Early Detection of Stroke Risk Through Intelligent Health Data Modeling and Machine Learning

R Venkatesh[1], K Dhanamjay [2], K Yatheendra[3]

[1]P.G Scholar, Department of MCA, Sri Venkatesa Perumal College of Engineering & Technology, Puttur, E-mail: ravvalavenkatesh725@gmail.com , ORC-ID: https://orcid.org/0009-0009-1581-9821

[2] Assistant Professor, Department of CSE(AI & ML), Sri Venkatesa Perumal College of Engineering & Technology, Puttur, E-mail: kanipakkamdhanamjay@gmail.com

[3]Assistant Professor, Department of CSE(AI & ML), Sri Venkatesa Perumal College of Engineering & Technology, Puttur, E-mail: k.yatheendra84@gmail.com, ORC-ID: https://orcid.org/0009-0003-1382-8587

To Cite this Article

R Venkatesh, K Dhanamjay, K Yatheendra, "Early Detection of Stroke Risk Through Intelligent Health Data Modeling and Machine Learning", Journal of Science Engineering Technology and Management Science, Vol. 03, Issue 04, April 2026,pp: 308-318, DOI: http://doi.org/10.64771/jsetms.2026.v03.i04.pp308-318

Submitted: 28-02-2026 Accepted: 01-04-2026 Published: 08-04-2026

Abstract: Cerebral stroke is a big global health problem that causes a lot of death and long-term disability. This shows how important it is to find people who are likely to have a stroke early on so that they can be prevented and treated quickly. The study uses Kaggle's Stroke Prediction Dataset, which has information like gender, age, high blood pressure, heart disease, glucose level, body mass index (BMI), and smoking status to figure out how likely someone is to have a stroke. After label encoding, one-hot encoding, and Random Forest Imputation are used to fix missing BMI values in the data, it is visualized using class distribution charts and correlation matrices. Normalizing the dataset with MinMax, Standard, and Robust scalers is done, and to fix class imbalance, oversampling methods like SMOTE, ADASYN, and Random oversampling are used. A number of machine learning and deep learning models are used to improve performance. These include Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest, Support Vector Machine, Gaussian Naive Bayes, 1D-CNN, GRU, LSTM, ANN, and BiRNN. To make the predictions even better, ensemble learning methods are looked into. A Voting classifier that combines XGBoost, Random Forest, and AdaBoost does the best, with 99.4% accuracy, precision, recall, F1-score, and ROC-AUC. Explainable AI techniques like LIME and SHAP are also used to figure out how important each trait is, and a web-based interface built on the Flask framework and integrating SQLite is created to let users make predictions in real time.

"Index Terms - *Stroke prediction, Machine learning, Ensemble learning, Deep learning, Explainable AI, Flask framework, Data preprocessing, Health informatics".*

1. INTRODUCTION

A stroke, also called a cerebrovascular accident, is one of the most common and deadly neurological conditions in the world. It happens when the brain's blood flow is cut off, either by ischemia or hemorrhage [1]. Millions of new cases are reported every year across all age groups, making it the top cause of death and long-term disability [9]. Survivors often have serious neurological problems, which puts a lot of physical, emotional, and financial stress on people and healthcare systems. Finding people who are at risk early on is therefore important for successful prevention, prompt medical intervention, and better recovery outcomes [2]. More recent progress in AI and data analytics has made it possible to use predictive models in healthcare systems. This could lead to better early diagnosis and risk stratification [3].

Having access to large public datasets like the Stroke Prediction Dataset from Kaggle [10] and more people using electronic health records has sped up the use of data-driven methods to figure out who is more likely to have a stroke based on age, hypertension, heart disease, glucose level, body mass index (BMI), and smoking status [5]. While traditional clinical assessment tools can be useful in some situations, they often lack the ability to be scaled up, to be automated, and to deal with complex, nonlinear feature interactions. Because of this, machine learning and deep learning algorithms have become strong ways to find hidden connections between health parameters and make accurate predictions [4, 6].

Even though a lot of progress has been made, current prediction systems still have major problems, such as unequal access for different classes, difficulty in interpreting results, and limited usability by people who aren't experts. A lot of the tools that are already out there can only be used in labs or institutions, which makes them less useful in the real world [7]. Also, the fact that many deep learning models are not clear makes it harder for both doctors and patients to trust and understand them. Recently, explainable artificial intelligence (XAI) techniques like SHAP and LIME have gotten a lot of attention because they can explain model choices, which makes them more clear and easier for doctors to accept [6].

The study's goal is to create a framework for assessing stroke risk that is easy to understand, accurate, and available to everyone. The framework will use improved machine learning and deep learning models. The system uses ensemble learning and XAI to make accurate predictions and easy-to-understand visualizations through a web-based platform built with the Flask framework. This helps with evaluating health risks in real time and makes intelligent healthcare systems better [8].

2. RELATED WORK

In the past few years, advances in artificial intelligence, data analytics, and computer healthcare systems have brought a lot of attention to finding and predicting strokes. Several research studies have looked at how machine learning (ML) and deep learning (DL) can help find early signs of stroke, figure out risk factors, and make more accurate and understandable predictions about how patients will do. Putting together clinical, behavioral, and demographic data into predictive models has shown that it can change how early detection and preventive healthcare are done.

Several studies have used standard machine learning methods to predict the outcome of a stroke, showing that they can make accurate predictions using structured clinical datasets. Raja et al. [11] created a system for predicting strokes using algorithms like Random Forest (RF), Support Vector Machine (SVM), and Decision Tree (DT). They pointed out that RF performed better because it can handle complex and nonlinear relationships between features. In the same way, Shobayo et al. [12] used the Random Forest algorithm on demographic and behavioral datasets to predict the occurrence of a stroke, focusing on the role of lifestyle factors like smoking, not being active, and high blood pressure. Their results showed that RF did better than other models in terms of accuracy and recall, which means it can be used on medical datasets with classes that aren't balanced.

Deep learning-based studies have also shown promise in improving the early identification and classification of strokes. Researchers Al-Mekhlafi et al. [13] did a study comparing ML and DL models for finding stroke and bleeding early on. They found that deep neural networks (DNNs) were more accurate and sensitive than regular ML methods. The study showed that convolutional architectures have the ability to automatically pull out complex spatial patterns from biomedical images. This could lead to better diagnostic support. Similarly, Tusher et al. [14] looked into machine learning-based early brain stroke prediction methods and confirmed that ensemble learning methods could greatly improve diagnosis accuracy by lowering the number of false positives.

For preventive healthcare uses, ML has also been looked at as a way to be added to real-time warning systems. Malini et al. [15] created a sophisticated system for finding and alerting strokes that uses machine learning techniques to look at physiological factors and send out early warnings. Their system used IoT-enabled devices to keep an eye on patients all the time, which made early action more likely. These kinds of systems fill the void between old-fashioned medical exams and smart digital health tracking.

In the field of neurorehabilitation, Kim et al. [16] looked into the microstructure of the corticospinal tract as a biomarker for predicting muscle recovery after a long-term stroke. By using scanning data along with machine learning models, they showed that certain measures of white matter integrity could accurately predict how well stroke patients would recover from their injuries. This method stresses the growing importance of machine learning not only in diagnosis but also in predicting how well therapy will go, which helps doctors make more personalized recovery plans.

Also, Kashi et al. [17] suggested a machine learning model that could find compensatory movements that stroke patients make while they are doing rehabilitation tasks. Using motion capture data, their system automatically found movement deviations. This made automated feedback easier and reduced the need for human control. This new idea shows how machine learning can improve therapy and patient tracking by making things more automated and accurate.

Predicting how well a treatment will work is another area where deep learning is used. Bacchi et al. [18] used deep neural networks to guess how thrombolysis would affect the functional results of people who had an ischemic

stroke. The results showed that DL models, especially convolutional neural networks (CNNs), were better at using medical image data to predict how patients would do after treatment than traditional statistical models. This showed that DL could help people make decisions about how to treat an acute stroke by judging how well a treatment will work before it is given.

Sirsat et al. [19] did a thorough study of many machine learning methods for predicting and finding brain strokes. They compiled the latest developments in feature selection, data preprocessing, and ensemble methods. The study stressed that combining imaging data with clinical and lifestyle factors makes predictions much more accurate. But it also brought up problems like uneven data, overfitting, and the need to be able to explain things to gain professional trust.

This last study by Shoily et al. [20] compared different machine learning methods for finding strokes. They looked at models like Logistic Regression, Naïve Bayes, and Random Forest. Based on their research, they found that while Random Forest was good at making predictions, hybrid and ensemble methods could be even more accurate and reliable. They also said that making the data easier to understand and dealing with missing or noisy data are very important for deployment to work in clinical situations.

3. MATERIALS AND METHODS

The suggested system includes a full machine learning-based prediction framework created to figure out how likely it is that a person will have a brain stroke by looking at organized clinical and behavioral factors. The Stroke Prediction Dataset from Kaggle is used, which includes important factors like age, gender, high blood pressure, heart disease, glucose level, body mass index (BMI), and smoking status. Label encoding, one-hot encoding, and Random Forest Imputation are some of the things that are done to prepare the data before it is normalized and class-balanced using SMOTE and ADASYN methods. Grid Search optimization uses many methods, like Random Forest, SVM, and deep learning architectures like LSTM and BiRNN, to improve accuracy and generalization. This method focuses on being easy to understand by using AI-based feature analysis that can be explained. This helps us learn more about how health issues like high blood pressure, glucose metabolism, and obesity make people more likely to have a stroke [22, 23, 27].

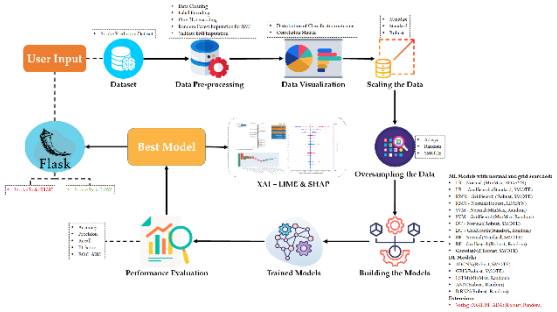


Fig.1 Proposed Architecture

Figure 1 shows a workflow for using a web tool to guess strokes using AI. The patient starts by entering information about their living into the tool. Based on their habits, our model then looks at this information to guess how likely it is that they will have a stroke. After seeing the model's result, the patient decides what to do next. Case 1: If the model says "Stroke likely," the person should see a doctor right away for a checkup. Case 2: If the model says "Stroke Unlikely," the person should only see a doctor if they have symptoms.

i) Dataset Collection:

The study uses the Stroke Prediction Dataset from Kaggle, which has 5,110 patient records with information about their gender, age, hypertension, heart disease, marital status, type of job, type of residence, average glucose level, BMI, smoking status, and the number of strokes they had. This dataset provides a wide range of samples that are fair and accurate, which is important for creating accurate models that can predict who will have a stroke. Each factor gives important information about the complex nature of stroke, since risk factors like high blood pressure, diabetes, and obesity are highly linked to cerebrovascular disorders [24]. The structured nature of the information makes it possible to look at both medical and behavioral factors that affect the risk of having a stroke. Figure 2 shows how the attributes in the Stroke Prediction Dataset used in this study are spread out and organized.

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

Fig.2 Dataset Collection

ii) Pre-processing:

During preprocessing, unstructured health data is turned into a format that can be used for predictive models. It makes sure that the data is correct, consistent, and reliable by filling in missing values, encoding categorical traits, and leveling scales. In this study, preprocessing improves the quality of the dataset by encoding, correction, scaling, and balancing. These steps are necessary to make learning more efficient and improve the performance of stroke prediction models.

***a) Data Processing*:** A series of steps called "refinement" are used in data processing to get the information ready for model training. Label Encoding is used to turn categorical factors like gender, marital status, and type of residence into numeric variables that can be used by any machine. One-Hot Encoding is used to avoid ordinal bias for nominal traits like the type of work and whether or not someone smokes. Random Forest Imputation fills in missing BMI numbers by guessing and replacing them based on how they relate to other data points. This makes the data more accurate. The assumed BMI numbers are checked to make sure they are statistically consistent with the original distributions. This all-encompassing processing approach reduces noise, makes features easier to understand, and makes sure that the dataset is balanced so that a strong model can be trained for predicting stroke risk.

***b) Data Visualization*:** Data visualization is used to look at the organization, distribution, and relationships between features in a dataset. Visualizing the results of the classification shows that there is an imbalance between the number of stroke and non-stroke cases, which makes it clear that resampling methods are needed. Histograms and box plots show how things like glucose level, BMI, and age change over time, helping you find trends and possible outliers. The correlation matrix shows how variables are related to each other and which health factors are most important for lowering the chance of stroke. High connections between high blood pressure, glucose levels, and age show how well they can predict what will happen. These visual studies help choose which features to use, how to preprocess them, and how to make the model work best. This makes sure that the predictive framework finds the important connections that are needed for a correct assessment of stroke risk and easy understanding.

***c) Scaling the Data*:** By putting all the numbers into a single range, scaling makes sure that they all add the same amount to training the model. Three types of scaling are used in this study: MinMax, Standard, and Robust. Normalizing data within a [0,1] range is possible with MinMax scaling, which makes it good for methods that need to handle changes in magnitude. Standard scaling makes features equal so that they have a mean of zero and a range of one. This helps distance-based models like SVM and KNN converge faster. By putting data in the middle of the median and interquartile range, robust scaling lessens the effect of outliers. By using these methods, you can make sure that the numbers stay stable and that gradient optimization works well in machine learning and deep learning models. This improves the accuracy of predictions and performance across a wide range of health factors related to stroke risk.

***d) Oversampling the Data*:** The Stroke Prediction Dataset has a class imbalance because there are a lot fewer strokes than non-strokes. To make sure that training is fair, oversampling methods are used. The SMOTE, ADASYN, and Random Oversampling methods are used in this work. SMOTE creates fake minority samples by interpolating real ones, which increases the variety of classes. ADASYN improves this even more by creating synthetic data that changes based on the difficulty of the class and focused on samples that are harder to learn. Random Oversampling copies current minority class instances to make the distribution more even. These methods stop model bias toward majority classes and make it easier to remember and generalize minority stroke estimates. Using these methods to get balanced datasets makes learning more fair, cuts down on classification mistakes, and makes it easier for the model to find people who are at high risk.

iii) Train & Test:

The dataset is split into training and testing subsets in an 80:20 split to make sure that building models and judging their success are both fair. The training set, which is made up of 80% of all the data, is used to find trends and correlations between factors that are linked to stroke risk. The last 20% is new data that is used to test how well the learned model can adapt to new situations. This split keeps the model from overfitting by making sure it doesn't just use trends it has learned from the training data. A lot of medical prediction studies use the 80:20 split because it strikes the best balance between having enough learning data and having enough testing coverage. This makes sure that the model is evaluated fairly and accurately so that it can be used in the real world.

iv) Algorithms:

Logistic Regression is a basic classification algorithm that uses a logistic function to describe the chance of having a stroke. It shows how health indicators are linked to desired results in a straight line. The model uses MinMax scaling and ADASYN oversampling to make sure that learning is fair across classes and that convergence is stable, which makes predictions more accurate. Because it is easy to understand and works well, it is a good starting point for testing how well more complicated models can tell the difference between things.

The equation predicts class probabilities using a logistic sigmoid function.

$$\hat{y}_i = \sigma(w^T x + b) = \frac{1}{1 + e^{-(w^T x + b)}} \qquad (1)$$

Systematic hyperparameter tuning is used in logistic regression optimized through grid search to improve performance and classification limits. The model evens out feature ranges and reduces class mismatch with Standard scaling and SMOTE oversampling. This setup improves coefficient estimation, making it more stable, robust, and generalizable while still being easy to understand across a range of data distributions.

The K-Nearest Neighbors algorithm sorts cases into groups based on how close they are to labeled samples in the feature space. Grid Search optimization tweaks things like the size of the neighborhoods and the distance between them. When combined with Robust scaling and SMOTE oversampling, it makes the system more resistant to errors and better at handling uneven data by making sure that local patterns are recognized correctly.

In this setup, KNN uses distance-based classification with the help of Robust scaling and ADASYN oversampling. It makes neighborhood-based decision boundaries stronger by creating fake cases in minority groups. The method does a good job of capturing non-linear relationships, and robust scaling stops distortions caused by extreme values. This makes the classification more stable and fair.

$$distance(x, X_i) = \sqrt{\sum_{j=1}^{d} \left(x_j - X_{i_j}\right)^2} \qquad (2)$$

Support Vector Machine creates the best dividing hyperplanes to increase class margins, which makes binary classification work well. Using MinMax scaling makes sure that the features are all the same, and Random oversampling makes sure that the representation of each class is fair. The configuration improves margin precision and reduces overfitting, which makes performance reliable on complicated data structures with many dimensions.

The Objective Function for Soft Margin SVM equation given below:

$$minimize\ \frac{1}{2}\ ||W||^2 + C\sum_{i=1}^{n} \xi_i \qquad (3)$$

SVM that has been improved with Grid Search carefully adjusts kernel parameters and regularization values to achieve better boundary separation. Random oversampling lowers bias toward dominant classes, and MinMax scaling keeps feature uniformity. This setup makes generalization and accuracy better, especially when it comes to finding small differences between feature groups that are closely linked.

To guess what will happen in a categorical situation, decision tree learning divides data into hierarchical layers based on feature limits. Robust scaling makes the model less sensitive to outliers, and SMOTE fixes class imbalance by making minority samples more common. The final model makes the data easier to understand and more accurate at classifying it, effectively capturing complicated decision patterns in structured health data.

$$I\ (i) = 1 - \sum_{i=1}^{k} p_i^2 \qquad (4)$$

To keep things from fitting too well, Decision Tree with Grid Search uses hyperparameter optimization for depth, split criteria, and minimum samples per leaf. Standard scaling makes sure that all features are the same, and random oversampling evens out the spread of samples. This setup improves the stability of decisions and the accuracy of predictions, which makes classification more robust and easier to understand.

Random Forest uses ensemble averaging to join several Decision Trees to improve the accuracy of classification and lower the variance. It makes sure that everyone learns the same way by using Standard scaling and SMOTE oversampling to balance how inputs are represented. The algorithm is very good at generalization, better at figuring out how important a trait is, and strong against noise and overfitting.

The Gini Equation given below:

$$Gini = 1 - \sum_{i=1}^{C} (Pi)^2 \qquad (5)$$

Random Forest improved by Grid Search uses split criteria and optimized tree configurations to get better results from the whole group. Outliers have less of an effect when robust scaling is used, and random oversampling makes the distribution of classes similar. By adding up different tree results to lower model uncertainty, the model becomes more accurate, consistent, and easy to understand.

The Gaussian Naïve Bayes method uses statistical reasoning based on the Gaussian distribution to sort data into groups where each group is conditionally independent. Robust scaling reduces distortion caused by extreme values, and SMOTE creates fake cases of minorities. The method is fast, gives statistical results that are easy to understand, and works reliably on datasets that are only slightly unequal.

The 1D Convolutional Neural Network uses convolutional filters to find patterns in a small area of space. This lets the network automatically learn new features. The sources are standardized by robust scaling, and the representation of classes is balanced by SMOTE. This deep learning method makes it easier to describe features in a hierarchical way and makes predictions more accurate by quickly finding patterns in sequential numerical attributes.

The Gated Recurrent Unit handles sequential dependencies by keeping important temporal information by using gates. Stable input sizes are guaranteed by robust scaling, and SMOTE makes class variety better. GRU improves the efficiency of dynamic learning by lowering disappearing gradients and making accurate predictions from feature sequences that change over time or are correlated.

Gated cells control how long memories are kept in Long Short-Term Memory networks, which can pick up on long-range connections. Random oversampling lessens the affects of imbalance, and MinMax scaling makes features more even. The model learns complex temporal relationships and keeps gradient propagation stable, which leads to better generalization and consistent performance across data that is organized in a sequential way. The equation below represents the hidden state update in LSTM.

$$h_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \cdot \tanh(C_t) \qquad (6)$$

The Artificial Neural Network uses many layers that are all linked to show how features are related in a way that is not linear. Random oversampling fixes class imbalance, and robust scaling evens out the spread of the data. The network improves its adaptability, robustness, and accuracy by changing its weights over and over again. This makes it better at making predictions across a wide range of data trends.

The equation below represents the output prediction in neural networks.

$$\hat{y} = \sigma(w^L \cdot a^{L-1} + b^L) \qquad (7)$$

The Bidirectional Recurrent Neural Network can handle sequences going both forward and backward, which lets it learn about the whole temporal context. Random oversampling makes sure that training is fair, and robust scaling keeps the numbers that are fed in stable. This two-way flow makes it easier to see how context affects predictions and makes predictions more accurate across both ordered and linked datasets.

The Voting Classifier uses majority or weighted voting to combine results from XGBoost, Random Forest, and AdaBoost, which are three base learners. Random oversampling improves representativeness, and robust scaling makes sure that inputs are treated consistently. The ensemble integration makes predictions more stable, accurate, and useful by using the best parts of each method that work together.

The equation below represents the majority voting process in classifiers.

$$\hat{y} = argmax_c \left(\sum_{i=1}^{n} II\ (\hat{y}_i = c) \right) \qquad (8)$$

v) Integration of XAI and Flask Framework:

Explainable Artificial Intelligence (XAI) is added to the suggested voting-based ensemble structure to make stroke vulnerability predictions easier to understand and more open. The system uses XAI techniques like LIME and SHAP to find and show the most important factors that affect each forecast made by the three models working together (XGBoost, Random Forest, and AdaBoost). This integration helps doctors understand why predictions are made, which builds trust and makes it easier to make clinical choices based on data.

The Flask framework is used to make a simple, interactive online interface for predicting things in real time. It links the trained model to an easy-to-use front end so that users can enter health information and get instant

estimates of their risk of having a stroke. The combination of Flask and SQLite makes it possible to handle data quickly and launch the predictive system without any problems.

4. RESULTS AND DISCUSSIONS

Accuracy: How well a test can tell the difference between sick and healthy people is called its accuracy. To get an idea of how accurate a test is, we should figure out what percentage of cases are true positives and true negatives. In terms of math, this can be written as

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (9)$$

Precision: Precision is the percentage of correctly classified cases or samples compared to those that were correctly classified as positives. So, here is the method to figure out the precision:

$$Precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} (10)$$

Recall: In machine learning, recall is a metric that shows how well a model can find all the important instances of a certain class. It shows how well a model captures instances of a certain class. It is calculated by dividing the number of correctly predicted positive observations by the total number of real positives.

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (11)$$

F1-Score: The F1 score is a way to rate the correctness of a machine learning model. It takes a model's accuracy and recall scores and adds them together. The accuracy metric counts how many times, across the whole dataset, a model made a correct guess.

$$F1\ Score = 2 * \frac{Recall\ \text{X}\ Precision}{Recall + Precision} * 100 \quad (12)$$

AUC-ROC Curve: The AUC-ROC Curve shows how well a classification problem is solved at different benchmark levels. The True Positive Rate is plotted against the False Positive Rate by ROC. AUC measures how well the model can tell the difference between classes; a higher AUC means the model works better.

$$AUC = \sum_{i=1}^{n-1} (FPR_{i+1} - FPR_i) \cdot \frac{TPR_{i+1} + TPR_i}{2} \quad (13)$$

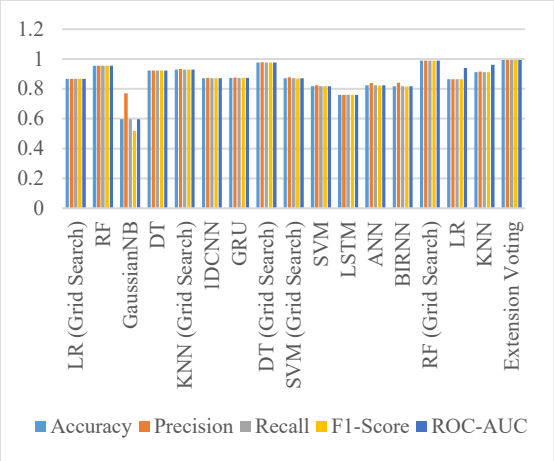
Table.1 Performance Evaluation

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
LR (Grid Search)	0.867	0.867	0.867	0.867	0.867
RF	0.954	0.954	0.954	0.954	0.954
GaussianNB	0.596	0.771	0.596	0.518	0.596
DT	0.922	0.922	0.922	0.922	0.922
KNN (Grid Search)	0.930	0.933	0.930	0.930	0.930
1DCNN	0.871	0.874	0.871	0.871	0.871
GRU	0.874	0.876	0.874	0.874	0.874
DT (Grid Search)	0.977	0.978	0.977	0.977	0.977
SVM (Grid Search)	0.871	0.877	0.871	0.870	0.871
SVM	0.818	0.825	0.818	0.818	0.818
LSTM	0.760	0.760	0.760	0.760	0.760
ANN	0.825	0.840	0.825	0.823	0.825
BIRNN	0.818	0.841	0.818	0.815	0.818
RF (Grid Search)	0.989	0.989	0.989	0.989	0.989
LR	0.864	0.864	0.864	0.864	0.941
KNN	0.912	0.917	0.912	0.912	0.962

Extension Voting	**0.994**	**0.994**	**0.994**	**0.994**	**0.994**

Table 1 shows how well different machine learning and deep learning models predict the risk of having a stroke.

Fig.3 Comparison Graph



Accuracy (blue), Precision (orange), Recall (gray), F1-Score (yellow), and ROC-AUC (dark blue) are the performance measures that are shown in Figure 3.

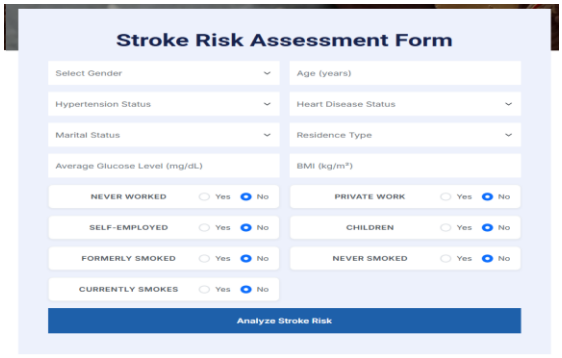


Fig.4 Stroke Risk Assessment Form

In Fig. 4, the interface shows a Stroke Risk Assessment Form that users fill out with medical, demographic, and lifestyle information to figure out how likely it is that they will have a stroke.

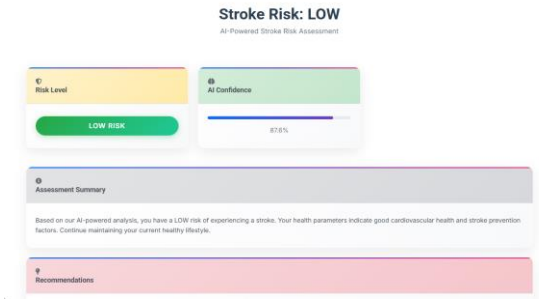


Fig.5 Prediction Results – Positive

Figure 7 shows the result interface, which shows that the expected stroke risk is "Low," which means that there is a very small chance that the stroke will happen based on the factors that were looked at.

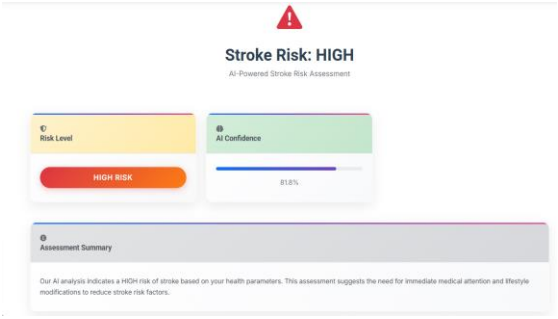


Fig.6 Prediction Results – Negative

Figure 6 shows the result interface, which shows that the predicted stroke risk is "High," which means that there is a much higher chance of having a stroke based on the health signs that were looked at.

5. CONCLUSION

The system successfully shows how machine learning and deep learning can accurately predict a person's risk of having a brain stroke using the Stroke Prediction Dataset from Kaggle. This dataset includes important health indicators like age, gender, hypertension, heart disease, glucose level, BMI, and smoking status. A lot of work went into preprocessing the data, including label encoding, one-hot encoding, and Random Forest Imputation for BMI. Next, correlation matrices and distribution plots were used to see how the features were related. Using MinMax, Standard, and Robust feature scalers along with oversampling techniques like SMOTE, ADASYN, and Random oversampling fixed the problem of uneven data and made sure the model was consistent. Grid Search was used to learn and improve many algorithms for better accuracy. These included Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest, Support Vector Machine, Gaussian Naive Bayes, 1D-CNN, GRU, LSTM, ANN, and BiRNN. To get even better results, ensemble learning was used with a Voting classifier that mixed XGBoost, Random Forest, and AdaBoost. This achieved amazing accuracy, precision, recall, F1-score, and ROC-AUC of 99.4%. Using Explainable AI techniques like LIME and SHAP made model interpretation clear, and the Flask-based web interface with SQLite integration let users interact with the system and see their predicted stroke risk in real time. This made the system very useful for preventive healthcare because it was effective, easy to understand, and easy to use.

In the future, this system will be able to make better predictions by adding bigger and more varied medical datasets, such as genomic data for personalized analysis and real-time patient tracking. Adding advanced ensemble learning and deep hybrid architectures could make the system even more accurate and useful for a wide range of groups. The web-based interface can be turned into a mobile app so that more people can use it and healthcare can be integrated from afar. Real-world deployment with cloud-based data pipelines, ongoing model retraining, and advanced Explainable AI frameworks will also improve the ease of clinical interpretation, help with decision-making, and widespread use of digital health environments.

REFERENCES

[1] Sarkar, M. M. R., & Sarkar, P. (2025). Brain Stroke Prediction Using Machine Learning.

[2] Alsieni, M., & Alyoubi, K. H. (2025). Artificial intelligence with feature fusion empowered enhanced brain stroke detection and classification for disabled persons using biomedical images. Scientific Reports, 15(1), 29224.

[3] Uddin, K. M. M., Chowdhury, A., Druvo, M. M. R., Islam, M. S., & Uddin, M. A. (2025). Web-Based Early Dementia Detection Using Deep Learning, Ensemble Machine Learning, and Model Explainability Through LIME and SHAP. IET Software, 2025(1), 5455082.

[4] Ganesh, B. R., B M, P., Prasad K, K., Swapna, G., & G, Viswanath. (2025). Data Mining-Driven Multi-Feature Selection for Chronic Disease Forecasting. Journal of Neonatal Surgery, 14(5S), 108–124. https://doi.org/10.52783/jns.v14.1993

[5] Abdi, A. M., Abdi, M. A., Isak, M. M., Omar, M. A., Ali, M. A., & Ahmed, B. A. (2024, December). Web-Based Brain Stroke Prediction System Using Machine Learning Classifiers. In 2024 International Conference on Recent Progresses in Science, Engineering and Technology (ICRPSET) (pp. 1-7). IEEE.

[6] K. Moulaei, L. Afshari, R. Moulaei, B. Sabet, S. M. Mousavi, and M. R. Afrash, "Explainable artificial intelligence for stroke prediction through comparison of deep learning and machine learning models," *Scientific Reports*, vol. 14, no. 1, p. 31392, 2024.
[7] R. Tehseen, U. Omer, R. Javaid, M. Mehr, M. Yousaf, and A. Zaheer, "Prediction of brain stroke using federated learning," *International Journal of Innovative Science and Technology*, vol. 6, pp. 1995–2013, 2024.
[8] N. Patil and A. Sumarsono, "Stroke prediction using machine learning," *Journal of Research in Engineering and Computer Science*, vol. 2, no. 1, pp. 61–72, 2024.
[9] National Institute of Neurological Disorders and Stroke (NINDS), "Stroke," May 29, 2024. [Online]. Available: https://www.ninds.nih.gov/health-information/disorders/stroke
[10] Kaggle, "Stroke Prediction Dataset," May 29, 2024. [Online]. Available: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset
[11] Swapna, G., Sreenivasulu, K., Deepika, M., Baseer, K. K., Neerugatti, V., & Viswanath, G. (2025). Brain tumour detection using MRI images in CNN. Advances in Science, Engineering and Technology, 6.
[12] O. Shobayo, O. Zachariah, M. O. Odusami, and B. Ogunleye, "Prediction of stroke disease with demographic and behavioural data using random forest algorithm," *Analytics*, vol. 2, no. 3, pp. 604–617, 2023.
[13] Z. G. Al-Mekhlafi *et al.*, "Deep learning and machine learning for early detection of stroke and haemorrhage," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 775–796, 2022.
[14] A. N. Tusher, M. S. Sadik, and M. T. Islam, "Early brain stroke prediction using machine learning," in *Proc. 11th Int. Conf. System Modeling & Advancement in Research Trends (SMART)*, pp. 1280–1284, IEEE, 2022.
[15] T. Malini, M. Deepalakshmi, B. Dhivyaa, P. Karthikeswari, and N. Kavipriya, "Advanced stroke detection and alert system using machine learning," in *Proc. 7th Int. Conf. Communication and Electronics Systems (ICCES)*, pp. 1084–1089, IEEE, 2022.
[16] B. Kim, N. Schweighofer, J. P. Haldar, R. M. Leahy, and C. J. Winstein, "Corticospinal tract microstructure predicts distal arm motor improvements in chronic stroke," *Journal of Neurologic Physical Therapy*, vol. 45, no. 4, pp. 273–281, 2021.
[17] S. Kashi, R. F. Polak, B. Lerner, L. Rokach, and S. Levy-Tzedek, "A machine-learning model for automatic detection of movement compensations in stroke patients," *IEEE Trans. Emerging Topics in Computing*, vol. 9, no. 3, pp. 1234–1247, 2021.
[18] S. Bacchi, T. Zerner, L. Oakden-Rayner, T. Kleinig, S. Patel, and J. Jannes, "Deep learning in the prediction of ischaemic stroke thrombolysis functional outcomes," *Academic Radiology*, vol. 27, no. 2, pp. e19–e23, 2020.
[19] M. S. Sirsat, E. Fermé, and J. Câmara, "Machine learning for brain stroke: A review," *J. Stroke Cerebrovascular Diseases*, vol. 29, no. 10, p. 105162, 2020.
[20] T. I. Shoily, T. Islam, S. Jannat, S. A. Tanna, T. M. Alif, and R. R. Ema, "Detection of stroke disease using machine learning algorithms," in *Proc. 10th Int. Conf. Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–6, IEEE, 2019.
[21] G Loge, T Sunil Kumar Reddy, G Swapna, & G Viswanath. (2025). Interpretable AI for Precision Brain Tumor Prognosis: A Transparent Machine Learning Approach. In International Journal of Health Sciences and Pharmacy (IJHSP) (Vol. 9, Number 1, pp. 180–195). Zenodo. https://doi.org/10.5281/zenodo.15523628
[22] M. J. Cipolla, D. S. Liebeskind, and S.-L. Chan, "The importance of comorbidities in ischemic stroke: Impact of hypertension on the cerebral circulation," *J. Cerebral Blood Flow & Metabolism*, vol. 38, no. 12, pp. 2129–2149, 2018.
[23] M. J. Haley and C. B. Lawrence, "Obesity and stroke: Can we translate from rodents to patients?," *J. Cerebral Blood Flow & Metabolism*, vol. 36, no. 12, pp. 2007–2021, 2016.
[24] J. Wang, X. Ning, L. Yang, J. Tu, H. Gu, C. Zhan, W. Zhang, and T.-C. Su, "Sex differences in trends of incidence and mortality of first-ever stroke in rural Tianjin, China, from 1992 to 2012," *Stroke*, vol. 45, no. 6, pp. 1626–1631, 2014.
[25] S. Zhang, W. Zuo, X.-F. Guo, W.-B. He, and N.-H. Chen, "Cerebral glucose transporter: The possible therapeutic target for ischemic stroke," *Neurochemistry International*, vol. 70, pp. 22–29, 2014.
[26] A D Venkatesh, K Bhaskar, G Swapna, & G Viswanath. (2025). Advanced Hybrid Learning Architecture for Precision Cardiovascular Risk Assessment. In International Journal of Health Sciences and Pharmacy (IJHSP) (Vol. 9, Number 1, pp. 50–61). Zenodo.

[27] G. Faraco and C. Iadecola, "Hypertension: A harbinger of stroke and dementia," *Hypertension*, vol. 62, no. 5, pp. 810–817, 2013.
[28] D. A. Brenner, R. M. Zweifler, C. R. Gomez, B. M. Kissela, D. Levine, G. Howard, B. Coull, and V. J. Howard, "Awareness, treatment, and control of vascular risk factors among stroke survivors," *J. Stroke Cerebrovascular Diseases*, vol. 19, no. 4, pp. 311–320, 2010.
[29] R. L. Sacco, B. Boden-Albala, R. Gan, X. Chen, D. E. Kargman, S. Shea, M. C. Paik, and W. A. Hauser, "Stroke incidence among white, black, and hispanic residents of an urban community: The Northern Manhattan Stroke Study," *Am. J. Epidemiology*, vol. 147, no. 3, pp. 259–268, 1998.
[30] H. Jørgensen, H. Nakayama, H. O. Raaschou, and T. S. Olsen, "Stroke in patients with diabetes: The Copenhagen stroke study," *Stroke*, vol. 25, no. 10, pp. 1977–1984, 1994.