Machine Learning-Based Prediction and Prevention of Brain Stroke Analysis

Manish Deep, Rithish R, P Geetha and T Grace Shalini
Department of Computational Intelligence, SRMIST, Chennai, Tamil Nadu, India

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

Abstract: Machine Learning in the Medical domain, especially in brain strokes prediction and prevention is one of the most sophisticated techniques. Stroke in the brain occurs when blood supply to part of the brain is interrupted which causes possible cell death. Brain strokes are of two types: hemorrhagic, whenever bleeding occurs inside the brain, and ischemic, which originate from a blocked artery-ship. When treated early, strokes can be managed, preserving life and resulting in less disability and better medical outcomes. Innovative machine learning techniques have been implemented to create a smart framework for stroke classification and forecasting as presented in this paper. It will consist of major parts like data cleaning, graphical presentation of data, and various machine learning models to achieve high accuracy in prediction. You are procced on the data until October 2023. Meaningful stroke-related narratives Using visualization techniques process the information complexities into a more understandable form. We employ a set of machine learning classifiers to effectively detect and predict stroke events. Data quality, model interpretability issues and privacy challenges are critical to successful deployment of ML-based stroke prevention systems. It is also implemented with a Django-based web platform where health-care professionals can make accurate decisions on prediction made by the model.

1 INTRODUCTION

A brain stroke develops when blood flow stops temporarily in any area of the brain thus depriving brain cells of oxygen before they die. The world-wide rate of brain strokes constitutes a major global health challenge because these medical conditions cause substantial disability together with being frequent causes of death. The process of early intervention results in reduced stroke severity and enhanced patient recovery outcomes after such incidents. The traditional assessment methods used for stroke predictions through clinical examinations and imaging do not efficiently detect minimal early warning indications. Through medical data analysis the ML models process comprehensive information such as both biomedical and clinical and imaging and biomarker information to make predictions about potential strokes. The implementation of patient data systems faces critical privacy and ethical barriers that must receive proper solutions before becoming widely accepted. The predictive models built from dark box systems face challenges because clinicians need to understand the derivation process behind the predictions before accepting them as valid. Weaknesses in prediction and brain stroke prevention through machine learning will be examined in this paper with emphasis on the techniques used along with data sources and technology benefits and implementation barriers. However, strokes continue to be one of the major causes of long term disability and premature death and require fast and accurate diagnosis. Leveraging AI for Brain Stroke Classification and Detection. The main idea behind the project is to develop and evaluate AI models that are able to predict what type of stroke a person had based on information available from detailed medical images and clinical records. This project hope to enhance the capabilities of healthcare professionals to make more data informed treatment decision and better care for the patients. Beyond classification, this research prioritizes optimizing preprocessing frameworks and refining feature engineering strategies for better predictive performance. By applying deep learning architectures such as CNNs and multi-layer perceptron's, to enhance the system's ability to recognize stroke indicators accurately. Automation reduces manual intervention, making the

Deep, M., R., R., Geetha, P. and Shalini, T. G.
Machine Learning-Based Prediction and Prevention of Brain Stroke Analysis.
DOI: 10.5220/0013940100004919

In *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT'25 2025) - Volume 5*, pages 634-640
ISBN: 978-989-758-777-1

diagnostic process more efficient and error-free. The developed real time predictive model helps at mitigating the virus exponentially by assisting early intervention and better patient management. The main goal of this initiative is to hit on the world's need for an affordable, scalable tool that can detect strokes quickly and facilitate discovery in neurological research and care.

2 LITERATURE REVIEW

Sivapalan et.al, (2022), discusses the utilization of multiple machine learning models for stroke classification. For this study, logistic regression, SVM, CatBoost, Random forest, Multi-Layer Perceptron (MLP), Naïve Bayes and K nearest Neighbors (NN) were used as the seven-machine learning technique. It was found that CatBoost performed the better accuracy, precision, recall, F1 score. Koh H C et.al, (2011), discusses the growing role of machine learning in fields like healthcare, security, and data analytics. This study employs data mining techniques to analyze stroke-related information from both linguistic and syntactic perspectives, making it possible to extract crucial patient information efficiently.

Yoo et. al, (2012), propose a method where patient symptoms are extracted from medical case sheets which are used to train a classification model. 507 patient case sheets were collected from Sugam Multispeciality Hospital, Kumbakonam, Tamil Nadu, India which was the data collection, phase. We processed these documents using maximum entropy techniques and tagging methods and extracted features using a customized stemmer for effective stroke type classification. Meschia et. al, [4], demonstrate the skill of machine learning in classifying stroke. Therefore, in this study, Artificial Neural Networks (ANN), Support Vector Machines (SVM), Boosting and Bagging techniques and Random Forests were created from the dataset. In these cases, the best result (accuracy of 95% with standard deviation of 14.69) was achieved with the ANN model trained via Stochastic Gradient Descent (SGD).

Harmsen et. al. (2006), highlight the need for faster stroke diagnosis. The proposed model in this study serves as the first line of defense for a simple and quick identification of a stroke case using imaging data in which Support Vector Machines (SVMs), Decision Trees, and Deep Learning Models, at its foundation, enable effective identification of stroke cases. Nwosu et. has introduced multiple machine learning architectures such as Random Forest, K-Nearest Neighbors (KNN), and Convolutional Neural Networks (CNNs) at diagnosing strokes through brain scans. Nwosu C.S et al., (2019) Models were trained and validated on a large dataset of labeled images of the brain to ensure robust model performance.

Pathan et.al, (2020), discuss how morphological operations and feature extraction improve stroke detection. These post-processing techniques help in fine-tuning stroke regions, ultimately boosting accuracy. Jeena et.al, (2016) explore the classification of stroke prediction models into four categories. A systematic review of research studies revealed Support Vector Machines (SVMs) as the most optimal model in 10 different studies, emphasizing their high accuracy in stroke detection.

Luk et.al, (2010) identify that most stroke-related research focuses on diagnosis, while fewer studies address treatment, revealing a research gap. Additionally, CT scans were found to be the most widely used dataset for stroke classification. Findings suggest that SVM and Random Forest models remain among the most efficient approaches in machine learning-based stroke detection. (Luk J.K et.al, 2006), proposed Stroke remains one of the leading causes of illness and mortality worldwide. The objective of this study is to explore effective methods to improve the accuracy of stroke classification and diagnosis. The Kaggle stroke dataset was used in this research, where preprocessed data significantly aids in enhancing patient outcomes. Strokes are broadly classified into ischemic and hemorrhagic types, and machine learning algorithms are employed to categorize individuals accordingly.

After a stroke, the brain dynamically reorganizes its functional networks to compensate for lost or impaired abilities. Previous studies based on static modularity analysis have shed light on overarching behavioral changes, but little is known about time-resolved reorganization of these networks. Resting-state functional MRI data were analyzed from 15 stroke patients (mild (n = 6); severe (n = 9)). A separate control group who were age-matched (n = 15) and healthy was also included as a comparison. In this context, the study employed a multilayer temporal network approach to discover time-sensitive modules and quantified dynamic network attributes such as recruitment, integration, and flexibility to give insight regarding post-stroke neural plasticity. By contrast, severe stroke patients showed less recruitment with greater inter-network connectivity, and mild stroke patients showed less flexibility and less inter-network integration. Interestingly, prior

static network analysis approaches did not identify such differences in network dynamics based on the severity. From a clinical perspective, these findings provide insight into how stroke-induced changes in posterior cortical connectivity impact motor, speech and cognitive function. This approach enables not only an assessment of normal brain activity but also predictions for future outcomes. This limitation is that although there were 15 stroke patients, and it is small, it is not generalizable in the future. And although dynamic modular analysis provides a fresh perspective, static techniques still yield critical insights into the overarching functional organization of global brain networks. Furthermore, focusing on dynamic metrics such as recruitment and integration may distract from other crucial components in recovery, such as individualized/targeted rehabilitation protocols and non-neural mechanisms of recovery.

3 PROPOSED METHODOLOGY

The steps involved in the methodology are: preparing the Data, Visualization, the prediction model and Integration with Django to make it user accessible. Step 1: Preprocessing: In this phase, the brain imaging data is cleaned and standardized to ensure data with the highest quality is utilized into machine learning algorithms. After this, data visualization tools will be used to create informative graphical representations that allow for better understanding of underlying patterns and distributions. Multiple machine learning algorithms will be used for more precise classification of stroke types and better prediction of subsequent functional outcomes. Finally, a Django coded web-based system will be used as UI interface to make predictions and view results. The application of this framework is packaged with the help of machine learning-based techniques to create a powerful predictive model that maximizes efficiency and effectiveness. Histograms, plots, and graphs are used to analyze data for effective visualization of trends. Several algorithms were tested and compared to identify the most accurate prediction model.

3.1 Data Pre-processing

This machine learning model's error rate, validation techniques are implemented, ensuring it aligns closely with the actual dataset error. If a dataset is comprehensive enough to depict the full population, validation may not be essential. However, in practical scenarios, data samples often fail to represent the entire dataset accurately. These techniques facilitate the identification of missing data, duplicate values, and the classification of numerical data types (integer or float). The validation process provides an objective measure of model performance on training data while adjusting hyperparameters for optimal results.

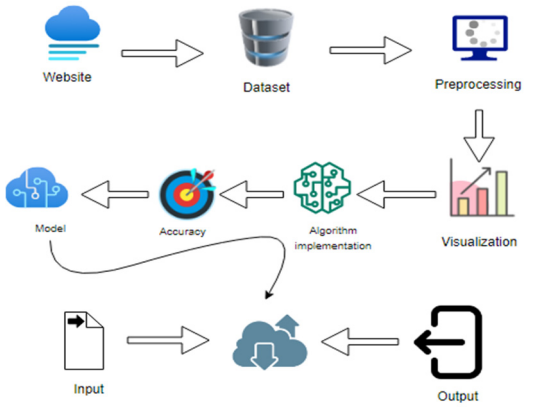

Figure 1: Proposed Architecture.

This Figure 1. Shows the evaluation tends to become less objective as reliance on the validation dataset grows during model configuration. The validation set plays a key role in assessing a model but is also frequently utilized for hyperparameter optimization. Data sourcing, examination, and improving its content, accuracy, and arrangement can be a labor-intensive process. Gaining insights into data properties early on assists in choosing the most effective algorithm for model implementation.

3.2 Data Validation

The essential library packages are imported, and the provided dataset is loaded. Variable analysis includes examining the dataset's structure, data types, and detecting missing or duplicate values. A validation dataset is separated from the training dataset to evaluate the model's predictive ability while adjusting its parameters. Effective techniques optimize the use of validation and test datasets during model evaluation. This phase is called data preparation where the dataset is renamed, useless field is dropped and even analysed with uni-variate, bi variate and multivariate methods. Since each dataset demands different methods of cleaning, the goal is to sort out inconsistencies and raise the quality of the data to make.

3.3 Comparative Study of Algorithms

In this proposed methodology 3 different algorithms are compared likely KNN, AdaBoost Classifier and Random Forest Classifier Each machine learning algorithm undergoes evaluation using the K-fold cross-validation approach, which ensures consistent data division by utilizing the same random seed. Before comparing different techniques, the Scikit-Learn library is installed to build the model. This package supports preprocessing, applies logistic regression for linear modeling, performs cross-validation with the K-Fold technique, utilizes random forests for ensemble learning, and employs decision trees for classification. The dataset is further divided into training and testing subsets to predict outcomes and analyze model accuracy effectively.

3.3.1 K-Nearest Neighbors (K-NN)

K-Nearest Neighbors (K-NN) is a widely used supervised learning algorithm that operates under the assumption that similar data points belong to the same category. Rather than developing a predictive model during training, K-NN retains the entire dataset and classifies new data based on proximity. It is applicable in both classification and regression but is mainly utilized for classification problems. As a non-parametric technique, K-NN does not impose constraints on data distribution. Additionally, it is a lazy learning algorithm, meaning it processes data only when a classification task arises. KNN algorithm measures similarity between points using distance metrics such as Euclidean and Manhattan distances. During training, feature data and their respective labels are stored. When predicting a new instance, K-NN computes distances to all existing points and selects k-nearest ones, where k is a user-specified hyperparameter. In classification, the most frequent class among these k neighbors is assigned, while in regression, the output is calculated as the average of their values. Selecting the right k-value is essential—too small a k leads to overfitting, while a large k results in excessive generalization. Cross-validation helps determine an optimal k. K-NN is highly dependent on proper data scaling, necessitating normalization or standardization for accurate distance measurements. Although it effectively models non-linear decision boundaries, the algorithm struggles in high-dimensional spaces due to sparse data distribution. In addition, K-NN is computationally very expensive for large sets of data as it is an exhaustive one, where distance calculations are performed for all the records computed. sklearn's implementation of K-NN is simple yet a powerful one, suitable for any machine learning applications.

This figure 2 shows the performance results of confusion matrix, cross validation test result, accuracy and hamming loss results for KNN classifier algorithm.

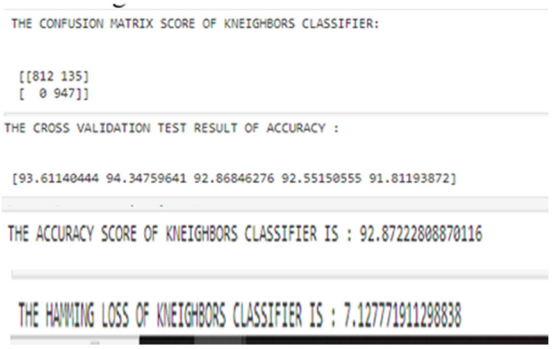

Figure 2: KNN Classifier Algorithmic Results.

3.3.2 AdaBoost Classifier Algorithm

The AdaBoost algorithm starts by training a weak learner on the full dataset. A weak learner is defined as a model that performs just slightly better than random selection. Initially, all training samples are given the same weight. After training the first weak learner, the algorithm calculates its error rate by comparing its predictions to the true labels. Misclassified instances are assigned greater significance, and the weak learner's influence is adjusted based on its performance—the fewer errors it makes, the greater its weight. This process is repeated, with each successive weak learner focusing more on challenging cases. The iterations continue for a set number of rounds or until the model reaches peak accuracy. The final AdaBoost model integrates multiple weak classifiers into a single, powerful predictor. Each weak learner contributes by making weighted predictions, where more accurate models hold greater voting power. AdaBoost's adaptability ensures that it progressively prioritizes difficult samples, thereby refining model accuracy. Key hyperparameters include the number of weak learners, the learning rate (which controls each learner's influence), and the choice of weak model. Because of its structure, AdaBoost has a lower risk of overfitting and generalizes well to unseen data. Though mainly designed for binary classification, AdaBoost can also be applied to regression tasks. It is widely utilized in fields such as handwriting recognition, spam filtering, and anomaly detection. Python's scikit-learn library provides an accessible

implementation of AdaBoost, making it an effective tool for model training and evaluation.

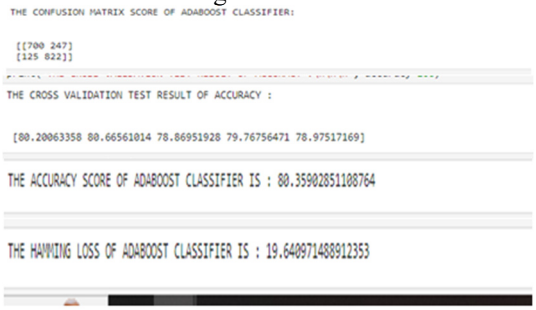

Figure 3: Ada Boost Classifier Algorithmic Results.

This figure 3 shows the performance results of confusion matrix, cross validation test result, accuracy and hamming loss results for AdaBoost classifier algorithm.

3.3.3 Random Forest Classifier Algorithm

The Random Forest Classifier is an ensemble algorithm that boosts model reliability by merging multiple decision trees. This approach enhances accuracy while limiting overfitting by combining the results of several independent trees. Each tree in the Random Forest is trained using a unique subset of data and features, ensuring model diversity. The algorithm applies Bootstrap Aggregation (bagging), where trees are trained on randomly sampled data with replacement. This technique allows certain instances to appear multiple times in the training set while others are excluded. Additionally, during the training process, only a limited number of features are chosen at each node-splitting stage, minimizing correlation between trees and improving predictive accuracy.

Each tree in the ensemble learns distinct characteristics of the data. Once trained, the individual predictions from all trees are aggregated to form the final decision. In classification problems, the model selects the class with the most votes, whereas in regression tasks, it computes the average prediction across all trees. Model accuracy is evaluated using the Out-of-Bag (OOB) error estimation, which measures performance based on the data excluded from training.

The effectiveness of the Random Forest model is influenced by several hyperparameters, including the number of trees, the number of features considered per split, the maximum allowable tree depth, and the minimum required samples for splitting nodes. Proper hyperparameter tuning ensures a balance between model complexity and accuracy. One of Random Forest's key strengths is its ability to generalize well by averaging predictions, reducing the risk of overfitting, and making it a suitable choice for large-scale and high-dimensional datasets.

Figure 4: Random Forest Classifier Algorithmic Results.

This figure 4 shows the performance results of confusion matrix, cross validation test result, accuracy and hamming loss results for Random Forest classifier algorithm.

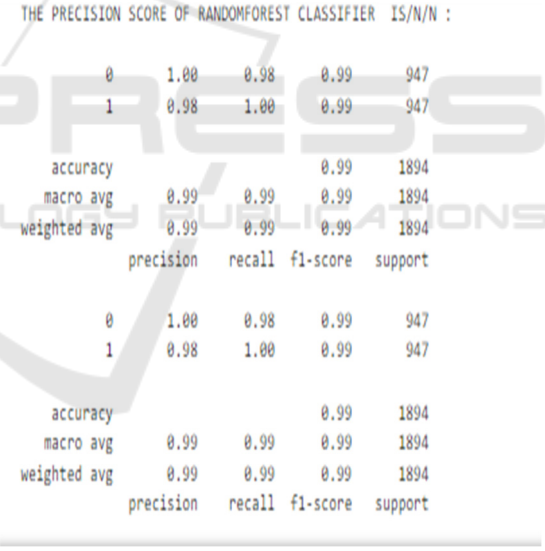

Figure 5. Accuracy Score of KNN, Adaboost, RF Algorithms.

This figure 5 shows the performance results accuracy results for KNN, AdaBoost, RF algorithms.

Based on the results of accuracy results of KNN, AdaBoost RF algorithms the website has been developed to describe how machine learning methods serve to forecast and minimize occurrences of brain strokes.

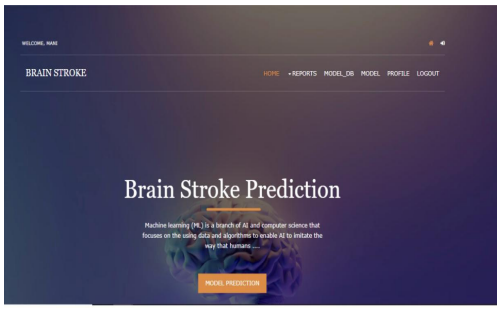

Figure 6: Design Page of Brain Stroke Prediction.

In figure 6, shows the webpage of front page. It is shown to evaluate technology methods for early stroke detection alongside prevention recommendations. Its primary objective is to develop patient data analysis tools which forecast stroke probabilities as well as enhance healthcare results and decrease stroke risk levels. Users should encounter input forms with spaces to provide their health information such as age and smoking status and medical background for predictive purposes. The website provides complete information about the stroke prediction algorithms including adaboost, KNN and random forest algorithms.

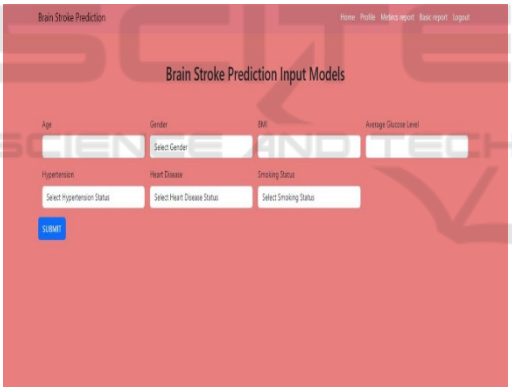

Figure 7: Design Page of Brain Stroke Prediction Input Models.

In figure 7, shows after registration of users, to use of the website can use the tool to input medical data and receive stroke risk estimates. The software demonstrates practical functionality to users by means of this feature. Users can find the details about all data used in model training including dataset origin, list of features along with data processing methods. Users receive customized prevention and risk reduction measures through the site depending on their predicted stroke risk after examination.

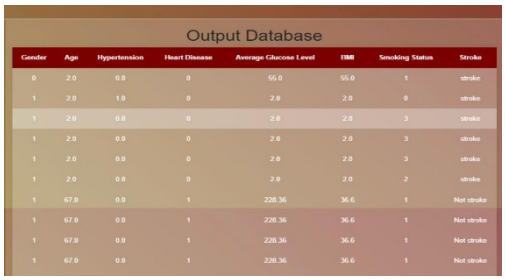

Figure 8: Design Page of Brain Stroke Prediction Output Database.

Users can access information about prediction model accuracy levels and performance enhancement procedures which include methods like hyper parameter tuning or cross-validation on the website.

In figure 8, The website presents information about ongoing system developments which include equipment updates and extra data collection features and performance improvement of prediction algorithms. The website contains details about the programming languages that power its construction along with the model development (Python, Tensor Flow, Flask, Django) and the website development technologies.

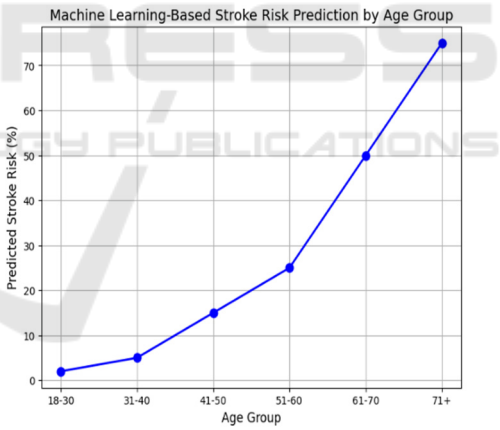

Figure 9: Predicting and Preventing Brain Strokes Based on Ages.

In figure 9 shows the graph results for predicting and preventing brain strokes based on different age groups and plot a hypothetical distribution of stroke risk predictions across various age groups. The graph would show how stroke prediction risk increases or varies with ages. The curve starts low in the younger age groups (18-30) and increases sharply as age progresses (71+), reflecting the higher risk of stroke in older individuals.

4 CONCLUSIONS

The summary of this project, the study of machine learning methodologies for stroke classification has revealed considerable potential in enhancing diagnostic reliability and patient care. By employing algorithms such as Support Vector Machines (SVMs) and Random Forests, this research highlights how these techniques can identify complex imaging patterns associated with stroke classification and severity. The results suggest that deep learning approaches and ensemble techniques surpass traditional classification methods, leading to increased accuracy. These findings indicate that integrating modern machine learning advancements into clinical applications could result in faster and more precise stroke diagnoses, ultimately improving patient treatment and management. Further research should focus on refining these models using larger datasets and extensive cross- validation to improve their robustness. Additionally, tackling biases and embracing real-world diversity will be crucial for the widespread adoption of these systems in various healthcare settings.

The future work focused on, supplementing stroke prediction models with insights from external datasets could lead to improved predictive performance. This project also intends to gather an institutional dataset to further evaluate and compare different machine learning techniques. As a part of our planned future work, also aim to conduct external validation of our proposed framework.

REFERENCES

Sivapalan G., Nundy K., Dev S., Cardiff B., Deepu J. ANNet: a lightweight neural network for ECG anomaly detection in IoT edge *sensorsIEEE Transactions on Biomedical Circuits and Systems (2)* (2022)

Koh H.C., Tan G., *et al.* Data mining applications in healthcareJ. Healthc. Inf. Manage., 19 (2) (2011), p. 65

Yoo I., Alafaireet P., Marinov M., PenaHernandez K., Gopidi R., Chang J.-F., Hua L. Data mining in healthcare and biomedicine: a survey of the literatureJ. Med. Syst., 36 (4) (2012), pp. 2431-2448

Meschia J.F., Bushnell C., BodenAlbala B., Braun L.T., Bravata D.M., Chaturvedi S., Creager M.A, Eckel R.H., Elkind M.S., Fornage M., *et al.* Guidelines for the primary prevention of stroke: a statement for healthcare professionals from the American heart association/American stroke association Stroke, 45 (12) (2014), pp. 3754-3832

Harmsen P., Lappas G., Rosengren A., Wilhelmsen L. Long-term risk factors for stroke: twenty-eight years of follow-up of 7457 middle-aged men in goteborg, swedenStroke, 37 (7) (2006), pp. 1663-1667

Nwosu C.S., Dev S., Bhardwaj P., Veeravalli B., John D. Predicting stroke from electronic health records, 2019 41st Annual International Conference of the *IEEE Engineering in Medicine and Biology Society (EMBC), IEEE (2019),* pp. 5704-5707

Pathan M.S., Jianbiao Z., John D., Nag A., Dev S. Identifying stroke indicators using rough *setsIEEE Access, 8 (2020)*, pp. 210318-210327

Jeena R.S., Kumar S. Stroke prediction using SVMProc. *International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT) (2016*), pp. 600-602, 10.1109/ ICCICCT.2016.7988020Hanifa S.-M., Raja-S K.

Stroke risk prediction through non-linear support vector classification modelsInt. J. Adv. Res. Comput. Sci., 1 (3) (2010)

Luk J.K., Cheung R.T., Ho S., Li L. Does age predict outcome in stroke rehabilitation? A study of 878 Chinese subjectsnCerebrovasc. Dis., 21 (4) (2006), pp. 229-234