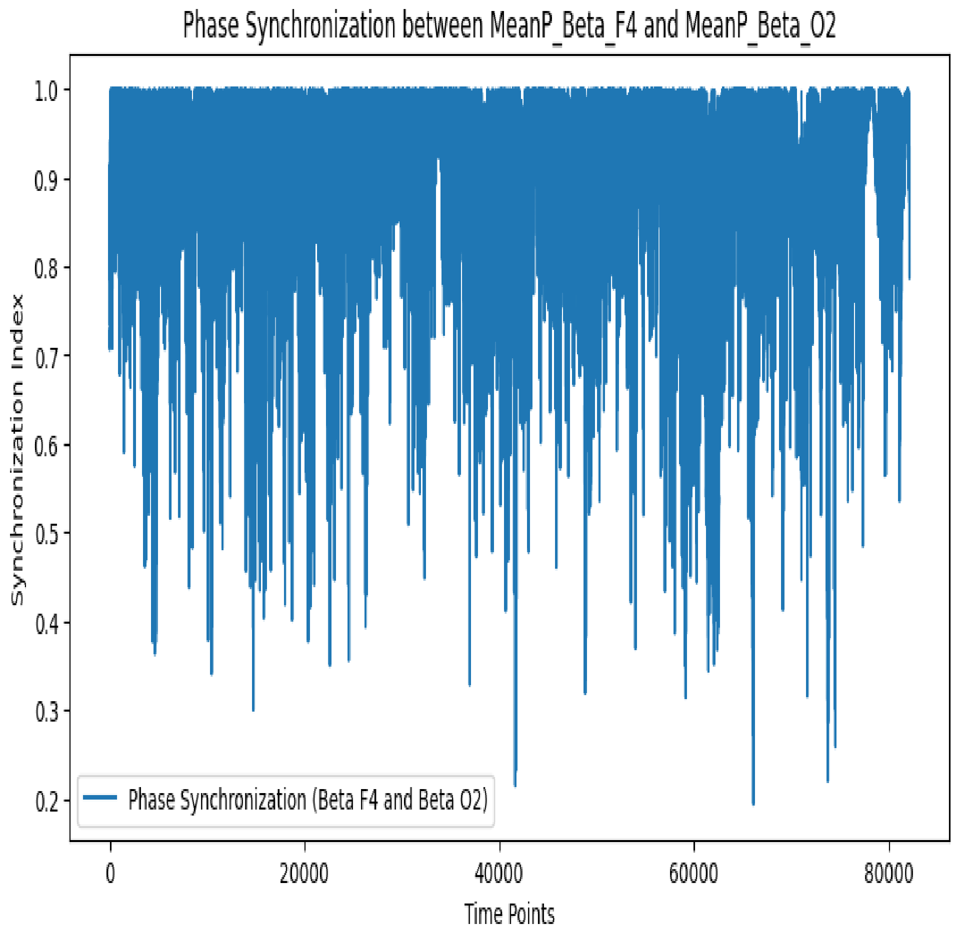

Figure 3.3: Phase Synchronization

This graph shows the beta frequency band phase synchronization between two brain areas, MeanP_Beta_F4 and MeanP_Beta_O2. From 0.2 to 1 is the range of the synchronization index. Points in time are shown on the x-axis, while the synchronization index is shown on the y-axis. The two brain areas that coordinate their activity in the Beta band—which is linked to focused attention and active thinking—are shown below. Throughout the observation time, brain activity fluctuated, as seen by the map, which displays periods of high and low synchronization.

3.12 Coherence between Beta Band

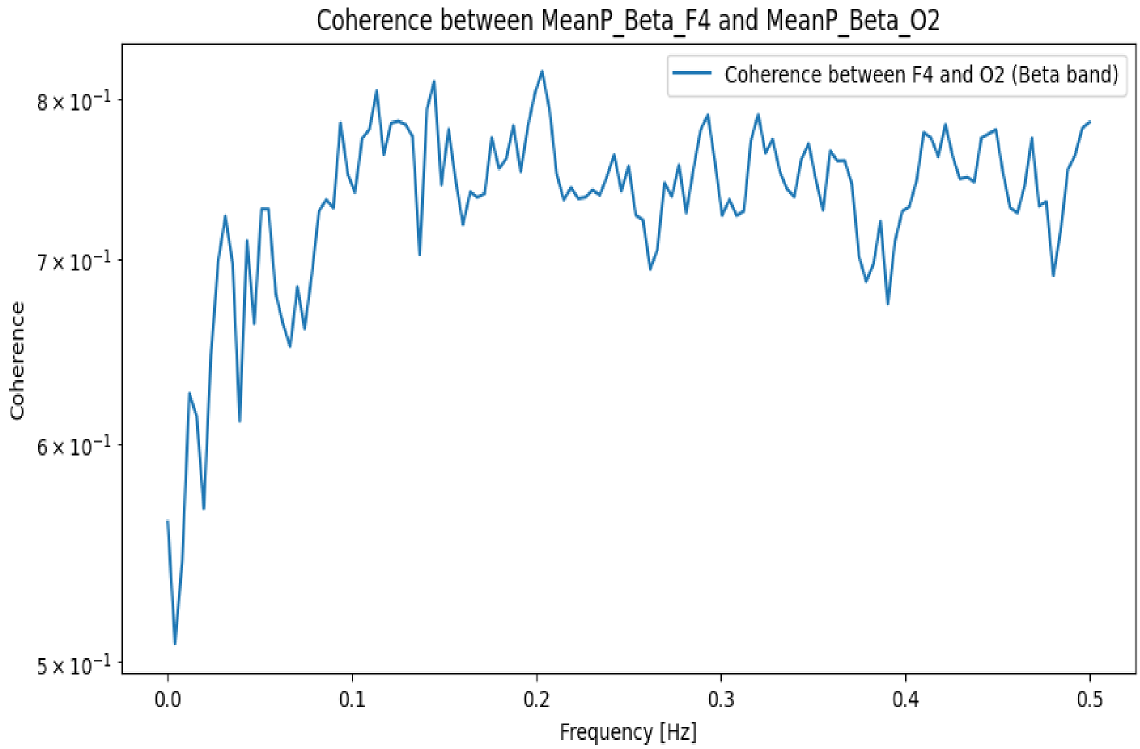

Figure 3.4: Coherence between Beta Band

See how well MeanP_Beta_O2 and MeanP_Beta_F4 work together in the Beta frequency range in this graphic. Coherence, a measure of the degree of correlation between the two signals at various frequencies, is shown on the y-axis, while frequency, in Hertz (Hz), is shown on the x-axis. connection strength is shown by coherence values closer to 1, while values closer to 0 imply weak or no connection at all. Based on the figure, it seems that certain areas in the Beta band are largely coherent, with some frequency-dependent fluctuations.

3.13 Coherence between Alpha Band

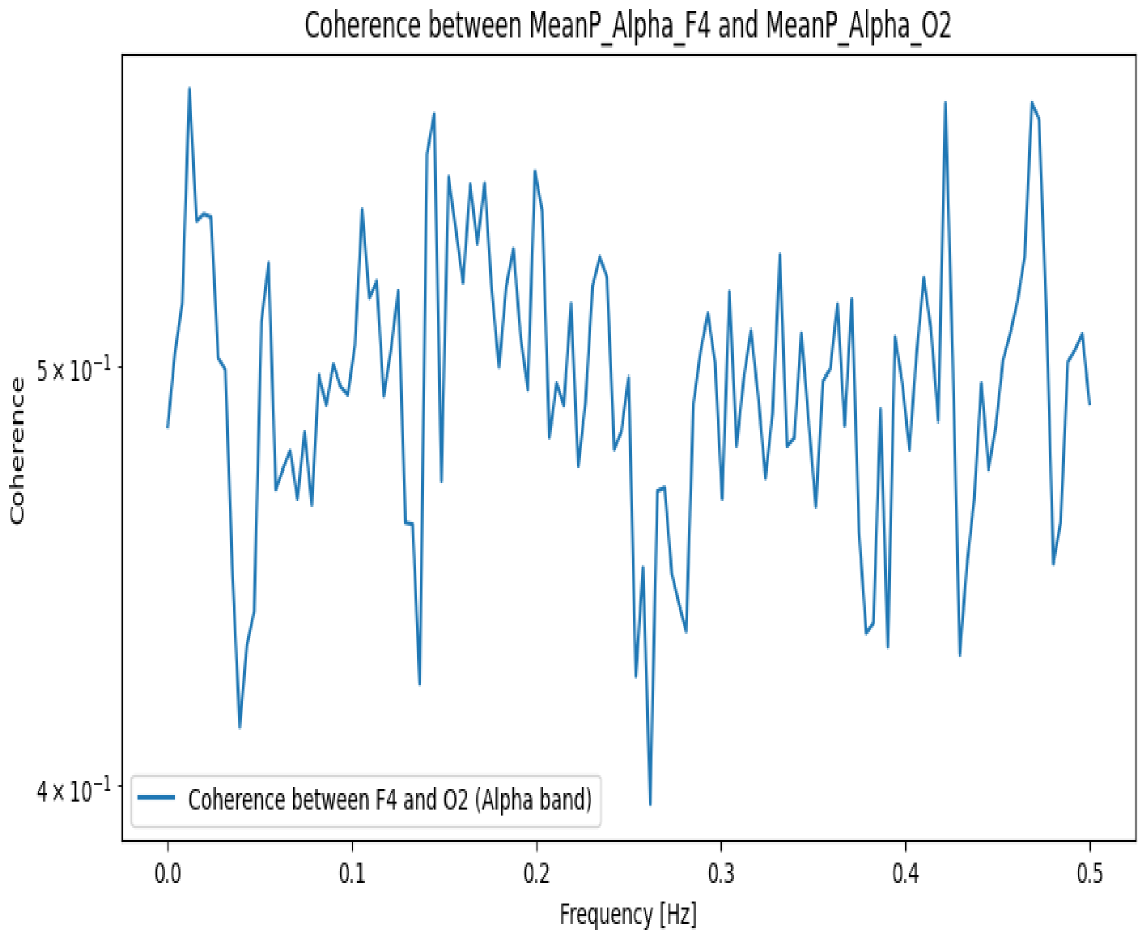

Figure 3.5: Coherence between Alpha Band

Here we see, in the Alpha frequency range, the same kind of coherence as in the prior figure, between MeanP_Beta_F4 and MeanP_Beta_O2. Again, the coherence values here reflect the degree to which certain brain areas are correlated, but at various frequencies within the Alpha band. Coherence is usually lower in the Alpha band than in the Beta band because the former is associated with calm and relaxed states and the latter with active processing, both of which may not have as high inter-regional synchronization.

3.14 The Mean P_Beta_F4 Moving Average

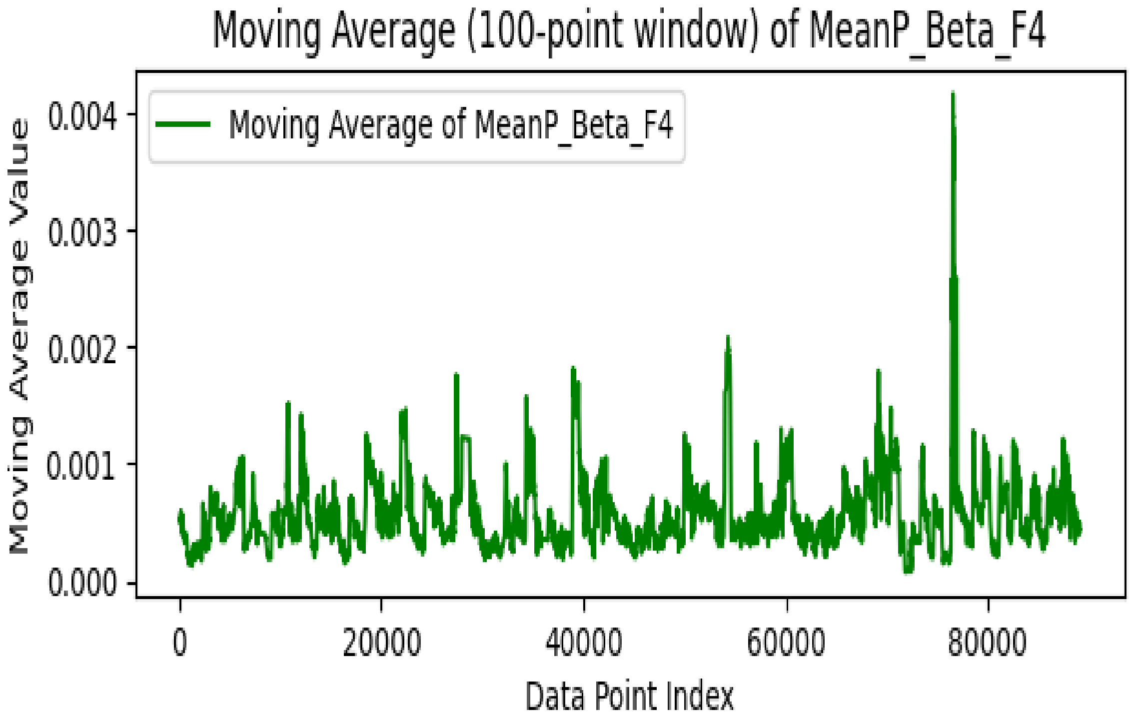

Figure 3.6: The Mean P_Beta_F4 Moving Average

To better see patterns and reduce high-frequency noise, this figure displays the MeanP_Beta_F4 signal's moving average with a 100-point window. Data point indices are shown on the x-axis, while the moving average value is shown on the y-axis. By reducing short-term oscillations and emphasizing long-term changes in the data, this moving average helps us evaluate the general direction of Beta activity over time.

3.15 MeanP_Alpha_F4 With a 100-point Window and Moving Average

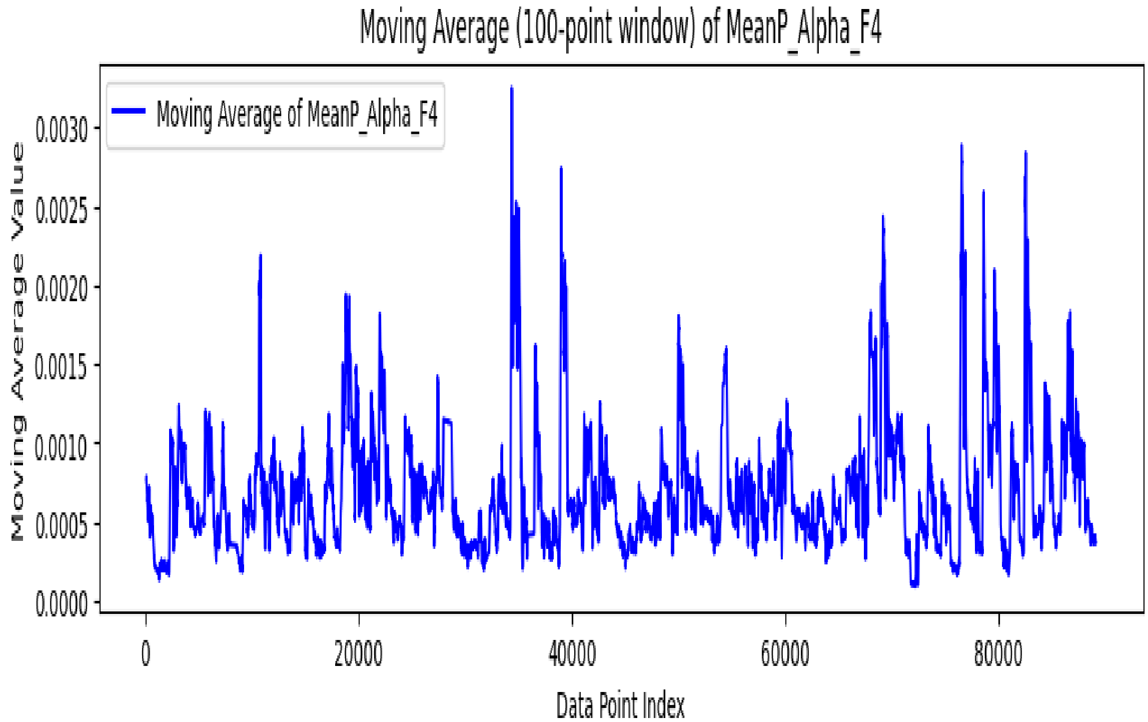

Figure 3.7: MeanP_Alpha_F4 With a 100-point Window and Moving Average

The MeanP_Alpha_F4 signal, which represents brain activity in the Alpha frequency band, is shown by the moving average in this figure, which is otherwise identical to the preceding one. The moving average in this case serves to smooth out volatility, revealing more regular patterns in the data, and the Alpha band is usually linked with relaxation. You can see the long-term changes in Alpha activity and how it changes over time by looking at this graph.

3.16 Summary of LSTM Model

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 75, 64)	16,896
dropout (Dropout)	(None, 75, 64)	0
lstm_1 (LSTM)	(None, 32)	12,416
dropout_1 (Dropout)	(None, 32)	0
dense (Dense)	(None, 1)	33

Figure 3.8: Summary of LSTM Model

You can see the structure of an LSTM model that processes EEG signals in the table. The LSTM in the first layer processes a sequence of 64 units and 75 time steps, as seen by its output form of (None, 75, 64). The 16,896 parameters that make up this layer teach the model its biases and weights. To avoid overfitting, the second layer is a parameterless Dropout layer that randomly sets certain input units to zero during training. Layer three uses an additional LSTM, this time with 32 units and 12,416 parameters. Dropout layers repeat themselves on the fourth layer. As a last step in the prediction or classification process, the 33 parameters are applied to a Dense layer with a single output unit.

CHAPTER IV

Result Analysis

4.1 ROC Curve

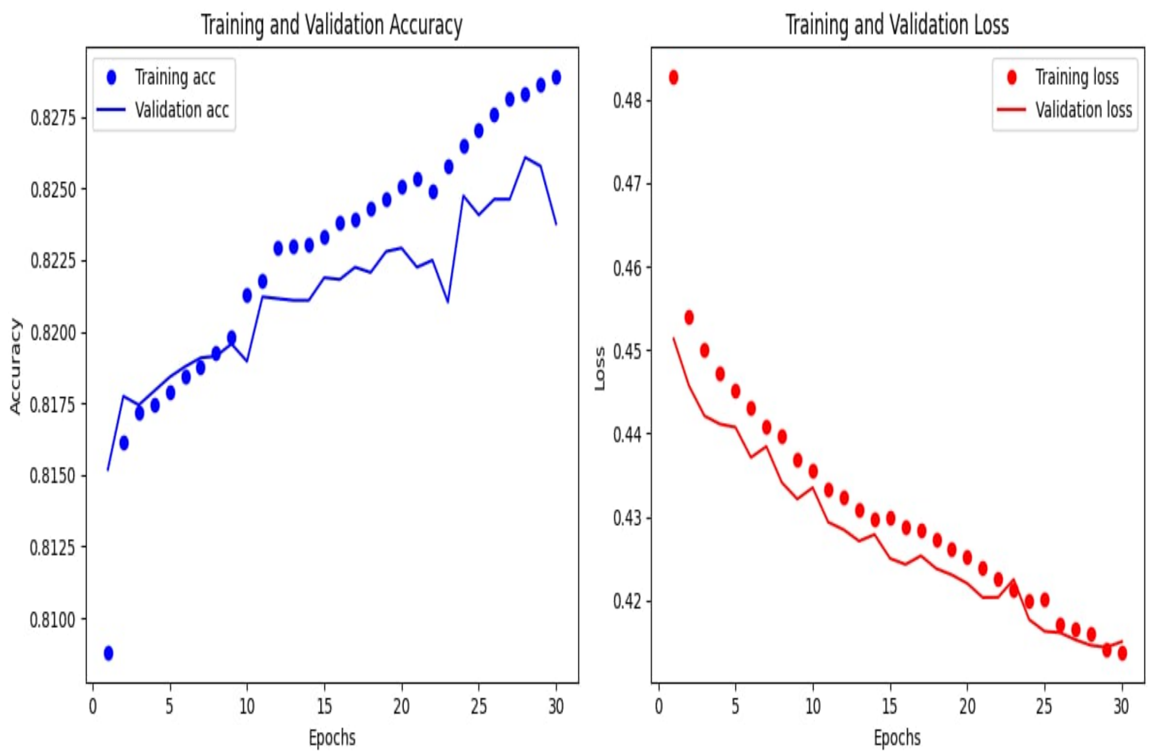

Figure 4.1: ROC Curve

The model's accuracy during training and validation throughout many epochs (training iterations) is shown in the first graph. The accuracy % is shown on the y-axis, while the number of epochs is shown on the x-axis. An indication that the model is learning from the data is the fact that the plot shows a rise in both the training and validation accuracy with time. Consistent gains in the blue training accuracy curve indicate that the model is becoming better at what it does best on the training set. You can see how well the model is doing at certain times in time by looking at the blue dots, which represent the accuracy at each epoch.

This graph shows the training and validation loss as a function of time; it is a measure of the accuracy with which the model predicts future events. The y-axis displays the loss amount, while the x-axis indicates the number of epochs. As time progresses, the loss goes down, which means the model is becoming better at making predictions. As the model begins to learn, the red curve shows the training loss, which first drops sharply before leveling out. The loss levels at each epoch are shown by the red dots on the curve.

4.2 User Interface

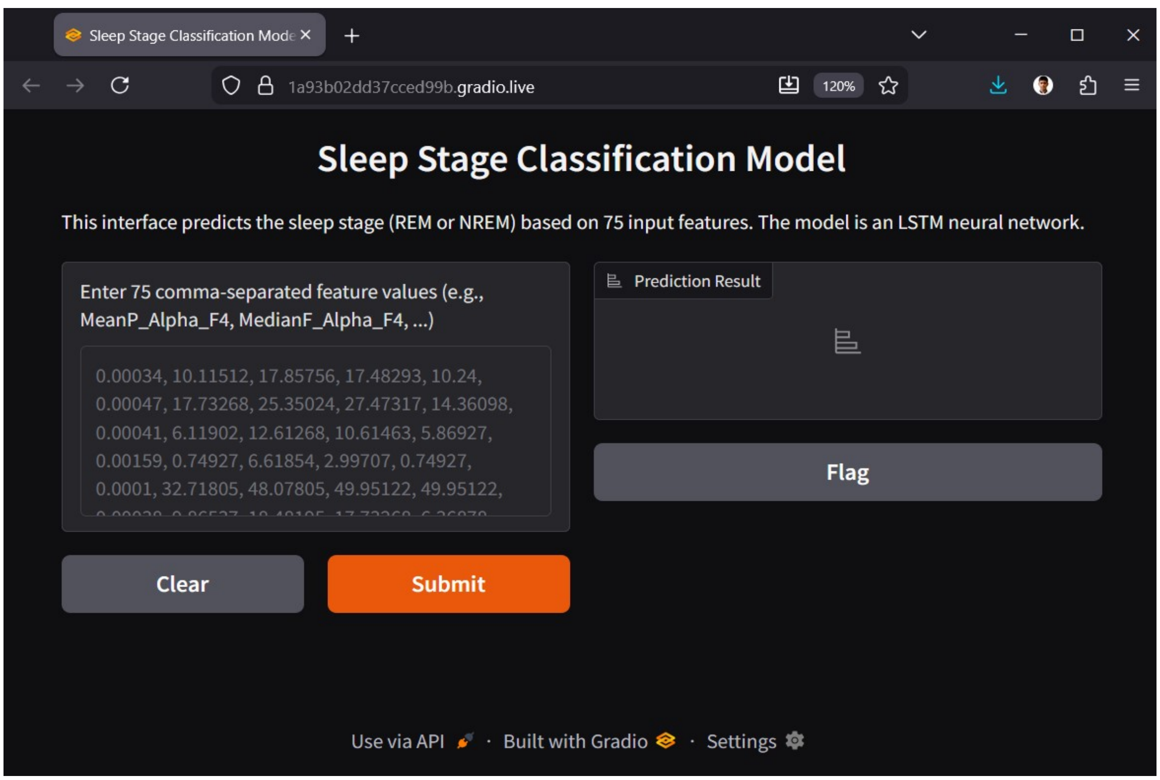

Figure 4.2: User Interface

The picture shows the user interface of a Gradio-built sleep stage classification model. This model uses 75 input variables extracted from EEG data to forecast the phases of sleep, whether rapid eye movement (REM) or non-REM (NREM). numbers for characteristics like MeanP_Alpha_F4, MedianF_Alpha_F4, etc., must be entered by the user in a text field with commas between the numbers. Clicking Submit after data entry activates the prediction; the model, which employs an LSTM neural network, will then categorize the sleep state and provide the outcome. Using the input characteristics, the model can predict the sleep stage and provide a probability, which helps to distinguish whether the sleep state is REM or NREM. If a person thinks the outcome is wrong, they may also click Flag. Users may easily enter their EEG data and get real-time forecasts using the interface.

4.3 REM Prediction

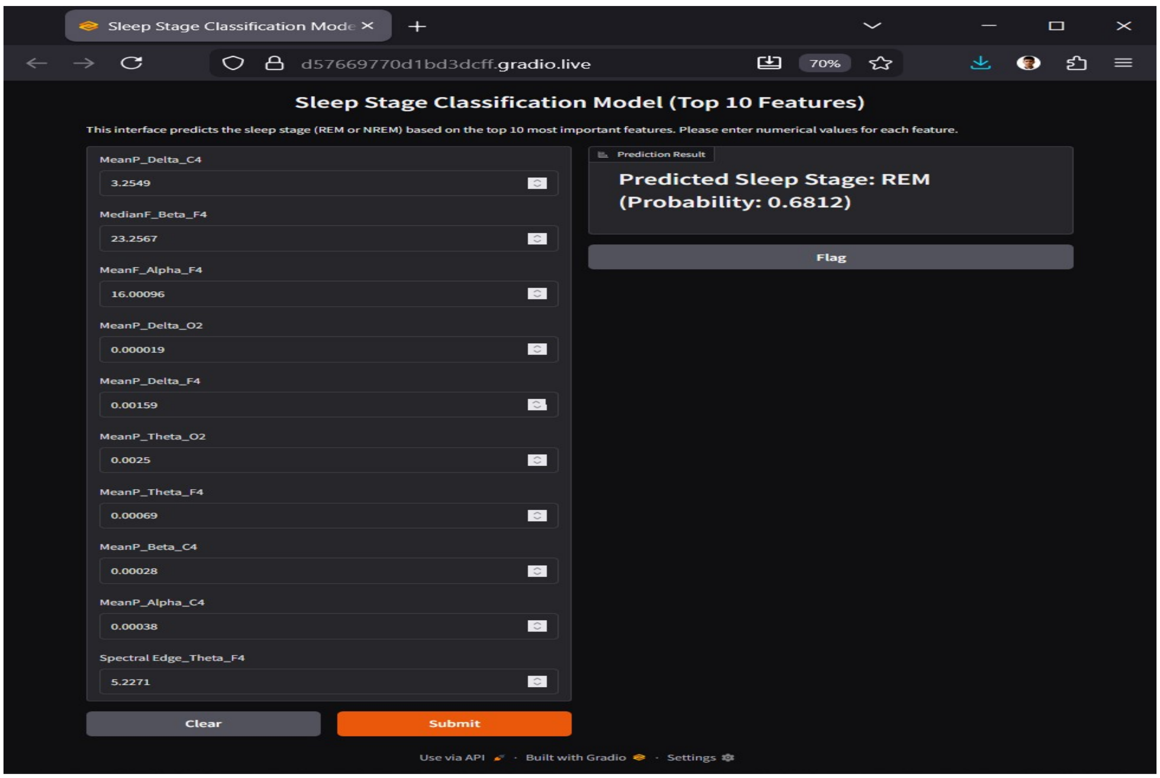

Figure 4.3: REM Prediction

we can see the user interface of the Sleep Stage Classification Model, which allows them to enter the top ten EEG characteristics in order to forecast the phases of sleep (REM or NREM). Here, the model uses the input characteristics to predict REM sleep with a probability of 0.6812. Key characteristics, including MeanP_Delta_C4, MedianF_Beta_F4, MeanP_Alpha_F4, and others, reflect distinct statistical measurements of brain activity in different frequency bands, and the user has input numerical values for these features. After the user inputs the values, they may submit them and the model will provide a forecast using an LSTM neural network. A probability value of 0.6812 for the REM prediction here shows how confident the model is in this categorization. Users may indicate their disapproval of the forecast by clicking the "Flag" button. If the user wants a more efficient and quick forecast, they should only enter the most important features.

4.4 NREM Prediction

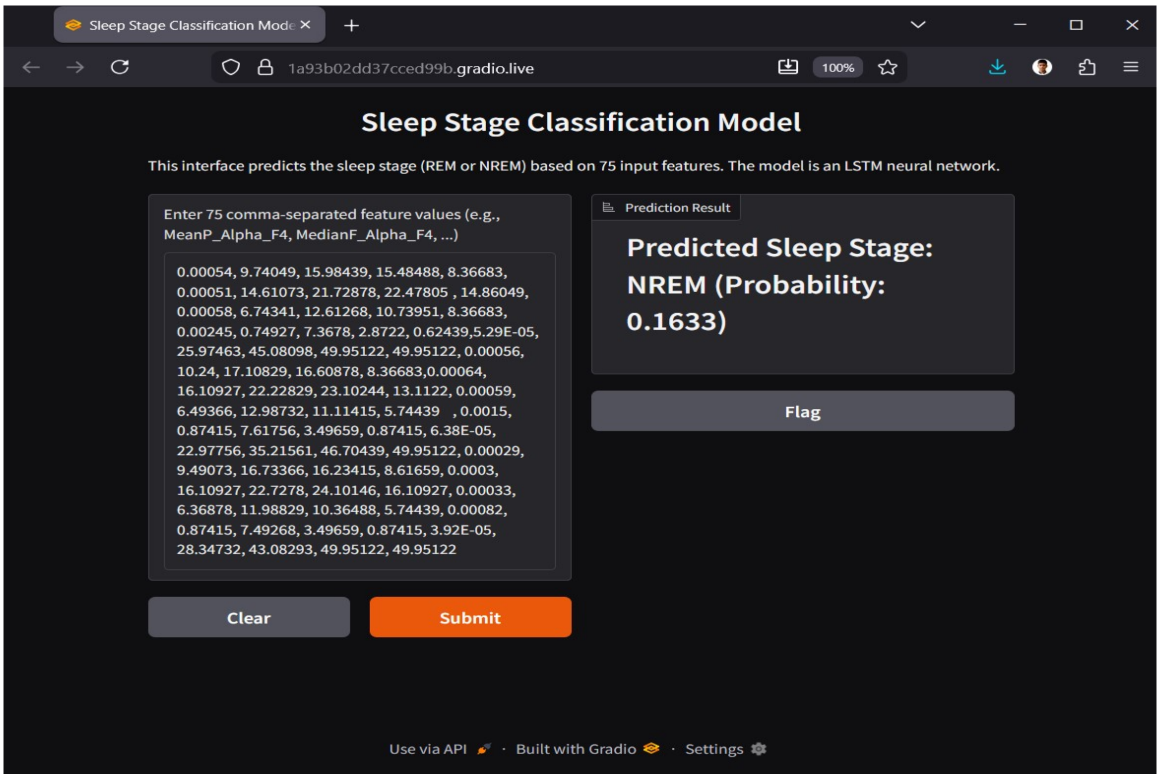

Figure 4.4: NREM Prediction

The user may enter 75 feature values for EEG data, separated by commas, into the Sleep Stage Classification Model interface. These feature values indicate brain activity and include MeanP_Alpha_F4, MedianF_Alpha_F4, and others. The LSTM neural network model then uses the user-provided data to forecast the user's sleep state. For this instance, the model's prediction for NREM (Non-Rapid Eye Movement) sleep is 0.1633. With a lower probability indicating less certainty, the probability shows how confident the model is in the forecast. The user may enter information from their EEG records into the interface, and it will identify their sleep state as REM or

NREM. You may also mark forecasts for correction or more scrutiny by clicking the Flag button.

CHAPTER V

Future Work & Conclusion

5.1 Future Work

1. Gathering additional data from a wider range of people will improve the model's ability to generalize. 2. To enhance the model's precision and performance, we will investigate alternative feature extraction methods and machine learning algorithms. 3. Creating a tool that can be used by both medical professionals and patients to monitor and analyze sleep patterns in real time.

5.2 Conclusion

There are many IDS model for IoT network. But here we have considered a SDN based IoT network model where SDN provides the security of network and Application layer. So here an efficient, simple human perception rule based FIS has been designed to protect the physical layer from passing any kind of malicious network traffic. The FIS has been designed in a centralized manner by using concept of ML to be implemented in SDN Controller. Through this model the four most common attacks of IoT perception layer has been detected which has not been detected in the same way before. Further analysis can be done to make this model more efficient to use in a real network model.

References

[1] K. A. Aboalayon, W. S. Almuhammadi, and M. Faezipour, “A comparison of different machine learning algorithms using single channel eeg signal for classifying human sleep stages,” in *2015 Long Island Systems, Applications and Technology*, pp. 1–6, IEEE, 2015.

[2] H. W. Loh, C. P. Ooi, J. Vicnesh, S. L. Oh, O. Faust, A. Gertych, and U. R. Acharya, “Automated detection of sleep stages using deep learning techniques: A systematic review of the last decade (2010–2020),” *Applied Sciences*, vol. 10, no. 24, p. 8963, 2020.

[3] K. Stuburić, M. Gaiduk, and R. Seepold, “A deep learning approach to detect sleep stages,” *Procedia Computer Science*, vol. 176, pp. 2764–2772, 2020.

[4] S. Santaji and V. Desai, “Analysis of eeg signal to classify sleep stages using machine learning,” *Sleep and Vigilance*, vol. 4, no. 2, pp. 145–152, 2020.

[5] A. Smith, H. Anand, S. Milosavljevic, K. M. Rentschler, A. Pocivavsek, and H. Valafar, “Application of machine learning to sleep stage classification,” in *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 349–354, IEEE, 2021.

[6] P. Chriskos, D. S. Kaitalidou, G. Karakasis, C. Frantzidis, P. T. Gkivogkli, P. Bamidis, and C. Kourtidou-Papadeli, “Automatic sleep stage classification applying machine learning algorithms on eeg recordings,” in *2017 IEEE 30th international symposium on computer-based medical systems (CBMS)*, pp. 435–439, IEEE, 2017.

[7] S. K. Satapathy and D. Loganathan, “A study of human sleep stage classification based on dual channels of eeg signal using machine learning techniques,” *SN Computer Science*, vol. 2, no. 3, p. 157, 2021.

[8] Y.-H. Cheng, M. Lech, and R. H. Wilkinson, "Simultaneous sleep stage and sleep disorder detection from multimodal sensors using deep learning," *Sensors*, vol. 23, no. 7, p. 3468, 2023.

[9] T. S. Alshammari, "Applying machine learning algorithms for the classification of sleep disorders," *IEEE Access*, vol. 12, pp. 36110–36121, 2024.

[10] A. Malafeev, D. Laptev, S. Bauer, X. Omlin, A. Wierzbicka, A. Wichniak, W. Jernajczyk, R. Riener, J. Buhmann, and P. Achermann, "Automatic human sleep stage scoring using deep neural networks," *Frontiers in neuroscience*, vol. 12, p. 781, 2018.

[11] N. Surantha, T. F. Lesmana, and S. M. Isa, "Sleep stage classification using extreme learning machine and particle swarm optimization for healthcare big data," *Journal of Big Data*, vol. 8, no. 1, p. 14, 2021.

Appendix