Identification of Delirium in patients from Electronic Health Records using Machine Learning

A PROJECT REPORT

Submitted by

S. Archanashreee (913119205006)

G. Jeevitha Sai (913119205014)

B. Maheswari (913119205023)

In Partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

VELAMMAL COLLEGE OF ENGINEERING AND TECHNOLOGY,

AUTONOMOUS

MADURAI-625 009

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2023

VELAMMAL COLLEGE OF ENGINEERING AND TECHNOLOGY, VIRAGANOOR, MADURAI

DEPARTMENT OF INFORMATION TECHNOLOGY

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"Identification of Delirium in patients from Electronic Health Records using Machine Learning"** is the bonafide work of **"S. Archanashreee (913119205006)**, **G. Jeevitha Sai (913119205014), B. Maheswari (913119205023)"** who carried out the project work under my supervision.

HEAD OF DEPARTMENT	**SUPERVISOR**
Dr. R. Kavitha, M.E.,Ph.D.	Dr. R. Kavitha, M.E.,Ph.D.
Professor and Head	Professor and Head
Department of Information Technology	Department of Information Technology
Velammal College of Engineering and Technology, Madurai – 625 009	Velammal College of Engineering and Technology, Madurai – 625 009

Submitted for the Project Viva-Voce Examination held on___________at Velammal College of Engineering and Technology, Madurai.

INTERNAL EXAMINER EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We thank the almighty for giving us the moral strength to work on the project for the past few months.

We would like to express our sincere thanks to, **Shri. M. V. MUTHURAMALINGAM,**

Founder of the institution for giving on opportunity to do our B.Tech Information Technology.

Our sincere thanks to our respected Principal **Dr. P. Alli,** for the encouragement and motivation they offered us during the period of our research study.

Our heartfelt gratitude to **Dr. R. Kavitha,** Head of the Department, Information Technology for his valuable guidance, inspiration and professor appreciation, which helpedus a lot in completing this project in time.

We would also wish to extend our sincere gratefulness to **all faculty members** of the Department of Information Technology for their valuable guidance throughout the course of ourproject. We also thank to our **parents** and **friends** who provided moral and physical support.

ABSTRACT

Delirium is an abrupt, variable, and typically reversible disruption of mental function. It is characterized by a lack of focus, disorientation, difficulty thinking coherently, and varying levels of attentiveness (consciousness). Inattention and generalized cognitive impairment are the main traits of acute neuropsychiatric illness delirium, which is both frequent and dangerous and is often associated with poor outcomes. Delirium frequently complicates a patient's stay in the intensive care unit and poses a serious risk for negative consequences. Delirium diagnosis takes time and needs specialist expertise. Delirium can be avoided by early identification of those who are at risk for it. Once a condition is identified, treatment is time-consuming and involves many teams. This paper aims to demonstrate a model that recognizes delirium, based on EHR data and using a Machine Learning algorithm.

TABLE OF CONTENTS

LIST OF FIGURES

Abbreviations

i. EHR – Electronic Health Record
ii. ML – Machine Learning
iii. SVM – Support Vector Machine
iv. EEG – Electro Encephalogram

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Delirium is a common neuropsychiatric syndrome that often affects patients in hospital settings. It is a serious condition characterized by acute confusion, disorientation, and changes in attention, perception, and cognitive function. Delirium can have serious consequences, including increased morbidity, longer hospital stays, and higher healthcare costs. Traditionally, delirium has been diagnosed through clinical observation and evaluation by healthcare professionals. However, with the increasing availability of electronic health records (EHRs), there is an opportunity to develop automated methods for identifying delirium using machine learning (ML) algorithms. Machine learning algorithms can be trained to recognize patterns in large datasets, including EHRs.

By training ML algorithms on data from patients with delirium and from patients without delirium, these algorithms can learn to identify the specific features or markers that are associated with delirium. Once trained, the ML algorithms can be used to predict the likelihood of delirium in new patients based on their EHR data. Identifying delirium using EHRs and ML algorithms has several potential benefits. It may help healthcare professionals to identify patients who are at risk of developing delirium, allowing for early intervention and improved outcomes. It may also help to reduce the burden on healthcare professionals who are responsible for monitoring patients for signs of delirium, freeing up time and resources for other tasks. However, there are also challenges to using EHR data for delirium identification. EHR data can be complex and heterogeneous; and may contain missing or inaccurate information. Additionally, there may be issues with data privacy and security that need to be addressed. The use of ML algorithms to identify delirium from EHR data shows promise as a valuable tool for improving patient outcomes and healthcare efficiency.

1.2 MOTIVATION

The identification of delirium in patients is important for several reasons. Delirium is a common condition in hospitalized patients, and it can have serious consequences, including increased morbidity, longer hospital stays, and higher healthcare costs. Identifying patients with delirium early can help healthcare professionals provide appropriate interventions to prevent complications and improve outcomes.

However, delirium can be challenging to identify, as it is often underdiagnosed and misdiagnosed. Healthcare professionals may not have enough time to closely monitor all patients for signs of delirium, and symptoms may be subtle or fluctuate over time. This is where machine learning (ML) algorithms can be helpful.ML algorithms can analyze large amounts of data quickly and accurately, which may help identify patients who are at risk of developing delirium. By using electronic health record (EHR) data to train ML algorithms, healthcare professionals can develop models that can detect patterns and markers associated with delirium. These models can then be used to identify patients who may be at risk of delirium and may benefit from closer monitoring or early interventions. The use of ML algorithms to identify delirium from EHR data also has the potential to improve healthcare efficiency. By automating the process of delirium identification, healthcare professionals can save time and resources that can be directed towards other patient care activities. The motivation for identifying delirium in patients from EHR using ML algorithms is to improve patient outcomes, enhance healthcare efficiency, and reduce the burden on healthcare professionals.

1.3 OBJECTIVE

The main objective of identifying delirium in patients from electronic health records (EHRs) using machine learning (ML) algorithms is to improve patient outcomes by providing early detection and timely interventions for delirium. Specifically, the objectives include:

- Developing accurate ML algorithms ML algorithms need to be trained on large datasets of EHRs from patients with and without delirium to identify patterns and markers that are associated with delirium. The accuracy of these algorithms is critical to ensure that patients are correctly identified as being at risk of delirium.

- Creating a prediction model The ML algorithms can be used to create a prediction model that can identify patients who are at risk of developing delirium. This model can be used by healthcare professionals to prioritize patients for closer monitoring or early interventions.
- Improving early detection Early detection of delirium can help prevent complications and improve patient outcomes. By identifying patients who are at risk of delirium, healthcare professionals can intervene early and prevent the development of delirium or reduce its severity.
- Enhancing healthcare efficiency Automating the process of delirium identification using ML algorithms can help healthcare professionals save time and resources, allowing them to focus on other patient care activities.

In summary, the objective of identifying delirium in patients from EHRs using ML algorithms is to improve patient outcomes, enhance healthcare efficiency, and reduce the burden on healthcare professionals.

1.4 SCOPE OF THE PROJECT

- The scope for identifying delirium in patients from electronic health records (EHRs) using machine learning (ML) algorithms is broad, as it has the potential to benefit many stakeholders in the healthcare ecosystem. The scope includes:
- Hospitals and healthcare providers The use of ML algorithms to identify delirium can help hospitals and healthcare providers improve patient outcomes by providing early detection and interventions for delirium. This can also improve healthcare efficiency by automating the process of delirium identification.
- Patients Early detection and interventions for delirium can improve patient outcomes, reduce the length of hospital stays, and lower healthcare costs.
- Researchers The use of ML algorithms to identify delirium can help researchers better understand the risk factors, causes, and outcomes of delirium. This can inform the development of new interventions and treatments for delirium.
- Data scientists and engineers Developing ML algorithms to identify delirium from EHRs requires expertise in machine learning, data analytics, and software engineering. There is a scope for data scientists and engineers to develop new algorithms and tools

to improve the accuracy and efficiency of delirium identification.

- Regulatory agencies The use of EHR data for delirium identification raises issues related to data privacy, security, and regulatory compliance. There is a scope for regulatory agencies to develop guidelines and standards for the use of EHR data in ML algorithms.

1.5 LIMITATIONS

- Limited data availability EHR data may not be available for all patients, or it may not contain all the necessary information for delirium identification. This can limit the accuracy and effectiveness of ML algorithms.
- Data quality and completeness The quality and completeness of EHR data can vary widely, which can affect the accuracy of ML algorithms. Missing or incomplete data can lead to biased or inaccurate models.
- Lack of standardization EHR data may not be standardized across different healthcare systems, which can make it challenging to develop models that can be applied universally.
- Ethical and legal concerns The use of EHR data for delirium identification raises ethical and legal concerns related to patient privacy, data security, and regulatory compliance.
- Human involvement ML algorithms can assist healthcare professionals in identifying delirium, but they cannot replace clinical judgment. Delirium is a complex condition, and healthcare professionals must still be involved in the diagnosis and treatment of patients.

1.6 IMPORTANCE OF THE PROJECT

The identification of delirium in patients from electronic health records (EHRs) using machine learning (ML) is important for several reasons:

- Early detection Delirium is a serious medical condition that can cause complications and increase mortality rates. Early detection of delirium using ML algorithms can help healthcare professionals intervene and prevent or reduce the severity of complications.
- Improved patient outcomes By identifying patients who are at risk of delirium using ML algorithms, healthcare professionals can intervene early and improve patient outcomes. This can result in shorter hospital stays, reduced healthcare costs, and improved quality of life for patients.
- Increased efficiency Automating the process of delirium identification using ML algorithms can save time and resources for healthcare professionals. This can free up time for other patient care activities and improve the overall efficiency of healthcare delivery.
- Better understanding of delirium ML algorithms can help researchers better understand the risk factors, causes, and outcomes of delirium. This can inform the development of new interventions and treatments for delirium.
- Personalized care ML algorithms can be used to develop personalized care plans for patients who are at risk of delirium. This can help healthcare professionals tailor interventions to the specific needs of each patient.

1.7 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between outputs and inputs.

1.7.1 HARDWARE REQUIREMENTS

- Hard Disk: 500GB and Above
- RAM: 4GB and Above
- Processor: I 3 and Above

1.7.2 SOFTWARE REQUIREMENTS

- Operating system: Windows 11
- Technology Use: Machine Learning
- IDE: Python
- Database: Microsoft Excel
- Language used: Python

1.8 NON FUNCTIONAL REQUIREMENTS

A non-functional requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. The plan for implementing non-functional requirements is detailed in the system architecture because they are usually architecturally significant requirements.

Non-functional requirements are in the form of "system shall be <requirement>", an overall property of the system as a whole or of a particular aspect and not a specific function. The system's overall properties commonly mark the difference between whether the development project has succeeded or failed. Non-functional requirements are often called "Quality Attributes" of a system. Other terms for non-functional requirements are "Qualities", "Quality Goals", "Quality of Service Requirements" or "Technical Requirements". Informallythese are sometimes called the "Ilities", from attributes like stability and portability.

Nonfunctional requirements are those requirements that elaborate the performance characteristic of the system and define the constraints on how the system will do.

- Defines the constraints, targets, or control mechanisms for the new system.
- Describes how well or to what standard a function should be provided.
- They are sometimes defined in terms of metrics to make them more tangible.
- Identify realistic, measurable target values for each service level.
- These include reliability, performance, service availability, responsiveness, throughput, and security.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

Delirium is a major medical issue that has a poor outcome and requires early identification and prevention. A model for predicting the onset of delirium in hospitals was developed and evaluated using data from inpatient visits from the Electronic Health Record (EHR) and the Confusion Assessment Method (CAM) screening. For demographic information, comorbidities, drugs, procedures, and physiological measurements, a Random Forest machine learning method was applied. This machine learning method can provide a clinically useful prediction model for earlier intervention in patients who are at a greater risk of developing delirium [1]. A prospective cohort study of internal medicine wards of a tertiary care hospital in China was conducted to create a machine learning model that predicts delirium risk in geriatric internal medicine inpatients. Blinded observers used the Confusion Assessment Method to evaluate delirium (CAM). The model was trained on the training set and assessed on the test set. It took into account five characteristics out of 32 possible predictors: depression, cognitive impairment, drug use, nutritional status, and daily exercise (ADL). This model enhances clinical judgment and enables more accurate targeting of delirium prevention [2].

Delirium has a wide range of complicated underlying medical issues that makes it challenging to diagnose. This study compared the performance of four rule-learning algorithms to screen for psychomotor delirium in adult patients. The major objective was to create a model that can predict delirium in adult patients and offer findings that can be explained. The study discovered that acute consciousness in S-CAM evaluation is strongly connected with several predictors for screening three psychomotor behaviors of delirium, with the LEM2 model being the strongest predictor for accurately identifying adult patients at delirium risk [3]. Machine learning is used to predict delirium in cardiac surgery patients. This study used machine learning techniques to predict delirium in cardiac surgery patients. Models based on logistic regression, ANN, SVM, BBN, naive Bayesian, random forest, and decision trees were created using a clinical dataset. The findings of this study suggest that a machine learning model's ability to detect patients who are prone to have postoperative delirium will be enhanced by correcting class imbalance on the training dataset and exposing hidden patterns. Cost savings, problem avoidance, and patient outcomes optimization will result from this [4].

Electronic Health Records (EHR) are used to record patient data while they are hospitalized, providing a valuable and affordable data source for medical research. In this study, preoperative delirium was predicted using seven machine learning models. Random forests and generalized additive models performed better than other models at predicting delirium, driven by factors such as age, alcohol or drug misuse, socioeconomic level, underlying medical condition, the severity of the condition, and the attending surgeon [5]. To avoid negative outcomes in intensive care units, delirium prediction is necessary. This study evaluated the effectiveness of a cutting-edge method to predict delirium in an intensive care unit. Five prediction models and 168 combinations of study design parameters were used to examine data from patients admitted to the unit. Patients without a history of neuropsychiatric diseases and those brought to the ICU without delirium had the highest performance. A logistic regression classifier was the foundation of this prediction model [6]. Cognitive impairment is a potential side effect of cardiac surgeries involving acute hypothermic circulatory arrest. This study aims to identify biomarkers in the intraoperative electroencephalogram (EEG) that are indicative of postoperative delirium, which is linked to long-term health issues. To make this easier, artifact identification and feature normalization techniques were used to predict 16 instances accurately in 14 cases [7]. A study aimed to develop a classification model that, when applied to retrospective EHR data, can recognize delirium was proposed by Jae Hyun Kim et al. The logistic regression and multi-layer perceptron models were trained using characteristics such as age, sex, the Elixhauser comorbidity score, drug exposures, and diagnoses. The clinical notes from HER were processed to complete the missing characteristics from the structured data, [8].

Patients at risk might be given priority using screening technologies. This project aims to create a delirium prediction model that may be used as a screening tool. Three algorithms, Logistic Regression, Random Forest, and Bidirectional Long Short-Term Memory, were used to analyze the models. Compared to more recent models with fewer variables, the model performed comparably. The drawbacks of prior models were addressed using sliding windows, changing the threshold to improve recall, and feature ranking for interpretability [9]. Machine learning models can improve predictive accuracy in test datasets, but not in clinical workflows. A random forest-based algorithm was used to identify hospitalized patients at high risk for delirium, and its effectiveness in a clinical context was assessed. Delirium was predicted for the most recent hospital admission. This study shows the effectiveness of a machine learning system as a delirium predictor and provides insights into how it can be integrated into clinical workflow [10]. It is important to identify and prevent delirium in critically ill patients as soon as possible. This study uses data from electronic health records to create and verify a delirium prediction model within 24 hours after ICU admission. The prediction of ICU Delirium is the name of the algorithm (PRIDE). The model's RF, XGBoost, DNN, and LR successfully predicted delirium [11]. Machine learning (ML) is a novel method of using data from electronic health records (EHRs) for POD prediction. A POD risk prediction model was established, internally validated, and compared to models created using conventional logistic regression. Two ML models predicted POD in a large perioperative sample with strong accuracy. To enhance the perioperative care of surgical patients at risk for POD, the models should be used in the best way possible to result in automated, real-time delirium risk categorization [12].

This study compared the prevalence, delirium medication therapy, and outcomes of individuals with behavioral disturbances evaluated using Natural Language Processing (NLP) vs those diagnosed using the Confusion Assessment Method for intensive care units (CAM-ICU). Data were collected on demographics, antipsychotic drug use, and outcomes in three mixed medical-surgical ICUs. NLP-Dx-BD positive patients outnumbered CAM-ICU positive patients, and NLP identified more individuals who are likely to be prescribed antipsychotic drugs [13]. Due to its negative effects, such as a prolonged hospital stay, postoperative delirium is a challenging complication. This study aimed to create a machine-learning prediction model and identify preoperative risk variables of postoperative delirium following knee arthroplasty. Data from pre-operative electronic hospital records were used to create the model, which can be used to anticipate delirium before surgery and support medical professionals' efforts to prevent it [14]. Long-term care institutions lack an electronic health record system, making it difficult to monitor and treat delirium. This project designed a web-based tool to prevent delirium in long-term care institutions (LCFs). It is supported by evidence and is simple to deploy to help LCFs provide care for delirious patients. It can increase delirium detection and risk prediction and may be expanded to forecast different risk variables and use preventative measures. Its adoption can raise patient safety and care standards [15].

2.2 INFERENCE OF THE LITERATURE REVIEW

- Delirium is a significant medical condition that is associated with increased mortality rates and healthcare costs.
- The diagnosis of delirium is complex and requires expertise from healthcare professionals. ML algorithms can assist healthcare professionals in identifying patients who are at risk of delirium, which can lead to early detection and intervention.
- ML algorithms can be trained on large datasets of EHRs to identify patterns and risk factors associated with delirium. These algorithms can improve the accuracy and efficiency of delirium identification compared to traditional methods.
- ML algorithms have been shown to be effective in identifying delirium from EHRs in several studies. These algorithms can detect delirium with high accuracy and have the potential to improve patient outcomes and reduce healthcare costs.
- There are several challenges to the use of ML algorithms for delirium identification, including limited data availability, data quality and completeness, lack of standardization, limited generalizability, ethical and legal concerns, and the need for human involvement.
- Despite these challenges, the use of ML algorithms for delirium identification is an active area of research, and there is significant potential for future development and implementation of these algorithms in clinical practice.

CHAPTER 3

PROPOSED SYSTEM

3.1 PROPOSED SYSTEM

The proposed system is an ML model to predict delirium patients by collecting their EEG reports from EHR. An EHR contains all the data of the patient like time of admission, personal details, medication history, treatment history, vital signs (such as blood pressure, heart rate, temperature, and respiratory rate), laboratory test results, progress notes from healthcare providers (including physician notes, nursing notes, and other clinical documentation), immunization records (including vaccination history and current status), care plans (including treatment plans, medication lists, and discharge instructions), appointment schedules and information about past and upcoming appointments, billing and insurance information (including insurance coverage and payment information). In our model, a delirium patient is identified by testing and training the given dataset using the SVM model.

3.2 CONSTRUCTION OF SVM

The Support Vector Machine aims to find a hyperplane that separates the data points into different classes with the largest possible margin between them. The formula for calculating data using SVM is as follows:

$$f(x) = w \wedge t\, x + b$$

where x - input data,

w- weight vector, b - bias term.

In our project, x is the input data from the patient i.e eeg data, and the weight vector is the PO4 data which is the main category for detecting delirium.

3.3 SVM ALGORITHM

The below figure shows the construction of the SVM algorithm. We used the SVM algorithm to predict delirium because the data extracted from patients may be in negative values. This algorithm also considers the negative values for prediction.

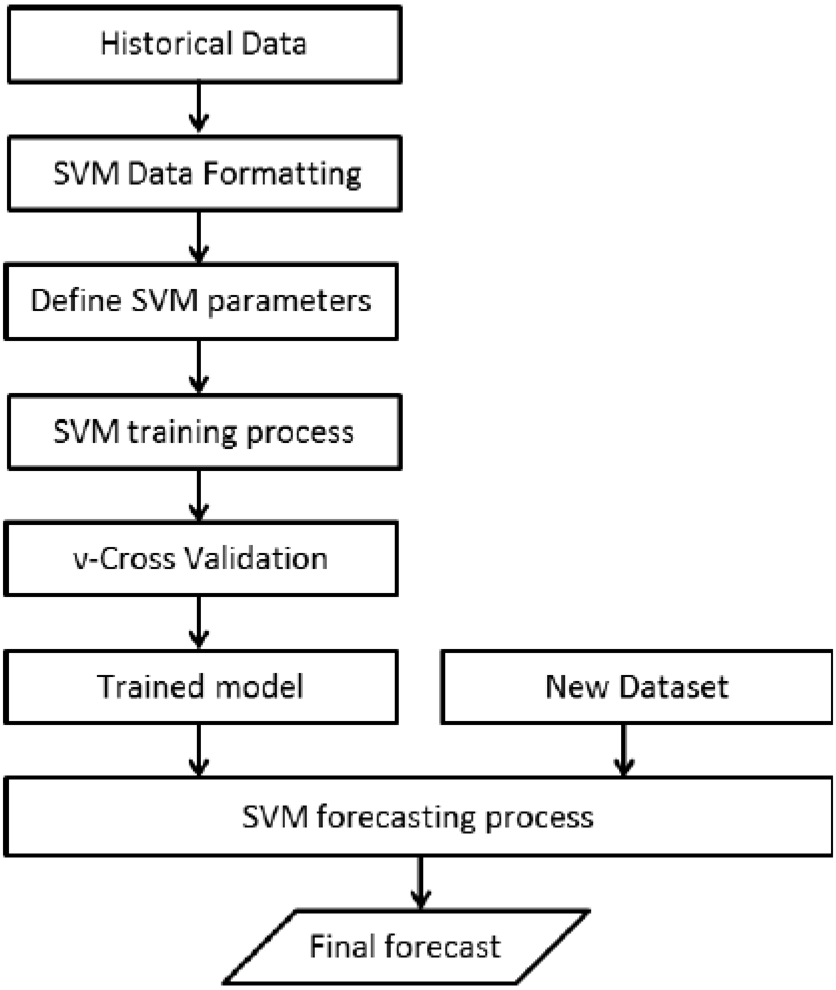

Figure 1 Flow chart for SVM Algorithm

3.4 PROPOSED ALGORITHM

1. Import the necessary package
2. Read the CSV file
3. Process the data
4. Split the training data and test data separately using the train_test_split() function
5. fit the model using the fit() method
6. Read the new data to predict whether the patient has delirium or not
7. Predict the test data using the predict function
8. Print the results

CHAPTER 4
ARCHITECTURE OF THE PROJECT

4.1 OVERALL ARCHITECTURE OF THE PROJECT

A new approach to detect delirium is to monitor physiologic alterations. A new approach to detect delirium is to monitor physiologic alterations. A medical examination that measures the electrical activity of the brain is called an electroencephalogram (EEG). The scalp receives several electrode stimulations. Several diseases, such as epilepsy, insomnia, brain tumors, and delirium can be diagnosed with the use of EEG. Multiple electrodes provide the EEG signals, and different characteristics can be generated from the data. It has become vital to find a solution to the confusion of how to choose electrodes and characteristics to enhance classification performance. Early research showed that the presence and severity of delirium can be correlated with EEG.

This paper demonstrates that using a classifier SVM which operates only with a dataset of EEG correlation signals can classify the delirious and non-delirious groups with high accuracy.

This paper proposes a methodology to identify patients with or without delirium using a supervised machine learning algorithm known as Support Vector Machines (SVM).

SVMs are very useful for splitting data points into two groups when trying to solve problems with binary classification. Finding an ideal hyperplane to divide the data points of several classes in a high-dimensional space is the fundamental concept of SVM. The hyperplane is selected in a manner that maximizes the margin, which is the separation between the hyperplane and the closest data points for each class. Support vectors, which are the data points nearest to the hyperplane, are very important in setting the decision boundary.

This paper focuses on the two important EEG signal data generated by PO4 and O2 electrodes which are crucial factors causing delirium. The data is grouped based on the unique values in the 'O2'. Then the mean value for each group is calculated. This allows us to compare the average values of different columns for each group. The target variable (dependent variable) is separated from the rest of the dataset.

These variables can be used for further analysis, pre-processing, or model training. splits the data into random train and test subsets. These subsets can be used to train a machine learning model on the training data and evaluate its performance on the test data.

The SVM classifier uses a linear decision boundary to separate the classes. The model is finally trained and is used to make predictions on the data. When the input is passed to the trained model, the mean is iteratively checked if it matches the mean of the dataset which is grouped based on unique values in O2. If it is true, then the patient is said to have delirium.

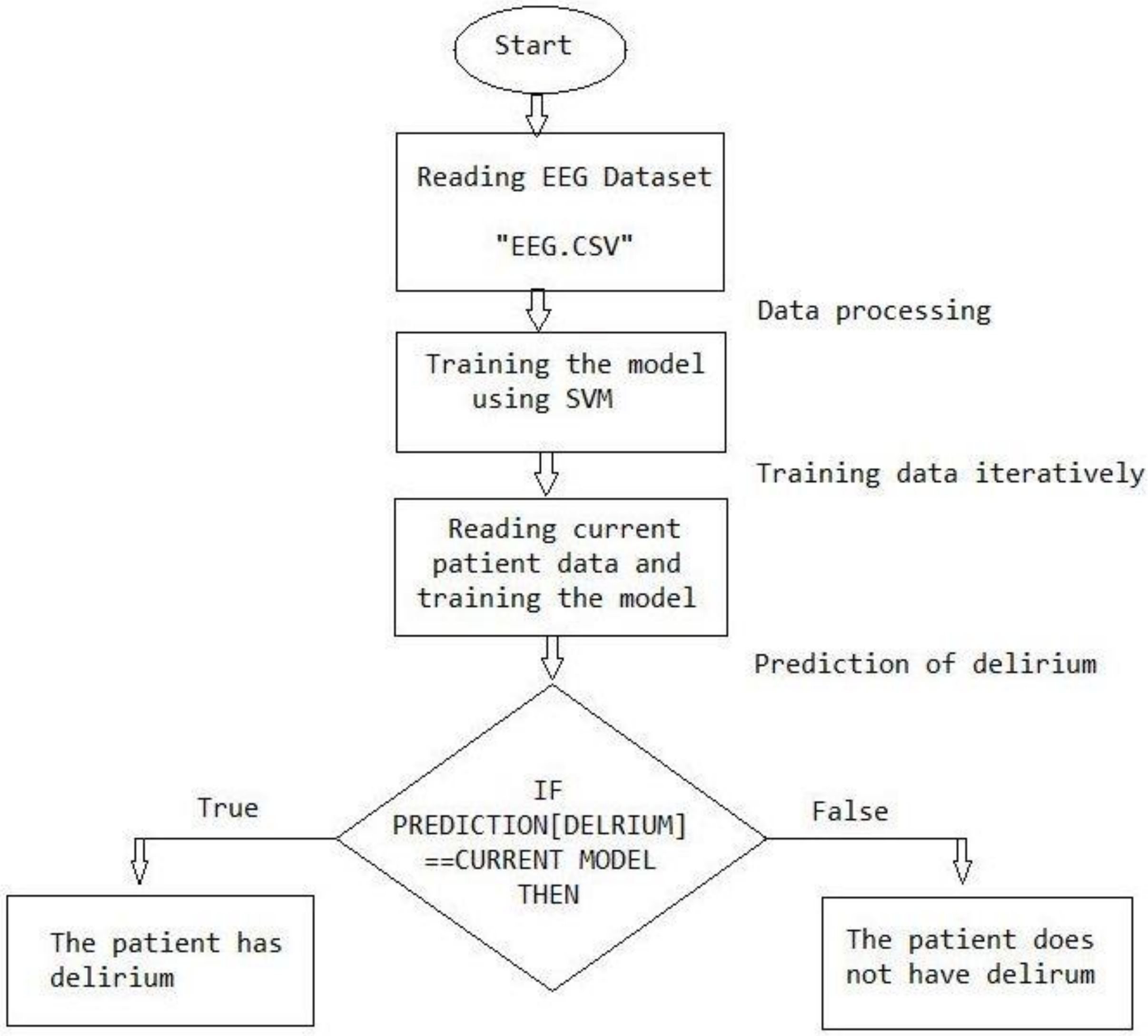

Figure-2 Architecture of the system

CHAPTER 5

IMPLEMENTATION

5.1 HARDWARE

- Hard Disk: 500GB and Above
- RAM: 4GB and Above
- Processor: I 3 and Above

5.2 SOFTWARE

- Operating system: Windows 11
- Technology Used: Machine Learning
- IDE: Jupiter Notebook
- Database: Microsoft Excel
- Language used: Python

5.3 TECHNOLOGY USED

5.3.1 MACHINE LEARNING

In the 1970s and 1980s, ML research faced several setbacks due to limitations in computing power and a lack of high-quality data. However, the field started to gain momentum in the 1990s, when advancements in computing power and the availability of large datasets allowed researchers to develop more sophisticated ML algorithms. The 2000s saw the emergence of new techniques such as deep learning, which uses neural networks with many layers to learn from data. This led to breakthroughs in areas such as image recognition and natural language processing. Today, ML has become an integral part of many industries, including healthcare, finance, and retail. The field continues to evolve rapidly, with researchers developing new algorithms and techniques to address challenges such as bias and interpretability.

Machine learning is a type of artificial intelligence (AI) that enables computer systems to learn from data and improve their performance on a specific task over time, without being explicitly programmed. In other words, machine learning algorithms can identify patterns and relationships in data, and use that knowledge to make predictions or decisions. There are three main types of machine learning: supervised learning, unsupervised learning, and reinforcement learning.

Supervised learning involves training a model on labeled data, where the correct output is already known. The model uses this data to learn how to make predictions or decisions on new, unseen data.

Unsupervised learning involves training a model on unlabeled data, where the correct output is not known. The model identifies patterns and relationships in the data on its own, without guidance, and can be used for tasks such as clustering or anomaly detection.

Reinforcement learning involves training a model to make decisions based on feedback from the environment. The model receives rewards or punishments for its actions, and learns to maximize rewards over time.

Machine learning has numerous applications, including image and speech recognition, natural language processing, fraud detection, recommendation systems, and predictive modeling in healthcare and finance. Machine learning (ML) has enormous potential to revolutionize the medical field. Here are some examples of how ML is being used in healthcare:

Diagnosis and Prognosis Machine learning algorithms can analyze patient data and help doctors diagnose and predict disease. For example, ML can be used to analyze medical images such as X-rays and MRI scans to detect tumors and other abnormalities that might be difficult to see with the naked eye.

Personalized Treatment Machine learning can help doctors develop personalized treatment plans based on a patient's unique health profile. ML algorithms can analyze data from electronic health records and patient-generated data to identify patterns that can help predict how a patient is likely to respond to a particular treatment.

Drug Discovery ML can be used to accelerate the drug discovery process by identifying potential drug candidates and predicting their efficacy and safety.

Remote Patient Monitoring Machine learning algorithms can analyze data from wearable devices and other remote monitoring devices to track a patient's health and detect early warning signs of a potential health problem.

While there are many potential benefits to using ML in healthcare, there are also challenges and limitations to consider. ML algorithms are only as good as the data they are trained on, and there is a risk of bias in the algorithms if the data used to train them is not diverse or representative. There are also concerns around the interpretability and explainability of ML algorithms, as well as issues related to data privacy and security.

Advantages

Automation Machine learning can automate many tasks that would otherwise require human expertise, time, and resources, such as image and speech recognition, natural language processing, and predictive analytics.

Efficiency Machine learning can process large amounts of data quickly and accurately, which can improve the speed and quality of decision-making and reduce errors and biases.

Scalability Machine learning can scale up or down depending on the size and complexity of the data and the computational resources available, which can improve the flexibility and adaptability of the system.

Generalization Machine learning can learn from a diverse range of data and generalize to new data, which can improve the robustness and reliability of the predictions and reduce the need for manual interventions.

Innovation Machine learning can drive innovation by discovering new insights, patterns, and relationships in the data that were previously unknown or unexplored.

Disadvantages

Data quality Machine learning depends on the quality, completeness, and representativeness of the data. Poor-quality data, missing data, or biased data can lead to inaccurate or unfair predictions and decisions.

Bias Machine learning can amplify and propagate biases that are present in the data or the algorithms, such as racial, gender, or socioeconomic biases. This can lead to discriminatory or unjust outcomes and reinforce existing inequalities.

Interpretability Machine learning can be difficult to interpret or explain, especially for complex models such as deep learning. This can limit the transparency and accountability of the system and make it difficult to identify and correct errors or biases.

Overfitting Machine learning can overfit to the training data by memorizing the data rather than learning the underlying patterns. This can lead to poor generalization and performance on new data and require regularization or validation techniques.

Security and privacy Machine learning can pose security and privacy risks by exposing sensitive data, vulnerabilities, or malicious attacks. This can compromise the confidentiality, integrity, and availability of the data and the system.

5.4 DEVELOPMENT TOOLS

5.4.1 PYTHON

Python is a high-level, interpreted programming language that was first released in 1991 by Guido van Rossum. It has become one of the most popular programming languages in the world, thanks to its simplicity, versatility, and powerful libraries.

Python is known for its clean syntax and easy-to-read code, which makes it a popular choice for beginners and experts alike. It supports multiple programming paradigms, including object-oriented, procedural, and functional programming.

Python has a vast library of modules and packages that can be used for various tasks, such as web development, data analysis, machine learning, and scientific computing. Some of the popular libraries for data analysis and machine learning include NumPy, Pandas, Scikit-learn, TensorFlow, and PyTorch.

Python is also known for its strong community support, with a large and active community of developers who contribute to open-source projects and provide support and resources to fellow programmers. Python was first released in 1991 by Guido van Rossum, a Dutch programmer who was working at the National Research Institute for Mathematics and Computer Science in the Netherlands. Van Rossum wanted to create a programming language that was easy to use, readable, and could handle complex tasks.

The name "Python" was inspired by the British comedy group Monty Python, of which van Rossum was a fan. The first version of Python, version 0.9.0, was released in February 1991. Over the years, Python evolved and became more powerful, with the release of several major versions. Python 2.0 was released in 2000, which included new features such as garbage collection and improved Unicode support. Python 3.0 was released in 2008, which included significant changes to the language, such as removing features that were considered obsolete or problematic. Python has become one of the most popular programming languages in the world, with a large and active community of developers. It is used in many industries and fields, including web development, data analysis, scientific computing, and machine learning. Python's popularity can be attributed to its simplicity, readability, versatility, and powerful libraries. It has a vast library of modules and packages that can be used for various tasks, and it supports multiple programming paradigms, including object-oriented, procedural, and functional programming. Python's community support is also strong, with a large number of open-source projects and resources available for developers.

5.4.2 JUPYTER NOTEBOOK

Jupyter Notebook is an open-source web application that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. It is a popular tool for data analysis, scientific computing, and machine learning.

The name "Jupyter" is a combination of the three programming languages it supports Julia, Python, and R. However, it also supports many other programming languages. Jupyter Notebook provides an interactive computing environment that allows users to write and run code, visualize data, and create rich multimedia documents. The notebooks are organized into cells, which can contain code, markdown, or raw text. Users can execute code cells to see the output immediately, without having to run a separate script. Jupyter Notebook is a versatile and powerful tool for data analysis, scientific computing, and machine learning, and one of the reasons for its popularity is its support for a wide range of libraries and frameworks. These libraries and frameworks are pre-written code that can be used to perform specific tasks such as data manipulation, visualization, and machine learning.

Here are some of the most popular libraries and frameworks used in Jupyter Notebook:

NumPy NumPy is a Python library used for working with arrays. It provides a powerful array object and functions for working with arrays, such as mathematical operations, indexing, and reshaping.

Pandas Pandas is a Python library used for data manipulation and analysis. It provides powerful data structures, such as DataFrames and Series, and functions for working with data, such as merging, filtering, and grouping.

Matplotlib Matplotlib is a Python library used for data visualization. It provides a wide range of functions for creating charts and graphs, such as line plots, scatter plots, and histograms.

Seaborn Seaborn is a Python library built on top of Matplotlib. It provides a higher-level interface for creating statistical visualizations, such as heatmaps, bar charts, and violin plots.

Scikit-learn Scikit-learn is a Python library used for machine learning. It provides a wide range of functions for building and evaluating machine learning models, such as classification, regression, and clustering.

TensorFlow TensorFlow is a popular open-source machine learning framework developed by Google. It provides a wide range of functions for building and deploying machine learning models, such as deep neural networks, convolutional neural networks, and recurrent neural networks.

PyTorch PyTorch is another popular open-source machine learning framework used in Jupyter Notebook. It provides a dynamic computational graph and functions for building and training machine learning models, such as deep neural networks and convolutional neural networks.

One of the main advantages of Jupyter Notebook is that it allows users to share their work easily. Notebooks can be saved in various formats, including HTML, PDF, and Markdown, and can be shared with others via email or online platforms such as GitHub or Jupyter Notebook Viewer. Overall, Jupyter Notebook is a powerful and flexible tool for interactive computing, data analysis, and scientific research.

5.4.3 MICROSOFT EXCEL

Microsoft Excel is a spreadsheet program developed by Microsoft Corporation. It is widely used for tasks such as data analysis, financial modeling, and data visualization. Excel allows users to create and manipulate spreadsheets organized into rows and columns and can contain text, numbers, formulas, and functions. Excel has a wide range of features and capabilities that make it a powerful tool for working with data. These include:

Data organization Excel provides a variety of tools for organizing data, such as sorting, filtering, and grouping.

Formulas and functions Excel has a vast library of built-in formulas and functions that can be used to perform calculations and analysis of data.

Data visualization Excel allows users to create charts, graphs, and other visualizations to help them understand and present data.

Data analysis Excel has a range of features for data analysis, such as pivot tables, data tables, and what-if analysis.

Macros and automation Excel allows users to create macros and automate tasks, saving time and increasing efficiency.

Excel is widely used in many industries and professions, including finance, accounting, marketing, and engineering. It is also a popular tool for personal use, such as creating budgets or managing personal finances.

Microsoft Excel is often used as a database, especially for small to medium-sized businesses and organizations that do not have the need or resources for a full-scale database management system. While it is not designed specifically as a database program, Excel has many features that make it suitable for storing and managing data, including:

Spreadsheet organization Excel uses a spreadsheet format, which makes it easy to organize and store data in rows and columns.

Data validation Excel provides tools for validating data, such as drop-down lists, data entry restrictions, and conditional formatting, which help ensure that data is entered correctly.

Sorting and filtering Excel allows users to sort and filter data, making it easy to find and extract specific information.

Formulas and functions Excel has a wide range of built-in formulas and functions that can be used to perform calculations and analysis of data.

Charts and graphs Excel provides tools for creating charts and graphs, which can be used to visualize data and identify trends.

While Excel has many benefits as a database, it is important to note that it has some limitations compared to more robust database management systems. For example, Excel has limited security features and can only handle a certain amount of data before performance is affected. Additionally, it may not be suitable for more complex data structures or multi-user environments. Excel may not be the most powerful database management system available, it can be a useful tool for storing and managing data in many business and organizational contexts.

5.5 RESULT

A. *Dataset*

The dataset used to identify Delirium patients from Electronic Health Records (EHR) is represented inFigure-1 below. EEG data from the EHR was collectedto identify delirium, and the dataset was created by gathering patient details through the reference link https://archive.ics.uci.edu/ml/datasets/eeg+database. The system was developed using this data.

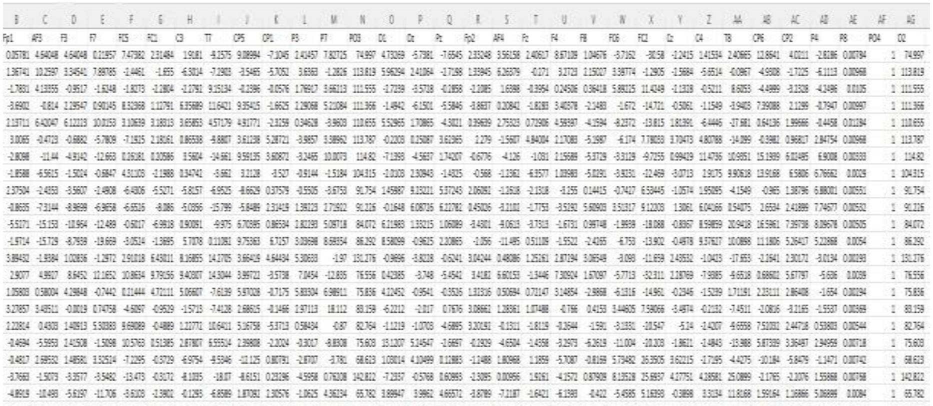

Figure-3 Data set description

B. Importing libraries

The required imports such as numpy (as np), pandas (as pd), matplotlib.pyplot (as plt),train_test_split and StandardScaler from sklearn,svm from sklearn for implementing Support Vector Machines (SVM) with Scikit-Learn for classification jobs are performed.

C. Loading the data

The data from a CSV file named 'eeg1.csv' is loaded into a Pandas DataFrame called delirium_data. Data preprocessing and modeling steps are executed next.

D. *Data preprocessing*

To determine the number of rows and columns in the delirium_data DataFrame, you can use the shape attribute.The structure of the data frame is examined by using delirium_data.head(). The statistical summary for each numerical column in the Data Frame, including the count, mean, standard deviation, and quartiles is obtained.

	Fp1	AF3	F3	F7	FC5	FC1	C3	T7	CP5	CP1	...	FC2
count	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	...	195.000000
mean	-0.106505	-0.318698	-0.042240	-0.299562	0.226797	0.050570	0.262757	-0.339075	0.499166	-0.019999	...	-1.176044
std	5.413082	40.818598	4.231414	28.327196	32.842429	9.909672	31.506629	42.499395	57.459542	14.832331	...	196.030158
min	-18.735390	-174.162043	-10.964000	-117.254905	-95.516772	-29.704726	-93.189754	-175.751521	-169.456961	-59.674319	...	-839.742441
25%	-3.400272	-12.026347	-3.186378	-10.372932	-11.161279	-5.009656	-9.012655	-14.709759	-16.210154	-5.761297	...	-46.760695
50%	0.405876	0.186423	-0.126543	-0.265751	0.214436	-0.073445	0.666480	-0.410979	1.870917	0.246369	...	-2.669080
75%	3.057775	11.600970	2.561254	10.780359	9.818905	4.674052	8.440832	17.121559	14.427592	5.516463	...	54.802399
max	14.212123	113.828207	11.656166	73.831522	140.707711	38.094272	133.738083	118.616808	247.950231	43.397580	...	580.377956

8 rows × 32 columns

Figure-4 Statistical summary

By eliminating the mean and scaling to unit variance, the StandardScaler preprocessing step is frequently used to standardize characteristics. This transformation is frequently applied to make sure that the scales of the characteristics are similar and to stop some features from predominating over others throughout the learning process. The features will have zero mean and unit variance after data standardization, which is frequently advantageous for many machine learning methods.

E. Grouping data based on target variable

The data in the delirium_data are grouped based on the 'O2' column and calculate the mean value for each group. This assumes that 'O2' is a categorical variable representing the target variable. This allows us to observe the average values of the other columns corresponding to different categories of the 'O2' variable.

F. Separating the features and the target

The feature matrix containing all the columns from delirium_data except 'name' and 'PO4' is termed as X, and the target variable, which is the 'name' column is termed as Y.

G. Splitting data into Test and Training Data

The train-test split on the feature matrix X and the target variable Y is performed. The train-test split separates the data into training and testing sets, enabling us to assess how well the model performs on data that was previously unknown. The percentage of the data that should be assigned to the test set is set to 0.2 indicating that 20% of the data would be used for testing and the remaining 80% for training. Four separate variables are obtained as training features, testing features, training target, and testing target. We train the SVM model on these variables and assess its effectiveness.

H. Model Training

For classification, a linear decision boundary is employed. The SVM model is now fitted to the training set of data in order to assess its performance.

The training data will be used to train the SVM model, which will then use the learned parameters to predict patterns and make predictions. We can assess how effectively the model generalizes to new cases by analyzing how well it performs on the testing data.

```
model = svm.SVC(kernel='linear')
```

```
#Training SVM model with training data
model.fit(X_train, Y_train)
```

```
SVC
SVC(kernel='linear')
```

Figure-5 Training the linear SVM model

I. *Model Evaluation*

We can assess your SVM model's performance by looking at the accuracy ratings for both the training and testing sets of data. The model had a training accuracy of 80.12% and a test accuracy of 79.48%. Finally, we predict the class label for the standardized input data using the trained SVM model (model. predict()). Then whether the individual has delirium or not is displayed based on the predicted result.

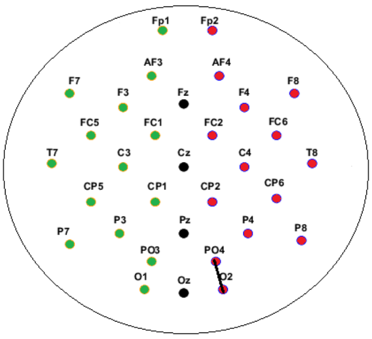

Figure-6 Most effective electrode combinations for detecting delirium

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

The use of machine learning algorithms for the identification of delirium in patients from electronic health records has shown promising results. It can potentially improve the accuracy and efficiency of delirium diagnosis, leading to better patient outcomes and reduced healthcare costs. However, it is important to note that machine learning algorithms are not perfect and may have limitations in terms of data availability and biases. Therefore, further research and validation studies are needed to ensure the reliability and validity of these algorithms. Additionally, the use of machine learning algorithms should not replace clinical judgment and expertise but rather be used as a tool to aid and support clinical decision-making. Overall, the application of machine learning in healthcare, including the identification of delirium, has the potential to improve patient care and outcomes.

6.2FUTURE SCOPE

The future scope for the identification of delirium in patients from electronic health records using machine learning is promising. Here are some potential areas of development:

Improving the accuracy and specificity of the algorithms With more data and more refined machine learning algorithms, the accuracy and specificity of delirium diagnosis can be improved.

Developing real-time monitoring tools Machine learning algorithms can be integrated into real-time monitoring tools that continuously analyze electronic health records to identify patients at risk for delirium.

Combining different types of data Machine learning algorithms can be applied to a combination of different types of data, including clinical, laboratory, and imaging data, to better predict delirium.

Incorporating natural language processing Natural language processing can be used to extract information from clinical notes and reports, which can be used to improve the accuracy and specificity of delirium diagnosis.

Developing personalized interventions Machine learning algorithms can be used to develop personalized interventions for patients at risk for delirium, such as targeted medications, behavioral interventions, and cognitive training.

The future of machine learning in the identification of delirium is bright, with many opportunities for development and refinement. With continued research and validation studies, machine learning algorithms can become valuable tools for improving patient care and outcomes.

CHAPTER 7

APPENDIX

```
/*Importing Libraries*/
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression

/*Data Collection and Analysis

#Loading the data from .csv file to a Pandas Dataframe
delirium_data = pd.read_csv('eeg1.csv')

#Printing the 1st 5 rows of the Dataframe
delirium_data.head()

#No. of rows and columns in the Dataframe
delirium_data.shape

#Getting more infor about the dataset
delirium_data.info()

#Checking for missing values in each column
delirium_data.isnull().sum()

import seaborn as sns
sns.countplot(delirium_data['PO4'])
sns.countplot(delirium_data['O2'])
```

```
#Getting statistical measures about the data
delirium_data.describe()

#Distribution of target Variable
delirium_data['PO4'].value_counts()

#Grouping data based on target variable
delirium_data.groupby('O2').mean()

/* Data Pre-Processing
/*Separating the Features and Target

X = delirium_data.drop(columns=['name','PO4'], axis=1)
Y = delirium_data['name']
print(Y)

/*Splitting data into Test and Training Data
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)

print(X.shape, X_train.shape, X_test.shape)

/*Data Standardization
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
print(X_train)

/* Model Training
/*Support Vector Machine

model = svm.SVC(kernel='linear')
```

```
#Training SVM model with training data
model.fit(X_train, Y_train)
/*Model Evaluation
/*Accuracy score
X_train_prediction = model.predict(X_train)
training_data_accuracy = (accuracy_score(Y_train, X_train_prediction)*100)
print('Accuracy score of training data : ', training_data_accuracy)
X_test_prediction = model.predict(X_test)
test_data_accuracy = (accuracy_score(Y_test, X_test_prediction)*100)
print('Accuracy score of test data : ', test_data_accuracy)

/*Building a Predictive System

input_data=
(95.730,132.068,91.754,0.00551,0.00006,0.00293,0.00332,0.00880,0.02093,0.191,0.01073,0.
01277,0.01717,0.03218,0.01070,21.812,0.615551,0.773587,5.498678,0.327769,2.322511,0.2
31571,0.00006,0.00293,0.00332,0.00880,0.02093,0.191,0.01073,0.01277,0.01070,)

input_data_as_numpy_array = np.asarray(input_data)

# reshape the numpy array
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

# standardize the data
std_data = scaler.transform(input_data_reshaped)
prediction = model.predict(std_data)
if (prediction[0] == 0):
  print("The Person does not have Delirium")
else:
  print("The Person has Delirium")
```

CHAPTER 8

REFERENCES

[1] Corradi, J.P., Thompson, S., Mather, J.F. et al.“Prediction of Incident Delirium Using a Random Forest classifier”,14 November 2018.

[2] Qinzheng Li, Yanli Zhao, Yu Chen, Jirong Yue, Yan Xiong, “Developing a machine learning model to identify delirium risk in geriatric internal medicine inpatients”, European Geriatric Medicine, 2021.

[3] Chang-Sik Son, Won-Seok Kang, Jong-Ha Lee, Kyoung Ja Moon, “Machine Learning to Identify Psychomotor Behaviors of Delirium for Patients in Long-Term Care Facility”, IEEE Journal of Biomedical and Health Informatics, vol.26, no.4, pp.1802-1814, 2022.

[4] Hani Nabeel Mufti, Gregory Marshal Hirsch, Samina Raza Abidi, Syed Sibte Raza Abidi, “Exploiting Machine Learning Algorithms and Methods for the Prediction of Agitated Delirium After Cardiac Surgery: Models Development and Validation Study”, JMIR Medical Informatics, vol.7, no.4, pp.e14993, 2019.

[5] Anis Davoudi, Tezcan Ozrazgat-Baslanti, Ashkan Ebadi, Alberto C. Bursian, Azra Bihorac, Parisa Rashidi, “Delirium Prediction using Machine Learning Models on Preoperative Electronic Health Records Data” Bioinformatics and Bioengineering, 2017.

[6] Filipe R. Lucini, Kirsten M. Fiest, Henry T. Stelfox and Joon Lee, “Delirium prediction in the intensive care unit: a temporal approach” Engineering in Medicine & Biology Society (EMBC),2020

[7] Owen Ma, Arindam Dutta, Daniel W. Bliss, Amy Z. Crepeau “Predicting postoperative delirium in patients undergoing deep hypothermia circulatory arrest” Signals, Systems, and Computers,2017.

[8] Jae Hyun Kim, May Hua, Robert A. Whittington, Junghwan Lee, Cong Liu, Casey N. Ta, Edward R. Marcantonio, Terry E. Goldberg, and Chunhua Weng “A machine learning approach to identifying delirium from electronic health records” JAMIA Open, Volume 5, Issue 2, July 2022.

[9] Anirban Bhattacharyya, Seyedmostafa Sheikhalishahi, Heather Torbic, Wesley Yeung, Tiffany Wang, Jennifer Birst, Abhijit Duggal, Leo Anthony Celi, and Venet Osmani "Delirium prediction in the ICU: designing a screening tool for preventive interventions", JAMIA Open, Volume 5, Issue 2, July 2022.

[10] Stefanie Jauk, Diether Kramer, Birgit Großauer, Susanne Rienmu¨ later, Alexander Avian, Andrea Berghold, Werner Leodolter, and Stefan Schulz "Risk prediction of delirium in hospitalized patients using machine learning: An implementation and prospective evaluation study", Volume 27, Issue 9, September 2020.

[11] Sujeong Hur, Ryoung-Eun Ko, Junsang Yoo, Juhyung Ha, Won Chul Cha, Chi Ryang Chung, "A Machine Learning–Based Algorithm for the Prediction of Intensive Care Unit Delirium (PRIDE): Retrospective Study", Vol 9, No 7 (2021): July.

[12] Bishara, A., Chiu, C., Whitlock, E.L. et al. "Postoperative delirium prediction using machine learning models and preoperative electronic health record data", 03 January 2022.

[13] Young, M., Holmes, N., Kishore, K, et al. "Natural language processing diagnosed behavioral disturbance vs confusion assessment method for the intensive care unit: prevalence, patient characteristics, overlap, and association with treatment and outcome". Intensive Care Med 48, 559–569 (2022).

[14] Jung, J.W., Hwang, S., Ko, S. et al. "A machine-learning model to predict postoperative delirium following knee arthroplasty using electronic health records", 27 June 2022.

[15] Moon, K.J., Son, CS., Lee, JH. et al. "The development of a web-based app employing machine learning for delirium prevention in long-term care facilities in South Korea", 17 August 2022.