Predicting Brain Stroke Risk Using Machine Learning and Deep Learning Models.

[1] Anitha Rajathi, [2]Parvathy A,[3]Abinaya B,[4]Shivani M

[1]Assistant Professor, Department of CSBS, R.M.D Engineering College, anitha.csbs@rmd.ac.
[2]UG Scholar, Fourth Year, Department of CSBS, R.M.D Engineering College, 22202036@rmd.ac.in
[3]UG Scholar, Fourth Year, Department of CSBS, R.M.D Engineering College, 22202060@rmd.ac.in
[4]UG Scholar, Fourth Year, Department of CSBS, R.M.D Engineering College, 22202050@rmd.ac.in

Abstract

This project aims to develop a comprehensive system for the classification and prediction of brain strokes using advanced machine learning and deep learning techniques The project encompasses data preprocessing, data visualization, and the implementation of various algorithms to achieve high accuracy in stroke prediction. Data preprocessing involves cleaning and preparing datasets, addressing missing values, and performing feature engineering to enhance model performance. Data visualization techniques are employed to uncover insightful patterns and trends related to stroke, aiding in the interpretation of complex medical data. In addition to traditional machine learning algorithms, this system incorporates deep learning techniques using Convolutional Neural Networks (CNNs) for effective stroke detection from imaging data. CNNs are utilized to extract spatial features and detect stroke-related anomalies in medical images, significantly improving the system's diagnostic accuracy. The integration of CNN-based deep learning ensures robust analysis, especially when dealing with high-dimensional data such as brain scans. The complete solution is deployed within the Django web framework, providing a user-friendly interface for medical professionals. This web application enables healthcare providers to make informed, real-time decisions based on predictive analytics and model outputs, thus contributing to timely stroke diagnosis and intervention.

Keywords: Brain Stroke Classification, Machine Learning, Data Preprocessing, Data Visualization, Deep Learning, Detection, CNN, Django Framework.

.

I.INTRODUCTION

A. Problem Definition:

Stroke is one of the leading causes of death and long-term disability worldwide. Early and accurate detection of stroke is critical for timely medical intervention and improved patient outcomes. However, traditional diagnostic methods rely heavily on manual analysis of medical imaging and clinical data, which can be time-consuming, prone to human error, and dependent on expert availability.

With the increasing volume of medical data, including brain imaging scans and patient health records, there is a need for an automated, accurate, and real-time prediction system that can assist healthcare professionals in stroke diagnosis. Existing systems often lack integration between structured clinical data analysis and image-based detection, limiting their diagnostic efficiency.

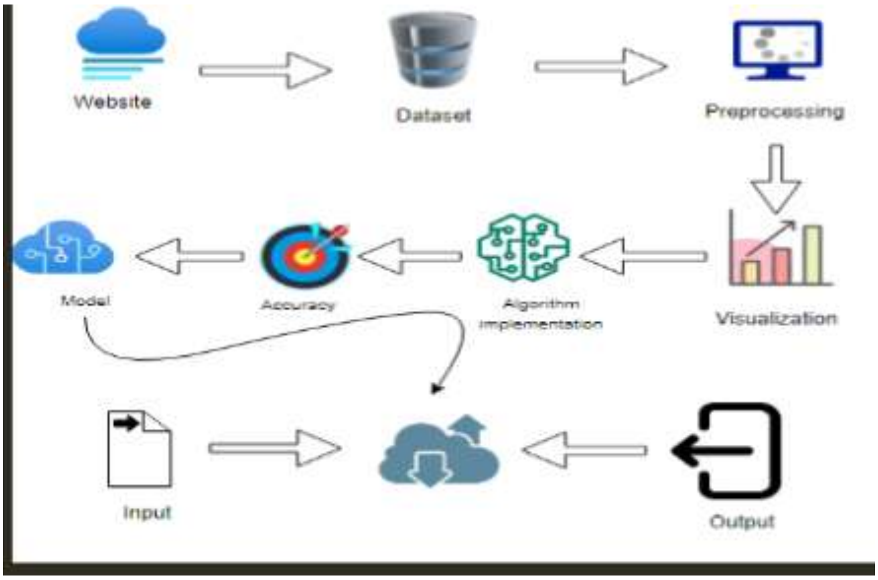
Therefore, the problem is to develop a comprehensive, intelligent system that can preprocess medical data, analyze patterns using machine learning techniques, detect stroke-related abnormalities from brain images using Convolutional Neural Networks (CNNs), and provide real-time predictions through a user-friendly web interface to support medical decision-making.

B. Problem-Solving: The proposed system integrates machine learning and deep learning techniques for accurate stroke classification and prediction. Data preprocessing is performed to clean, normalize, and prepare clinical and imaging datasets for analysis. Multiple machine learning algorithms are applied to predict stroke risk based on patient health parameters, while Convolutional Neural Networks (CNNs) are used to analyze brain images and detect stroke-related abnormalities.

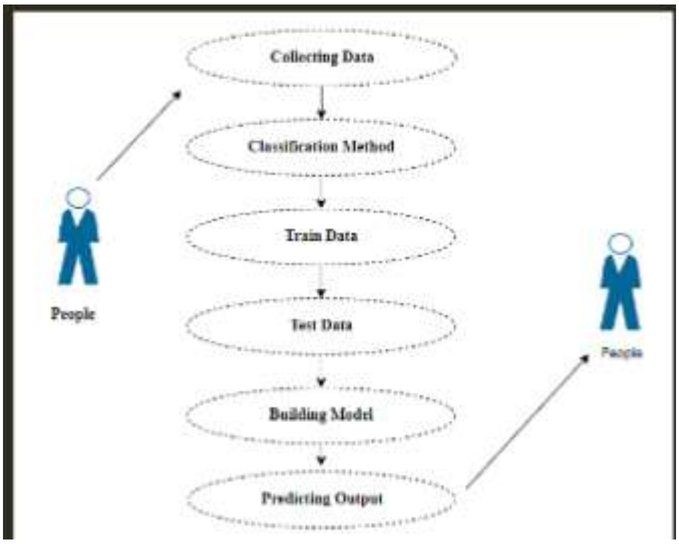
Te complete system is deployed using the Django web framework, providing a user-friendly platform for real-time stroke prediction and decision support for healthcare professionals.

II.METHODOLOGY

A. SYSTEM ARCHITECTURE



B.USE CASE DIAGRAM:



III.LIST OF MODULES:

Data Pre-processing

Data Analysis of Visualization

KNN

Adaboost classifier

Random Forest Classifier

Deployment

IV.MODULE DESCRIPTION:

MODULE- 1
Data Pre-processing:

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the
population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and
description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters.

MODULE-2:
DATA VISUALIZATION:

Data visualization is an important skill in applied statistics and machine learning.Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and
can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data
analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

MODULE – 3
KNN CLASSIFIER:

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.

K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

K-NN algorithm can be used for Regression as well as for Classification but mostly it I used for the Classification problems.

K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.

MODULE – 4:
ADABOOST CLASSIFIER:

This method operates iteratively, identifying misclassified data points and adjusting their weights to minimize the training error. The model continues to optimize sequentially until it yields the strongest predictor.

AdaBoost is implemented by combining several weak learners into a single strong learner. The weak learners in AdaBoost take into account a single input feature and draw out a single split decision tree called the decision stump. Each observation is weighted equally while drawing out the first decision stump.

The results from the first decision stump are analyzed, and if any observations are wrongfully classified, they are assigned higher weights. A new decision stump is drawn by considering the higher-weight observations as more significant. Again if any observations are misclassified, they're given a higher weight, and this process
continues until all the observations fall into the right class.

MODULE – 5:
RANDOM FOREST CLASSIFIER:

The Random Forest Classifier is an ensemble learning method that combines multiple decision trees to create a robust and accurate model. It operates by aggregating the results from various decision trees to improve prediction accuracy and control overfitting.

1. Decision Trees:

- A Random Forest is composed of multiple decision trees, which are individual models that make predictions based on features in the dataset. Each decision tree is trained on a random subset of the data and features.

2. Bootstrap Aggregating (Bagging):

- Random Forest employs a technique called bagging (Bootstrap Aggregating). Each decision tree is trained on a random sample of the training data, sampled with replacement. This means some data points may be repeated in the sample, and some may be omitted.

MODULE-6:
DEPLOYMENT

Django

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your
app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

V.ABOUT THE PROJECT

Brain stroke, a leading cause of severe disability and death worldwide, demands rapid and accurate diagnostic methods for effective treatment. "Deep Stroke:Exploring Machine Learning Techniques for Accurate Brain Stroke Classification"aims to address this need by leveraging advanced machine learning algorithms to enhance stroke diagnosis. This project focuses on developing and evaluating Machine learning models, and other cutting-edge techniques, to accurately classify stroke types from medical imaging and clinical data. By integrating these technologies, we seek to improve diagnostic precision, support clinicians in decision-making, and ultimately contribute to better patient outcomes in stroke management.

..

VI.APPLICATION OF PROJECT

The proposed system can be applied in the following healthcare contexts:

- Early-stage neurodegenerative disease screening
- Clinical risk assessment support
- MRI-based diagnostic assistance
- Research environments for AI-based medical analysis

VII.EXISTING SYSTEM

Emotion recognition based on electroencephalogram (EEG) is fundamentally associated with human-like intelligence system. However, due to the noise-sensitive characteristics of EEGs and the individual variability of emotions, it is very challenging to extract inherent emotion dependent patterns from emotional EEG signals. In this work, we propose a L1-norm space defined discriminative brain network manifold learning model (L1-SGL), in which the EEG noise outliers can be effectively separated and the pseudo labeled samples caused by subjective feelings can be automatically corrected. Off-line experimental results consistently indicate that the L1-SGL can effectively suppress the influence of noise and achieve an incomparable superiority performance over other existing methods in EEG emotion recognition. Besides, benefiting from the time efficiency of the L1-SGL, an online emotion monitoring and regulation system is further implemented in this work. On-line emotion decoding experimental results (86.30%) of 25 participants prove that the L1-SGL can effectively satisfy the real-time requirements of on-line emotional monitoring applications, and the significant negative emotion regulation experimental results ($p < 0.001$) further confirm the feasibility and effectiveness of L1-SGL model in real-time
emotion regulation and interactive applications. Overall, the L1-SGL provides a promising solution
for the Realtime online affective brain-computer interfaces (aBCIs) and the intelligent clinical closed-loop treatments

Disadvantages:

□ Limited Sample Size: The study only included 15 stroke patients, which may not be sufficient to generalize findings across a broader population or different stroke severities.

□ Static vs. Dynamic Analysis: While dynamic modular analysis offers deeper insights, the static modularity analysis used in previous studies might still be useful for capturing global network behaviors that could complement dynamic findings.

□ Potential Overemphasis on Network Metrics: Focusing heavily on dynamic network metrics like recruitment and integration might overlook other critical factors influencing stroke recovery, such as individual patient variability or non-network-related aspects of rehabilitation.

VIII.PROPOSED SYSTEM

The proposed system follows a comprehensive workflow that includes data preprocessing, visualization, algorithm implementation, and web deployment. Initially, brain imaging data is cleaned, normalized, and preprocessed to ensure high-quality input for predictive modeling. Proper preprocessing improves model reliability and reduces errors caused by noise or missing values.

Data visualization techniques such as histograms, plots, and graphs are used to understand patterns and trends related to stroke occurrence. Both traditional machine learning algorithms and deep learning techniques are implemented to classify stroke cases and predict outcomes accurately.

Convolutional Neural Networks (CNNs) are specifically used for analyzing medical imaging data. CNNs automatically extract complex spatial features from brain scans and detect stroke-related abnormalities with high precision. This deep learning integration enhances the system's ability to interpret image-based medical data effectively.

The entire system is deployed using the Django framework, which provides a user-friendly web interface. Medical professionals can input patient data, view predictions, and access diagnostic insights in real time. This creates a robust, intelligent, and accessible platform for stroke classification and analysis.

- **MERITS**
- Framework-based web application for easy deployment
- Implementation of machine learning techniques for predictive modeling
- Improved accuracy and performance
- Data analysis performed using histograms, plots, and graphical visualization
- Comparison of multiple algorithms to achieve better accuracy

ENVIRONMENTAL REQUIREMENTS

- 1. Software Requirements
- Operating System: Windows
- Development Tools: Anaconda, Jupyter Notebook
- Framework: Django
- 2. Hardware Requirements
- Processor: Pentium III / IV or higher
- Hard Disk: Minimum 100 GB
- RAM: Minimum 4 GB

IX. RESULT AND DISCUSSION

The performance evaluation of the proposed stroke detection system indicates that deep learning models, particularly Convolutional Neural Networks (CNNs), achieved higher accuracy compared to traditional machine learning algorithms. Effective preprocessing, feature engineering, and image enhancement techniques significantly improved prediction performance and minimized misclassification.

The integration of machine learning and deep learning approaches resulted in a reliable and efficient prediction system. The deployed Django-based platform enables real-time stroke classification, assisting healthcare professionals in making timely and informed clinical decisions.

X. CONCLUSION

This study demonstrates the effectiveness of machine learning techniques in brain stroke classification and prediction. Algorithms such as Support Vector Machines (SVM), Random Forests, and deep learning models successfully identified complex patterns in brain imaging data related to stroke occurrence and severity.

The findings reveal that ensemble methods and deep learning approaches provide improved classification accuracy compared to conventional techniques. Integrating advanced machine learning models into clinical systems can support faster and more precise diagnoses, ultimately enhancing patient care and treatment planning.

Future work should focus on expanding the dataset size, applying cross-validation techniques, reducing model bias, and ensuring robustness across diverse healthcare environments. These improvements will enhance the system's generalizability and practical applicability in real-world clinical settings**.**

XII.FUTURE WORK

- The added background knowledge from other datasets can also possibly improve the accuracy of stroke

prediction models as well.

- We intend to collect our institutional dataset for further benchmarking of these machine learning

methods for stroke prediction.

- We also plan to perform external validation of our proposed method, as a part of our upcoming planned

work.

REFERENCES

[1] D. Li, S. Huang, L. Xie, Z. Wang, and J. Xu, "Neuron perception inspired EEG emotion recognition with parallel contrastive learning," IEEE Trans. Neural Netw. Learn. Syst., early access, Mar. 14, 2025, doi: 10.1109/TNNLS.2025.3546283.

[2] M. M. Shanechi, "Brain–machine interfaces from motor to mood,"Nature Neurosci., vol. 22, no. 10, pp. 1554–1564, Oct. 2019.

[3] S. K. Khare and V. Bajaj, "Time–Frequency representation and convolu-tional neural network-based emotion recognition," IEEE Trans. NeuralNetw. Learn. Syst., vol. 32, no. 7, pp. 2901–2909, Jul. 2021.

[4] Z. Gao et al., "EEG-based spatio-temporal convolutional neural network for driver fatigue evaluation," IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 9, pp. 2755–2763, Sep. 2019.

[5] S. Sakhavi, C. Guan, and S. Yan, "Learning temporal information for brain–computer interface using convolutional neural networks," IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 11, pp. 5619–5629, Nov. 2018.

[6] J. Shen, X. Zhang, G. Wang, Z. Ding, and B. Hu, "An improved empiri-cal mode decomposition of electroencephalogram signals for depression detection," IEEE Trans. Affect. Comput., vol. 13, no. 1, pp. 262–271, Jan. 2022.

[7] J. Shen et al., "Exploring the intrinsic features of EEG signals via empirical mode decomposition for depression recognition," IEEE Trans.Neural Syst. Rehabil. Eng., vol. 31, pp. 356–365, 2023.

[8] J. Shen et al., "Depression recognition from EEG signals using an adap-tive channel fusion method via improved focal loss," IEEE J. Biomed.Health Informat., vol. 27, no. 7, pp. 3234–3245, Jul. 2023.

[9] T. Gu, Z. Wang, X. Xu, D. Li, H. Yang, and W. Du, "Frame-level teacher–student learning with data privacy for EEG emotion recognition," IEEE Trans. Neural Netw. Learn. Syst., vol. 34, no. 12,pp. 11021–11028, Dec. 2022.

[10] Y. Hou et al., "GCNs-net: A graph convolutional neural network approach for decoding time-resolved EEG motor imagery signals," IEEE Trans. Neural Netw. Learn. Syst., vol. 35, no. 6, pp. 7312–7323, Jun. 2024.