APPLYING MACHINE LEARNING FOR SLEEP DISORDER CLASSIFICATION

A Project report submitted in the partial fulfillment of the Requirements for the Award of the Degree Of

MASTER OF COMPUTER APPLICATIONS

Submitted By

GHANTA SAI VARSHINI

Y24MC166042

Under the esteemed guidance of

Mr. J.SOMAIAH MCA.,

Assistant Professor

Department of Master of Computer Applications

MEDARAMETLA ANJAMMA MASTAN RAO PG COLLEGE

(Approved by AICTE, New Delhi, Recognized by Govt. of A.P.,
and Affiliated to Acharya Nagarjuna University)

Kesanupalli :: Narasaraopet :: Palnadu (Dt)

2023-2025

MEDARAMETLA ANJAMMA MASTAN RAO PG COLLEGE

(Approved by AICTE, New Delhi, Recognized by Govt. of A.P.,
and Affiliated to Acharya Nagarjuna University)
Kesanupalli :: Narasaraopet :: Palnadu (Dt)

Department of Master of Computer Applications

CERTIFICATE

This is to certify that the Project entitled **"APPLYING MACHINE LEARNING FOR SLEEP DISORDER CLASSIFICATION".** This is being submitted by **GHANTA SAI VARSHINI (Y24MC166042)** in the partial fulfillment for the award of the degree of **"Master of Computer Applications"** is a record Bonafide work carried out by me under the guidance supervision during the academic year 2024-2025 and found worthy of acceptance to requirements of the university.

Internal Guide
Mr. J.SOMAIAH M.C.A.,
Assistant Professor

Head of the Department
Mr. CH. NAVEEN M. Tech.,
Associate Professor

External Examiner

DATAPOINT

Date: 01.05.2025

CERTIFICATE

This is to certify that **Ms. Sai Varshini** bearing the **Roll No. Y24MC166042** from **M.A.M College, Narasaraopet,** has successfully completed the Project titled **"Applying Machine Learning For Sleep Disorder Classification"** as part of his academic curriculum in our organization.

She carried out the project using **Python with Machine learning Domain,** during the Period **11th February, 2025 to 301h April, 2025** under the guidance and supervision of our Development Division at Data-point **1T** & **Hardware Tech Pvt. Ltd., Hyderabad.**

She successfully completed the assigned project within the stipulated time frame. He has been sincere, hardworking, and his conduct during the project period has been commendable.

We wish her all the best in his future endeavors.

Thanking You,

For Data point IT & Hardware Tech Pvt. Ltd.,

MEDARAMETLA ANJAMMA MASTAN RAO PG COLLEGE

(Approved by AICTE, New Delhi, Recognized by Govt. of A.P.,
and Affiliated to Acharya Nagarjuna University)
Kesanupalli :: Narasaraopet :: Palnadu (Dt)

Department of Master of Computer Applications

DECLARATION

I hereby declare that this is dissertation entitled **"APPLYING MACHINE LEARNING FOR SLEEP DISORDER CLASSIFICATION"** being submitted to the Department of Computer Science, Approved by AICTE, affiliated to ANU, Guntur for the award of Master of Computer Applications, is a record of Bonafied work done by me and it has not been submitted to any other Institute or University of the award of any other degree or prize.

Place:
Date:

By

Ghanta Sai Varshini

Y24MC166042

ACKNOWLEDGEMENT

I wish to express my thanks to various personalities who are responsible for the completion of the project. I am extremely thankful to our chairman **Sri Mr. M.R SESHAGIRI RAO, M.Sc.Ed., M.A.(Ed)., M.Phill.,** Chairman of MEDARAMATLA ANJAMMA MASTAN RAO PG COLLEGE Kesanupalli, Narasaraopet, who took keen interest onus every effort throughout this course.

I express our deep felt and gratitude to our Director **Mrs. DARUVURI SRAVYA** M.Tech.,(Ph.D)., of MEDARAMATLA ANJAMMA MASTAN RAO PG COLLEGE for helping us providing all necessary facilities, support and encouragement in successful completion of the project.

I am grateful to our principal **Dr. S. RAMA RAO., M.Tech (NITW), Ph.D., ISTE.,** for providing me all the necessary facilities for the completion of this project in specified time.

I express our sincere gratitude to **Mr. CH.NAVEEN M.Tech.,** Head of the Department of Master Computer Applications, for his support and encouragement during the period of the project work.

I express our sincere thanks to my guide **Mr. J.SOMAIAH M.C.A.,** Assistant Professor for his valuable guidance and constant encouragement which enabled us to accomplish my project successfully in the time. His vast Experience, profound knowledge and willingness have been a constant source of inspiration for me throughout this project work.

I extend my sincere thanks to all other teaching and non-teaching staff of the Dept.of Master of Computer Applications for their cooperation and encouragement during the MCA degree. I have no words to acknowledge the worm affection, constant inspiration and encouragement that i received from my family.

Ghanta Sai Varshini

Y24MC166042

INDEX

TITLES **PAGE NO**

ABSTRACT:

Sleep disorder classification is crucial in improving human quality of life. Sleep disorders and apnoea can have a significant influence on human health. Sleep-stage classification by experts in the field is an arduous task and is prone to human error. The development of accurate machine learning algorithms (MLAs) for sleep disorder classification requires analysing, monitoring and diagnosing sleep disorders. This paper compares deep learning algorithms and conventional MLAs to classify sleep disorders. This study proposes an optimised method for the Classification of Sleep Disorders and uses the Sleep Health and Lifestyle Dataset publicly available online to evaluate the proposed model. The optimisations were conducted using a genetic algorithm to tune the parameters of different machine learning algorithms. An evaluation and comparison of the proposed algorithm against state-of-the-art machine learning algorithms to classify sleep disorders. The dataset includes 400 rows and 13 columns with various features representing sleep and daily activities. The k-nearest neighbours, support vector machine, decision tree, random forest and artificial neural network (ANN) deep learning algorithms were assessed. The experimental results reveal significant performance differences between the evaluated algorithms. The proposed algorithms obtained a classification accuracy of 83.19%, 92.04%, 88.50%, 91.15% and 92.92%, respectively. The ANN achieved the highest classification accuracy of 92.92%, and its precision, recall and F1-score values on the testing data were 92.01%, 93.80% and 91.93%, respectively. The ANN algorithm that achieved high accuracy than other tested algorithms.

INTRODUCTION

Sleep is a vital physiological function necessary for physical and mental health. Sleep helps strengthen the body and consolidate the brain and memories. Sleep quality affects cognitive functions, particularly in children and older drivers at increased risk of accidents. Sleep deprivation can affect the human body and cause health problems like heart disease, diabetes and obesity. Physicians, doctors, medical professionals and experts must manually evaluate polysomnography (PSG) records, which can lead to different assessments of sleep stages. Manual classification is prone to human error and is time-consuming for sleep-stage classification [1], [2].

Philips conducts an annual World Sleep Day survey on sleep-related attitudes and behaviours. In 2021, the survey polled more than 13,000 adults in 13 countries. Only 55% of adults were satisfied with their sleep, and the rest were dissatisfied with their sleep quality. They suffered from sleep quality because of such factors as the coronavirus disease 2019 (COVID-19) pandemic, sleep apnoea and insomnia. The statistics revealed that 37% said the pandemic negatively influenced their ability to sleep well. Moreover, 37% of participants reported suffering insomnia, while 29% snore, 22% have a shift-work sleep disorder, and 12% experience sleep apnoea [1], [2].

SYSTEM STUDY

FEASIBILITY STUDY:

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

PROBLEM ANALYSIS

Deep learning based abnormal event detection in pedestrian pathways aims to identify unusual occurrences like non-pedestrian entities or atypical movement patterns in pedestrian areas. This can be challenging due to factors like cluttered scenes, varying lighting conditions, and the diversity of pedestrian behaviors. The problem analysis focuses on these challenges, which include accurately distinguishing between normal and abnormal activities, handling complex video data, and ensuring the system's robustness and reliability.

Problem Analysis:

1. Data Complexity and Variability:

Pedestrian pathways exhibit a high degree of variability in traffic density, lighting conditions, and background clutter, making it difficult to train robust models.

2. Distinguishing Normal vs. Abnormal:

Defining what constitutes "normal" pedestrian behavior can be subjective and context-dependent, making it difficult to train models to distinguish normal from truly abnormal events.

3. Handling Cluttered Scenes:

Pedestrian pathways often have numerous objects, including other pedestrians, vehicles, and static objects, which can interfere with the detection of anomalies.

4. Computational Requirements:

Deep learning models, especially those using object detection and pose estimation, can be computationally intensive, requiring significant processing power and memory.

PROPOSED SYSTEM :

Typically involves a multi-stage approach. Initially, edge devices process the video data, and only potentially abnormal events are sent to a central server for more detailed analysis using deep learning algorithms. The system may use techniques like optical flow analysis, CNNs, or other deep learning models to identify and classify abnormal behaviors like unusual pedestrian movements or the presence of non-pedestrian entities.

Here's a more detailed breakdown of the proposed system:

1. Data Acquisition and Preprocessing:

Video Cameras:

Install CCTV cameras with high resolution and frame rates to capture pedestrian activity in the designated pathway.

Preprocessing:

- **Motion Detection**: Identify moving objects within the video frames.
- **Object Tracking:** Track the movement of identified objects over time.
- **Normalization:** Ensure consistent input data for the deep learning model.

2. Edge Device Analysis:

Initial Analysis:

Edge devices (e.g., smaller computers at camera locations) perform a preliminary analysis of the video data.

Feature Extraction:

Extract relevant features from the video data, such as motion direction, speed, and object trajectories.

Early Warning:

Based on pre-trained models, the edge devices identify potentially abnormal behaviors, such as rapid changes in movement or unusual object presence.

3. Central Server Analysis:

Data Transmission:

Only potentially abnormal events (identified by the edge devices) are transmitted to the central server for detailed analysis.

Deep Learning Models:

Employ deep learning models, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), to classify and analyze the events.

Classification:

The deep learning models classify the event as either "normal" or "abnormal" based on the extracted features.

Advanced Analysis:

Utilize techniques like optical flow analysis to capture subtle changes in pedestrian movement and identify potential anomalies.

4. System Control and Management:

Integration Control Server (ICS):

Coordinate the communication between edge devices and the central server, ensuring efficient data flow and load management.

Real-time Processing:

The system should provide real-time analysis and alerts for immediate action.

Scalability and Flexibility:

The system should be designed to accommodate large-scale CCTV systems and adapt to various environments.

5. Example Deep Learning Models:

- **Mask-RCNN with DenseNet:** Used for object detection and anomaly identification.

- **Autoencoders:** Learn to reconstruct normal behavior and identify anomalies as deviations from the learned representation.
- **Convolutional Neural Networks (CNN's):** Effective for analyzing video data and identifying patterns.
- **Recurrent Neural Networks (RNN's):** Well-suited for analyzing sequential data like video frames and capturing temporal dependencies.

6. Training Data:

- **UCSD Pedestrian Data set:** Used for training and evaluating anomaly detection models.
- **Synthetic Data:** Generated data that simulates normal and abnormal behaviors for model training.
- **Real-world Data:** Collected from existing CCTV systems in pedestrian pathways.

7. System Evaluation:

- **Accuracy:** Measure the system's ability to correctly identify abnormal events.
- **False Positive Rate:** Determine the rate at which normal events are incorrectly flagged as abnormal.
- **Processing Time:** Evaluate the system's ability to provide real-time analysis.

HARDWARE AND SOFTWARE REQUIREMENTS :

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

REQUIREMENT SPECIFICATION :

Functional Requirements

- Graphical User interface with the User.

Software Requirements

For developing the application the following are the Software Requirements.

1. Python
2. Django

Operating Systems supported

1. Windows 10 64 bit OS

Technologies and Languages used to Develop

1. Python

Debugger and Emulator

- Any Browser (Particularly Chrome)

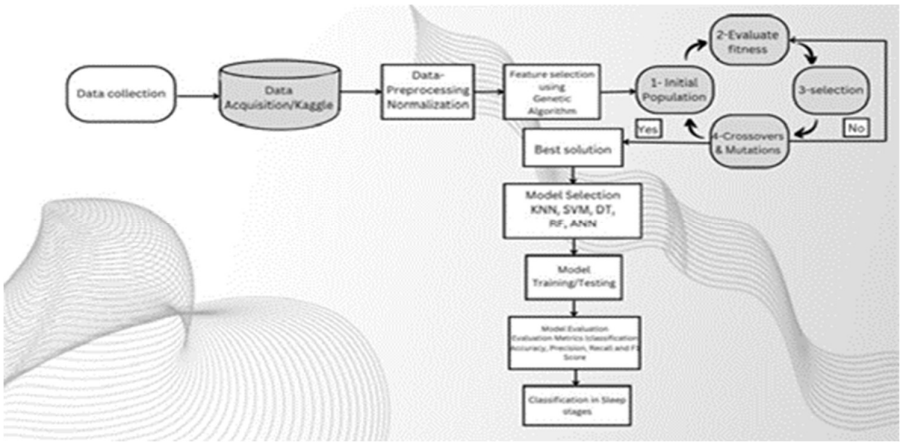
Hardware Requirements

For developing the application the following are the Hardware Requirements:

- Processor: Intel i9
- RAM: 32 GB
- Space on Hard Disk: minimum 1 TB

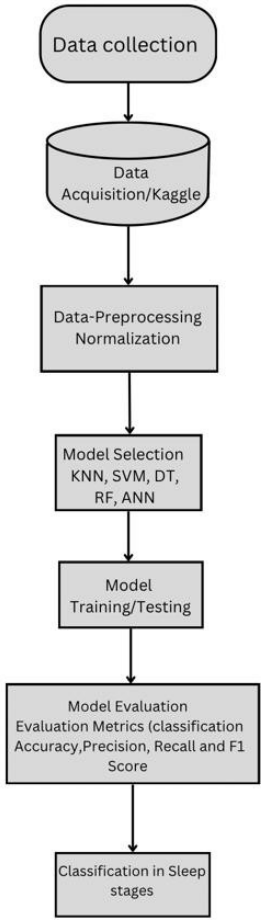
SYSTEM DESIGN

SYSTEM ARCHITECTURE:



DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent as system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.
5. DFD is also known as bubble chart. A DFD may be used to represent as system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail



UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

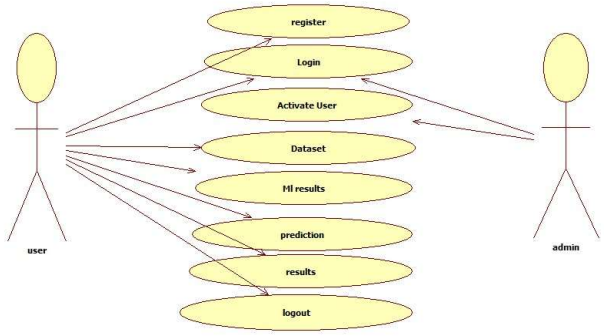
GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

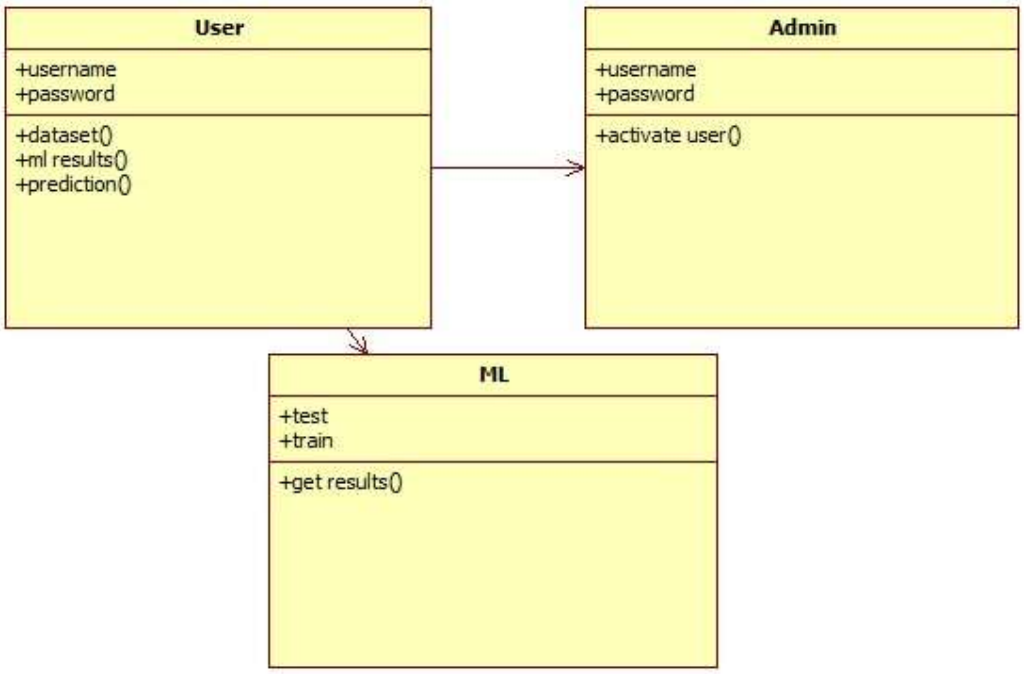
USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



CLASS DIAGRAM:

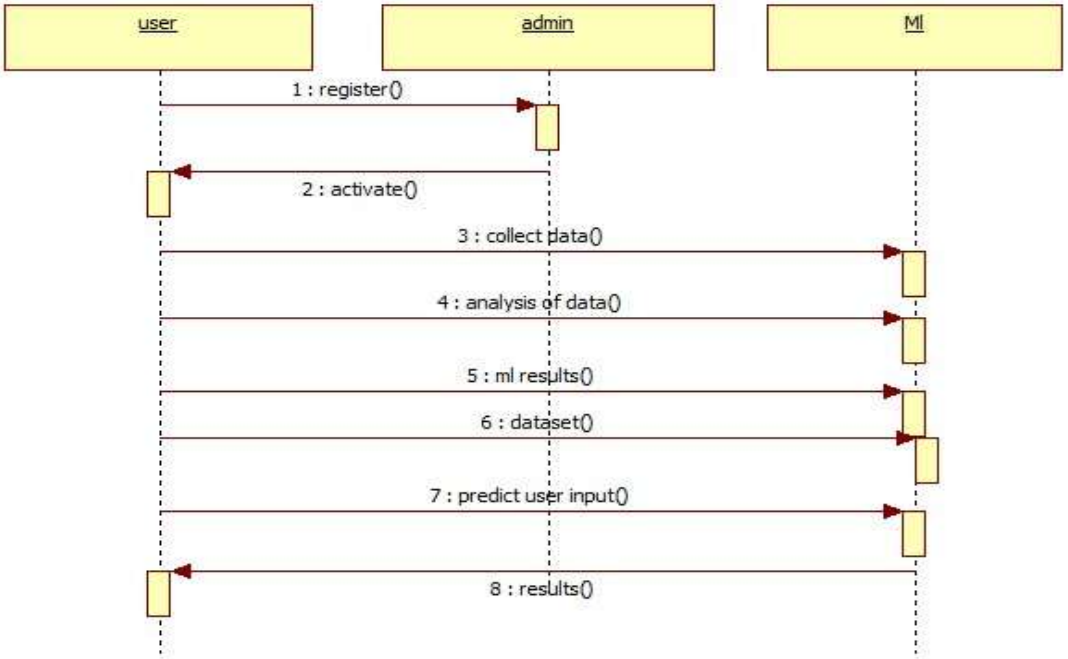
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships



among the classes. It explains which class contains information.

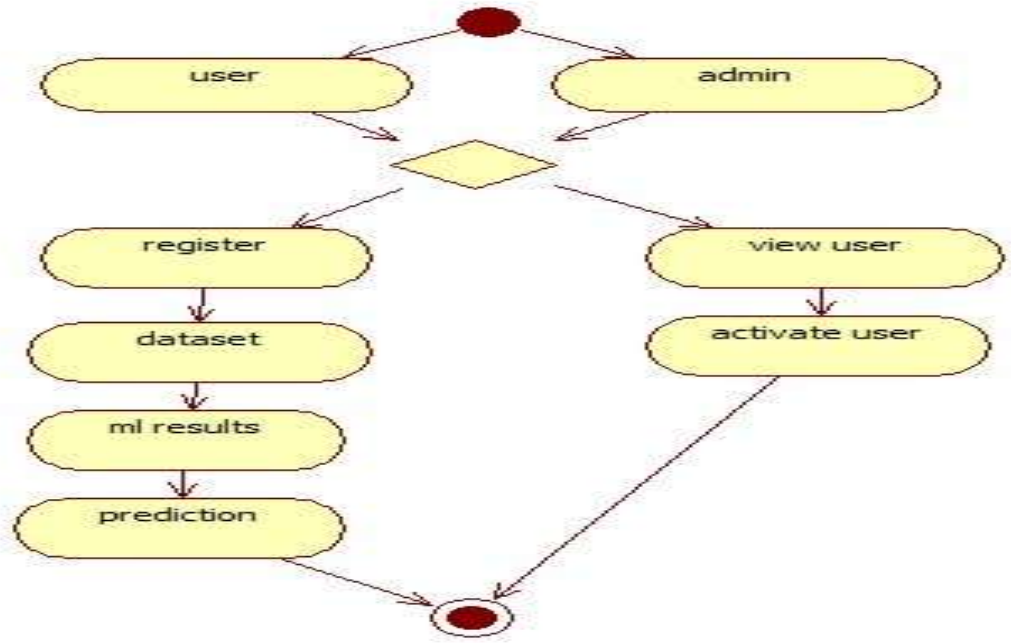
SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



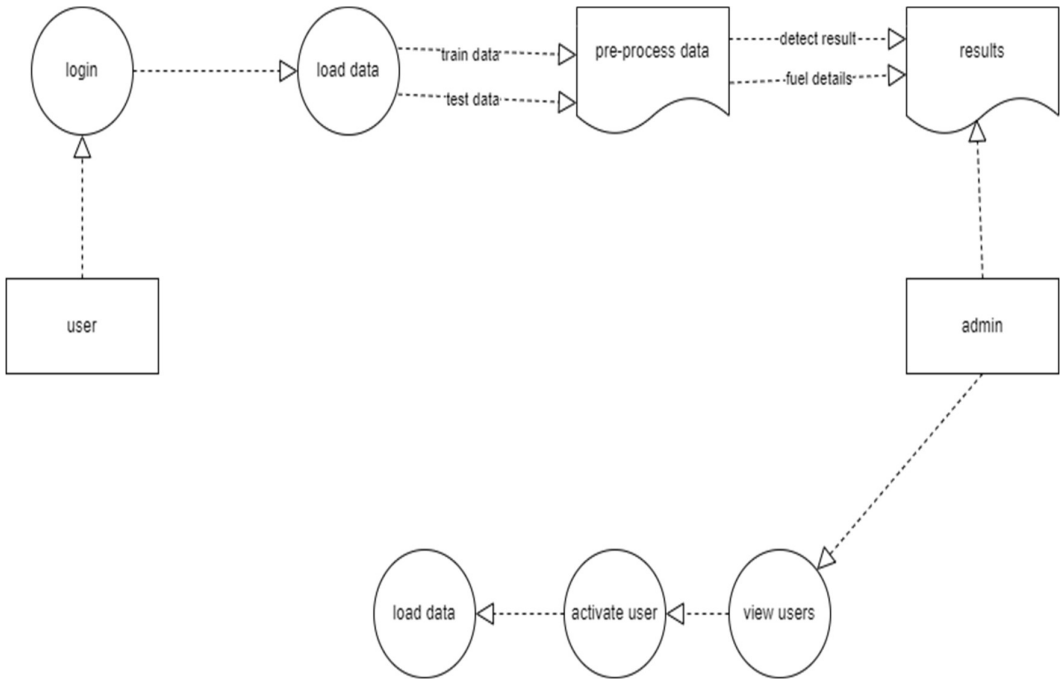
ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



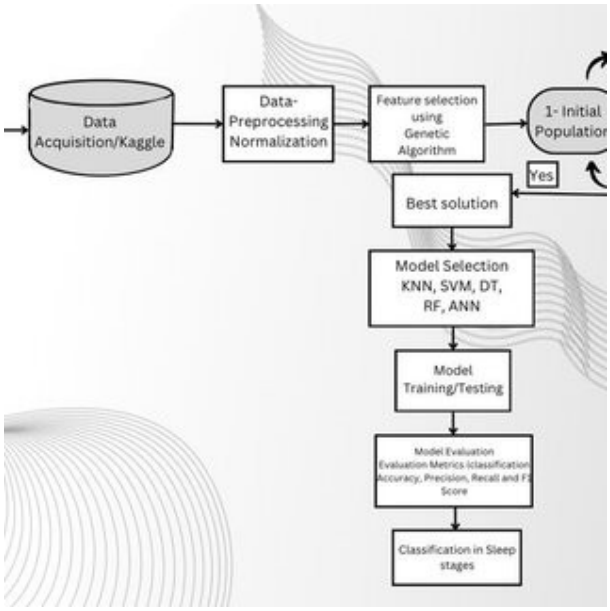
DATA FLOW DIAGRAM

6. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
7. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
8. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
9. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



Context Diagrams :

Context diagrams illustrate the external entities that a system interacts with



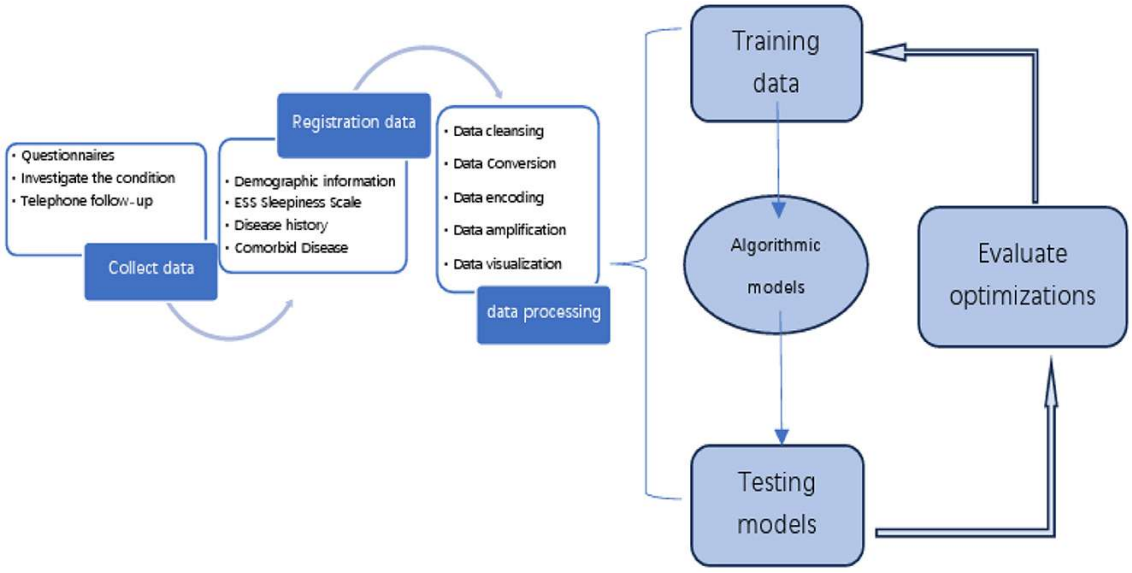
Entities:

- **Patient:** The source of data, including clinical information, biochemical readings, and imaging results.
- **Clinician:** The recipient of the diagnostic insights and results generated by the machine learning model.
- **Database:** Stores all the patient data, including the raw data and the diagnostic results generated by the machine learning model.

- **Flow:**
- The patient provides data to the system.
- The machine learning model processes the data.
- The model generates diagnostic insights and results.
- These results are sent to the clinician.
- Both the data and the results are stored in the database

Control flow diagrams

A control flow diagram for applying machine learning to sleep disorders would outline the steps from data acquisition to model deployment. It would typically include steps for data collection, preprocessing, model selection and training, evaluation, and ultimately, model deployment for prediction or diagnosis.



1. Data Acquisition:

- **Source:** This includes patient data, such as sleep studies (polysomnography), patient questionnaires, and clinical records.
- **Collection:** Gather data from various sources, ensuring data integrity and accuracy.

2. Data Preprocessing:

- **Cleaning:** Handle missing data, outliers, and inconsistencies.
- **Feature Engineering:** Create new features from existing data to improve model performance.
- **Normalization/Standardization:** Prepare data for model input by scaling values.

3. Model Selection:

- **Choice:** Select appropriate machine learning algorithms (e.g., classification, regression) based on the task and data.
- **Considerations:** Evaluate factors like model complexity, accuracy, and interpretability.

4. Model Training:

- **Training Data:** Use a portion of the data to train the chosen model.
- **Hyperparameter Tuning:** Optimize model parameters for better performance.

5. Model Evaluation:

- **Testing Data:** Use a separate portion of the data to evaluate the model's performance.
- **Metrics:** Calculate metrics like accuracy, precision, recall, F1-score, and AUC to assess model effectiveness.

6. Model Deployment:

- **Implementation:** Integrate the model into a system for diagnosis or prediction.
- **Monitoring:** Continuously monitor model performance and retrain it as needed.

7. Use Cases:

- **Sleep Stage Classification:** Automatically identify sleep stages (e.g., REM, NREM).
- **Sleep Apnea Detection:** Identify individuals with sleep apnea based on physiological signals.
- **Predicting Severity:** Predict the severity of sleep disorders based on patient data.
- **Personalized Treatment:** Provide personalized treatment recommendations based on predicted sleep patterns.

INPUT AND OUTPUT DESIGN :

INPUT DESIGN :

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things.

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

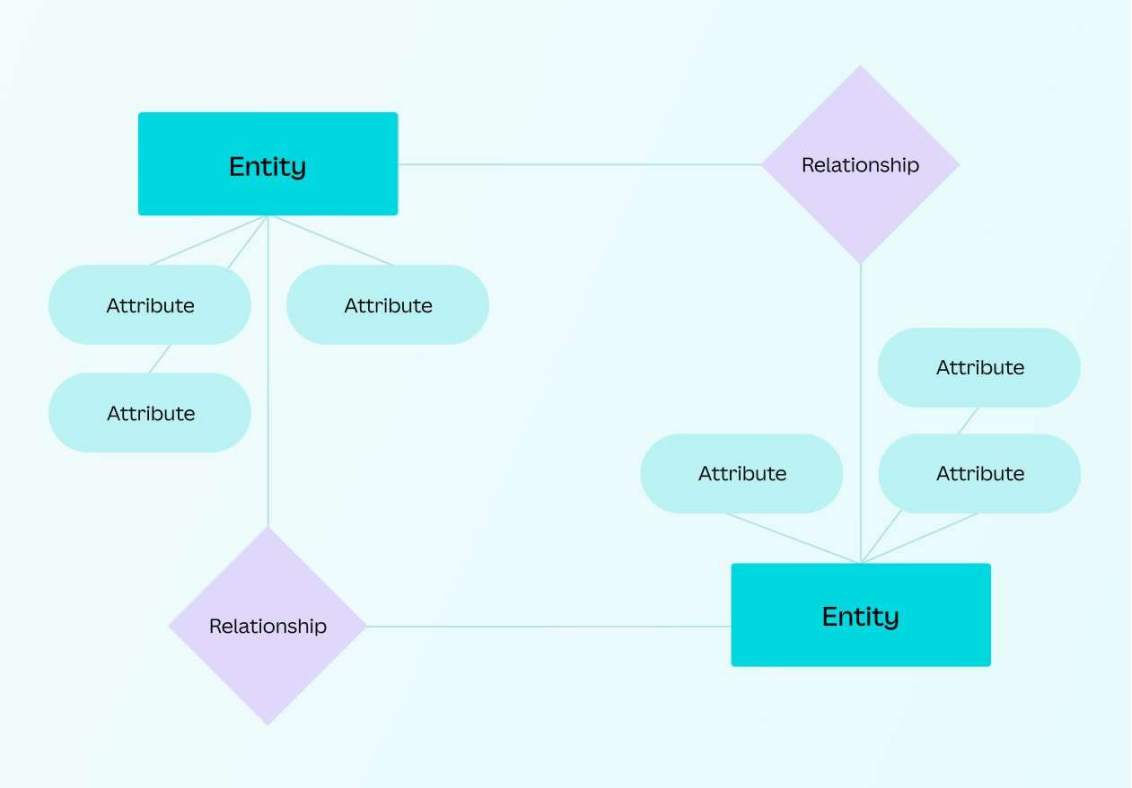
The output form of an information system should accomplish one or more of the

following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

E-R DIAGRAMS

An ER diagram visually represents the entities (data objects) and their relationships in a system. For the Applying Machine Learning for Sleep Disorder Classification, a simple ER diagram might look like this:



Entities:

- **Patient:**

  This entity would store patient demographics (ID, name, age, etc.), and potentially any relevant clinical history.

- **Sleep Data:**

  This entity would contain the actual sleep data, which could be gathered from various sources like polysomnography (PSG), wearable devices, or other sensors. It would include parameters like heart rate, brainwaves (EEG), breathing rate, airflow, etc.

- **Sleep Disorder Diagnosis:**

  This entity would record the diagnosis of a sleep disorder (e.g., sleep apnea, insomnia, restless legs syndrome).

- **Machine Learning Model:**

  This entity would represent the machine learning model used for diagnosis or prediction. It would store information about the model's architecture, parameters, and training data.

- **Feature:**

  This entity would store the features extracted from the sleep data and used as input to the machine learning model (e.g., AHI, respiratory rate, etc.).

- **Prediction:**
  This entity would store the predictions made by the machine learning model, including the predicted diagnosis and confidence level.
  Relationships:
- **Patient to Sleep Data:**

  A "one-to-many" relationship, meaning one patient can have multiple sets of sleep data.

- **Sleep Data to Feature:**

  A "one-to-many" relationship, meaning one set of sleep data can have multiple extracted features.

- **Feature to Machine Learning Model:**

  A "many-to-one" relationship, meaning multiple features can be used by a single machine learning model.

- **Machine Learning Model to Prediction:**

  A "one-to-many" relationship, meaning a machine learning model can make multiple predictions.

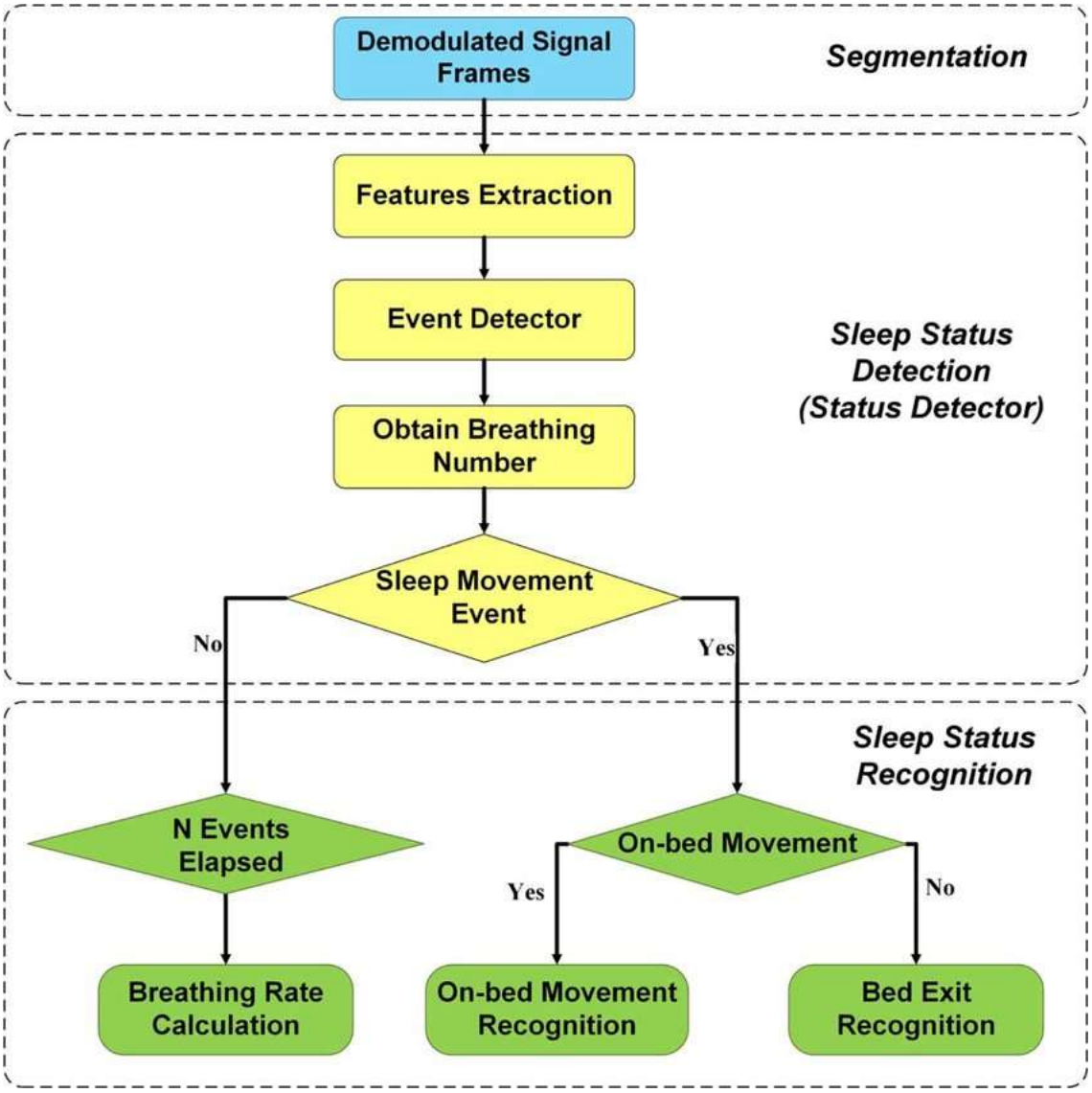
- **Patient to Sleep Disorder Diagnosis:**

  A "one-to-many" relationship, meaning one patient can have multiple diagnoses over time.

- **Sleep Disorder Diagnosis to Prediction:**
  A "one-to-one" or "one-to-many" relationship, depending on the specific application. A diagnosis might be associated with a single prediction, or multiple predictions if the model makes ongoing predictions.

Flow Chart:

A flowchart for sleep disorder diagnosis using machine learning data typically starts with data collection, followed by preprocessing and feature extraction from EEG signals or other relevant



1. Data Collection:

- Gather relevant data from various sources, including polysomnography (PSG) studies, wearable devices, and patient-reported outcomes.
- PSG records physiological signals like EEG, EMG, EOG, heart rate, and respiratory effort.

2. Data Preprocessing:

- Clean and prepare the raw data, removing noise and artifacts.

- Segment EEG signals into epochs (e.g., 30 seconds) for analysis.
- Handle missing values and inconsistencies in the data.

3. Feature Extraction:

- Extract relevant features from the preprocessed data, such as Hjorth parameters (activity, mobility, complexity) from wavelet decomposition of EEG signals.
- Identify patterns in the data that are indicative of different sleep stages and disorders.

4. **Model Development**:

- Select an appropriate machine learning algorithm for classification (e.g., support vector machines, random forests, or deep learning models).
- Train the model using the extracted features and labeled data.

5. Model Validation and Testing:

- Evaluate the model's performance using various metrics (e.g., accuracy, precision, recall).
- Ensure the model generalizes well to unseen data.

6. Automated Diagnosis and Treatment Planning:

- Use the trained model to automatically classify sleep disorders in new patients.
    - Potentially integrate the model into a system for personalized treatment planning and monitoring.

SYSTEM TEST

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing :

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software life cycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

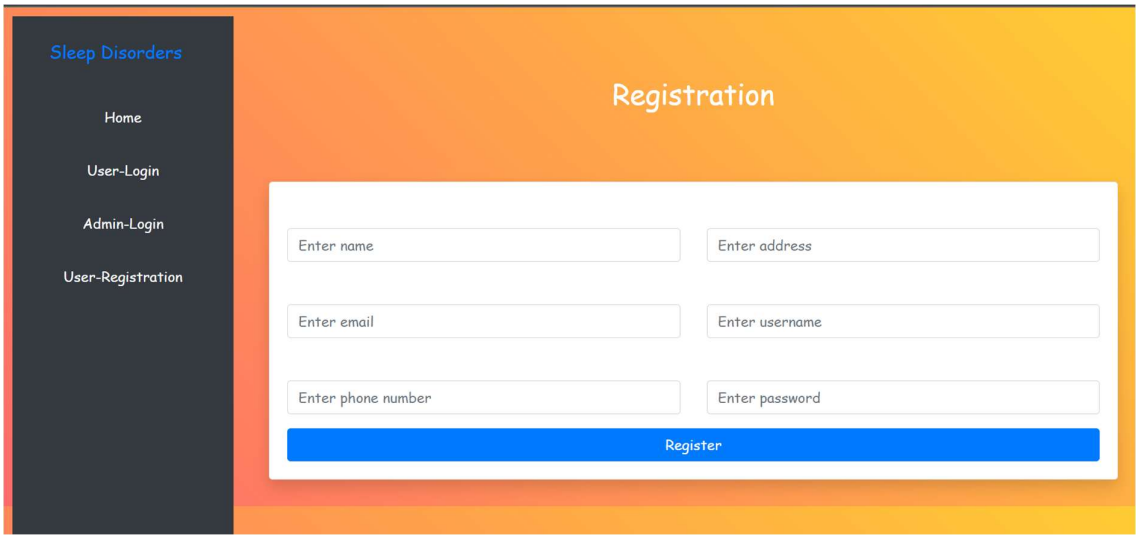
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

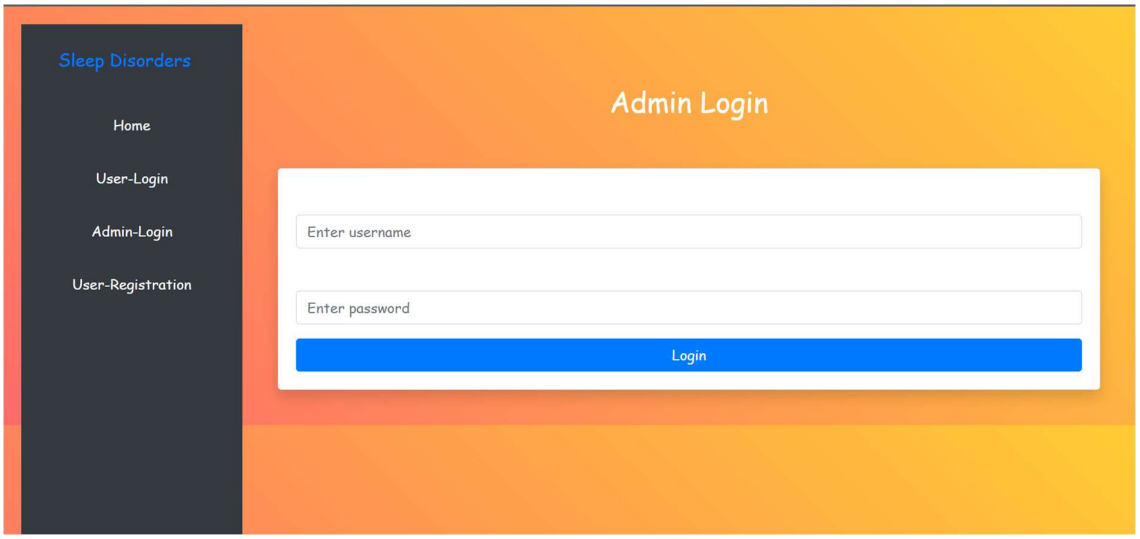
SCREENS & *RESULTS*

Home page:

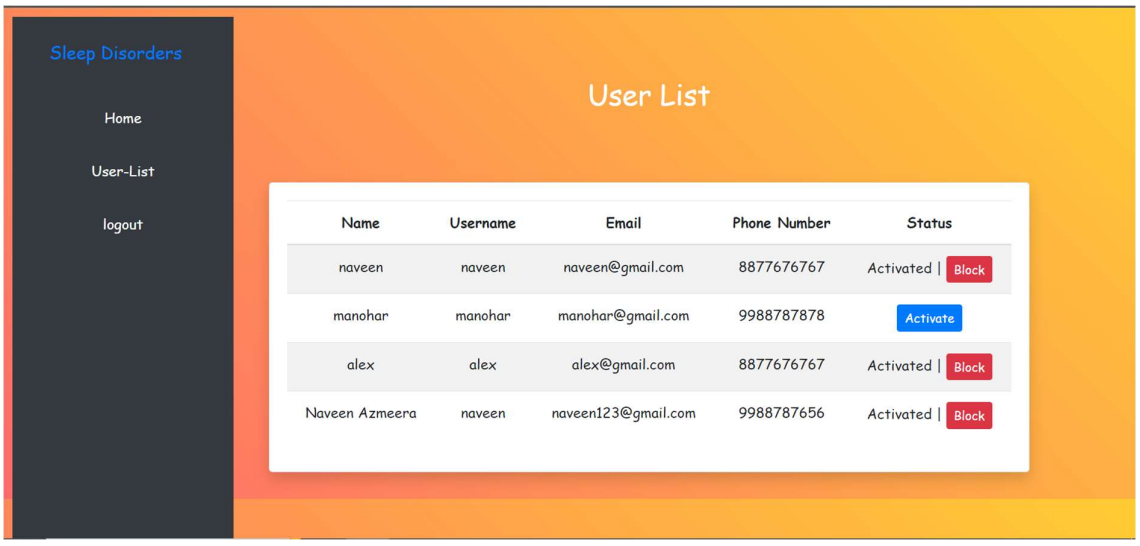
User register:



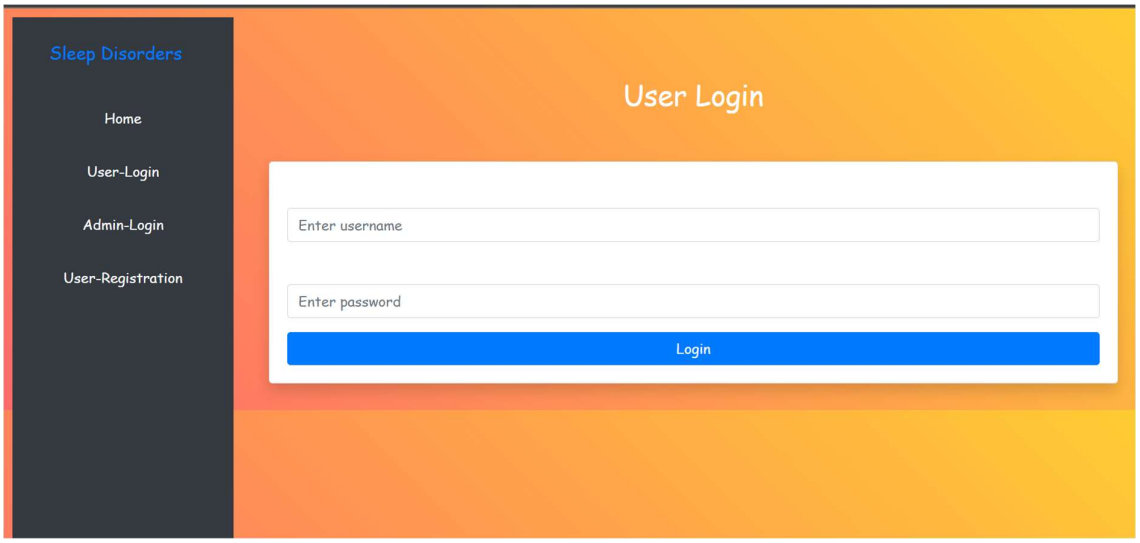
Admin login:



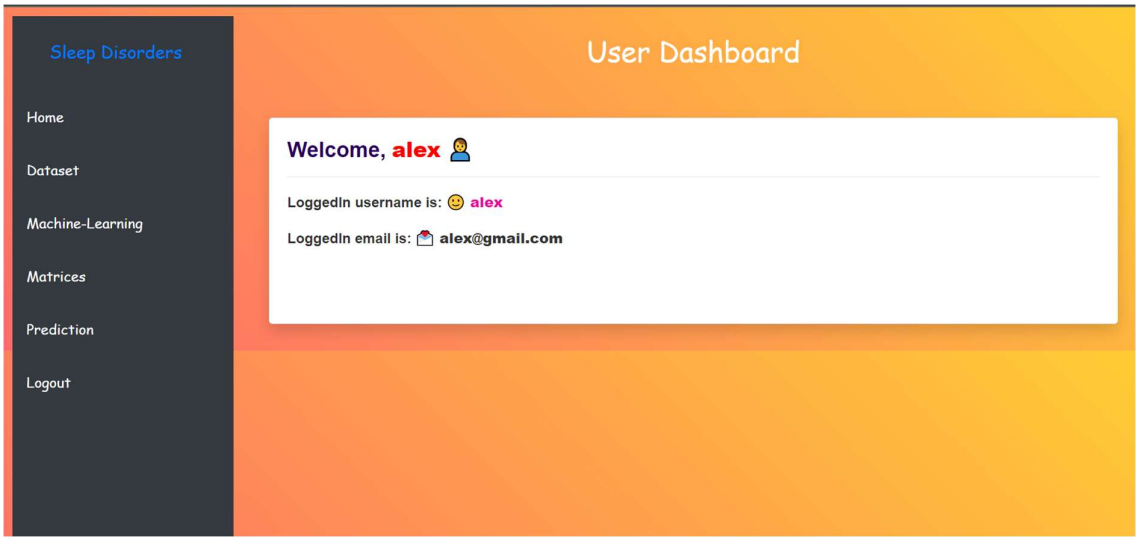
Activate User:



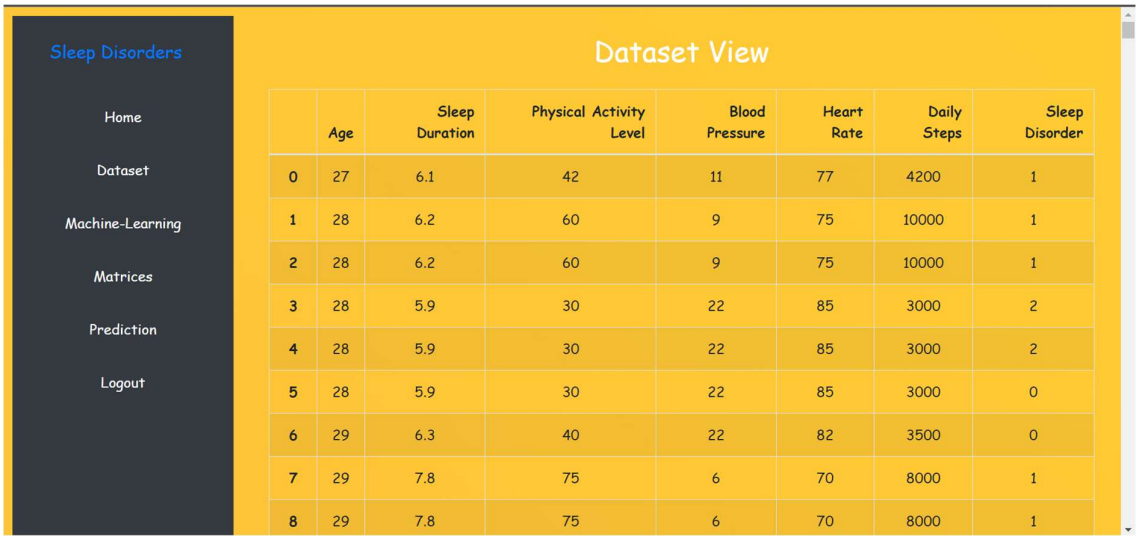
User login Page:



User home Page:

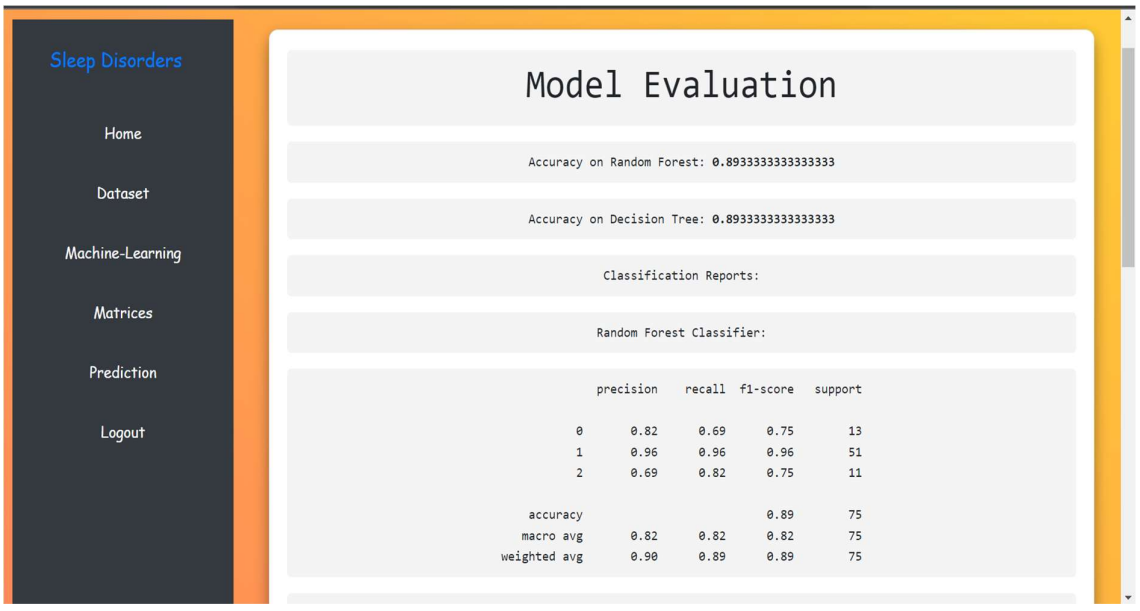


Data Set:

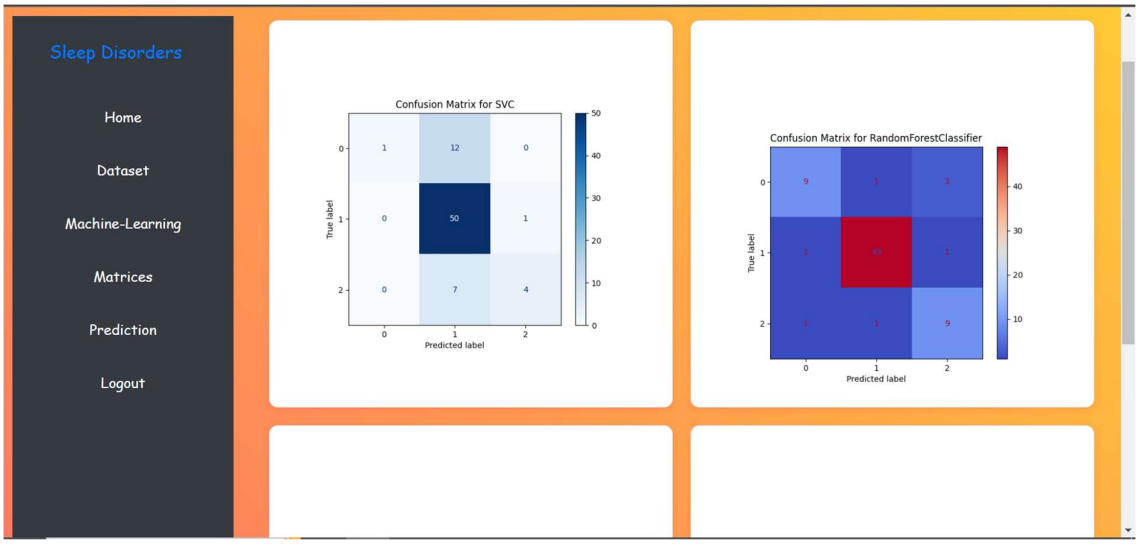


	Age	Sleep Duration	Physical Activity Level	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder
0	27	6.1	42	11	77	4200	1
1	28	6.2	60	9	75	10000	1
2	28	6.2	60	9	75	10000	1
3	28	5.9	30	22	85	3000	2
4	28	5.9	30	22	85	3000	2
5	28	5.9	30	22	85	3000	0
6	29	6.3	40	22	82	3500	0
7	29	7.8	75	6	70	8000	1
8	29	7.8	75	6	70	8000	1

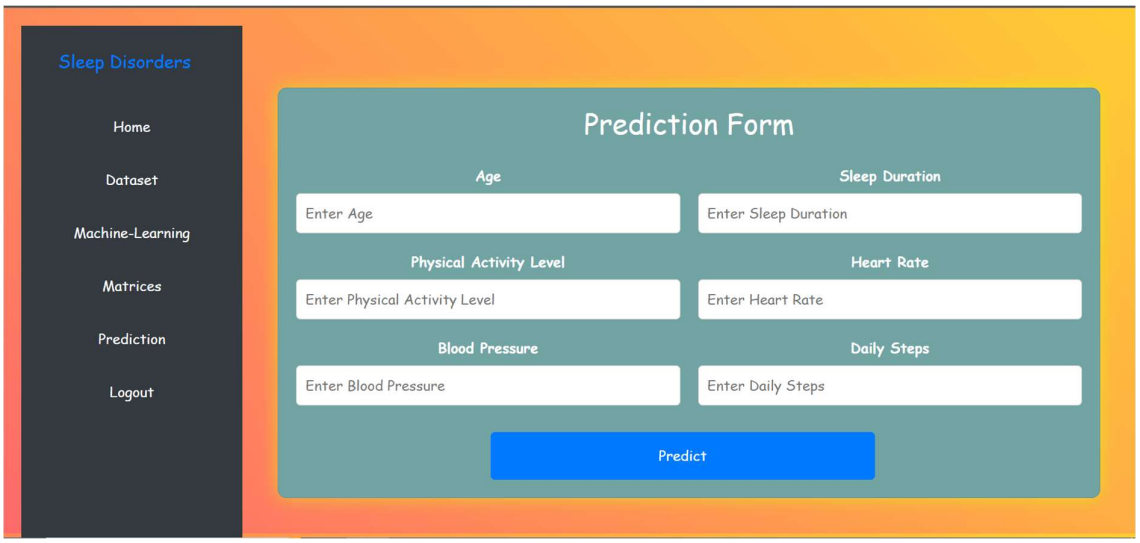
ML Results :



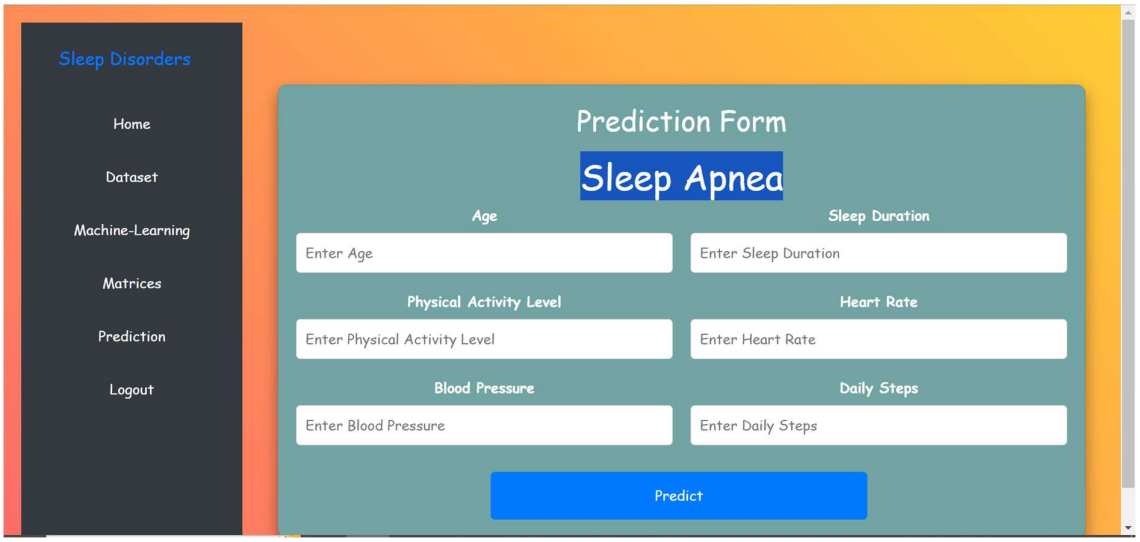
Machine Learning Metrics:



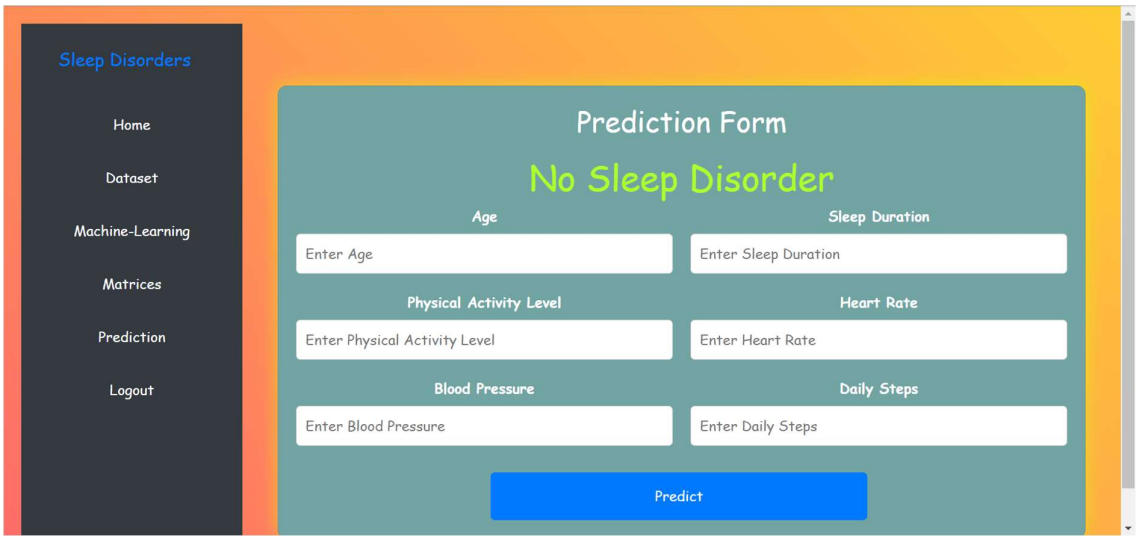
Prediction Form:



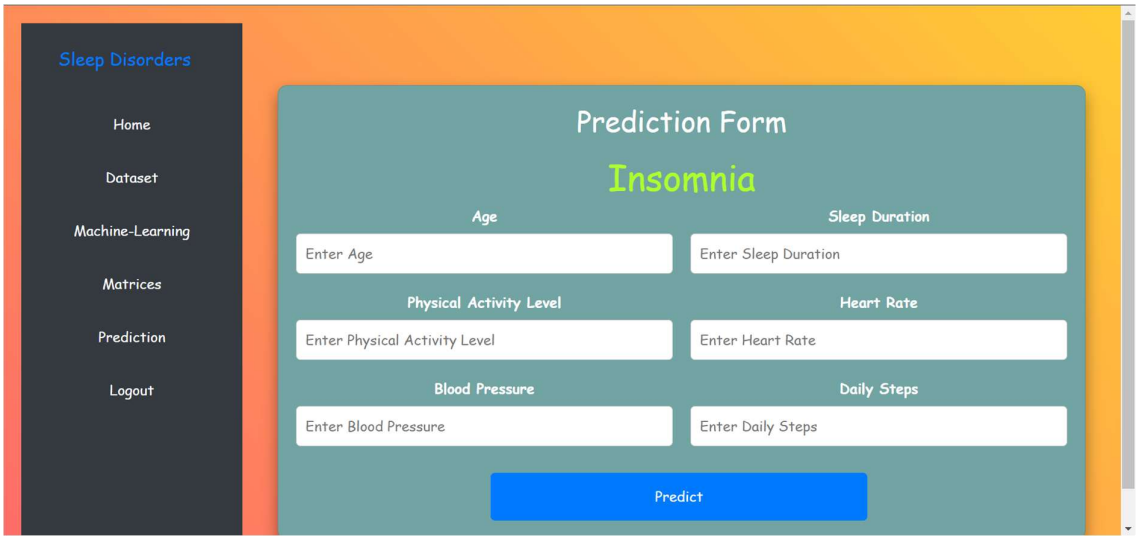
Prediction Label 1:



Predicted Label 2:



Predicted Label 3:



REPORTS

S.no	Test Case	Excepted Result	Result	Remarks (IF fails)
1	User Register	If User registration successfully.	Pass	If already user email exists then it fails.
2	User Login	If Username and password is correct then it will getting valid page.	Pass	Un Register Users will not log in.
3	deep learning algorithm	Here we used deep learning algorithm. ANN and GA	Pass	The request will be not accepted otherwise its failed
4	Model training	model under training results calculated.	Pass	Results not true failed
5	Model training results	Model training results calculated and displayed	Pass	**data is considered for testing**
6	Model accuracy and model loss	Model accuracy and model loss will be displayed by the user	Pass	Results not true failed
7	Prediction results	Detect Sleep Disorder.	Pass	Results not true failed
8	Admin login	Admin can login with his login credential. If success he get his home page	Pass	Invalid login details will not allow here
9	Admin can activate the register users	Admin can activate the register user id	Pass	If user id not found then it won't login.

USER MANUAL

Objective:

To guide users in operating the Loan Prediction system effectively.

System Requirements:

- Python 3.x installed
- Required libraries: pandas, numpy, sklearn, matplotlib, seaborn
- Jupyter Notebook or any Python IDE

Steps to Use:

1. Home Page

The landing page of the application provides a navigation menu that includes links to key sections like Register, Login (Admin/User), and Prediction.

---

2. Register Form

New users must register before accessing the application:

- **Fields Required**: Username, Email, Password, etc.
- After registration, admin approval may be required to activate the account.

---

3. Admin Login Page

Admins can log in using their credentials to:

- View and manage users
- Activate user accounts
- Monitor usage and results

---

4. Activate User

After a user registers, the admin must:

- Navigate to the "Activate User" section
- Approve or deny registration requests
- Activated users will be able to log in and use the system

---

5. User Login Page

Registered and activated users can log in using:

- **Username/Email**
- **Password**
  Once authenticated, they are redirected to their dashboard.

---

6. User Home Page

From here, users can:

- View available datasets
- Access prediction forms
- View previous ML results and metrics

---

7. Dataset

This section displays:

- Input data used for training and testing ML models
- Details such as features, labels, and data types

---

8. ML Results

Users can view:

- Machine learning model predictions
- Outputs generated based on user input

---

9. Machine Learning Metrics

This section shows evaluation results of the model, such as:

- Accuracy
- Precision
- Recall
- F1 Score

---

10. Prediction Form

Users input relevant values into a form to get predictions. The form dynamically interacts with the ML model.

CONCLUSION

An optimised model for sleep disorder classification is proposed that implements MLAs with a genetic algorithm to explore optimal hyperparameter values for each model and obtain good results. This paper analysed the performance of MLAs for sleep disorder classification and evaluated many state-of-the-art MLAs on the real-world Sleep Health and Lifestyle Dataset. In addition, MLAs can learn from high-dimensional sleep data and attempt to classify sleep disorders without depending on expert-defined features. The proposed optimised ANN with GA achieved the highest accuracy over the other MLAs at 92.92%. The precision, recall, and F1-score values on the testing data were 92.01%, 93.80% and 91.93%, respectively. Even with a limitation in the amount of data. This study addressed the challenges in implementing MLAs for classification sleep disordering. However, large datasets are still needed for training and evaluating models in this field. The MLAs with GA can significantly improve the accuracy of sleep disorder classification. Future work will focus on developing MLAs using unsupervised learning in addition to assessing the dataset on a new model and comparing its performance against existing state-of-the-art models.

REFERENCES

[1] F. Mendonça, S. S. Mostafa, F. Morgado-Dias, and A. G. Ravelo-García, ''A portable wireless device for cyclic alternating pattern estimation from an EEG monopolar derivation,'' *Entropy*, vol. 21, no. 12, p. 1203, Dec. 2019.

[2] Y. Li, C. Peng, Y. Zhang, Y. Zhang, and B. Lo, ''Adversarial learning for semi-supervised pediatric sleep staging with single-EEG channel,'' *Methods*, vol. 204, pp. 84–91, Aug. 2022.

[3] E. Alickovic and A. Subasi, ''Ensemble SVM method for automatic sleep stage classification,'' *IEEE Trans. Instrum. Meas.*, vol. 67, no. 6, pp. 1258–1265, Jun. 2018.

[4] D. Shrivastava, S. Jung, M. Saadat, R. Sirohi, and K. Crewson, ''How to interpret the results of a sleep study,'' *J. Community Hospital Internal Med. Perspect.*, vol. 4, no. 5, p. 24983, Jan. 2014.

[5] V. Singh, V. K. Asari, and R. Rajasekaran, ''A deep neural network for early detection and prediction of chronic kidney disease,'' *Diagnostics*, vol. 12, no. 1, p. 116, Jan. 2022.

[6] J. Van Der Donckt, J. Van Der Donckt, E. Deprost, N. Vandenbussche, M. Rademaker, G. Vandewiele, and S. Van Hoecke, ''Do not sleep on traditional machine learning: Simple and interpretable techniques are competitive to deep learning for sleep scoring,'' *Biomed. Signal Process. Control*, vol. 81, Mar. 2023, Art. no. 104429.

[7] Y. Yang, Z. Gao, Y. Li, and H. Wang, ''A CNN identified by reinforcement learning-based optimization framework for EEG-based state evaluation,'' *J. Neural Eng.*, vol. 18, no. 4, Aug. 2021, Art. no. 046059.

[8] Y. J. Kim, J. S. Jeon, S.-E. Cho, K. G. Kim, and S.-G. Kang, ''Prediction models for obstructive sleep apnea in Korean adults using machine learning techniques,'' *Diagnostics*, vol. 11, no. 4, p. 612, Mar. 2021.

[9] Z. Mousavi, T. Y. Rezaii, S. Sheykhivand, A. Farzamnia, and S. N. Razavi, ''Deep convolutional neural network for classification of sleep stages from single-channel EEG signals,'' *J. Neurosci. Methods*, vol. 324, Aug. 2019, Art. no. 108312.

[10] S. Djanian, A. Bruun, and T. D. Nielsen, ''Sleep classification using consumer sleep technologies and AI: A review of the current landscape,'' *Sleep Med.*, vol. 100, pp. 390–403, Dec. 2022.

[11] N. Salari, A. Hosseinian-Far, M. Mohammadi, H. Ghasemi, H. Khazaie, A. Daneshkhah, and A. Ahmadi, ''Detection of sleep apnea using machine learning algorithms based on ECG signals: A comprehensive systematic review,'' *Expert Syst. Appl.*, vol. 187, Jan. 2022, Art. no. 115950.

[12] C. Li, Y. Qi, X. Ding, J. Zhao, T. Sang, and M. Lee, ''A deep learning method approach for sleep stage classification with EEG spectrogram,'' *Int. J. Environ. Res. Public Health*, vol. 19, no. 10, p. 6322, May 2022.

[13] H. Han and J. Oh, ''Application of various machine learning techniques to predict obstructive sleep apnea syndrome severity,'' *Sci. Rep.*, vol. 13, no. 1, p. 6379, Apr. 2023.

[14] M. Bahrami and M. Forouzanfar, ''Detection of sleep apnea from single- lead

ECG: Comparison of deep learning algorithms,'' in *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, Jun. 2021, pp. 1–5.

[15] S. Satapathy, D. Loganathan, H. K. Kondaveeti, and R. Rath, ''Performance analysis of machine learning algorithms on automated sleep staging feature sets,'' *CAAI Trans. Intell. Technol.*, vol. 6, no. 2, pp. 155–174, Jun. 2021.

[16] M. Bahrami and M. Forouzanfar, ''Sleep apnea detection from single-lead ECG: A comprehensive analysis of machine learning and deep learning algorithms,'' *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022. J. Ramesh, N. Keeran, A. Sagahyroon, and F. Aloul, ''Towards validating the effectiveness of obstructive sleep apnea classification from electronic health records using machine learning,'' *Healthcare*, vol. 9, no. 11, p. 1450, Oct. 2021.

[17]S. K. Satapathy, H. K. Kondaveeti, S. R. Sreeja, H. Madhani, N. Rajput, and D. Swain, ''A deep learning approach to automated sleep stages classification using multi-modal signals,'' *Proc. Comput. Sci.*, vol. 218, pp. 867–876, Jan. 2023.

[18]O. Yildirim, U. Baloglu, and U. Acharya, ''A deep learning model for automated sleep stages classification using PSG signals,'' *Int. J. Environ. Res. Public Health*, vol. 16, no. 4, p. 599, Feb. 2019.

[19]S. Akbar, A. Ahmad, M. Hayat, A. U. Rehman, S. Khan, and F. Ali, ''IAtbP-Hyb-EnC: Prediction of antitubercular peptides via heterogeneous feature representation and genetic algorithm based ensemble learning model,'' *Comput. Biol. Med.*, vol. 137, Oct. 2021, Art. no. 104778.

[20](2023). *Sleep Health and Lifestyle Dataset*. [Online]. Available: http://www.kaggle.com/datasets/uom190346a/sleep-health-and-lifestyle- dataset

[21] F. Ordóñez and D. Roggen, ''Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition,'' *Sensors*, vol. 16, no. 1, p. 115, Jan. 2016.

[22] D. M. W. Powers, ''Evaluation: From precision, recall and F- measure to ROC, informedness, markedness and correlation,'' 2020, *arXiv:2010.16061*.

[23] F. Pedregosa, ''Scikit-learn: Machine learning in Python,'' *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[24] M. Bansal, A. Goyal, and A. Choudhary, ''A comparative analysis of K- nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning,'' *Decis. Anal. J.*, vol. 3, Jun. 2022, Art. no. 100071.

[25] M. Q. Hatem, ''Skin lesion classification system using a K-nearest neighbor algorithm,'' *Vis. Comput. Ind., Biomed., Art*, vol. 5, no. 1, pp. 1–10, Dec. 2022.

[26] V. G. Costa and C. E. Pedreira, ''Recent advances in decision trees: An updated survey,'' *Artif. Intell. Rev.*, vol. 56, no. 5, pp. 4765–4800, May 2023. P. Tripathi, M. A. Ansari, T. K. Gandhi, R. Mehrotra, M. B. B. Heyat,

F. Akhtar, C. C. Ukwuoma, A. Y. Muaad, Y. M. Kadah, M. A. Al-Antari, and J. P. Li, ''Ensemble computational intelligent for insomnia sleep stage detection via the sleep ECG signal,'' *IEEE Access*, vol. 10, pp. 108710–108721, 2022.

[27] Y. You, X. Zhong, G. Liu, and Z. Yang, ''Automatic sleep stage classification: A light and efficient deep neural network model based on time, frequency and fractional Fourier transform domain features,'' *Artif. Intell. Med.*, vol. 127, May 2022, Art. no. 102279.

[28] S. Kuanar, V. Athitsos, N. Pradhan, A. Mishra, and K. R. Rao, ''Cognitive analysis of working memory load from eeg, by a deep recurrent neural network,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2576–2580.

[29] A. Hichri, M. Hajji, M. Mansouri, K. Abodayeh, K. Bouzrara, H. Nounou, and M. Nounou, ‘‘Genetic-algorithm-based neural network for fault detection and diagnosis: Application to grid-connected photovoltaic systems,’’ *Sustainability*, vol. 14, no. 17, p. 10518, Aug. 2022.
[30] I. A. Hidayat, ‘‘Classification of sleep disorders using random forest on sleep health and lifestyle dataset,’’ *J. Dinda : Data Sci., Inf. Technol., Data Anal.*, vol. 3, no. 2, pp. 71–76, Aug. 2023.

TALAL SARHEED ALSHAMMARI received the B.Sc. degree in computer science from Al Jouf University, Saudi Arabia, in 2007, the M.Sc. degree in computer science from California Lutheran University, Thousand Oaks, CA, USA, in 2012, and the Ph.D. degree from Staffordshire University, U.K., in 2019. He is currently an Assistant Professor with the College of Computer Science and Engineering, University of Ha’il. His main research interests include artificial intelligence, machine learning, and data science.