**Innovations in Stroke Identification: A Machine**

**Learning-Based Diagnostic Model**

**Using Neuroimages**

*Report submitted to SASTRA Deemed to be University As per the requirement for the course*

### CSE300: MINI PROJECT

*Submitted by*

#### VARIKUTI PRUTHVINATH REDDY

**(Reg. No.: 126003221, B. Tech Computer Science and Engineering)**

**RUDRA DHANUSH**

**(Reg. No.: 126003221, B. Tech Computer Science and Engineering)**

**NELLI LAKSHMI SRINIVAS**

**(Reg. No.: 126003181, B. Tech Computer Science and Engineering)**

# MAY 2025



## SCHOOL OF COMPUTING

#### THANJAVUR, TAMIL NADU, INDIA – 613 401



**SCHOOL OF COMPUTING THANJAVUR – 613 401**

##### Bonafide Certificate

This is to certify that the report titled **“Innovation in Stroke Detection: A Machine- Based Diagnostic Model Using Neuroimages”** submitted as a requirement for the course, **CSE300 : MINI PROJECT** for B.Tech. is a bonafide record of the work done by **Mr. VARIKUTI PRUTHVINATH REDDY** (Reg. No.: 126003285, B. Tech Computer Science and Engineering), **Mr. RUDRA DHANUSH** (Reg. No.: 126003221, B. Tech Computer Science and Engineering) and **Mr. NELLI LAKSHMI SRINIVAS** (Reg. No.: 126003181, B. Tech Computer Science and Engineering) during the academic year 2024-25, in the School of Computing, under my supervision.

##### Signature of Project Supervisor :

**Name with Affiliation :** Dr.Senthil Selvan.N, AP- III, SOC

**Date :** 17.04.2025

Mini Project *Viva voice* held on

**Examiner 1 Examiner 2**

### ACKNOWLEDGEMENTS

We would like to thank our Honorable Chancellor **Prof. R. Sethuraman** for providing us with an opportunity and the necessary infrastructure for carrying out this project as a part of our curriculum.

We would like to thank our Honorable Vice-Chancellor **Dr. S. Vaidhyasubramaniam** and **Dr. S. Swaminathan**, Dean, Planning & Development, for the encouragement and strategic support at every step of our college life.

We extend our sincere thanks to **Dr. R. Chandramouli**, Registrar, SASTRA Deemed to be University for providing the opportunity to pursue this project.

We extend our heartfelt thanks to **Dr. V. S. Shankar Sriram**, Dean, School of Computing, **Dr. R. Muthaiah,** Associate Dean, Research, **Dr. K.Ramkumar**, Associate Dean, Academics, **Dr. D. Manivannan,** Associate Dean, Infrastructure, **Dr.R. Algeswaran**, Associate Dean, Students Welfare

Our guide **Dr. SENTIL SELVAN.N** , Assistant Professor - III, School of Computing was the driving force behind this whole idea from the start. Her deep insight in the field and invaluable suggestions helped us in making progress throughout our project work. We also thank the project review panel members for their valuable comments and insights which made this project better.

We would like to extend our gratitude to all the teaching and non-teaching faculties of the School of Computing who have either directly or indirectly helped us in the completion of the project.

We gratefully acknowledge all the contributions and encouragement from my family and friends resulting in the successful completion of this project. We thank you all for providing us an opportunity to showcase our skills through this project.

**List of Figures**

|  |  |  |
| --- | --- | --- |
| Figure No | Title | Page No |
| 1.1 | Study Framework Diagram | 15 |
| 1.2 | Implementation modules | 16 |
| 1.3 | Data Split | 18 |
| 3.1 | ML Models – Data Preprocessing andData Split | 29 |
| 3.2 | ML Models – Model Architectures | 30 |
| 3.3 | DL Models – Data Processing and Data Split | 32 |
| 3.4 | Implementation of Genetic Algorithm | 35 |
| 3.5 | Feature Extraction Using CNN Models | 36 |
| 3.6 | Genetic Algorithm based Feature Selection | 39 |
| 3.7 | GA+LSTM Implemetation | 42 |
| 3.8 | GA + BiLSTM Implementation | 46 |
| 4.1 | Data Set Images | 49 |
| 4.2.1 | ML Models - Evaluation Results | 49 |
| 4.2.2 | DL Models – Evaluation Results including ROC Curves and Learning Curves | 52 |
| 4.2.3 | GA+LSTM – Evaluation Results along with ROC Curves and Learning Curves | 55 |
| 4.2.4 | GA+BiLSTM – Evaluation Results along with ROC Curves and Learning Curves | 56 |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| Table No | Table Name | Page No |
| 1.1 | Description of STROKE PREDICTION Dataset | 13 |

**Abbreviations**

|  |  |
| --- | --- |
| ML | Machine Learning |
| DL | Deep Learning |
| CT | Computed Tomography |
| LSTM | Large Short-Term Memory |
| BILSTM | Bidirectional Long Short-Term Memory |
| CNN | Conventional Neural Network |
| ANN | Artificial Neural Network |
| DNN | Deep Neural Network |
| RNN | Recurrent Neural Network |
| SVM | Support Vector Machine |
| GA | Genetic Algorithm |
| AUC | Area Under The Curve |
| ROC | Receiver Operating Characteristic Curve |
| NASNET | Neural Architecture Search Network |
| VGG | Visual Geometry Group |
| RGB | Red Green Blue |

## ABSTRACT

Cerebrovascular diseases like stroke is one of the main causes of death worldwide, making a huge impact on patients and families. Detection at an early stage is important to save lives and enhance the outcome of recovery. But conventional stroke detection relies heavily on CT brain image analysis manually conducted by medical practitioners, which is time-consuming and susceptible to errors owing to human factors. Recent improvements in machine learning have introduced new chances for stroke identification. The traditional methods of logistic regression, decision trees, random forests, naive Bayes, and support vector machines have been employed here. These techniques have been found to be promising but suffer from problems in choosing a good set of features, dealing with high-dimensional data effectively, and attaining high accuracy. To address these issues, we introduce a novel stroke detection system that is hybrid in nature. Genetic algorithm (GA) along with a bidirectional long short-term memory (BiLSTM) model. The genetic algorithm will select the most relevant features by simplifying the data and enhancing the efficiency of the BiLSTM model. The system will be evaluated using cross-validation and metrics like accuracy, precision, recall, F1 score, ROC (Receiver Operating Characteristic Curve), and AUC (Area Under The Curve). Our goal is to achieve high accuracy, with the aim to surpass current models. This is a suggested system that will assist physicians with an accurate tool to take better decisions and plan treatment for stroke patients. Future refinements will involve the utilization of larger databases, having high-quality data, and the utilization of deep convolutional neural networks (CNNs) in order to make the system more reliable.

**KEY WORDS**: Stroke, Feature selection, Genetic Algorithm, Long short-term memory(LSTM), Bidirectional long short-term memory(BiLSTM), CT images, Machine Learning

**Table of Contents**

**TITLE PAGE NO**

[BONAFIDE CERTIFICATE ii](#_TOC_250010)

[ACKNOWLEDGEMENTS iii](#_TOC_250009)

[LIST OF FIGURES iv](#_TOC_250008)

[LIST OF TABLES v](#_TOC_250007)

[ABBREVIATIONS vi](#_TOC_250006)

[ABSTRACT vii](#_TOC_250005)

[CHAPTER 1 SUMMARY OF BASE PAPER 1](#_TOC_250004)

* 1. INTRODUCTION 1
  2. RELATED WORK 2
  3. PROBLEM STATEMENT 3
  4. OBJECTIVE 3
  5. PROPOSED SOLUTION AND SYSTEM ARCHITECTURE 3
  6. METHODOLOGY AND IMPLEMENTATION 6

CHAPTER 2 MERITS AND DEMERITS OF BASE PAPER 10

* 1. MERITS 10
  2. DEMERITS 10

[CHAPTER 3 SOURCE CODE 11](#_TOC_250003)

* 1. MODULE 1: MACHINE LEARNING MODELS 11
  2. MODULE 2: MODEL ARCHITECHTURES 13
  3. MODULE 3: DEEP LEARNING MODELS 15
  4. MODULE 4: IMPLEMENTATION OF GENETIC ALGORITHM
  5. MODULE 5: FEATURE EXTRACTION USING CNN MODELS
  6. MODULE 6 : GENETIC ALGORITHM-BASED FEATURE SELECTION USING NEURAL NETWORKS
  7. MODULE 7: GA+LSTM
  8. MODULE 8: GA+BILSTM

[CHAPTER 4 SNAPSHOTS 18](#_TOC_250002)

* 1. DATA SET IMAGES 18
  2. EVALUATION RESULTS

4.2.1 MACHINE LEARNING MODEL PERFORMANCES

4.2.2 DEEP LEARNING MODEL PERFORMANCES

4.2.3 GA+LSTM MODEL

4.2.4 GA+BILSTM MODEL

CHAPTER 5 CONCLUSION AND FUTURE PLANS 58

* 1. CONCLUSION 58
  2. FUTURE PLANS 58

CHAPTER 6 REFERENCES 60

**CHAPTER 1**

**SUMMARY OF BASE PAPER**

|  |  |  |
| --- | --- | --- |
| **Title** | : | Innovation in Stroke Identification: A Machine Learning-Based Diagnostic Model Using Neuroimages |
| **Publisher** | : | Institute of Electrical and Electronics Engineers (IEEE) |
| **Year** | : | 2024 |
| **Journal** | : | IEEE Access |
| **Indexing** | : | SCIE / Scopus |
| **Base paper URL** | : | [**https://ieeexplore.ieee.org/document/10445193**](https://ieeexplore.ieee.org/document/10445193) |

#### INTRODUCTION

Identification of a stroke is important to timely identification and effective control. Traditional models rely on clinical tests alone and possess limited abilities to detect a stroke at an early stage. The present work suggests a machine learning-based diagnostic model, supported by neuroimages to improve the detection of strokes. Integrating deep learning mechanisms and multimodal analysis, the accuracy and reliability in the detection of strokes are improved. This method strengthens medical diagnosis, facilitates early treatment, and allows for individualized treatment planning. Stroke is one of the main causes of death and disability globally, with more than 6.2 million deaths occurring every year. Stroke happens either as a result of obstruction of blood flow (ischemic stroke) or breakage of a blood vessel (hemorrhagic stroke). Though ischemic strokes are more prevalent, hemorrhagic strokes tend to be more critical. Conventional clinical practices cannot identify strokes early or effectively, which results in a delay in receiving treatment. Thus, machine learning and deep learning enable an effective alternative to fully automatic, quicker, and more accurate diagnosis. In this research, a stroke diagnosis system has been implemented based on CT image databases. Genetically enhanced Convolutional Neural Networks (CNNs) are employed to extract significant features from neuroimages. These attributes are then labeled with sophisticated deep learning algorithms like Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM), which can learn intricate patterns and time dependencies in the information. To enhance model performance, feature selection methods are employed to ensure just the most relevant data features are utilized. The suggested GA-BiLSTM system is contrasted with other machine learning and deep learning techniques to test its accuracy, reliability, and efficiency. The incorporation of genetic algorithms aids in the optimization of network parameters and provides the system with enhanced capability to identify stroke types more accurately. The effectiveness of the model is implemented on real-world datasets, and its performance is compared with current clinical risk prediction methods. In total, this study makes a major contribution to the science by providing an image-based, automated stroke detection model with better accuracy and aiding clinical decision-making. It is intended to help medical professionals diagnose strokes at an earlier stage, decrease reaction time, and allow for better individualized treatment protocols. By integrating medical imaging, genetic optimization, and deep learning, the system overcomes the shortfalls of traditional diagnostics and enables the creation of smart, scalable healthcare solutions.

#### 1.2 RELATED WORK

* Machine learning algorithms have been used to develop highly accurate stroke risk detection models. These models use historical data from electronic medical records to predict stroke risk in hypertensive patients.
* Various machine learning algorithms, such as Extreme Gradient Boosting (XGBoost), were used to achieve optimal performance in predicting stroke risk.
* The use of machine learning models, including ensemble models that combine different approaches, has shown promising results in identifying individuals at high risk of stroke.
* The authors of thoroughly review skull stripping techniques, including more modern deep learning-based techniques. This review divides the existing approaches to skull stripping into two categories: Deep learning or convolutional neural networks and traditional or classical approaches. Several techniques are emphasized for their potential as they can be integrated into common clinical imaging techniques.
* The authors present a time-based link prediction model called TDRL that uses deep reinforcement learning techniques to learn from a real-world evolving crime dataset. The experiments show that the TDRL model trained on a temporal dataset has better prediction accuracy than other machine learning models trained on a single point in time. Although the TDRL-CNA model accurately predicted most edges in the network topology during the year of the bombing, it failed to predict disappearing edges. Feature analysis revealed that the TDRL-CNA model was least affected by learning based on the removed edges.
* Criminal network analysis (CNA) is difficult due to incomplete datasets, and most machine learning techniques rely on supervised learning. The authors investigated the use of deep reinforcement learning (DRL) to build a model for predicting hidden connections in criminal networks. The results show that this approach performs better than traditional supervised machine learning techniques.
* A study in examined clinical brain data CT and predicted multiple stroke scores after 24 hours or a Modified Rankin Scale score from 0 to 1 over 90 days (“mRS90”) using a National Institutes of Health-derived CNN hybrid structure for artificial neural networks. With this structure, they achieved a detection accuracy of 74% for the mRS90.
* An integrated wavelet entropy-based spider network graph was used with a probabilistic neural network to classify MRI images of the brain into normal brains, strokes, and degenerative diseases. The authors systematically examined diseases, infectious diseases, and brain tumors as part of their study. The first step was to use a discrete wavelet transform to process brain images in two dimensions (2D). They extracted features using spider web diagrams and classified them using a probabilistic neural network. They reported that they achieved a classification accuracy of 100%.
* The authors want to improve the diagnosis and management of strokes by analyzing various factors in electronic health records. To better predict strokes, they use statistics and principal component analysis. Strokes are most commonly found in patients with advanced age, heart disease, average blood sugar levels, and high blood pressure. The paper also suggests using a perceptron neural network with these four attributes to achieve the highest accuracy and lowest error rate compared to other benchmarking algorithms. In addition, the authors address the problem of unbalanced datasets by presenting results on a balanced dataset created using sub-sampling techniques.
* The deep learning framework proposed in has been shown to be highly accurate in detecting different subtypes of intracranial hemorrhage (ICH) in CT images of the head. The system achieves an average accuracy of 96.21% for three types of hemorrhages - epidural, subdural, and intraparenchymal. Compared to existing work, the false positive rate is significantly reduced. In addition, the system includes a quantitative scoring algorithm that automatically measures the thickness and volume of hemorrhagic lesions. This enables clinically relevant quantification, which is important for the decision on emergency surgical treatment.
* A hybrid feature selection approach uses transformed CT image features and grayscale co-occurrence matrix texture features. Features are extracted using discrete wavelet transforms, discrete cosine curves, and GLCMs. The machine learning algorithms Random Tree, Random Forest, and REPTree are used for classification, with Random Forest achieving the highest accuracy of 87.97% for the combination of discrete wavelet transforms and GLCM features.
* An approach to classify CT images with intracranial hemorrhages using machine learning techniques is presented. A set of common features consists of the Gray-Level Co-Occurrence Matrix (GLCM), Discrete Wavelet Techniques (DWT), and Discrete Cosine Techniques (DCT). SMOTE is used to solve the oversampling problem, and sequential forward selection of features is used to obtain subsets of features. Classification accuracy is evaluated using a confusion matrix, precision, and recall. Random Forest achieved the highest accuracy of 87.22 percent in combination with the proposed feature extraction mechanism.
* The authors measure the performance of a deep learning model for detecting intracranial hemorrhage on CT head scans and compare the effects of different preprocessing and model design methods. Implementing preprocessing techniques and a CNN-RNN framework significantly improved the model’s performance and demonstrated its potential as a radiologist decision-support tool.

Based on the literature review conducted, we found that current research can classify strokes with less accuracy. The proposed methods were very computationally intensive. However, we propose a diagnostic system that uses a genetic algorithm with bidirectional long short-term memory (BiLSTM) to predict strokes by analyzing brain images on a CT scan. The proposed method solves the problems of lower accuracy and higher computational complexity of the previously proposed models.

**1.3 PROBLEM STATEMENT**

Stroke is a major cause of death and disability globally, necessitating quick and precise diagnosis for successful treatment. Traditional clinical approaches are not effective in detecting strokes at early stages, particularly in distinguishing between stroke types. Intelligent systems that use neuroimaging data to improve early detection and classification accuracy are needed. Manual analysis of CT images is labor-intensive and susceptible to human error, influencing diagnosis speed and reliability. This project seeks to overcome these limitations by creating a deep learning-based model with optimized feature selection for precise stroke detection from neuroimages.

**1.4 OBJECTIVE**

A method for better prediction of strokes with neuroimages, especially CT brain scans, is suggested. The research identifies the best feature selection method based on Genetic Algorithm (GA) to enhance the model's performance. Through the selection of most meaningful features, the model's capacity to forecast stroke risk is improved. Moreover, the study examines appropriate classification techniques that perform best in processing CT brain scans for accurate stroke detection. Using the integration of Bidirectional Long Short-Term Memory (BiLSTM) is expected to boost the accuracy of predictions further, as it will identify both the forward and backward dependencies of brain image information. In conclusion, the technique will help doctors with early stroke diagnosis and treatment.

**1.5 PROPOSED SOLUTION AND SYSTEM ARCHITECTURE:**

**1.5.1 STUDY AREA**

The scope of research here is the stroke prediction and diagnosis from neuroimages, more specifically CT brain scan images, utilizing convolutional neural networks (CNN) for feature extraction and genetic algorithms (GA) for feature selection. The study employs neuroimage data of stroke patients, along with significant medical parameters and image data, to ascertain the likelihood of a stroke. Strokes, particularly ischemic and hemorrhagic strokes, are significant sources of death and disability worldwide, and therefore require early detection for optimal outcomes in patients. Traditional diagnosis using the CT scan and MRI can be costly and requires much expertise, which may not always be readily available. By utilizing CNNs to efficiently extract features and GA to select the most discriminative features, the research aims at enhancing the predictive accuracy. In addition, the research examines hybrid models by incorporating both GA and LSTM as well as BiLSTM for stroke prediction improvement. This approach provides an early non-invasive, cost-effective, and accurate stroke detection method, overcoming data paucity problems and improving model performance. The long-term goal is to assist clinicians in their accurate and timely diagnoses in order to improve the outcomes of patients.

**1.5.2 STROKE PREDICTION DATASET**

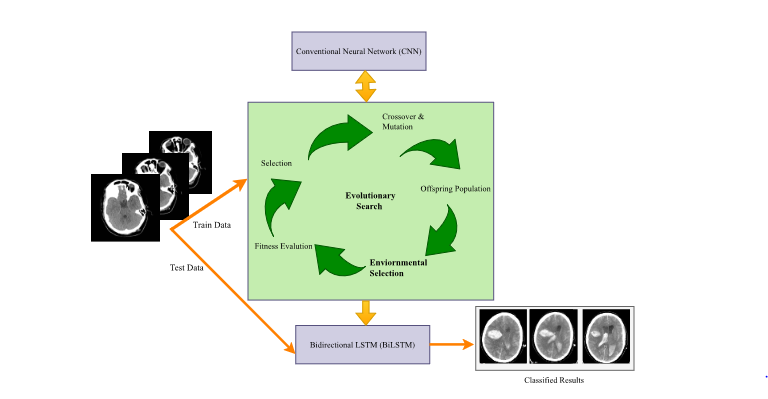
|  |  |  |
| --- | --- | --- |
| **S.NO** | **FEATURES** | **DESCRIPTION** |
| 1 | Image Data | CT brain scan images of stroke and non-stroke patients |
| 2 | Age | The age of the patient (in years) |
| 3 | Gender | The gender of the patient (Male/Female) |
| 4 | Hypertension | Whether the patient has hypertension (1: Yes, 0: No) |
| 5 | Heart Disease | Whether the patient has heart disease (1: Yes, 0: No) |
| 6 | Smoke Status | Smoking status of the patient (1: Smoker, 0: Non-smoker) |
| 7 | Stroke Type | The type of stroke (Ischemic, Hemorrhagic, Transient) |

**Table 1.1 Description of STROKE PREDICTION Dataset**

The Stroke Prediction dataset used in this study contains CT scan images of stroke patients with medical features significant in evaluating stroke risk. The dataset includes brain scan images labeled into two groups: stroke and non-stroke, where stroke images are sub-divided into ischemic, hemorrhagic, or transient. Apart from the image data, the dataset contains some of the important medical features like the age of the patient, gender, status of hypertension, status of heart disease, and history of smoking. The dataset consists of 1551 normal (non-stroke) images and 950 stroke images, representing a balanced sample of the two classes. The main purpose of this dataset is to train machine learning models, mainly convolutional neural networks (CNNs), to extract essential features from the images of the CT scans. A genetic algorithm (GA) is also used for best feature selection and assist in identifying relevant features for stroke prediction. The target of classification is the stroke diagnosis, which labels both the CT scan images and patient medical data as stroke or non-stroke. Due to the complexity of brain images, sophisticated methods like CNN-based feature extraction are utilized to learn automatically and extract features from the images. For further improving prediction accuracy, hybrid models including GA together with Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) are used. These models assist in capturing spatial and temporal dependencies of the data, enhancing the overall predictive capability. Preprocessing techniques like image resizing, normalization, and feature selection are used to optimize the training process of the model so that efficient learning occurs and the models' predictive power for stroke is enhanced to high accuracy.

**1.5.3 Study Framework Diagram**

The dataset employed within this project consists of brain CT scan images and corresponding patient medical data, which are subjected to systematic preprocessing to improve data quality and readiness for deep learning algorithms. These include image resizing to a common dimension, grayscale conversion, pixel value normalization, and cleaning or encoding corresponding medical attributes like age, gender, hypertension, heart disease, and smoking status. Feature extraction is accomplished by utilizing several CNN models to extract spatial features, and a Genetic Algorithm (GA) is used to choose the most important features to use for classification. The database is split into three sets—70% to train the models, 15% for validating the models' performance in training, and 15% for testing. These classification models like GA+LSTM and GA + BiLSTM are used to improve the prediction accuracy by learning sequential patterns from the extracted features. The entire process of using this stroke prediction system is shown in Fig 1.1**.**

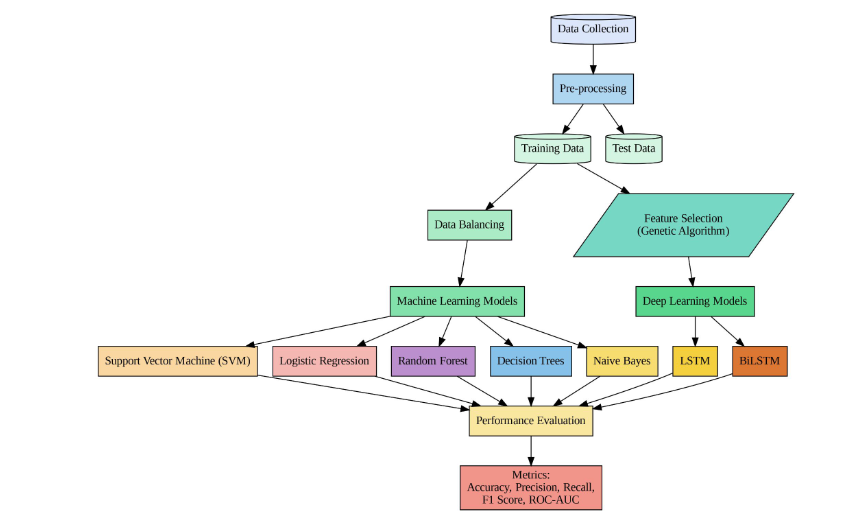


**Fig 1.1 Study Framework Diagram**

#### 

#### 1.6 METHODOLOGY AND IMPLEMENTATION

The methodology is segregated into three modules for proper implementation. The pictorial representation of this module segregation is given in Fig 1.2



**Fig 1.2 Implementation modules**

#### 1.6.1 MODULE 1: DATA PREPROCESSING AND FEATURE ENGINEERING

Module 1 contains the Preprocessing of data is a key step towards preparing the neuroimage dataset for effective and precise stroke detection. The raw dataset of grayscale CT brain images with "Normal" or "Stroke" labels needs systematic conversion prior to its application in deep learning models. All images were resized to a consistent resolution of 227×227 pixels and normalized to [0, 1] range to keep things consistent and ensure model compatibility. Labels were converted to numerical form using one-hot encoding to make them classification-ready.

To meet data quality, preprocessing entailed removing missing images or corrupted files, normalizing image formats and sizes, and balancing the dataset by providing equal representation of both classes. Outlier images or noisy scans that might mislead model training were removed during inspection and normalization.

Advanced feature extraction was performed using five CNN architectures—AlexNet, InceptionV3, VGG19, NASNetLarge, and ShuffleNet. Each model generated high-level features, which were then concatenated into a single feature vector representing each image. These combined features were further refined using a Genetic Algorithm (GA), which selected the most informative features by evaluating them through a small neural network as a fitness function.

Feature Engineering involved dimensionality reduction via GA and exploration of new derived features like pixel intensity change and texture patterns in order to improve model performance. The chosen features were fed into LSTM and BiLSTM models, which were trained with dropout, batch normalization, and regularization to prevent overfitting. Class imbalance was addressed using weighted loss functions based on label distribution.

This module also included designing the entire pipeline, such as data loading from remote locations (e.g., Google Drive), preprocessing, feature extraction, GA-based selection, and final model training and validation. Once prepared, the dataset was divided into training, validation, and testing sets, and models were configured for training under optimized neural parameters.

##### 1.6.2 MODULE 2: MODEL TRAINING AND EVALUATION

Module 2covers Model Training and Evaluation. In this step, the preprocessed CT neuroimage dataset was divided into 70% training, 15% validation, and 15% testing. Deep features were extracted from five convolutional neural networks—AlexNet, InceptionV3, VGG19, NASNetLarge, and ShuffleNet—and concatenated into a common high-dimensional feature vector. The combined features were optimized by a Genetic Algorithm (GA), which chose the most important features by testing their classification power through a light-weight neural network as the fitness function.



**Fig 1.3 Data split**

The chosen features were subsequently utilized for training complex models such as LSTM and BiLSTM architectures. The models were provided with regularization techniques such as dropout and L2 penalty, and class imbalance was solved utilizing calculated class weights. Mechanisms such as early stopping, model check pointing, and learning rate scheduling were employed for stabilizing and enhancing the process of learning.  
  
At the evaluation phase, models were evaluated on unseen testing and validation data, and their performance was measured using conventional classification metrics including accuracy, precision, recall, F1-score, and AUC. Visualizing learning curves and ROC curves gave additional information about model behavior. The specified architecture and training regimens resulted in good generalization and correct classification.  
  
This module assisted in the identification of the strongest and best-performing model pipeline for stroke classification based on deep neuroimage features and temporal learning frameworks.

##### ALEXNET model:

AlexNet is a groundbreaking deep CNN model that has played a major role in the success of deep learning in image classification problems. It is made up of a number of convolutional layers followed by fully connected layers.

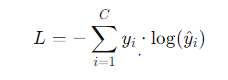
**Architecture:**  
  
Convolutional Layers: 5 convolutional layers (Conv1 to Conv5). These layers extract spatial hierarchies by applying filters on input images.

Pooling: Max pooling layers are present after some convolutional layers to decrease spatial dimensions.  
  
Fully Connected Layers: Following the convolution and pooling layers, there are 3 fully connected (FC) layers.  
  
Dropout: A dropout regularization strategy is used to prevent overfitting during training.  
  
**Activation:** ReLU Activation Function: Used after each of the convolution and fully connected layers to provide non-linearity to the model.  
  
**Feature Extraction:** The output from the network's last fully connected layer (FC6) is a 4096-dimensional feature vector. The feature vector extracts high-level image information.

**Feature Output:** The output from the network's last fully connected layer (FC6) is a 4096-dimensional feature vector. The feature vector extracts high-level image information.  
  
**Prediction:** For classification prediction, the final output is given after applying the softmax function, and the most likely class is taken:

**AlexNetpred​=argmax(softmax(f(x)))**

f(x) is the output of the final fully connected layer.

**Loss Function:** The model employs **categorical cross-entropy** as the loss function for classification. This computes the difference between the true label  
  
 

yi​ is the true label (one-hot encoded).

yi^​​ is the predicted probability for class i.

C is the number of classes (2 in this case: Stroke, Normal).

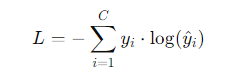
##### Output Features: The feature vector extracted after going through the AlexNet architecture is 4096-dimensional, utilized for further classification.

**INCEPTIONV3 model**

InceptionV3 is a more complex architecture that aims to enhance computational efficiency. It employs inception modules, which enable the model to learn various filter sizes at the same time.  
  
**Architecture:**  
  
Inception Modules: These are parallel convolutional towers (1x1, 3x3, 5x5) that extract various spatial information from the input image simultaneously.  
  
Dimensionality Reduction: InceptionV3 employs 1x1 convolutions to decrease the number of parameters and computation.  
  
Global Average Pooling (GAP): Following the convolutional layers, a global average pooling operation is used to decrease the spatial dimensions and produce a compact representation.  
  
**Activation:** ReLU Activation Function: Used after every convolutional and fully connected layer.  
  
**Feature Extraction:** The output of the global average pooling layer is a 2048-dimensional feature vector.  
  
**Feature Output:** The feature vector extracted after the InceptionV3 network is 2048-dimensional.  
  
**Prediction:** The prediction is obtained by applying the softmax function to the output of the model at the end:

**Inceptionpred​=argmax(softmax(g(x)))**

g(x) is the output of the global average pooling layer.

**Loss Function:** The loss function employed for optimization is **categorical cross-entropy**, as in AlexNet:  
 

yi​ is the true label (one-hot encoded).

yi^​​ is the predicted probability for class i.

C is the number of classes (2 in this case: Stroke, Normal).

**Output Features:** The last feature vector obtained by InceptionV3 is of 2048 dimension after global average pooling.

**VGG-19 model**

VGG-19 is a deep convolutional neural network with 19 layers, using small 3×3 filters. It is known for its simple and uniform architecture.

**Architecture:**

Convolutional Layers: 16 convolutional layers with 3×3 filters, grouped in blocks with increasing depth (64, 128, 256, 512, 512).

Pooling: MaxPooling layers are added after each block to reduce spatial size.

Fully Connected Layers**:** Not included in this custom version; the network ends after the last MaxPooling layer.

**Activation:** ReLU Activation Function is used after each convolutional layer to add non-linearity.

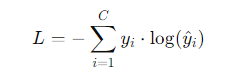
**Feature Extraction:** After the last MaxPooling layer, the output is flattened using a Flatten layer.The flattened feature vector is used as the final representation.

**Feature Output:** The final extracted feature vector is 25088-dimensional (for input size 227×227×3).

**Prediction:** The prediction is obtained by applying the softmax function to the output of the model at the end:

**VGGpred​=argmax(softmax(v(x)))**

v(x) is the flattened feature vector.

**Loss Function:** The loss function employed for optimization is **categorical cross-entropy**, as in AlexNet:  
 

yi​ is the true label (one-hot encoded).

yi^​​ is the predicted probability for class i.

C is the number of classes (2 in this case: Stroke, Normal).

**Output Features:** The final output feature vector is 25088-dimensional after flattening.

**NASNET-LARGE model**

NASNet-Large is a neural network architecture found through Neural Architecture Search (NAS). It is optimized for performance while maintaining computational efficiency.

**Architecture:**

Convolutional Layers: 3 convolutional layers with filters (32, 64, 128), using 3×3 and 1×1 filters, with the first layer having a stride of 2. Followed by 3 normal cell blocks that concatenate outputs from 1×1 and 3×3 convolutions.

Pooling: A GlobalAveragePooling2D layer is applied after the last block to reduce spatial dimensions and produce a 1D feature vector.

**Activation:** ReLU Activation Function is applied after every convolutional layer.

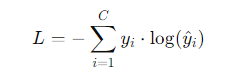
**Feature Extraction:** The final GlobalAveragePooling2D layer produces a compact feature vector for the input image.

**Feature Output:** Output is a 1D feature vector whose size depends on the number of filters in the last concatenated block (128 in this case after 3 blocks with concatenation).

**Prediction:** The prediction is obtained by applying the softmax function to the output of the model at the end:

**NASNetpred​=argmax(softmax(v(x)))**

v(x) is the flattened feature vector.

**Loss Function:** The loss function employed for optimization is **categorical cross-entropy**, as in AlexNet:  
 

yi​ is the true label (one-hot encoded).

yi^​​ is the predicted probability for class i.

C is the number of classes (2 in this case: Stroke, Normal).

**Output Features:** The final output is a 1D feature vector (e.g., 128 or 256 depending on concatenation depth).

**SHUFFLENET model**

ShuffleNet is a light-weight convolutional neural network for efficient computation on mobile and embedded systems. It applies pointwise group convolution and channel shuffling to minimize computational cost without compromising accuracy.

**Architecture:**

Convolutional Layers: 5 convolutional layers with filters (24, 144, 144, 288), using 3×3 and 1×1 filters. The first 3×3 convolution has a stride of 2, followed by a 3×3 max pooling layer with stride 2. The remaining layers use a combination of 1×1 and 3×3 convolutions to extract features efficiently.

Pooling: Global average pooling is applied after the final convolutional layer.

**Activation:** ReLU Activation Function is applied after every convolutional layer to provide non-linearity.

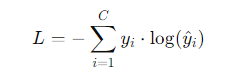
**Feature Extraction:** The output of the last convolutional layer is fed into a global average pooling layer to extract a 288-dimensional feature vector.

**Feature Output:** The final extracted feature vector from ShuffleNet is 288-dimensional.

**Prediction:** Prediction is obtained using a softmax function over the extracted feature vector:

**ShuffleNet\_pred = argmax(softmax(v(x)))**

v(x) is the flattened feature vector.

**Loss Function:** The loss function employed for optimization is **categorical cross-entropy**, as in AlexNet:  
 

yi​ is the true label (one-hot encoded).

yi^​​ is the predicted probability for class i.

C is the number of classes (2 in this case: Stroke, Normal).

**Output Features:** The last feature vector from ShuffleNet is 288-dimensional after global average pooling.

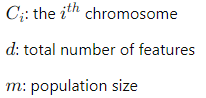
**GA-Based Feature Selection:**

Genetic Algorithm (GA) is a meta-heuristic that follows the natural process of selection. It is used to solve optimization problems and specifically excels at feature selection tasks. GA identifies the most significant subset of features in high-dimensional data by mimicking the evolution process — comprising **selection**, **crossover**, **mutation**, and **survival of the fittest**.  
  
In feature selection, every person (**chromosome**) in the population is a possible solution — a set of features. A chromosome's performance is measured with a fitness function, generally model accuracy.

**Steps to implement:**

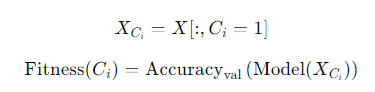
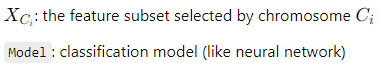
Step 1: Population Initialization: Each chromosome is presented as a binary string of the same length as the number of features d. A `1` means the feature is chosen, and `0` means otherwise.





A set of such chromosomes is created randomly to initiate the search space exploration.

Step 2: Fitness Evaluation: For every chromosome, the set of chosen features is employed to train a classifier (e.g., Neural Network). Classification accuracy on validation data is employed as the fitness score.

In this step, the utility of each feature subset is judged. More accurate chromosomes are fitter to reproduce.

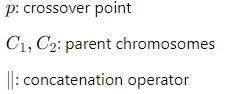
Step 3: Parent Selection: Choose the best k chromosomes with the best fitness scores to be parents of the next generation.



This step is an imitation of natural selection, where only the top-performing solutions get to reproduce.

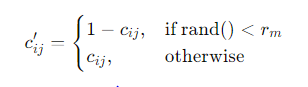
Step 4: Crossover (Recombination): Two parents are chosen to produce new offspring by applying single-point crossover. A crossover point is chosen, and bits from each parent are pooled together.

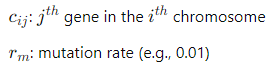




This introduces diversity by combining genetic information (feature subsets) from two parents to create a new child.

Step 5: Mutation: To avoid premature convergence, mutation is introduced. Each bit in the chromosome has a small probability rm of being flipped (0 to 1 or 1 to 0).





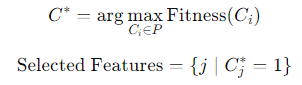
Mutation ensures exploration of new parts of the search space and prevents the algorithm from getting stuck in local optima.

Step 6: Forming New Generation: The chosen parents and the newly created mutated offspring constitute the new population for the subsequent generation.



This new generation iterates the process for a specified number of generations or until a stopping condition (e.g., no improvement) is reached.

Step 7: Final Solution: After running all generations, the chromosome with the best fitness score is chosen as the solution. The indices of features chosen by this chromosome are returned.



These selected features are the optimal subset, providing the best validation accuracy when used to train the model.

**LSTM:**

Long Short-Term Memory (LSTM) networks are a variant of Recurrent Neural Network (RNN) that is specifically used to identify long-range dependencies in sequential data. LSTM overcomes the vanishing gradient issue by employing a cell state and several gates to control information flow. Implementation employs LSTM for classifying a subset of reduced features that have been derived using GA.

LSTM Cell Architecture: An LSTM cell holds a **cell state Ct**​ and a **hidden state** **ht.** It uses three gates to control the flow of information:

**Forget Gate** ft

**Input Gate** it​

**Output Gate** ot

**Candidate Cell State** Ct~

Gate-Level Operations:

xt​: input at time step t

ht−1: hidden state from previous step

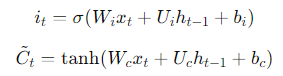
Ct−1​: cell state from previous step

W,U,b: weights and biases of the gates

Forget Gate: Controls what information to forget from the previous cell state.



Input Gate: Controls what new information to incorporate into the cell state.

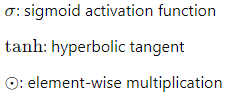


Cell State Update: Merges old and new information



Output Gate: Decides what to output



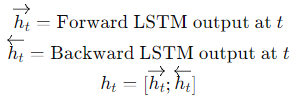


**BILSTM:**

A Bidirectional LSTM (BiLSTM) is a kind of Recurrent Neural Network (RNN) that enhances the performance on sequence prediction tasks because it enables the network to contain both backward and forward information on the sequence for each time step. It involves two LSTMs: one with input in the forward direction and the other in the backward direction.

Internal Architecture of a BiLSTM Cell: There are two LSTM layers in a BiLSTM.  
  
**Forward LSTM:** Processes the input sequence from t=1 to t=n.  
  
**Backward LSTM:** Processes the input sequence from t=n to t=1.

At each time step t, the final hidden state is the concatenation of both:



#### 1.6.3 MODULE 3: MODEL PERFORMANCES AND VALIDATION

In this stage, deep learning models were tested through features chosen through Genetic Algorithm (GA) from an ensemble CNN feature set (AlexNet, InceptionV3, VGG19, NASNet, ShuffleNet). Of 29,792 raw features,GA picked 14,834most pertinent ones for dimensionality reduction and better model efficiency. They were scaled and reshaped for sequential input into LSTM and BiLSTM models.  
  
Both BiLSTM and LSTM models were trained using dropout, early stopping, L2 regularization, and batch normalization. The BiLSTM model, utilizing bidirectional context, performed slightly better on validation. Performance was tracked across 100–500 epochs using accuracy, precision, recall, and ROC-AUC, while handling class imbalance using weighted loss functions. GA-based selection with BiLSTM overall resulted in effective stroke classification performance.

# CHAPTER 2

## MERITS AND DEMERITS OF BASE PAPER

#### MERITS

* Integrated features from five strong CNN models (AlexNet, InceptionV3, VGG19, NASNet, ShuffleNet) guaranteeing good feature representation.
* Genetic Algorithm (GA) effectively reduced dimensionality from 29,792 to 14,834 features, improving model performance and avoiding overfitting.
* Applied both LSTM and BiLSTM architectures with regularization, dropout, and early stopping for stable sequence modeling.
* Obtained high classification accuracy, reflecting good model generalization.
* Handled class imbalance with class weights and tested with precision, recall, and ROC-AUC scores.
* Clean and scalable design with good application of deep learning best practices and callbacks for efficient training.

#### DEMERITS

Although the project is well-structured and shows promise, there are a few areas where it could be improved:

* Genetic Algorithm feature selection process is computationally intensive and might not scale well to very large datasets.
* No explicit comparison with conventional feature selection techniques (e.g., PCA, mutual information).
* Temporal modeling might be overkill for single-step inputs, possibly introducing unnecessary complexity.
* Fixed training parameters do not generalize across datasets without re-tuning.
* No explainability or feature importance evaluation, restricting interpretability in clinical use.
* Single medical dataset evaluation; no cross-domain or cross-modal validation.

**CHAPTER 3**

**SOURCE CODE**

#### 3.1 MODULE 1: MACHINE LEARNING MODELS

#### 3.1.1 DATA PREPROCESSING

#### 

#### 3.1.2 SPLITTING THE DATA FOR TRAINING

#### 

#### 3.2 MODULE 2 : MODEL ARCHITECHTURES

#### 3.2.1 SVM MODEL IMPLEMENTATION

#### 

#### 3.2.2 LOGISTIC REGRESSION MODEL IMPLEMENTATION

#### 

#### 3.2.3 DECISION TREE MODEL IMPLEMENTATION

#### 

#### 3.2.4 RANDOM FOREST MODEL IMPLEMENTATION

#### 

#### 3.2.5 NAÏVE BAYES MODEL IMPLEMENTATION

#### 

#### 3.3 MODULE 3: DEEP LEARNING MODELS

#### 3.3.1 DATA PREPROCESSING

#### 

#### 3.3.2 DNN ARCHITECHTURE

#### 

#### 3.3.3 COMPIALTION AND EVALUATION METRICS

#### 

#### 

#### 

#### 3.3.4 ANN ARCHITECHTURE

#### 

#### 3.3.5 RNN ARCHITECHTURE

#### 

#### 3.4 IMPLEMENTATION OF GENETIC ALGORITHM

#### 3.4.1 SETTING DATAPATHS

#### 

##### This code defines the file paths for loading brain CT images, separating them into two categories: normal and stroke cases.

##### 3.4.2 IMPORTING REQUIRED LIBRARIES

##### 

##### 3.4.3 LOAD AND PREPROCESS IMAGES

##### 

##### 3.4.4 SPLIT THE DATASET

##### 3.5 MODULE 5 : FEATURE EXTRACTION USING CNN MODELS

##### 3.5.1 ALEXNET ARCHITECHTURE

##### 

##### 3.5.2 INCEPTION V3 ARCHITECHTURE

##### 

##### 3.5.3 VGG-19 ARCHITECTURE

##### 

##### 3.5.4 NASNET ARCHITECHTURE

##### 

##### 3.5.5 SHUFFLENET ARCHITECHTURE

##### 

##### 3.5.6 COMBINING ALL MODELS

##### 

##### 3.5.7 DISPLAYING FEATURES

##### 

##### 3.6 MODULE 6 : GENETIC ALGORITHM-BASED FEATURE SELECTION USING NEURAL NETWORKS

##### 3.6.1 IMPORTS AND ENVIRONMENTAL SETUP

##### 

##### 3.6.2 GENETIC ALGORITHM PARAMETERS AND POPULATION INITIALIZATION

##### 

##### 3.6.3 FITNESS FUNCTION USING NEURAL NETWORK

##### 

##### 3.6.4 PARENT SELECTION AND CROSSOVER, MUTATION

##### 

##### 3.6.5 RUNNING THE GENETIC ALGORITHM AND EXTRACTING BEST FEATURE SUBSET

##### 

##### 3.7MODULE 7 : GA + LSTM

##### 3.7.1 IMPORTS AND CONFIGURATION

##### 

##### 3.7.2 DATA PREPARATION

##### 

##### 3.7.3 LSTM MODEL ARCHITECTURE

##### 

##### 3.7.4 MODEL COMPILATION, CALLBACKS

##### 

##### 3.7.5 MODEL TRAINING

##### 

##### 3.7.6 EVALUATION ROC CURVE AND LEARNING CURVES

##### 

##### 3.8 MODULE 8 : GA+BiLSTM

##### 3.8.1 IMPORTS AND CONFIGURATION

##### 

##### 3.8.2 DATA PREPARATION

##### 

##### 3.8.3 BILSTM ARCHUTECHTURE

##### 

##### 3.8.4 MODEL COMPILATION, CALLBACKS

##### 

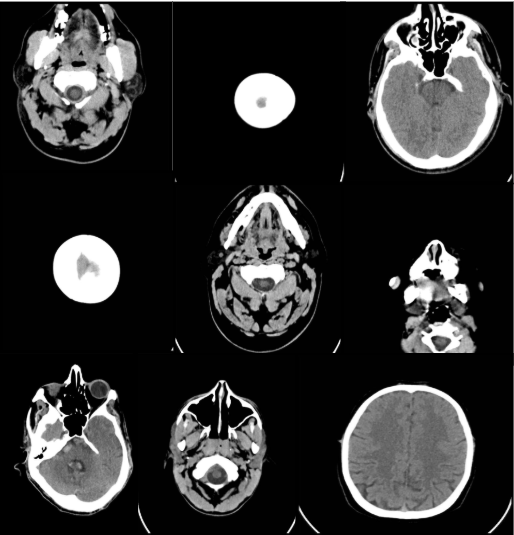
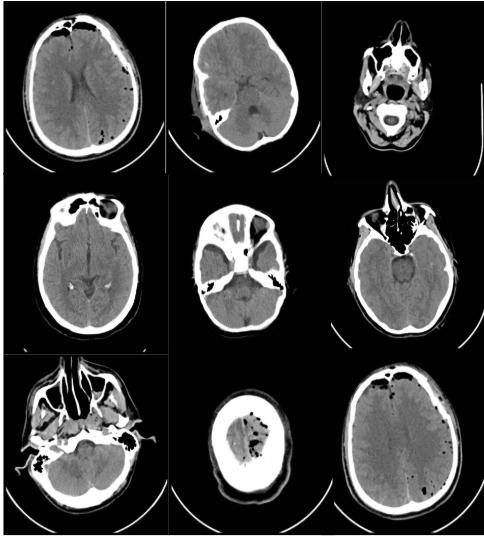
##### 3.8.5 EVALUATION, ROC CURVE AND LEARNING CURVES

##### 

##### 

## CHAPTER 4 SNAPSHOTS

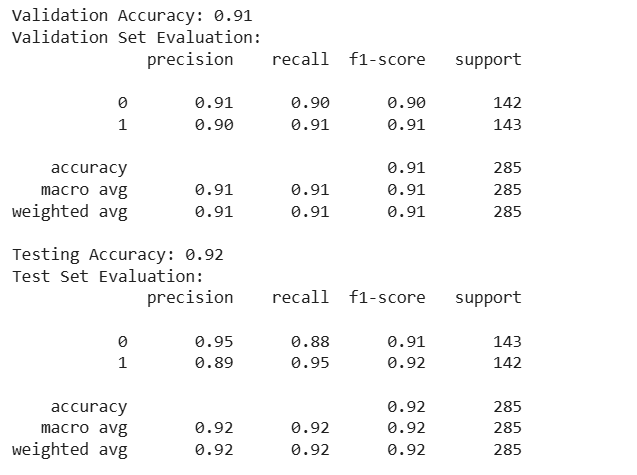
#### 4.1 DATA SET IMAGES

****

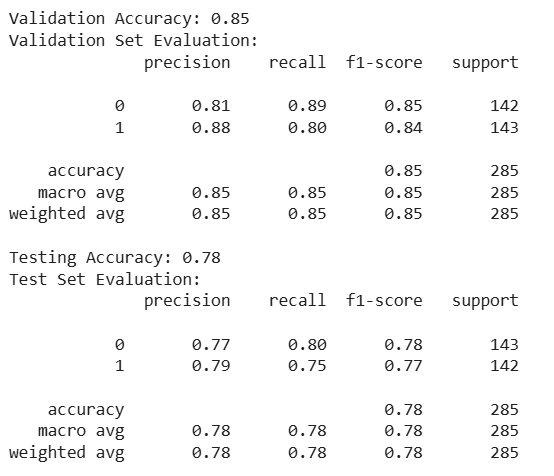
**Fig 4.1 Preprocessed STROKE Dataset**

**4.2 EVALUATION RESULTS**

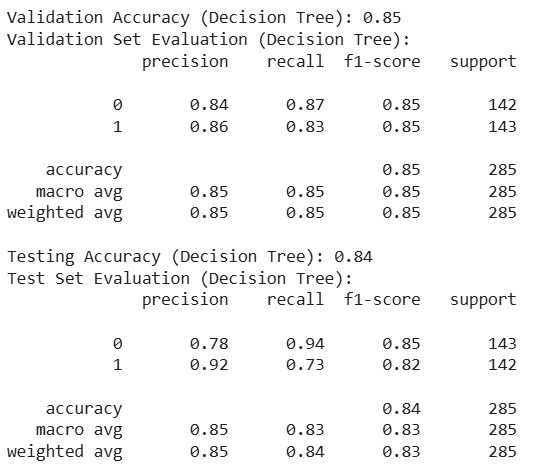
**4.2.1 MACHINE LEARNING MODEL RESULTS**

****

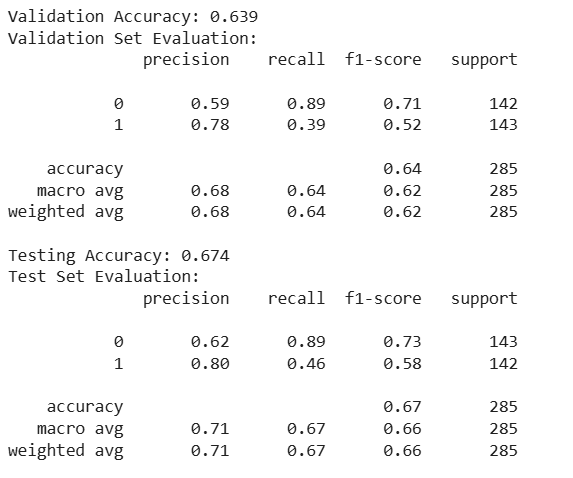
**Fig 4.2.1 Classification Report of SVM**

****

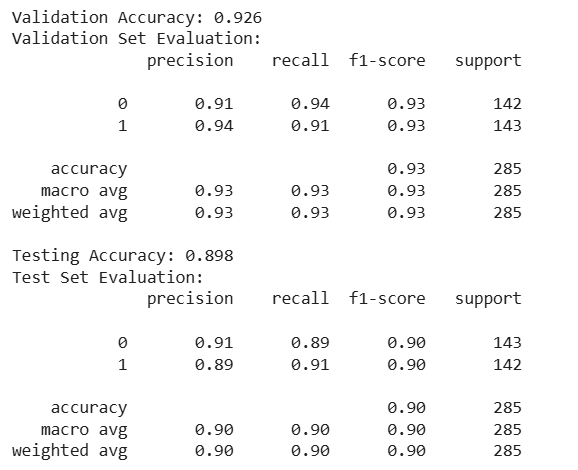
**Fig 4.2.2 Classification Report of Logistic Regression**

****

**Fig 4.2.3 Classification Report of Decision Tree**

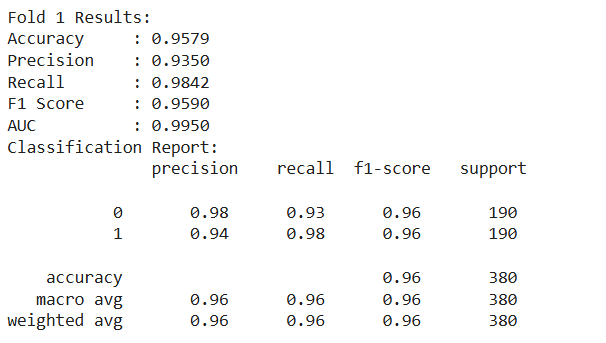
****

**Fig 4.2.4 Classification Report of Naive Bayes**

****

**Fig 4.2.1 Classification Report of Random Forest**

**4.2.2 DEEP LEARNING MODEL RESULTS**

****

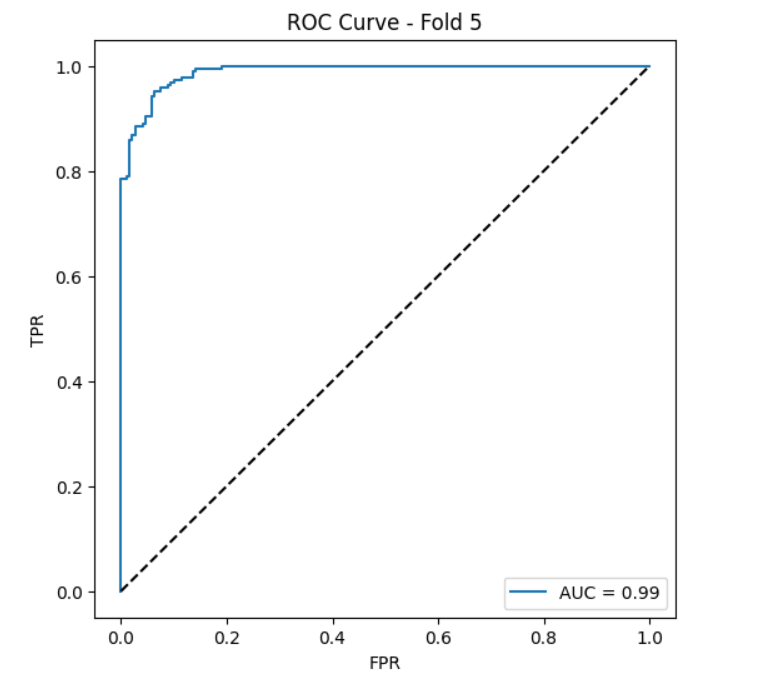
**Fig 4.2.2.1 Classification Report of DNN**

****

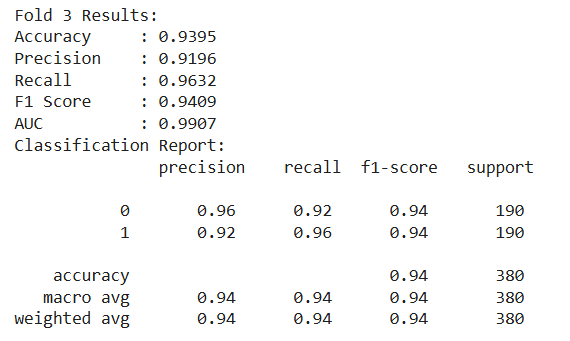
**Fig 4.2.2.2 ROC Curve of DNN**

****

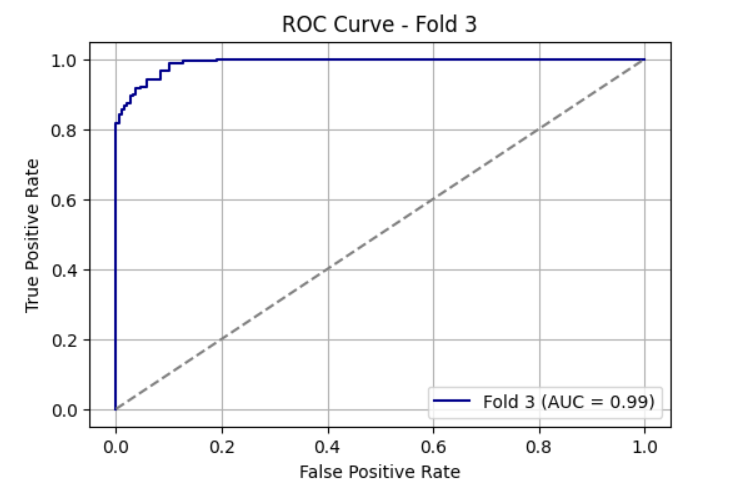
**Fig 4.2.2.3 Classification Report of ANN**

****

**Fig 4.2.2.4 ROC Curve of ANN**

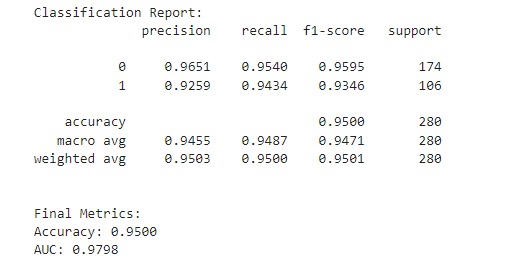
****

**Fig 4.2.2.5 Classification Report of RNN**

****

**Fig 4.2.2.6 ROC Curve of RNN**

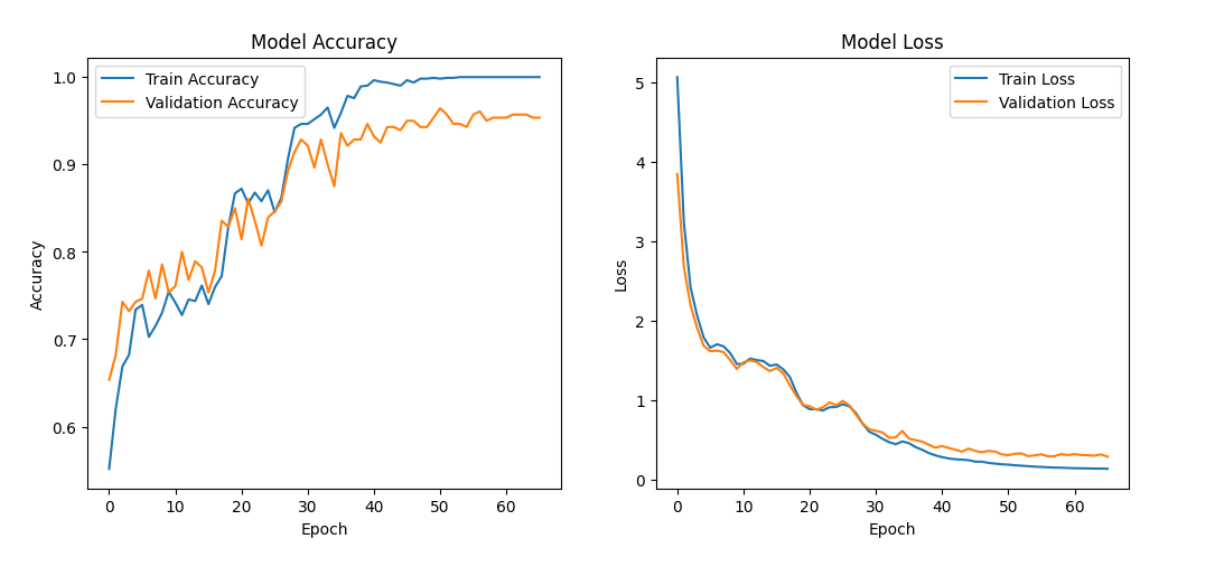
**4.2.3 GA + LSTM MODELS**



**Fig 4.2.3.1 Classification report of GA + LSTM**

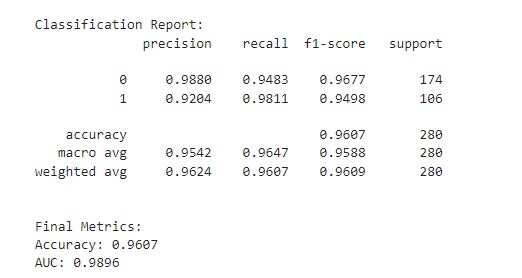
****

**Fig 4.2.3.2 ROC Curve of GA + LSTM**

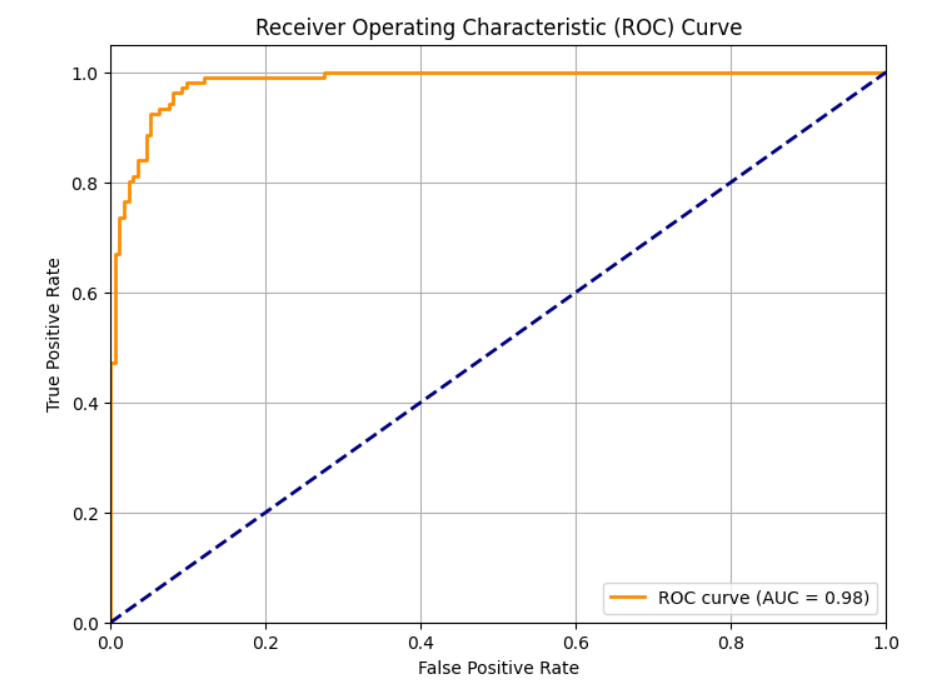
****

**Fig 4.2.3.3 Learning Curve of GA + LSTM**

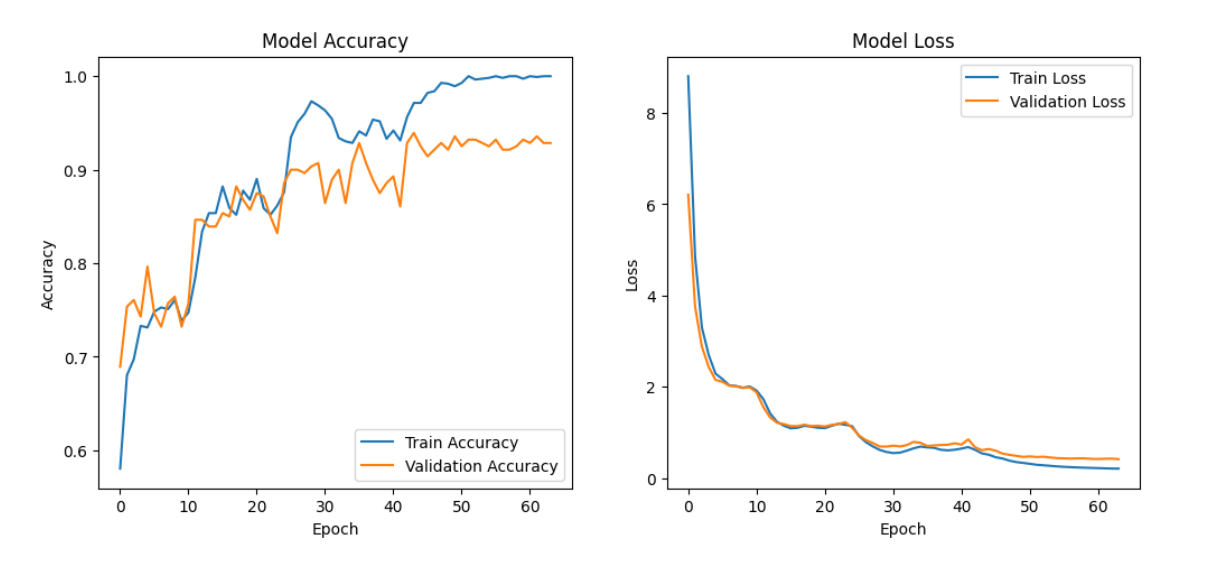
**4.2.4 GA + BiLSTM MODELS**



**Fig 4.2.4.1 Classification Report of GA + BiLSTM**

****

**Fig 4.2.4.2 ROC Curve of GA + BiLSTM**

** Fig 4.2.4.3 Learning Curve of GA + BiLSTM**

**CHAPTER 5**

**CONCLUSION AND FUTURE PLANS**

**5.1 CONCLUSION**

This project integrates machine learning techniques to enhance stroke detection from neuroimages. We evaluated BiLSTM, SVM, and logistic regression models, comparing their performance with and without genetic algorithm-based feature selection. Our findings indicate that the BiLSTM model with genetic algorithm-based feature selection achieves the highest accuracy, making it a promising tool for early stroke identification. This approach can aid medical professionals in making timely and accurate diagnoses, improving patient outcomes.

Furthermore, the employment of Genetic Algorithm (GA) feature selection not only reduced data dimensionality remarkably while preserving the core features but also improved the model’s efficiency. The GA\_LSTM and GA\_BILSTM models scored a 95.00% and 96.07% classification accuracy, respectively, affirming the performance of deep learning in the diagnosis of stroke from neuroimage.  
  
This project highlights the potential of machine learning in medicine, particularly in enhancing the accuracy of stroke diagnosis. Application of GA for feature selection not only enhances model performance but also ensures that only the most significant features are used, minimizing the computational requirements and making the process more efficient. Additionally, the results from this study hold promise to be applied in clinical practice, to help medical experts make early stroke diagnosis and treatment, ultimately enhancing healthcare outcomes.

**5.2 FUTURE PLANS**

Based on the encouraging outcomes of this study, several directions are planned to further improve the precision, scalability, and usability of the stroke detection system:

* **Dataset Enlargement and Diversity:** In the present scenario, the dataset for stroke detection is concentrated on a particular collection of images. Future work will include larger and more diverse datasets, such as data from diverse demographic populations, different levels of stroke severity, and images across various imaging modalities, to improve model robustness and generalizability.
* **Real-Time Integration Monitoring:** Successive versions of the system will look into integration with healthcare systems in real time to enable around-the-clock monitoring of patients' neuroimages. Using IoT-based devices or cloud-based platforms, the system has the capability of delivering real-time stroke detection for early intervention of emergency medical care.
* **Integration of State-of-the-Art Deep Learning Models:** Although BiLSTM has demonstrated encouraging outcomes, future enhancement will involve investigating more sophisticated deep learning models like Convolutional Neural Networks (CNN), Transformer models, and hybrid models to capture more spatial and temporal dependencies in neuroimages.
* **Enhanced Feature Selection Methods:** Additional improvements will include testing other feature selection methods like Recursive Feature Elimination (RFE) or feature engineering methods that may enhance the model’s capability to identify nuanced patterns, minimize the chance of overfitting, and enhance generalization.
* **Multi-Modality Stroke Detection:** Future advances will include multi-modality stroke detection by examining a combination of neuroimaging modalities like CT scans, MRI, and PET scans. This may enhance stroke detection accuracy by exploiting complementary information from various imaging modalities.
* **Intuitive Interface for Healthcare Professionals**: To make it more accessible, a user-friendly interface will be created for healthcare professionals so that they can easily upload and analyze neuroimages, obtain stroke detection results, and view the decision-making process of the model. This will make the system more feasible for use in real-world hospitals and clinics.
* **Decision Support System Integration:** Subsequent releases of the stroke detection system will be supplemented with a decision support module that offers further clinical information, including the probable severity of the stroke, suggested treatments, and risk factors for the patient's condition. This will facilitate more informed decision-making in clinical environments.

By integrating these advancements, the stroke detection system is intended to become an all-around early diagnosis tool, treatment planning tool, and patient care tool, leading to improved stroke management outcomes and enhancing the application of AI in the healthcare sector.

## REFERENCES

S. Dev, H. Wang, C. S. Nwosu, N. Jain, B. Veeravalli, and D. John, “A predictive analytics approach for stroke prediction using machine learning and neural networks,” *Healthcare Analytics*, vol. 2, Nov. 2022, Art. no. 100032.

M. Saritha, K. P. Joseph, and A. T. Mathew, “Classification of MRI brain images using combined wavelet entropy based spider web plots and probabilistic neural network,” *Pattern Recognition Letters*, vol. 34, no. 16, pp. 2151–2156, Dec. 2013.

M. Yeo et al., “Evaluation of techniques to improve a deep learning algorithm for the automatic detection of intracranial haemorrhage on CT head imaging,” *European Radiology Experimental*, vol. 7, no. 1, p. 17, Apr. 2023.

G. Ntaios et al., “Machine-learning-derived model for the stratification of cardiovascular risk in patients with ischemic stroke,” *Journal of Stroke and Cerebrovascular Diseases*, vol. 30, no. 10, Oct. 2021, Art. no. 106018.

L. Ali et al., “An automated diagnostic system for heart disease prediction based χ2 on chi-square statistical model and optimally configured deep neural network,” *IEEE Access*, vol. 7, pp. 34938–34945, 2019.

A. Phaphuangwittayakul et al., “An optimal deep learning framework for multi-type hemorrhagic lesions detection and quantification in head CT images for traumatic brain injury,” *International Journal of Speech Technology*, vol. 52, no. 7, pp. 7320–7338, May 2022.

H. Z. U. Rehman, H. Hwang, and S. Lee, “Conventional and deep learning methods for skull stripping in brain magnetic resonance imaging: A review,” *IEEE Access*, vol. 9, pp. 103952–103976, 2021.

R. B. Dubey, M. Hanmandlu, and S. K. Gupta, “Region splitting and merging technique for brain tumor segmentation,” *Journal of Medical Engineering & Technology*, vol. 36, no. 8, pp. 399–403, 2012.

A. Al-Shabi, “Machine learning algorithms for classifying medical images of different modalities: A comparative analysis,” *Computer Methods and Programs in Biomedicine*, vol. 202, p. 105988, May 2021.

K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint*, arXiv:1409.1556, 2014.

C. Szegedy et al., “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, 2015, pp. 1–9.

A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, Jun. 2017.

Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, May 2015.

K. Suzuki, “Overview of deep learning in medical imaging,” *Radiological Physics and Technology*, vol. 10, no. 3, pp. 257–273, Sep. 2017.

A. T. Tzeng, J. E. Guest, and S. M. Pieper, “Image analysis tools for neuroimaging data,” *NeuroImage*, vol. 85, no. 1, pp. 210–226, Jan. 2014.

X. Zhang, S. Zhang, and Y. Yan, “Stroke detection using deep learning models based on CT images,” *Biocybernetics and Biomedical Engineering*, vol. 42, no. 2, pp. 610–620, 2022.

P. Rajpurkar et al., “CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning,” *arXiv preprint*, arXiv:1711.05225, 2017.

A. Rehman, I. Mehmood, and M. S. Riaz, “Deep learning-based framework for automatic diagnosis of COVID-19 using chest X-ray images,” *Neural Computing and Applications*, vol. 33, no. 10, pp. 4561–4575, May 2021.

A. Islam and Y. Zhang, “Brain tumor segmentation and volume estimation from MR images using deep learning,” *Neural Networks*, vol. 130, pp. 202–212, 2020.

W. H. Chang, H. S. Lin, and T. Y. Lin, “Automatic hemorrhage detection in CT images using 3D convolutional neural networks,” *Medical Image Analysis*, vol. 53, pp. 197–207, Jul. 2019.