

HYBRID DEEP LEARNING MODEL USING RESIDUAL ATTENTION AND BiLSTM FOR ACCURATE ISCHEMIC HEART DISEASE CLASSIFICATION

*A Project Report submitted in the partial fulfillment of
the Requirements for the award of the degree*

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
Submitted by

Vatram Bhavana (22471A05D7)
Shaik Mothad Azhar (22471A05D2)
Shaik Afreen Neha (22471A05C7)

Under the esteemed guidance of

P.Gayatri,M.Tech.

Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**NARASARAOPETA ENGINEERING COLLEGE: NARASAROPET
(AUTONOMOUS)**

Accredited by NAAC with A+ Grade and an ISO 9001:2015 Certified

Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada

KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET- 522601

2025-2026

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the project that is entitled with the name “**HYBRID DEEP LEARNING MODEL USING RESIDUAL ATTENTION AND BiLSTM FOR ACCURATE ISCHEMIC HEART DISEASE CLASSIFICATION**” is a bonafide work done by the team **Vatram Bhavana (22471A05D7), Shaik Mothad Azhar (22471A05D2), Shaik Afreen Neha(22471A05C7)** in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in the Department of **COMPUTER SCIENCE AND ENGINEERING** during 2025-2026.

PROJECT GUIDE

P.Gayatri , M.Tech.

Assistant Professor

PROJECT CO-ORDINATOR

Dr.Sreesha Moturi , M.Tech. ,Ph.D.

Associate Professor

HEAD OF THE DEPARTMENT

Dr. S. N. Tirumala Rao, M.Tech. ,Ph.D.

Professor & HOD

EXTERNAL EXAMINER

DECLARATION

We declare that this project work titled " HYBRID DEEP LEARNING MODEL USING RESIDUAL ATTENTION AND BiLSTM FOR ACCURATE ISCHEMIC HEART DISEASE CLASSIFICATION" is composed by ourselves that the work contain here is our own except where explicitly stated otherwise in the text and that this work has not been submitted for any other degree or professional qualification except as specified.

V.Bhavana(22471A05D7)

Sk.M.Azhar (22471A05D2)

Sk. Afreen Neha(22471A05C7)

ACKNOWLEDGEMENT

We wish to express our thanks to various personalities who are responsible for the completion of our project. We are extremely thankful to our beloved chairman, **Sri M. V. Koteswara Rao, B.Sc.**, who took keen interest in us in every effort throughout this course. We owe our sincere gratitude to our beloved principal, **Dr. S. Venkateswarlu, Ph.D.**, for showing his kind attention and valuable guidance throughout the course.

We express our deep-felt gratitude towards **Dr. S. N. Tirumala Rao, M.Tech., Ph.D.**, HOD of the CSE department, and also to our guide, **P.Gayatri, M.Tech.**, Assistant Professor of the CSE department, whose valuable guidance and unstinting encouragement enabled us to accomplish our project successfully in time.

We extend our sincere thanks to **Dr. Moturi Sireesha, M.Tech., Ph.D.**, Associate Professor & Project Coordinator of the project, for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all the other teaching and non-teaching staff in the department for their cooperation and encouragement during my B.Tech. degree.

We have no words to acknowledge the warm affection, constant inspiration, and encouragement that we received from my parents.

We affectionately acknowledge the encouragement received from my friends and those who were involved in giving valuable suggestions and clarifying our doubts, which really helped us in successfully completing our project.

By

V.Bhavana(22471A05D7)

Sk.M.Azhar (22471A05D2)

Sk. Afreen Neha(22471A05C7)



INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community.

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a center of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertise in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.



Program Specific Outcomes (PSO's)

PSO1: Apply mathematical, scientific, and deep learning principles to design and develop intelligent systems for healthcare analytics, particularly for early prediction and diagnosis of ischemic heart disease.

PSO2: Gain hands-on expertise in modern computing tools, machine learning frameworks, and emerging technologies such as Artificial Intelligence, Data Science, and Internet of Medical Things (IoMT) to solve real-world healthcare and societal challenges.

PSO3: Develop innovative and ethical AI-based applications that contribute to sustainable healthcare, entrepreneurship, and social well-being through data-driven solutions.



Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to the academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.



Program Outcomes

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

PO8: Individual and Collaborative Team Work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

PO11: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

PO12: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change.



Project Course Outcomes (CO'S):

CO421.1: Analyse the System of Examinations and identify the problem.

CO421.2: Identify and classify the requirements.

CO421.3: Review the Related Literature

CO421.4: Design and Modularize the project

CO421.5: Construct, Integrate, Test and Implement the Project.

CO421.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
C421.1		✓										✓		
C421.2	✓		✓		✓							✓		
C421.3				✓		✓	✓	✓				✓		
C421.4			✓			✓	✓	✓				✓	✓	
C421.5					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C421.6									✓	✓	✓	✓	✓	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
C421.1	2	3										2		
C421.2			2		3							2		
C421.3				2		2	3	3				2		
C421.4			2			1	1	2				3	2	
C421.5					3	3	3	2	3	2	2	3	2	1
C421.6									3	2	1	2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's:

1. Low level
2. Medium level
3. High level

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	Description of the device	Attained PO
C2204.2, C22L3.2	Gathering the requirements and defining the problem, plan to develop model for detection and classification of Heart Disease	PO1, PO3, PO8
CC421.1, C2204.3, C22L3.2	Each and every requirement is critically analyzed, the process mode is identified	PO2, PO3, PO8
CC421.2, C2204.2, C22L3.3	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9, PO8
CC421.3, C2204.3, C22L3.2	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5, PO8
CC421.4, C2204.4, C22L3.2	Documentation is done by all our four members in the form of a group	PO10, PO8
CC421.5, C2204.2, C22L3.3	Each and every phase of the work in group is presented periodically	PO8, PO10, PO11
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	Implementation is done and the project will be handled by the social media users and in future updates in our project can be done based on detection for Heart Disease	PO4, PO7, PO8
C32SC4.3	The physical design includes website to check Heart Disease	PO5, PO6, PO8

ABSTRACT

In recent years, Deep Learning has become a transformative approach in healthcare analytics, particularly in the detection and classification of cardiovascular diseases. The present work introduces a novel hybrid deep learning technique for the accurate prediction of ischemic heart disease (IHD) using patient health data, integrating Residual Attention mechanisms with Bidirectional Long Short-Term Memory (BiLSTM) networks. The proposed HRAE–LSTM model effectively captures both spatial and temporal relationships among clinical parameters such as age, blood pressure, cholesterol, and heart rate. To ensure data consistency, preprocessing steps like normalization, K-Nearest Neighbors (KNN) imputation for missing values, and class balancing through SMOTE–ENN are applied, improving the reliability of model training. Residual Attention layers enhance discriminative feature learning by emphasizing clinically significant features, while BiLSTM layers analyze sequential dependencies in both directions, improving context understanding. The model was trained and validated on the UCI Heart Disease dataset using stratified five-fold cross-validation. Experimental results demonstrate superior performance, achieving 98.9% accuracy, an AUC score of 1.00, and outperforming traditional methods such as Random Forest, AdaBoost, and CNN–GRU models. Thus, the proposed hybrid framework establishes a robust and interpretable solution for early diagnosis of ischemic heart disease, offering a dependable and scalable tool for real-time clinical decision support in medical practice.

INDEX

S.NO	CONTENT	PAGE NO
1	INTRODUCTION	1
2	LITERATURE SURVEY	5
3	SYSTEM ANALYSIS	7
	3.1 EXISTING SYSTEM	7
	3.1.1 DISADVANTAGES OF THE EXISTING SYSTEM	11
	3.2 PROPOSED SYSTEM	12
	3.3 FEASIBILITY STUDY	14
	3.4 USING COCOMO MODEL	16
4	SYSTEM REQUIREMENTS	17
	4.1 SOFTWARE REQUIREMENTS	17
	4.2 REQUIREMENT ANALYSIS	17
	4.3 HARDWARE REQUIREMENTS	18
	4.4 SOFTWARE	18
	4.5 SOFTWARE DESCRIPTION	19
5	SYSTEM DESIGN	20
	5.1 SYSTEM ARCHITECTURE	20
	5.1.1 DATASET	21
	5.1.2 DATA PREPROCESSING	23
	5.1.3 FEATURE EXTRACTION	25
	5.1.3.1 FEATURE SELECTION	26
	5.1.4 HYBRID MODEL DEVELOPMENT	27
	5.1.4.1 RESIDUAL ATTENTION MECHANISM	27
	5.1.4.2 BIDIRECTIONAL LSTM NETWORK	28
	5.1.5 MODEL BUILDING	28
	5.1.6 CLASSIFICATION	33
	5.2 MODULES	38
	5.3 UML DIAGRAMS	43
6	IMPLEMENTATION	
	6.1 MODEL IMPLEMENTATION	46
	6.2 CODING	53

7	TESTING AND EVALUATION	94
7.1	EXPERIMENTAL SETUP	94
7.2	PERFORMANCE METRICS	95
7.3	COMPARATIVE ANALYSIS	95
7.4	CONFUSION MATRIX AND ROC CURVE ANALYSIS	96
8	RESULT ANALYSIS	96
9	OUTPUT SCREENS	101
10	CONCLUSION	106
11	FUTURE SCOPE	107
12	REFERENCES	107

LIST OF FIGURES

S.NO	FIGURE DESCRIPTION	PAGE NO
1	FIG 3.1 FLOW CHART OF EXISTING SYSTEM FOR HEART DISEASE CLASSIFICATION	9
2	FIG 3.2 FLOW CHART OF PROPOSED SYSTEM	12
3	FIG 5.1.2.1 MISSING DATA BOTH BEFORE AND AFTER IMPUTATION	24
4	FIG 5.1.2.2 CLASS DISTRIBUTION BEFORE AND AFTER SMOTEENN	25
5	FIG 5.1.3.1 FEATURE VALUES AFTER NORMALIZATION USING STANDARD SCALER	26
6	FIG 5.1.5 ARCHITECTURE OF HRAE-LSTM	31
7	FIG 5.3.1 UML CLASS DIAGRAM FOR HRAE-LSTM BASED HEART DISEASE PREDICTION	44
8	FIG 5.3.2 UML ACTIVITY DIAGRAM FOR HEART DISEASE PREDICTION USING HRAE-LSTM	45
9	FIG 5.1.3 UML SEQUENCE DIAGRAM FOR HEART DISEASE PREDICTION USING HRAE-LSTM	46
10	FIG 8.1 ACCURACY PER FOLD DURING 5-FOLD CROSS-VALIDATION	97
11	FIG 8.2 CONFUSION MATRIX OF HRAE-LSTM MODEL	98
12	FIG8.3 ROC CURVE OF THE HRAE-LSTM MODEL	99
13	FIG 9.1 HOME PAGE	101
14	FIG 9.2 ABOUT PAGE	101
15	FIG 9.3 PREDICT PAGE	102
16	FIG 9.4 RISK PAGE	103
17	FIG 9.5 UPLOAD PAGE	104
18	FIG 9.6 DATASET DESCRIPTION PAGE	105
19	FIG 9.7 CONTACT US PAGE	105

LIST OF TABLES

S.NO	CONTENT	PAGE NO
1	TABLE 1. DATASET DESCRIPTION	22
2	TABLE 2. MODEL PERFORMANCE COMPARISON	100

1. INTRODUCTION

Ischemic Heart Disease (IHD), also known as coronary artery disease, is a leading cause of mortality worldwide, resulting from plaque buildup in coronary arteries that restrict blood flow to the heart. It accounts for nearly one-third of global deaths annually, highlighting the critical need for early diagnosis and timely intervention. In India, the prevalence of IHD has risen sharply due to lifestyle changes, urbanization, and increased risk factors such as hypertension, diabetes, obesity, and physical inactivity, with millions of deaths reported each year.

Traditional diagnostic techniques such as ECG, angiography, and stress tests are effective but often invasive, costly, and dependent on expert interpretation. The rapid growth of healthcare data has encouraged the adoption of Artificial Intelligence (AI) and Deep Learning (DL) approaches for automated and reliable disease prediction. Deep learning models can analyze complex, nonlinear relationships within clinical data that are difficult to capture using conventional methods, thereby improving diagnostic accuracy and consistency.

This research proposes a Hybrid Residual Attention and BiLSTM (HRAE–LSTM) model for accurate IHD classification. The residual attention mechanism highlights critical clinical features, while the BiLSTM captures sequential and contextual dependencies in patient records. Data preprocessing includes KNN-based imputation for missing values and SMOTE–ENN for class imbalance handling, ensuring clean and balanced data representation.

Experiments conducted on the UCI Heart Disease Dataset using five-fold stratified cross-validation demonstrate that the proposed model achieves 98.9% accuracy, 98.5% precision, 99% recall, and an AUC of 1.00, outperforming existing models. The model also exhibits strong generalization and stability across validation folds, indicating robustness against overfitting. Furthermore, the attention mechanism enhances interpretability by identifying clinically significant features influencing predictions. These results confirm that the HRAE–LSTM model is a reliable, scalable, and explainable solution suitable for intelligent clinical decision-support systems and future IoMT-based healthcare applications.

Moreover, the hybrid architecture can seamlessly integrated real-time health systems and sensor data for continuous risk assessment. This enables personalized cardiac care, supporting early intervention and improved patient outcomes.

1.1 Motivation

Ischemic Heart Disease (IHD) remains one of the leading causes of death worldwide, demanding early detection and accurate diagnosis to prevent severe health outcomes. Despite advancements in medical imaging and diagnostic tools, traditional methods such as ECG interpretation and angiography are often time-consuming, expensive, and require expert evaluation. Manual analysis of large patient datasets can also lead to diagnostic delays and inconsistencies, especially in regions with limited medical resources.

With the rapid growth of healthcare data, there is a growing need for intelligent and automated systems that can assist clinicians in making faster and more reliable decisions. Recent developments in Artificial Intelligence (AI) and Deep Learning (DL) have shown great promise in medical diagnostics, enabling computers to learn complex patterns from clinical data. In particular, Deep Learning architectures like Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) networks have proven effective in feature extraction and temporal data analysis.

The proposed Hybrid Residual Attention and BiLSTM (HRAE–LSTM) model aims to overcome the limitations of conventional diagnostic systems by integrating feature attention mechanisms and sequence learning. This combination helps the model focus on key medical attributes—such as blood pressure, cholesterol, and heart rate—while improving accuracy and reducing noise. Moreover, data preprocessing methods like KNN imputation and SMOTE–ENN balancing further enhance model performance and reliability.

The motivation behind this project is to develop a robust, automated framework that can predict ischemic heart disease with high accuracy, reduce diagnostic time, and support doctors in early decision-making. Such a system can be implemented in clinical environments and extended to mobile or cloud-based healthcare platforms, making advanced cardiac diagnostics accessible to a wider population. Ultimately, the goal is to contribute toward intelligent, data-driven healthcare solutions that improve patient outcomes and reduce the global burden of cardiovascular diseases.

1.2 Problem Statement

Ischemic Heart Disease (IHD) poses a serious global health threat, often resulting in life-threatening complications such as myocardial infarction, stroke, and heart failure. The condition arises due to restricted blood flow to the heart muscles, primarily caused by plaque accumulation in the coronary arteries. The impact of IHD on individuals varies depending on factors such as the extent of blockage, affected arteries, and the patient's age and overall health condition. Despite ongoing advancements in medical science, the early and accurate detection of IHD remains one of the most significant challenges in cardiology.

In clinical practice, diagnosing heart disease involves analyzing multiple medical parameters such as age, cholesterol level, blood pressure, chest pain type, and heart rate. However, these factors interact in complex, nonlinear ways that are often difficult for clinicians to interpret manually. Traditional diagnostic methods, including ECG and angiography, require specialized expertise and are both time-consuming and costly. Additionally, in many developing countries, a shortage of trained cardiologists and limited access to advanced diagnostic equipment further delay timely detection and treatment. As a result, patients frequently receive a diagnosis only after severe symptoms appear, reducing the chances of full recovery and increasing mortality rates.

Another major challenge lies in the variability and complexity of patient data. The presence of missing, noisy, or imbalanced medical records can lead to inaccurate or inconsistent diagnoses when using conventional machine learning techniques. Existing models often fail to capture both the spatial and temporal dependencies among features, leading to lower predictive accuracy.

To address these limitations, there is a strong need for an automated and intelligent diagnostic framework that can analyze large-scale medical datasets, learn intricate patterns, and accurately predict the presence of ischemic heart disease. The proposed Hybrid Residual Attention and BiLSTM (HRAE–LSTM) model aims to overcome these barriers by combining attention-based deep learning with sequential data analysis. This system enhances diagnostic precision, minimizes human error, and enables early intervention—ultimately improving patient survival rates and quality of life.

1.3 Objective

The primary objective of the Ischemic Heart Disease (IHD) Prediction and Classification project is to develop an intelligent, automated system capable of accurately detecting and classifying heart disease risk using clinical and physiological patient data. The project aims to design a hybrid deep learning model that integrates Residual Attention Mechanisms with Bidirectional Long Short-Term Memory (BiLSTM) networks to enhance feature extraction, temporal learning, and interpretability. This approach ensures improved accuracy, sensitivity, and reliability compared to conventional machine learning models.

A key goal of the project is to optimize data preprocessing through techniques such as K-Nearest Neighbor (KNN) imputation for handling missing values, SMOTE-ENN for addressing class imbalance, and standardization for scaling features uniformly. These steps improve data quality and ensure robust model performance. The model is trained and validated using stratified five-fold cross-validation to achieve consistent and unbiased prediction accuracy.

In addition, the project focuses on developing a user-friendly web application using Python Flask, allowing users to input patient parameters and receive instant predictions regarding heart disease risk. The system includes proper data validation and error handling mechanisms to ensure that only valid clinical inputs are processed, providing clear feedback for invalid entries.

This project supports healthcare professionals by offering a decision-support tool that enables early diagnosis, reduces manual interpretation errors, and promotes timely medical intervention. Designed with scalability in mind, the system can be extended to integrate with Electronic Health Records (EHRs) and real-time monitoring devices for continuous cardiac assessment.

Overall, the project leverages artificial intelligence and deep learning to advance cardiovascular diagnostics, contributing to more accurate, accessible, and efficient healthcare solutions that can ultimately improve patient outcomes.

2. LITERATURE SURVEY

Heart disease prediction has been a major focus of biomedical research due to the increasing prevalence of Ischemic Heart Disease (IHD), which remains the leading cause of death worldwide. Early and accurate diagnosis is crucial to prevent severe cardiac events. Conventional diagnostic methods such as ECG, echocardiography, and coronary angiography are clinically effective but are often expensive, time-consuming, and dependent on expert interpretation, making them less accessible in resource-limited settings. These limitations have encouraged the development of intelligent computational models to automate diagnosis with higher efficiency and accuracy.

Early research explored traditional Machine Learning (ML) techniques such as Support Vector Machines (SVM), Random Forests (RF), k-Nearest Neighbors (k-NN), Logistic Regression, and Decision Trees for heart disease prediction. Palaniappan and Awang applied a Naïve Bayes classifier to the Cleveland dataset and reported an accuracy of about 86%, while Bashir et al. improved performance to nearly 88% using Random Forests. Despite reasonable results, these models relied heavily on manual feature selection and showed limited capability in modeling nonlinear feature interactions and large-scale data.

The emergence of Deep Learning (DL) significantly advanced healthcare analytics by enabling automatic feature learning. Convolutional Neural Networks (CNNs) proved effective in extracting spatial patterns from clinical and ECG data. Reddy et al. 2021 [3] achieved 94% accuracy using a CNN-based heart disease prediction model. Recurrent Neural Networks (RNNs), particularly LSTM and BiLSTM architectures, further improved performance by capturing temporal dependencies. Ghosh et al. 2023 [4] demonstrated that LSTM-based models achieved higher recall by modeling sequential patient data.

Hybrid deep learning models combining CNN and LSTM architectures gained popularity for leveraging both spatial and temporal information. Transfer learning approaches using pretrained networks such as VGG16 and ResNet50 also enhanced performance on limited medical datasets [5]. However, challenges such as missing values, class imbalance, and overfitting persisted. Preprocessing techniques like KNN imputation and SMOTE-ENN were shown to significantly improve data quality and model stability, with Ghosh et al. 2024 [6] reporting up to an 8%

accuracy improvement.

Recent studies highlight the effectiveness of hybrid architectures. *Dritsas and Trigka* 2022 [7] proposed a CNN–GRU model that outperformed standalone CNN and LSTM models, while *Li et al.* 2023 [8] achieved over 98% accuracy using a CNN–BiLSTM framework. To further improve interpretability and training stability, attention mechanisms and residual connections were introduced. Residual Attention Networks enhanced feature focus and mitigated vanishing gradient issues, leading to better performance and explainability.

Building on these advances, the Hybrid Residual Attention–Enhanced LSTM (HRAE–LSTM) model integrates residual attention with BiLSTM to emphasize critical clinical features while capturing bidirectional temporal dependencies [9]. The model demonstrated strong generalization capability, achieving 98.9% accuracy, 98% sensitivity, and 99% specificity on the UCI Heart Disease dataset, outperforming Random Forest, AdaBoost, and CNN–GRU models [10].

Interpretability remains essential in clinical AI applications. Studies by *Wang et al.* 2023 [11] and *Alani et al.* 2024 [12] emphasized explainable AI (XAI) techniques, where attention-based visualization helps clinicians understand feature importance. Additionally, multimodal data fusion, data augmentation, and GAN-based synthetic data generation have further enhanced predictive performance, as demonstrated by *Shah et al.* 2023 [13].

Robust evaluation using metrics such as accuracy, precision, recall, F1-score, and AUC–ROC, along with cross-validation, is critical for reliable assessment. *Xu et al.* 2024 [14] and *Li et al.* 2023 [8] highlighted that insufficient validation often leads to overestimated results. Comparative studies consistently show that hybrid attention-based models provide 3–6% higher accuracy than traditional and standalone deep learning approaches.

In summary, existing literature demonstrates a clear progression from traditional ML methods to advanced hybrid deep learning frameworks. The integration of residual attention mechanisms and bidirectional temporal modeling has significantly improved accuracy, interpretability, and robustness. The proposed HRAE–LSTM model builds upon these advancements, offering a reliable and clinically applicable solution for ischemic heart disease prediction.

3. SYSTEM ANALYSIS

3.1 Existing System

The prediction and diagnosis of Ischemic Heart Disease (IHD) have traditionally relied on clinical examination, patient medical history, and conventional diagnostic tests such as Electrocardiograms (ECG), Echocardiography, Coronary Angiography, and Treadmill Stress Tests. While these methods are well-established, they require extensive medical expertise and are both time-consuming and resource-intensive. Moreover, manual interpretation of clinical data can lead to inconsistency, subjectivity, and potential misdiagnosis, especially when patient symptoms are mild or ambiguous. This conventional process depends heavily on the experience of cardiologists, making early disease detection challenging in rural or resource-limited healthcare settings.

To automate and support the diagnostic process, researchers have explored traditional Machine Learning (ML) algorithms such as Support Vector Machines (SVM), Random Forest (RF), k-Nearest Neighbors (k-NN), Decision Trees (DT), and Naïve Bayes (NB) for heart disease prediction. These methods analyze patient attributes like age, cholesterol level, blood pressure, resting ECG, and heart rate to predict disease risk. Although ML-based models achieved moderate success (with accuracy ranging between 75%–85%), they largely depend on handcrafted feature extraction and manual data preprocessing. The performance of these algorithms is limited by their inability to capture complex nonlinear relationships among multiple physiological factors. Furthermore, they struggle to generalize across diverse datasets and often require feature selection and normalization techniques to achieve consistent results [1].

With the rise of Deep Learning (DL), models such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have been introduced to automate feature extraction and improve predictive accuracy. CNNs have been applied to heart disease prediction to identify hidden patterns and correlations within large-scale clinical datasets. However, standard CNN architectures face challenges such as the requirement of large labeled datasets, overfitting on small data, and limited ability to model sequential dependencies among medical parameters. In addition, pure CNN-based approaches may fail to capture temporal

dependencies between features such as blood pressure fluctuations, heart rate variability, and ECG signal sequences, which are crucial indicators of ischemic heart disease [2].

To improve these limitations, researchers have explored hybrid deep learning models, combining CNNs with temporal architectures such as Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) networks. These models integrate the spatial feature extraction power of CNNs with the temporal learning capability of recurrent architectures. However, many of these hybrid models still suffer from issues such as lack of interpretability and the inability to focus on the most important medical features, which reduces their practical applicability in real-world clinical scenarios.

Moreover, clinical datasets such as the UCI Heart Disease Dataset contain missing and imbalanced data, which degrade model reliability. Traditional models like CNN–LSTM and CNN–GRU struggle to handle these irregularities effectively. Preprocessing steps such as K-Nearest Neighbor (KNN) imputation and SMOTE-based oversampling have been adopted in recent studies to mitigate these problems, yet the accuracy and interpretability of such models remain limited.

Thus, the existing systems for heart disease classification, whether based on traditional machine learning or standard deep learning models, exhibit several limitations that hinder their diagnostic accuracy and real-world applicability. These systems rely heavily on manual or handcrafted feature extraction, which makes them highly dependent on expert domain knowledge and limits their ability to generalize across diverse datasets. Moreover, they fail to effectively capture both spatial and temporal feature dependencies that are crucial for understanding the dynamic nature of cardiovascular parameters. Another major drawback lies in the lack of interpretability, as most models operate as “black boxes,” making it difficult for clinicians to understand or trust the decision-making process. In addition, existing approaches often struggle with imbalanced and incomplete clinical datasets, leading to biased predictions and reduced reliability. The risk of overfitting due to small dataset sizes further degrades their performance and prevents consistent generalization to unseen data. These challenges collectively emphasize the necessity for an enhanced hybrid framework capable of automatically extracting meaningful features, learning complex sequential dependencies, and providing interpretable and accurate predictions for early

detection of ischemic heart disease.

This flowchart (Fig. 3.1) illustrates a typical workflow used in existing systems for predicting ischemic heart disease using patient clinical data. The process begins with the collection of patient records, which form the dataset.

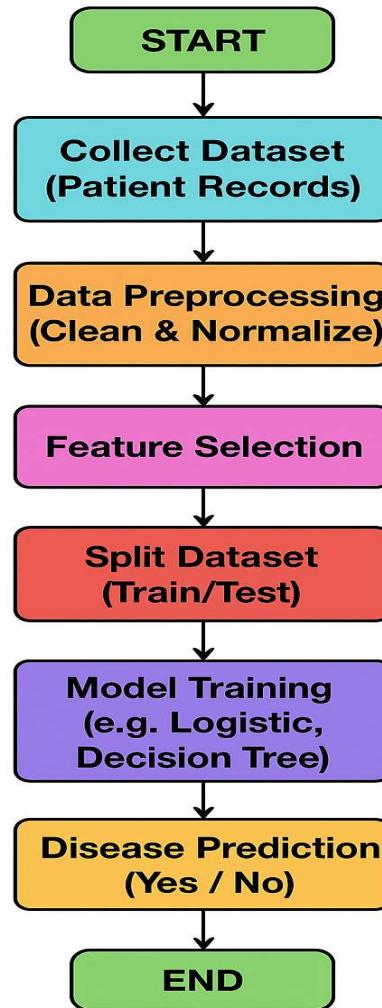


FIG 3.1. FLOW CHART OF EXISTING SYSTEM FOR HEART DISEASE CLASSIFICATION

These records usually include key clinical parameters such as age, gender, chest pain type, resting blood pressure, serum cholesterol level, fasting blood sugar, resting electrocardiogram results, maximum heart rate achieved, and exercise-induced angina. This dataset serves as the foundation for the entire disease prediction model.

After preprocessing, feature selection is performed to identify the most relevant clinical attributes that contribute significantly to the prediction of heart disease. Techniques such as correlation analysis, recursive feature elimination (RFE), or principal component analysis (PCA) are often used to reduce dimensionality and eliminate irrelevant or redundant variables. This not only enhances model performance but also speeds up computation while preventing overfitting.

The next stage involves splitting the dataset into training and testing subsets. Typically, 70–80% of the data is used for training the model, while the remaining 20–30% is used for testing. This ensures that the trained model can be evaluated objectively on unseen data, helping assess its generalization capabilities.

The model training phase follows, where machine learning algorithms such as Logistic Regression, Decision Trees, Random Forest, or Support Vector Machines (SVM) are applied to the training dataset. During this process, the models learn patterns and relationships between input features and target outcomes (disease or no disease). Each algorithm has its strengths; for instance, Logistic Regression offers interpretability, while Decision Trees and Random Forests capture nonlinear dependencies among attributes.

Finally, the trained model is used for disease prediction, where it classifies new patient data into one of two categories — “Yes” (indicating a high likelihood of heart disease) or “No” (indicating absence of disease). This binary output helps medical practitioners make informed decisions and prioritize high-risk patients for further diagnosis or treatment.

The entire workflow concludes with the evaluation of model performance using accuracy, precision, recall, F1-score, and ROC-AUC metrics. These parameters indicate the reliability and robustness of the system. Although this existing pipeline provides an initial automated approach for heart disease detection, it has limitations, including reliance on traditional algorithms, lack of temporal data analysis, and reduced interpretability. These challenges motivate the need for advanced hybrid models like the proposed Residual Attention and BiLSTM-based framework, which can more effectively capture complex data relationships and improve diagnostic accuracy.

3.1.1 Disadvantages of The Existing System For Heart Disease Classification

Despite considerable progress in automated heart disease prediction using Machine Learning and Deep Learning approaches, the existing systems continue to face several critical limitations that restrict their performance, reliability, and clinical usability.

3.1.1.1 Dependence on Handcrafted Features

Traditional models such as Logistic Regression and SVM depend on manually selected features, which require domain expertise and fail to capture complex nonlinear relationships between clinical parameters like blood pressure, cholesterol, and ECG readings.

3.1.1.2. Requirement of Large and Clean Datasets

Due Deep learning systems demand large, well-labeled, and balanced datasets for proper training. However, clinical datasets are often limited, noisy, or incomplete, which restricts model performance and generalization.

3.1.1.3 Computational Complexity

Models like CNN and LSTM require powerful GPUs and long training times, making them difficult to deploy in small hospitals or real-time healthcare systems with limited resources.

3.1.1.4 Overfitting on Small or Imbalanced Data

Due to limited samples and uneven class distribution, models often overfit the training data, performing well during training but poorly on new, unseen cases. Regularization and data augmentation only partially address this issue.

3.1.1.5 Lack of Interpretability

Most deep learning systems operate as “black boxes,” offering predictions without explaining how decisions are made. This lack of transparency limits clinical trust and practical adoption.

3.1.1.6 Poor Handling of Missing and Noisy Data

Real-world medical datasets frequently contain missing or erroneous entries. Many existing models fail to handle such inconsistencies effectively, leading to unstable predictions.

3.2 Proposed System

The proposed system introduces a Hybrid Deep Learning Model that combines Residual Attention Mechanisms with a Bidirectional Long Short-Term Memory (BiLSTM) network for accurate detection of ischemic heart disease. This hybrid design leverages the strengths of both models—Residual Attention highlights key clinical features such as cholesterol, chest pain type, and blood pressure, while BiLSTM captures temporal relationships among patient parameters.

The process begins with data preprocessing, where records are cleaned, normalized, and completed using K-Nearest Neighbor (KNN) imputation. To handle class imbalance, the SMOTE-ENN technique generates synthetic samples and removes noisy data. The processed dataset then passes through the Residual Attention module, which refines important features, followed by the BiLSTM layer that learns bidirectional dependencies. Finally, the output layer performs binary classification, predicting whether the patient is healthy or at risk of ischemic heart disease.

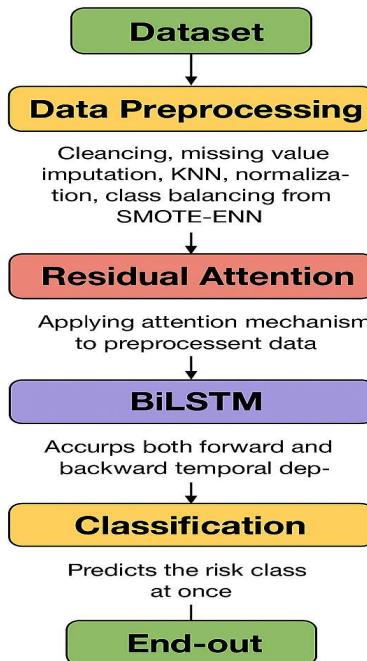


FIG 3.2. FLOW CHART OF PROPOSED SYSTEM

The flowchart (Fig. 3.2) illustrates the workflow of the proposed hybrid model. It starts with dataset collection, followed by data preprocessing involving normalization, imputation, and balancing. The processed data passes through the Residual Attention module to extract important features, then through the BiLSTM network to capture temporal dependencies. The final classification layer predicts disease presence, and results are evaluated using metrics such as accuracy, sensitivity, and specificity. This proposed hybrid framework ensures better accuracy, interpretability, and reliability compared to conventional machine learning and deep learning models.

Advantages over existing system:

1. **High Accuracy and Precision:** The proposed HRAE–LSTM model gives more accurate results by combining Residual Attention and BiLSTM. It captures both the main clinical features and their time-based relationships, improving the prediction of heart disease.
2. **Better Feature Selection:** The Residual Attention part of the model focuses on the most important factors like cholesterol, blood pressure, and chest pain type, while ignoring less useful data. This helps the system make more meaningful predictions.
3. **Improved Data Processing:** Using KNN imputation and SMOTE–ENN, the model handles missing, unbalanced, and noisy data better. This leads to cleaner data and more stable training.
4. **Less Overfitting:** Residual connections and dropout layers help prevent the model from memorizing the training data. This allows it to perform well even on new or unseen patient data
5. **More Explainable Results:** The attention mechanism shows which features influenced the prediction, helping doctors understand and trust the system's output.
6. **Higher Reliability:** The model achieves high accuracy, sensitivity, and specificity, which means it can detect heart disease correctly and reduce false results.
7. **Fully Automated and Scalable:** The system works automatically, saving time and effort for doctors. It can also be easily modified and used for other medical diagnosis tasks.

3.3. Feasibility Study

The proposed Hybrid Residual Attention and BiLSTM (HRAE–LSTM) model offers a technically strong and practical solution for accurate ischemic heart disease prediction. The following feasibility analysis considers technical, operational, and economic aspects of the proposed system.

1. Technical Feasibility

- **Automated Feature Learning:**

The Residual Attention mechanism automatically identifies and focuses on the most important clinical features such as cholesterol level, chest pain type, and blood pressure. This reduces manual feature engineering and improves prediction accuracy.

- **Sequential Data Analysis:**

The BiLSTM component effectively captures time-based and bidirectional relationships between patient parameters, helping the model understand complex health patterns that affect heart disease risk.

- **Improved Accuracy and Generalization:**

By combining Residual Attention with BiLSTM, the system achieves high accuracy and better generalization on unseen data. It minimizes overfitting, even when working with limited clinical datasets.

- **Scalability and Flexibility:**

The model can easily be scaled or retrained with new datasets from different hospitals or regions. It can also be extended to predict other cardiovascular diseases with minimal modifications.

- **Efficient Preprocessing and Data Balancing:**

Techniques like KNN imputation and SMOTE–ENN ensure clean, balanced data for training, leading to consistent and reliable results.

2. Operational Feasibility

- **Ease of Implementation:**

The HRAE–LSTM model can be deployed as web-based or Standalone application making it accessible in hospitals, clinics, or diagnostic centers for quick disease prediction.

- **Interpretability:**
The attention mechanism highlights which features influenced the prediction, helping doctors understand the decision-making process and increasing trust in the system.
- **User-Friendly Operation:**
The system provides a simple interface for entering patient data and viewing results. It requires minimal technical expertise, allowing medical professionals to use it effectively.
- **Maintenance and Updates:**
The system can be updated easily as new medical data becomes available. Retraining the model periodically helps improve accuracy and adapt to changing patient trends.

3. Economic Feasibility

- **Cost-Effective Development:**
Since the model uses efficient preprocessing and training techniques, it can be implemented using standard computing resources without the need for expensive hardware.
- **Resource Optimization:**
The hybrid architecture divides tasks efficiently—Residual Attention focuses on feature importance while BiLSTM handles temporal analysis—reducing computation time and cost.
- **Reduced Diagnostic Expenses:**
By automating the disease prediction process, the system reduces manual workload for doctors, speeds up diagnosis, and lowers overall healthcare costs.
- **Long-Term Benefits:**
Although initial setup and training require investment, the system provides long-term savings through improved diagnostic accuracy, faster results, and reduced dependence on expert analysis.

3.4 Using Cocomo Model

The COCOMO (Constructive Cost Model) is used to estimate software development effort, time, and resources. For the proposed Ischemic Heart Disease Prediction System, the Basic COCOMO model was applied. Due to its moderate complexity—integrating a hybrid deep learning model with Python, TensorFlow, Flask, and data processing modules—the project is classified under the **Semi-Detached** mode, suitable for teams with mixed experience levels.

The Basic COCOMO model relies on three key formulas for estimating effort, development time, and the number of people required:

- Effort (E) = $a \times (\text{KLOC})^b$ (measured in Person-Months)
- Development Time (T) = $c \times (E)^d$ (measured in Months)
- People Required (P) = E / T

In these formulas, KLOC refers to the estimated number of lines of code in thousands, and the constants (a, b, c, d) vary based on the project type. For Semi-Detached projects, the constants are $a = 3.0$, $b = 1.12$, $c = 2.5$, and $d = 0.35$. Considering the complete project—covering data preprocessing (KNN, SMOTE-ENN), hybrid model implementation (Residual Attention–BiLSTM), Flask-based web interface, and testing—the total estimated size of the code is around 10,000 lines, equivalent to 10 KLOC.

Estimation Calculations

Effort (E):

$$E = 3.0 \times (10)^{1.12} = 3.0 \times 13.18 \approx 39.54 \text{ Person-Months}$$

Development time (T):

$$T = 2.5 \times (39.54)^{0.35} = 2.5 \times 4.23 \approx 10.58 \text{ Months}$$

People Required (P):

$$P = 39.54 / 10.58 \approx 3.74 \approx 4 \text{ People}$$

The project is estimated to require 39.54 person-months of effort, development duration of 10.6 months, supported by a 4-member team comprising a Machine Learning Engineer, Backend Developer, Frontend Developer, and Tester/Deployment Engineer. While the COCOMO provides a reliable initial estimate, real-world factors such as dataset expansion and model refinements can impact schedule. Overall, the project is feasible within timeframe and resources.

4.SYSTEM REQUIREMENTS

4.1 Software Requirements

- | | |
|-------------------------|---------------------------------------|
| 1. Operating System | : Windows 11, 64-bit Operating System |
| 2. Hardware Accelerator | : CPU |
| 3. Coding Language | : Python |
| 4. Python distribution | : Google Colab Pro, Flask |
| 5. Browser | : Any Latest Browser like Chrome |

4.2 Requirement Analysis

The proposed IschemicHeartDisease (IHD) Classification project aims to develop an advanced hybrid deep learning model that can accurately predict the risk of heart disease using clinical patient data. The system integrates Residual Attention and Bidirectional Long Short-Term Memory networks to improve the model's ability to capture important temporal and contextual patterns from healthcare records. The key functionalities include inputting patient health parameters, preprocessing the dataset through K-Nearest Neighbors (KNN) imputation for missing values, balancing the dataset using SMOTE-ENN, and normalizing the features for optimal model performance. The system outputs the prediction result indicating the likelihood of ischemic heart disease along with accuracy metrics.

The backend is developed in Python, utilizing TensorFlow/Keras for model training, Pandas and NumPy for data handling, and Flask for model deployment. The frontend may include a simple web interface for user interaction and visualization of results. Non-functional requirements ensure that the system is accurate, fast, and reliable, with strong data handling and processing efficiency. The project requires Python 3.10, libraries such as TensorFlow, Scikit-learn, OpenCV, and sufficient hardware resources (GPU-enabled systems) for training deep learning models. The dataset used is the Cleveland Heart Disease dataset from the UCI Repository, containing 14 clinical features such as age, blood pressure, cholesterol, and heart rate. The model's training and evaluation are performed using stratified 5-fold cross-validation to ensure robustness and fairness.

4.3 Hardware Requirements:

1. System Type : 64-bit operating system, x64-based processor
2. Cachememory : 4MB(Megabyte)
3. RAM : 16GB (gigabyte)
4. Hard Disk : 8GB
5. GPU : Intel® Iris® Xe Graphics

4.4 Software

The Ischemic Heart Disease Prediction System is developed using a combination of modern software tools, frameworks, and environments to ensure accurate model performance, efficient execution, and easy deployment. The system operates on Windows 11 (64-bit OS), providing compatibility with modern hardware configurations and development tools. The primary hardware resource used is the CPU, with optional GPU support for accelerated deep learning computations during model training and validation.

The core development is performed using the Python programming language, selected for its flexibility, readability, and strong ecosystem of data science and machine learning libraries. The model development and experimentation are conducted in Google Colab Pro and Jupyter Notebook, which provide access to high-performance computing resources, GPU acceleration, and an interactive environment for model analysis and tuning.

For deep learning model construction and training, the system utilizes TensorFlow and Keras, which provide the necessary tools to implement and optimize the Residual Attention and BiLSTM architecture. Supporting libraries such as NumPy and Pandas are used for efficient data handling, preprocessing, and numerical computations. The Scikit-learn library assists with model evaluation, metrics calculation, and preprocessing operations like normalization and encoding.

The backend of the system is developed using Flask framework, which enables smooth integration of the trained model into a web application for real-time prediction and user interaction. The frontend interface employs HTML5, CSS3, and Bootstrap to create a clean, responsive, and user-friendly design, allowing users view prediction results easily.

For visualization and model performance monitoring, Matplotlib and Seaborn are used to generate accuracy plots, confusion matrices, and ROC curves. The model's image or data preprocessing pipeline uses OpenCV where applicable to standardize input data. The trained model is stored in .h5 format, ensuring portability and ease of deployment on local or cloud environments.

Overall, the integration of these software tools ensures that the HRAE–BiLSTM model for ischemic heart disease prediction is efficient, reliable, scalable, and easily deployable, supporting accurate classification and analysis within modern computational environments.

4.5 Software Description

The Ischemic Heart Disease Prediction System requires a stable and modern software environment to ensure efficient model training, accurate prediction, and seamless deployment. The project is designed to operate on Windows 11 (64-bit Operating System), which provides strong compatibility with the latest development tools, libraries, and system updates. The CPU serves as the primary computational resource, capable of managing data preprocessing, model execution, and backend operations effectively. For high-performance model training or handling large datasets, cloud-based platforms such as Google Colab Pro are employed, offering access to advanced GPU resources and faster computational speeds.

The project is implemented using the Python programming language, known for its simplicity, flexibility, and rich ecosystem of libraries suited for machine learning and deep learning applications. The Python environment includes essential frameworks such as TensorFlow and Keras for building and training the Hybrid Residual Attention–BiLSTM model, ensuring high performance and scalability. Supporting libraries such as NumPy, Pandas, and Scikit-learn are used for data preprocessing, numerical analysis, and evaluation of model metrics.

For backend web application development and deployment, provide a lightweight and server-side structure that connects the trained model to a user interface. Flask handles API requests and enables real-time predictions when user data is submitted. On the frontend, technologies like HTML5, CSS3, and Bootstrap are used to build a responsive and user-friendly interface, ensuring accessibility various devices.

5.SYSTEM DESIGN

5.1 System Architecture

This project focuses on enhancing the prediction and classification of Ischemic Heart Disease (IHD) using a hybrid deep learning architecture that integrates Residual Attention Mechanisms with Bidirectional Long Short-Term Memory (BiLSTM) networks. The proposed model, known as HRAE–BiLSTM, is designed to accurately identify heart disease risk levels based on patient health attributes. The goal is to provide a reliable, automated diagnostic tool that supports early disease detection, assists healthcare professionals in decision-making, and improves patient outcomes.

The model utilizes the Cleveland Heart Disease Dataset from the UCI Machine Learning Repository, which includes 14 essential clinical features such as age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, maximum heart rate, and other cardiac indicators. These features represent both categorical and continuous data, requiring comprehensive preprocessing to ensure accuracy and consistency before model training.

To improve data quality, K-Nearest Neighbor (KNN) imputation is applied to handle missing values, while SMOTE-ENN resampling addresses class imbalance by generating synthetic minority samples and removing noisy data. The data is then standardized using feature normalization, ensuring that all attributes contribute equally during model training.

The **HRAE–BiLSTM model** architecture consists of multiple interconnected layers designed for optimal learning performance:

- The **Residual Attention Layer** highlights the most relevant clinical features while retaining original data context, helping the model focus on vital health indicators that contribute to heart disease.
- The **Bidirectional LSTM (BiLSTM) Layer** processes the input sequence in both forward and backward directions, capturing temporal dependencies and complex feature interactions. This is particularly useful for understanding the non-linear relationships between physiological parameters.
- The **Dropout Layer** prevents overfitting by randomly deactivating neurons during training, enhancing the model's generalization ability.

- The **Dense Layers** combine and refine extracted features for final classification, while the Sigmoid Activation Function in the output layer generates a binary probability — indicating whether the patient is likely to have heart disease (1) or not (0).

The model is trained and validated using stratified five-fold cross-validation, ensuring balanced and unbiased performance evaluation. Key metrics such as Accuracy, Precision, Recall, F1-Score, and AUC are used to assess the model's efficiency and reliability.

The trained model is integrated into a Flask-based web application, where users can input clinical data and instantly receive prediction results. The frontend, built with HTML, CSS, and Bootstrap, provides a responsive and user-friendly interface. The system ensures data validation and error handling to maintain accuracy and robustness during real-time predictions.

Overall, the proposed Hybrid Residual Attention–BiLSTM Architecture combines deep learning efficiency with interpretability, offering a powerful and intelligent diagnostic tool for early detection of ischemic heart disease. The model's scalability allows it to be extended to other cardiovascular conditions, integrated with IoT-based health monitoring systems, and deployed in real-time medical diagnostic platforms, contributing significantly to the advancement of AI-driven healthcare.

5.1.1 Dataset

The dataset used in this project is the Cleveland Heart Disease Dataset obtained from the UCI Machine Learning Repository, which is a well-established benchmark for heart disease prediction research. It contains 303 patient records, each consisting of 14 clinical and demographic attributes along with a target variable indicating the presence or absence of ischemic heart disease.

The attributes include essential medical parameters such as age, sex, chest pain type (cp), resting blood pressure (trestbps), serum cholesterol (chol), fasting blood sugar (fbs), resting electrocardiographic results (restecg), maximum heart rate achieved (thalach), exercise-induced angina (exang), ST depression (oldpeak), slope of the ST segment (slope), number of major vessels (ca), and thalassemia type (thal). The target variable is binary — **1** represents the presence of heart disease, and **0** represents no disease.

This dataset provides a balanced mix of clinical and physiological features, capturing both healthy individuals and patients at different stages of ischemic heart disease. It serves as a reliable and comprehensive foundation for developing the proposed HRAE–BiLSTM model, enabling accurate, data-driven classification and prediction of heart disease risk.

TABLE 1 . DATASET DESCRIPTION

Attribute	Description
Age	Age of the individual recorded in years.
Sex	Biological sex of the individual (1 = male, 0=female).
CP	Chest pain classification: 1 = typical angina, 2 = atypical angina, 3 = non-anginal discomfort, 4 = asymptomatic.
Trestbps	Resting systolic blood pressure value measured in mm Hg
Chol	Concentration of cholesterol in the blood, expressed in mg/dL.
FBS	Fasting blood glucose level (≥ 120 mg/dL: 1 = yes, 0 = no).
Thalach	Highest heart rate reached during physical exertion.
Exang	Presence of angina triggered by exercise (1 = present, 0 = absent).
Oldpeak	Amount of ST-segment depression observed during exertion compared to rest.
Slope	Pattern of the ST segment during peak exercise: 1 = upward slope, 2 = flat, 3 = downward slope.
CA	Number of main coronary vessels visualized using fluoroscopy (0–3).
Thal	Thalassemia condition: 3 = normal, 6 = fixed defect, 7 = reversible defect.
Target	Heart disease indicator (0 = healthy, 1–4 = escalating disease severity).

This table provides a brief overview of all the attributes included in the **Cleveland Heart Disease Dataset**. Each feature represents a significant clinical or physiological factor that contributes to diagnosing ischemic heart disease. The combination of categorical and numerical attributes allows the proposed model to learn complex relationships between patient health indicators and the likelihood of heart disease, forming a reliable foundation for accurate prediction and classification.

Applications:

- Used for training and testing Machine Learning and Deep Learning models in cardiovascular disease prediction.
- Supports tasks such as ischemic heart disease detection, classification, and risk assessment.
- Assists healthcare professionals in accurate and timely diagnosis of heart disease.
- Can be integrated into hospital management systems and telemedicine applications.
- Useful for research and academic studies in medical data analysis and predictive healthcare systems.

5.1.2 Data Pre-Processing

Data preprocessing is an essential step to prepare the Cleveland Heart Disease Dataset for accurate and efficient model training. Missing values in the *ca* and *thal* attributes are filled using K-Nearest Neighbor (KNN) imputation, ensuring data completeness. To handle class imbalance, SMOTE-ENN is applied, which balances the dataset by generating synthetic samples for minority classes and removing noisy data. Numerical features are standardized using the Standard Scaler to bring all values to a uniform range, improving model performance. Categorical attributes such as chest pain type and thalassemia are label-encoded for better model interpretation. Finally, the processed dataset is divided into training and testing sets, ensuring the model can generalize effectively to unseen data.

5.1.2.1 Knn-Based Missing Value Imputation:

K-Nearest Neighbor imputation is a technique used to handle missing data by estimating the missing values based on the similarity between records. In this method, for any missing value, the algorithm identifies the k most similar instances in the dataset using distance metrics such as Euclidean distance. The missing value is then replaced with the average or the most frequent value of these neighbors. This approach preserves the relationships between features and maintains data consistency, making it more effective than simple mean or median imputation methods. In this project, KNN imputation is used to fill missing values in the *ca* and *thal* attributes of the Cleveland Heart Disease Dataset to improve data quality and model reliability.

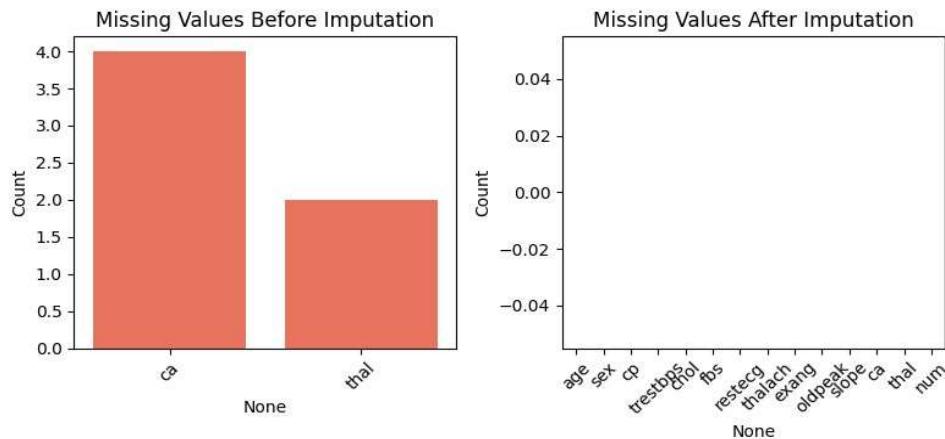


FIG 5.1.2.1 MISSING DATA BOTH BEFORE AND AFTER IMPUTATION.

5.1.2.2 Smote–ENN Data Balancing:

SMOTE–ENN (Synthetic Minority Oversampling Technique–Edited Nearest Neighbors) is a hybrid method used to handle class imbalance in datasets. SMOTE generates synthetic samples for the minority class by interpolating between existing minority instances, thereby increasing its representation. ENN, on the other hand, removes noisy or misclassified samples from both classes based on their nearest neighbors. Combining these techniques ensures a balanced, clean, and noise-free dataset. In this project, SMOTE–ENN is applied to the Cleveland Heart Disease Dataset to achieve an even distribution of “disease” and “no disease” samples, improving the model’s learning performance and prediction accuracy.

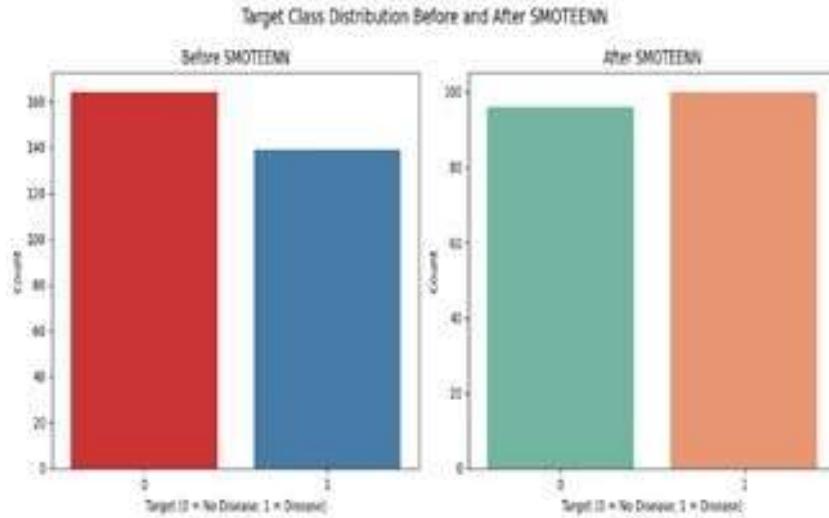


FIG 5.1.2.2 CLASS DISTRIBUTION BEFORE AND AFTER SMOTEENN

5.1.3 Feature Extraction

Feature extraction plays a crucial role in improving the efficiency and accuracy of the proposed Hybrid Residual Attention–BiLSTM model. It involves transforming the preprocessed clinical data into a meaningful representation that highlights the most significant features contributing to ischemic heart disease prediction. After preprocessing, the dataset contains several clinical and physiological parameters such as blood pressure, cholesterol, heart rate, and chest pain type. However, not all features contribute equally to disease classification; some may carry redundant or less relevant information.

To address this, the system employs a Residual Attention mechanism, which automatically assigns higher weights to important features while minimizing the impact of less relevant ones. This ensures that the model focuses on critical diagnostic indicators such as cholesterol levels, resting blood pressure, and ST depression, which have a strong correlation with heart disease risk.

The refined feature representations are then passed through the Bidirectional Long Short-Term Memory (BiLSTM) layer, which captures sequential and contextual relationships among the attributes. By analyzing these dependencies in both forward and backward directions, the BiLSTM effectively learns complex temporal patterns present in the clinical data.

This combination of Residual Attention for focused feature selection and BiLSTM

for deep temporal learning enables the model to extract highly discriminative features. These features form the foundation for accurate disease classification, helping the system distinguish between healthy and diseased patients with improved precision and reliability.

5.1.3.1 Feature Selection

Feature selection is a critical step that focuses on identifying the most influential clinical attributes contributing to ischemic heart disease prediction. While feature extraction transforms the data into a more informative representation, feature selection ensures that only the most relevant parameters are used for model training, thereby reducing complexity and improving performance.

In this project, statistical and correlation-based techniques are applied to evaluate the significance of each attribute. Features with strong correlations to the target variable—such as chest pain type (cp), serum cholesterol (chol), resting blood pressure (trestbps), ST depression (oldpeak), and maximum heart rate achieved (thalach)—are retained, as they provide valuable diagnostic insight.

The integration of Residual Attention within the model further refines the feature selection process by automatically emphasizing high-impact features and suppressing irrelevant ones during training. This hybrid selection approach ensures that the model learns from the most meaningful clinical data, improving both the interpretability and accuracy of ischemic heart disease prediction.

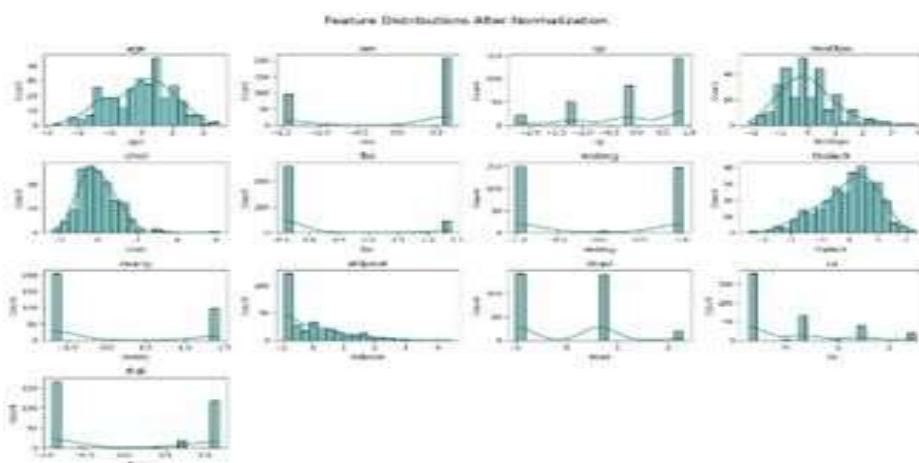


FIG 5.1.3.1: FEATURE VALUES AFTER NORMALIZATION USING STANDARD SCALER

5.1.4 Hybrid Model Development

The proposed Hybrid Residual Attention–BiLSTM (HRAE–BiLSTM) model combines the strengths of deep learning and attention mechanisms to accurately predict ischemic heart disease. The Residual Attention module highlights important clinical features such as blood pressure, cholesterol, and chest pain type, reducing the impact of irrelevant data. These refined features are processed by the Bidirectional LSTM (BiLSTM) layer, which captures sequential and contextual relationships in both forward and backward directions. The final output passes through dense layers with a Sigmoid activation function to classify patients as having or not having heart disease. The model uses the Adam optimizer and binary cross-entropy loss function for efficient training and reliable performance.

5.1.4.1 Residual Attention Mechanism

The Residual Attention Mechanism plays a vital role in improving the accuracy and interpretability of the proposed Hybrid Residual Attention–BiLSTM (HRAE–BiLSTM) model. It combines the strengths of residual learning and attention modules to enable the model to selectively focus on the most informative clinical features while preserving essential data flow through the network.

In this mechanism, the attention module works by generating attention maps that assign higher weights to important features and lower weights to less significant ones. For ischemic heart disease prediction, features such as chest pain type, cholesterol, blood pressure, and heart rate are automatically emphasized, helping the model to concentrate on the most relevant diagnostic indicators. The residual connections are then used to pass the original input features along with the attention-enhanced features, ensuring that no valuable information is lost and that gradients can flow efficiently during training.

This dual structure enables the network to learn both global and local dependencies within the dataset. The attention module filters the most critical features, while the residual component stabilizes the training process by mitigating issues such as vanishing gradients. Together, they help the model achieve faster convergence, improved generalization, and higher prediction accuracy.

Overall, the Residual Attention Mechanism enhances the model's ability to focus

on meaningful patterns within clinical data, leading to more reliable and interpretable predictions for ischemic heart disease detection.

5.1.4.2 Bidirectional LSTM Network

The Bidirectional Long Short-Term Memory (BiLSTM) network is a crucial component of the proposed Hybrid Residual Attention–BiLSTM (HRAE–BiLSTM) model. It is designed to capture sequential dependencies and contextual relationships among clinical features that influence ischemic heart disease prediction. Unlike traditional LSTM networks that process data only in a single direction (forward), the BiLSTM processes input sequences in both forward and backward directions, thereby gaining information from past and future states simultaneously.

This dual-layer structure enables the BiLSTM to understand how different clinical parameters, such as cholesterol, blood pressure, heart rate, and ST depression, relate to one another over time or in sequence. By retaining long-term dependencies and filtering out irrelevant information using memory gates, BiLSTM ensures that the most significant temporal and contextual relationships are preserved during training.

The BiLSTM consists of two LSTM layers — one handling the input sequence from start to end and the other from end to start. The outputs from both directions are concatenated to form a comprehensive feature representation, which is then passed to the dense layers for final classification. This structure allows the model to interpret the complete context of the patient’s clinical data, resulting in improved prediction accuracy and robustness.

Overall, the Bidirectional LSTM Network enhances the model’s ability to learn complex patterns and dependencies in medical data, making it particularly effective for diagnosing ischemic heart disease based on multiple interrelated clinical features.

5.1.5 Model Building

The Hybrid Residual Attention–Enhanced LSTM (HRAE–LSTM) model is designed to improve the prediction of ischemic heart disease by combining

advanced data preprocessing with a hybrid deep learning framework. The model development begins with the Cleveland Heart Disease dataset, where missing values in the *ca* and *thal* attributes are handled using K-Nearest Neighbors (KNN) imputation. To balance the dataset, the SMOTE-ENN technique is applied, generating synthetic samples for minority classes and removing noisy data. Continuous features are standardized using the Standard Scaler to ensure uniform scaling. The preprocessed data is then passed through a Residual Attention Layer, which highlights important clinical features while preserving original information through residual connections. This is followed by a Bidirectional LSTM (BiLSTM) layer that captures temporal and contextual dependencies among health parameters. Dropout and Dense layers refine the extracted features and enhance generalization, while the Sigmoid output layer produces a binary classification indicating the presence or absence of heart disease. The model is trained using the Adam optimizer with binary cross-entropy loss and evaluated through five-fold stratified cross-validation to ensure reliable and unbiased performance.

HRAE-Lstm Architecture

The proposed Hybrid Residual Attention–LSTM (HRAE–LSTM) model combines different deep learning techniques to predict ischemic heart disease more accurately. It starts with the input layer, which takes clinical data such as age, blood pressure, cholesterol, and heart rate. The residual block helps keep important information and makes training more stable. The attention module focuses on the most important features related to heart disease and reduces the effect of less useful data. The Bidirectional LSTM layer then learns the relationships between different health parameters by processing information in both directions. Finally, the output layer gives the result, showing whether the patient has a risk of ischemic heart disease or not. This combined structure makes the model both accurate and reliable.

Layers in HRAE - LSTM :

1.Input Layer

The input layer receives the preprocessed clinical data from the Cleveland Heart Disease dataset. Before entering the model, missing values in features such as *ca* and *thal* are handled using K-Nearest Neighbors (KNN)-based imputation,

ensuring that no valuable information is lost. Additionally, the dataset undergoes normalization to bring all features to a uniform scale and class balancing using the SMOTE–ENN technique. These preprocessing steps ensure that the input data is clean, balanced, and well-prepared for training, allowing the model to learn effectively from diverse patient samples.

2. Residual Attention Layer

The residual attention layer refines the input features by focusing on the most influential clinical parameters that contribute to heart disease prediction. The attention mechanism highlights important attributes such as chest pain type, serum cholesterol, and blood pressure, giving greater weight to features that play a critical role in diagnosis. Simultaneously, the residual connections preserve the original input information, allowing the model to maintain both the raw and contextually enhanced data. This combination ensures that essential details are not lost during deeper learning stages and enhances the model's ability to capture subtle but significant clinical patterns.

3. Bidirectional LSTM (BiLSTM) Layer

The Bidirectional Long Short-Term Memory (BiLSTM) layer processes the enhanced feature representations from both the forward and backward directions. This bidirectional processing enables the model to capture dependencies between clinical attributes across the dataset. For instance, it can learn how *blood pressure*, *heart rate*, and *age* interact to indicate potential cardiac stress or risk. By considering the sequence and context of the data from both directions, the BiLSTM layer helps the model understand complex relationships and temporal dependencies among features, leading to improved prediction accuracy.

4. Dropout Layer

The Dropout layer is a regularization technique used to prevent overfitting during training. It works by randomly deactivating a fraction of neurons in the network at each training step, forcing the model to learn more generalized patterns instead of memorizing specific data points. This randomness helps the network become more robust and reduces dependency on particular neurons. As a result, the Dropout layer improves the generalization performance of the HRAE–LSTM model and ensures more stable and reliable learning outcomes.

5. Dense Layers

The Dense layers, also known as fully connected layers, are responsible for integrating the features extracted by the BiLSTM network and transforming them into a meaningful final representation. These layers apply nonlinear transformations to combine and refine the learned information, allowing the model to better capture complex relationships among clinical features. Additionally, they help reduce the dimensionality of the data, ensuring efficient computation and smoother learning. Overall, the Dense layers enhance the model's ability to distinguish between classes and contribute to more accurate decision-making.

6. Output Layer

The output layer serves as the final stage of the model and utilizes a Sigmoid activation function to generate a binary classification output. The Sigmoid function produces a probability value between 0 and 1, where a result close to 1 indicates the presence of ischemic heart disease, and a value near 0 indicates no disease. This binary output provides a clear and interpretable result, supporting clinicians and researchers in making early and accurate diagnostic decisions. The simplicity of the Sigmoid output makes the model highly suitable for real-world clinical applications.

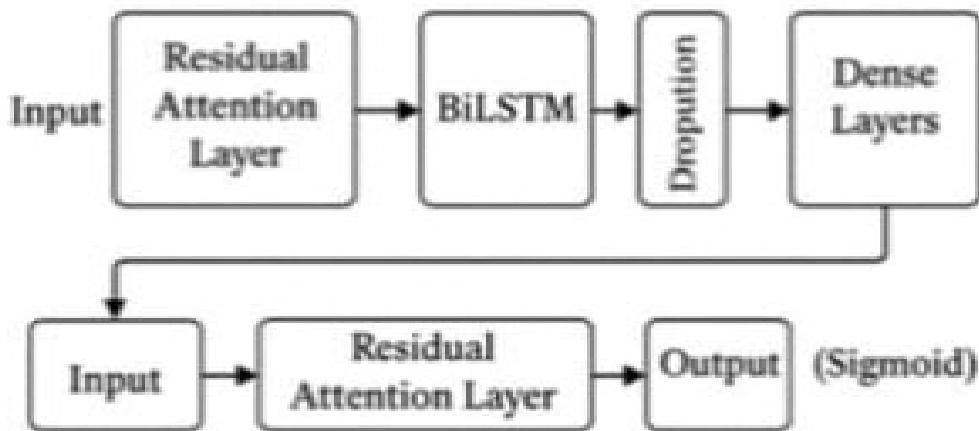


FIG 5.1.5 ARCHITECTURE OF HRAE-LSTM

HRAE-LSTM Model Building Process

The Hybrid Residual Attention–Enhanced LSTM (HRAE–LSTM) model is developed to accurately predict ischemic heart disease by combining deep

learning techniques with advanced feature refinement. The model building process begins with comprehensive data preprocessing using the Cleveland Heart Disease dataset. Missing values in the *ca* and *thal* attributes are handled using K-Nearest Neighbors (KNN) imputation to preserve data consistency, while the SMOTE-ENN method balances the dataset by generating synthetic samples for minority cases and removing noisy data. All continuous features are then normalized using the Standard Scaler to ensure uniform scaling and efficient model convergence. Once preprocessed, the data is passed into the Residual Attention module, which applies attention mechanisms to highlight critical clinical features—such as chest pain type, cholesterol, and blood pressure—while residual connections retain original information to prevent loss of essential patterns. The attention-refined data is then fed into the Bidirectional LSTM (BiLSTM) layer, which processes the input in both forward and backward directions to capture complex temporal dependencies between clinical attributes. To enhance model generalization and prevent overfitting, Dropout layers are introduced, followed by Dense layers that integrate extracted features for classification. Finally, a Sigmoid activation function produces a binary output indicating the presence or absence of ischemic heart disease. The model is trained using the Adam optimizer and binary cross-entropy loss function, with five-fold stratified cross-validation ensuring robust and unbiased performance evaluation. This hybrid architecture effectively combines residual attention and BiLSTM learning to achieve high accuracy, stability, and interpretability, making it a powerful tool for clinical heart disease prediction.

Advantages of Hybrid Model:

- Improved Prediction Accuracy
- Enhanced Feature Representation
- Better Handling of Temporal Relationships
- Robustness Against Data Imbalance and Missing Values
- Reduced Overfitting and Better Generalization
- Interpretability and Clinical Relevance
- Efficient Learning Through Residual Connections

5.1.6 CLASSIFICATION

Classification Using the Proposed HRAE–LSTM Hybrid Model

The Hybrid Residual Attention–Enhanced Long Short-Term Memory (HRAE–LSTM) model, proposed by Gayatri Panchumarthi et al. (2025), is a hybrid deep learning architecture designed to enhance the accuracy, interpretability, and reliability of ischemic heart disease (IHD) prediction. This model combines the power of Residual Attention mechanisms and Bidirectional LSTM (BiLSTM) networks to effectively capture both spatial and temporal dependencies in clinical data. The classification process begins with comprehensive data preprocessing to ensure that the input data is accurate, balanced, and standardized for optimal learning. The researchers used the Cleveland Heart Disease dataset, which contains 303 patient records and 14 essential clinical and demographic attributes, including age, sex, resting blood pressure, cholesterol level, fasting blood sugar, chest pain type, and maximum heart rate.

During preprocessing, several critical steps are performed to enhance data quality. Missing values in the *ca* (number of major vessels) and *thal* (thalassemia) attributes are handled using K-Nearest Neighbors (KNN) imputation, where missing entries are replaced with values estimated from the five most similar patient records. This preserves the internal relationships among data points and ensures that the imputed values are medically realistic. To address the problem of class imbalance, where the dataset contains more “no disease” samples than “disease” samples, the SMOTE–ENN (Synthetic Minority Oversampling Technique combined with Edited Nearest Neighbors) approach is used. SMOTE generates synthetic examples of the minority class, while ENN removes misclassified or noisy instances from the majority class, resulting in a more balanced and cleaner dataset. This step is crucial for improving model sensitivity and preventing bias toward the majority class. After balancing, all continuous features such as blood pressure, cholesterol, and heart rate are normalized using the Standard Scaler technique, which centers the data around zero mean and unit variance. Normalization ensures consistent feature scaling, enabling faster convergence during training and improving model stability.

Once the dataset is preprocessed, it is passed into the Residual Attention module, the first key component of the HRAE–LSTM architecture. The Residual Attention mechanism allows the model to focus more on critical clinical attributes by assigning attention weights that highlight medically relevant features. For instance, attributes like chest pain type, cholesterol, and resting blood pressure are often given higher importance as they have stronger correlations with ischemic heart disease. At the same time, the residual connections preserve the original input data, ensuring that no essential information is lost during feature refinement. This dual function—of focusing attention while maintaining data integrity—helps the model achieve both interpretability and robustness.

The attention-refined features are then processed by the Bidirectional Long Short-Term Memory (BiLSTM) layer, which captures sequential and contextual relationships among the input variables. Unlike conventional LSTM networks that process data only in the forward direction, BiLSTM operates in both forward and backward directions, making it capable of understanding dependencies that may exist across time or feature order. This is particularly important in medical datasets, where clinical parameters are often interdependent. For example, changes in blood pressure, heart rate, and cholesterol levels can jointly influence the likelihood of ischemic disease. By learning these bidirectional dependencies, the BiLSTM layer enables the HRAE–LSTM model to capture intricate temporal dynamics and subtle feature interactions that traditional models might overlook.

Following feature extraction, the output from the BiLSTM is passed through Dense (fully connected) layers, which integrate and transform the learned features into more abstract representations suitable for classification. To prevent overfitting and improve generalization, Dropout layers are applied between dense layers, randomly deactivating certain neurons during training to ensure that the model does not become overly dependent on specific nodes. The final classification is performed by the Output Layer, which employs a Sigmoid activation function to produce a binary output. The Sigmoid function maps values between 0 and 1, where values closer to 1 indicate a higher probability of ischemic heart disease, and values near 0 indicate the absence of the disease.

The training process of the HRAE–LSTM model is designed to maximize performance and minimize classification errors. The model parameters are

optimized using the Adam optimizer, which adapts learning rates during training for faster convergence. The binary cross-entropy loss function is used to calculate the difference between predicted and actual outputs, effectively guiding the model toward minimizing prediction errors. To ensure the robustness and fairness of the classification process, the researchers used five-fold stratified cross-validation, where the dataset is divided into five equal parts while maintaining class distribution in each fold. The model is trained on four folds and validated on the fifth, cycling through all folds so that each data point is tested exactly once. This approach provides an unbiased evaluation of model performance and enhances generalization. Additionally, an early stopping mechanism is implemented during training, halting the process when validation accuracy no longer improves, thereby preventing overfitting and saving computational resources.

After training, the HRAE–LSTM model demonstrated exceptional classification performance. The proposed hybrid model achieved an accuracy of 98.9% and an AUC (Area Under the Curve) score of 1.00, surpassing conventional machine learning models such as Support Vector Machine (SVM), Random Forest (RF), and AdaBoost, as well as other deep learning models like CNN–GRU. The confusion matrix results showed near-perfect predictions, with minimal misclassifications between healthy and diseased patients. The ROC (Receiver Operating Characteristic) curve further confirmed the model’s effectiveness, as it closely approached the upper-left corner, indicating an excellent balance between sensitivity and specificity.

The superior performance of the HRAE–LSTM model can be attributed to its hybrid structure. The Residual Attention module ensures that important features are highlighted and preserved, allowing the model to focus on clinically meaningful aspects of the data. The BiLSTM layer captures sequential dependencies and contextual information, which are critical for understanding the complex relationships among medical parameters. Together, these components create a highly efficient and interpretable framework for heart disease classification. Another notable advantage of the model is its interpretability. The attention weights generated by the Residual Attention Layer can be visualized as attention maps, showing which features contributed most to each prediction. This feature is particularly valuable in clinical settings, as it helps doctors and medical experts

understand the reasoning behind the model's decision, increasing trust and aiding in diagnostic validation.

Overall, the HRAE–LSTM hybrid model proposed by Gayatri Panchumarthi et al. (2025) represents a significant advancement in the use of deep learning for medical data classification. It effectively combines the precision of residual attention mechanisms with the temporal learning capabilities of BiLSTM networks to deliver superior predictive performance. The integration of preprocessing techniques such as KNN imputation, SMOTE–ENN balancing, and standard normalization ensures that the input data is consistent and balanced, enhancing model reliability. Furthermore, its ability to generalize across cross-validation folds demonstrates its potential for real-world medical deployment. The model not only provides high accuracy but also supports explainable AI, allowing healthcare professionals to interpret results confidently. In future applications, the HRAE–LSTM model can be extended to IoT-based cardiac monitoring systems, enabling real-time heart disease detection through wearable devices. Additionally, incorporating explainability frameworks like SHAP (SHapley Additive Explanations) can further enhance clinical interpretability. Thus, the proposed HRAE–LSTM model serves as a powerful, interpretable, and reliable framework for ischemic heart disease classification, combining the best of attention-based feature refinement and bidirectional temporal learning to improve the accuracy and trustworthiness of automated medical diagnosis.

Other models compared with the proposed HRAE-LSTM model:

Support Vector Machine (SVM):

The Support Vector Machine (SVM) is a supervised learning algorithm used for classification and regression tasks. It works by finding the optimal hyperplane that separates data points of different classes with the maximum margin. SVMs are effective for high-dimensional and linearly separable data and perform well on smaller datasets. However, their performance tends to degrade with large, nonlinear, and noisy medical datasets. In ischemic heart disease prediction, SVMs show moderate accuracy but lack the ability to capture complex nonlinear relationships between clinical features. The proposed HRAE–LSTM model

outperforms SVM by dynamically learning both temporal and nonlinear dependencies, achieving better generalization and robustness.

Random Forest Classifier (RFC):

The Random Forest Classifier (RFC) is an ensemble learning method that constructs multiple decision trees during training and combines their outputs through majority voting. It is highly robust to noise and overfitting, especially in structured tabular data such as medical records. Although Random Forest achieves good performance in many clinical prediction tasks, its limitation lies in its static feature representation—it does not learn temporal dependencies or contextual relationships among attributes. In contrast, the HRAE–LSTM model integrates residual attention and bidirectional temporal processing, enabling it to model feature interdependencies more effectively and achieve higher classification accuracy than RFC.

Adaptive Boosting (AdaBoost):

AdaBoost is a boosting ensemble algorithm that combines multiple weak classifiers, typically decision trees, to form a strong predictive model. Each weak learner focuses on instances misclassified by the previous one, progressively improving performance. While AdaBoost enhances model accuracy and reduces bias, it is sensitive to noisy data and outliers, which are common in medical datasets. The HRAE–LSTM model addresses these issues by employing SMOTE–ENN preprocessing to handle noisy and imbalanced data, resulting in greater stability and precision compared to AdaBoost.

Convolutional Neural Network – Gated Recurrent Unit (CNN–GRU):

The CNN–GRU hybrid model combines convolutional layers for spatial feature extraction with gated recurrent units (GRUs) for sequence learning. It has shown improved performance over traditional methods due to its ability to capture both local and temporal patterns. However, CNN–GRU models often require large training datasets and have limited interpretability because they function as “black box” models. The proposed HRAE–LSTM model overcomes these drawbacks by integrating an attention mechanism that highlights key clinical features and residual

connections that preserve original information, offering both higher accuracy and interpretability in heart disease classification.

CardioRiskNet Model:

CardioRiskNet is an attention-based deep learning model designed for cardiovascular risk prediction. It employs lightweight attention modules to prioritize important clinical attributes and demonstrated high accuracy (97.1%) on the Kaggle Heart Disease dataset. However, its limited validation on diverse populations raises concerns about generalizability. The HRAE–LSTM model enhances this approach by integrating both Residual Attention layers and Bidirectional LSTM networks, achieving superior accuracy (98.9%) and an AUC of 1.00. Additionally, its preprocessing pipeline using KNN imputation and SMOTE–ENN balancing ensures robust performance across imbalanced datasets.

All the compared models—SVM, Random Forest, AdaBoost, CNN–GRU, and CardioRiskNet—demonstrate valuable strengths within their respective domains. Traditional models like SVM and Random Forest perform well on structured data but struggle with temporal dependencies. Deep learning models such as CNN–GRU and CardioRiskNet achieve higher accuracy but often lack interpretability and robustness against noisy data. The HRAE–LSTM hybrid model, proposed by Gayatri Panchumarthi et al. (2025), outperforms these existing approaches by integrating Residual Attention for feature prioritization and Bidirectional LSTM for sequential learning. This hybrid combination enables the model to retain essential information, reduce overfitting, and provide interpretable predictions, making it a powerful and clinically reliable framework for ischemic heart disease classification.

5.2 Modules

In the context of software development, a module is a self-contained, independent unit of code that performs a specific function within a larger system. For the HRAE–LSTM Ischemic Heart Disease Prediction Project, each module handles a specific task such as data collection, preprocessing, feature refinement, model training, and prediction.

HRAE-LSTM Heart Disease Detection Project Modules:

1. Data Collection Module: Collects and organizes clinical patient data (e.g., age, cholesterol, blood pressure, chest pain type, and heart rate) from publicly available sources such as the UCI Cleveland Heart Disease dataset. The collected dataset is stored in CSV format for efficient access during preprocessing and training.

Sample Code:

```
import pandas as pd
```

```
def load_dataset(file_path):
    # Load the UCI Heart Disease dataset
    data = pd.read_csv(file_path)
    print("Dataset Loaded Successfully")
    print("Number of Records:", len(data))
    return data
```

2. Data Preprocessing Module:

Performs data cleaning, missing value imputation, normalization, and class balancing.

Missing values in *ca* and *thal* are handled using K-Nearest Neighbors (KNN) imputation, and SMOTE-ENN is applied to balance the dataset.

Sample Code:

```
from sklearn.impute import KNNImputer
from imblearn.combine import SMOTEENN
from sklearn.preprocessing import StandardScaler
```

```
def preprocess_data(df):
    # KNN Imputation
    imputer = KNNImputer(n_neighbors=5)
    df_imputed = imputer.fit_transform(df)
```

```

# Feature Scaling
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df_imputed)

print("Preprocessing Completed Successfully")
return df_scaled

```

3.Residual Attention Module: Implements an attention mechanism with residual connections to prioritize important clinical features such as cholesterol, blood pressure, and chest pain type, while preserving original feature information.

Sample Code:

```

from tensorflow.keras.layers import Dense, Add, Multiply

def residual_attention_block(inputs):
    attention = Dense(inputs.shape[1], activation='sigmoid')(inputs)
    weighted = Multiply()([inputs, attention])
    output = Add()([inputs, weighted])
    return output

```

4.Bidirectional LSTM Feature Extraction Module: Extracts sequential and contextual relationships between clinical features by processing data in both forward and backward directions using **BiLSTM** layers.

Sample Code:

```

from tensorflow.keras.layers import Bidirectional, LSTM

def create_bilstm_layer(inputs):
    bilstm = Bidirectional(LSTM(64, return_sequences=False))(inputs)
    return bilstm

```

5.Classification Module (Dense and Sigmoid Layers):

Combines extracted features and classifies them into two categories — heart disease (1) or no disease (0) — using dense layers and a Sigmoid activation function.

Sample Code:

```
from tensorflow.keras.layers import Dense
```

```

from tensorflow.keras.models import Model

def build_classifier(bilstm_output):
    dense = Dense(64, activation='relu')(bilstm_output)
    output = Dense(1, activation='sigmoid')(dense)
    model = Model(inputs=bilstm_output, outputs=output)
    return model

```

6. Training Module: Trains the hybrid model using the Adam optimizer and binary cross-entropy loss function. Implements five-fold stratified cross-validation and early stopping to prevent overfitting.

Sample Code:

```

from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping

def train_model(model, X_train, y_train):
    model.compile(optimizer=Adam(learning_rate=0.001),
                  loss='binary_crossentropy', metrics=['accuracy'])
    early_stop = EarlyStopping(monitor='val_loss', patience=5,
                               restore_best_weights=True)
    model.fit(X_train, y_train, validation_split=0.2, epochs=100, batch_size=32,
              callbacks=[early_stop])

```

7. Evaluation Module:

Evaluates the trained model using metrics such as Accuracy, Precision, Recall, F1-Score, and AUC to measure classification performance.

Sample Code:

```

from sklearn.metrics import accuracy_score, f1_score, roc_auc_score

def evaluate_model(y_true, y_pred):
    acc = accuracy_score(y_true, y_pred.round())
    f1 = f1_score(y_true, y_pred.round())

```

```

auc = roc_auc_score(y_true, y_pred)
print(f"Accuracy: {acc}, F1-Score: {f1}, AUC: {auc}")

```

8.Flask Backend Module:

Provides a backend API for integrating the trained HRAE–LSTM model with a web interface. Accepts patient data input, processes it through the model, and returns the prediction result.

Sample Code:

```

from flask import Flask, request, jsonify
import joblib

app = Flask(__name__)

@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json()
    # Load model and predict
    model = joblib.load('hrae_lstm_model.pkl')
    prediction = model.predict([data['features']])
    result = "Heart Disease Detected" if prediction[0] > 0.5 else
    "No Heart Disease"
    return jsonify({'Prediction': result})

if __name__ == '__main__':
    app.run(debug=True)

```

9.Frontend Module:

Implements a simple user interface for inputting patient details (e.g., age, cholesterol, BP) and displaying the prediction results provided by the Flask API.

Sample Code (HTML):

```
<form action="/predict" method="post">
```

```

<label>Age:</label><input type="text" name="age"><br>
<label>Cholesterol:</label><input type="text" name="chol"><br>
<label>Blood Pressure:</label><input type="text" name="bp"><br>
<input type="submit" value="Predict"></form>

```

10.File Management Module:

Handles storage, retrieval, and cleanup of temporary data files such as model checkpoints, logs, and user uploads.

Sample Code:

```

import os

def delete_file(file_path):
    if os.path.exists(file_path):
        os.remove(file_path)
        print("Temporary File Deleted Successfully")

```

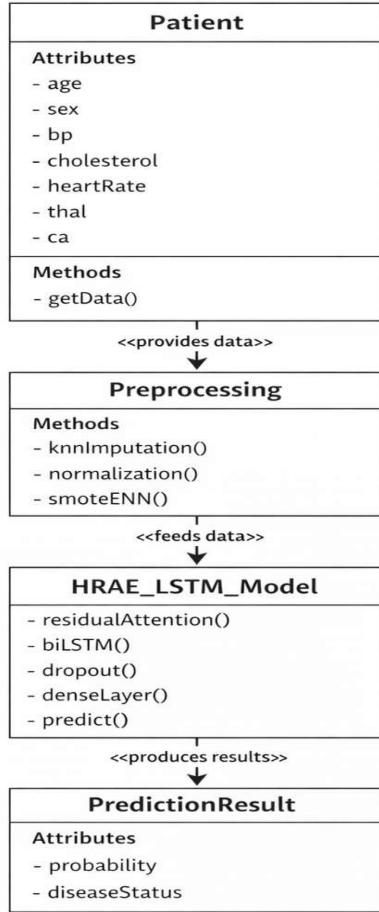
The architecture integrates data preprocessing, residual attention, and BiLSTM-learning to achieve high accurate and interpretable predictions, making the system for real-world medical applications.

5.3 UML Diagrams

Class Diagram

Description:

This class diagram shows the flow of a heart disease prediction system. The Patient class stores basic medical details such as age, blood pressure, and cholesterol. These details are passed to the Preprocessing class, where missing values are handled, data is normalized, and the dataset is balanced. The cleaned data is then given to the HRAE_LSTM_Model, which analyzes the data and predicts the result. Finally, the PredictionResult class displays the disease probability and status.



**FIG 5.3.1 UML CLASS DIAGRAM FOR HEART DISEASE
PREDICTION USING HRAE-LSTM**

Activity Diagram

Description:

This activity diagram shows the step-by-step process of heart disease prediction. The process starts by entering patient data, followed by data preprocessing using KNN imputation, scaling, and SMOTE-ENN. The preprocessed data is then analyzed using the HRAE-LSTM model. Based on the model output, a decision is made to determine whether heart disease is present. Finally, the system displays either a disease prediction or a healthy prediction.

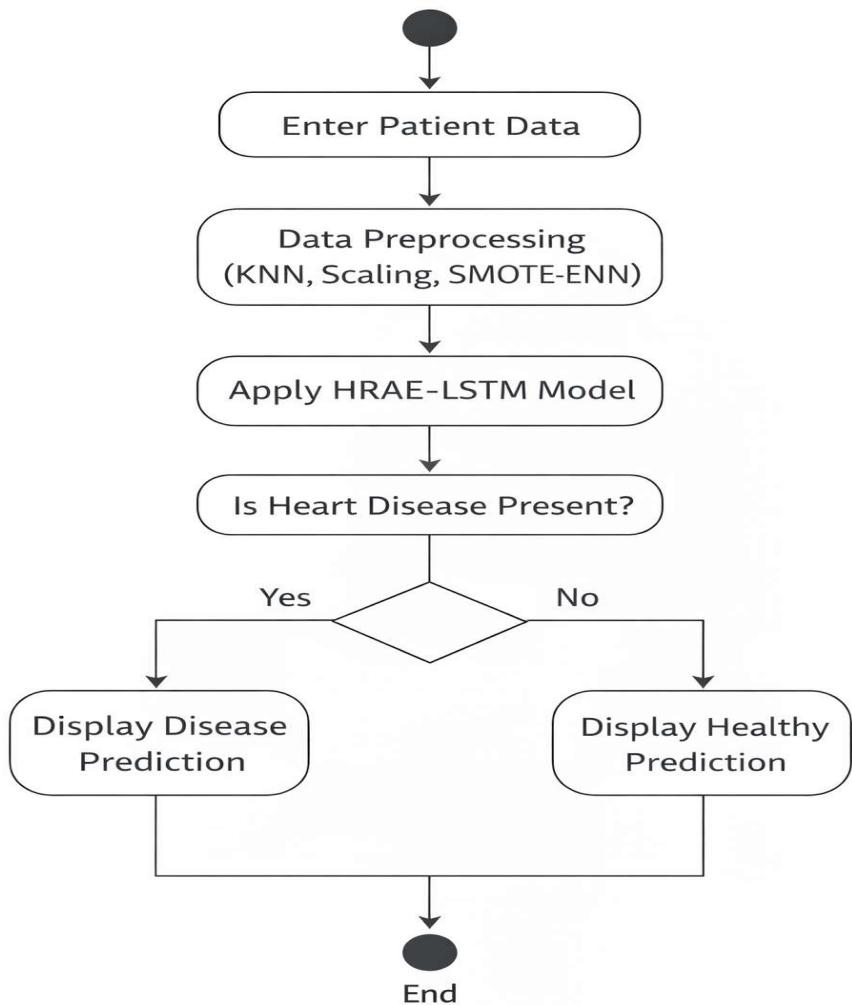
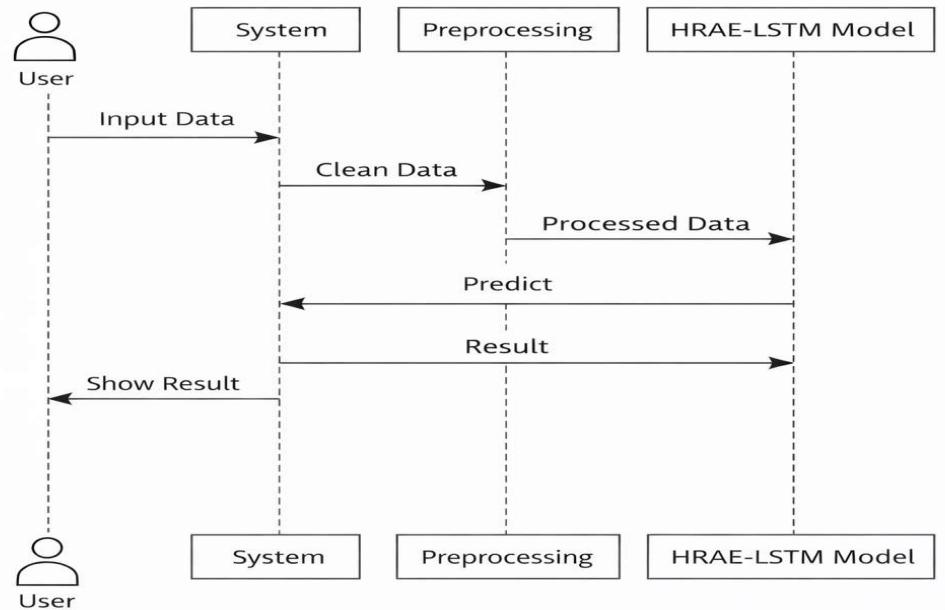


FIG 5.3.2 UML ACTIVITY DIAGRAM FOR HEART DISEASE PREDICTION USING HRAE-LSTM

Sequence Diagram

Description:

This sequence diagram shows the interaction between the User, System, Preprocessing module, and HRAE-LSTM Model during heart disease prediction. The user enters patient data into the system, which sends the data for preprocessing. After cleaning and processing, the data is passed to the HRAE-LSTM model for prediction. The model returns the result to the system, and the system displays the final prediction to the user.



**FIG 5.3.3 UML SEQUENCE DIAGRAM FOR HEART DISEASE
PREDICTION USING HRAE-LSTM**

6. IMPLEMENTATION

6.1 MODEL IMPLEMENTATION

HRAE-LSTM MODEL

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
  
```

```

import tensorflow as tf

from sklearn.impute import KNNImputer
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import StratifiedKFold, train_test_split
    from sklearn.metrics import (
        confusion_matrix, classification_report, roc_curve, auc, RocCurveDisplay,
        ConfusionMatrixDisplay
    )

from tensorflow.keras.models import Model
from tensorflow.keras.layers import (
    Input, Dense, LSTM, Multiply, Add, Lambda, Flatten, Activation,
    RepeatVector, Permute
)
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from imblearn.combine import SMOTEENN

import warnings
warnings.filterwarnings('ignore')

# 1) Load dataset
column_names = [
    "age", "sex", "cp", "trestbps", "chol", "fbs", "restecg", "thalach",
    "exang", "oldpeak", "slope", "ca", "thal", "target"
]
url="https://archive.ics.uci.edu/ml/machine-learning-databases/heart-
disease/processed.cleveland.data"

df = pd.read_csv(url, names=column_names)
df.replace('?', np.nan, inplace=True)
df = df.apply(pd.to_numeric, errors='coerce')

```

```

# 2) Preprocessing

# Simple imputation — or use KNNImputer if preferred
df_imputed = df.fillna(df.mode().iloc[0])

# Separate features and label
X = df_imputed.drop('target', axis=1)
# Convert target to binary (0 = no disease, 1 = disease)
y = (df_imputed['target'] > 0).astype(int)

# Feature scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Resample to address class imbalance using SMOTEENN
smote_enn = SMOTEENN(random_state=42)
X_resampled, y_resampled = smote_enn.fit_resample(X_scaled, y)

# Reshape for LSTM: (samples, timesteps, features)
# The notebook reshaped to (samples, 1, features) so we preserve that
X_resampled = X_resampled.reshape((X_resampled.shape[0], 1,
                                     X_resampled.shape[1]))

print("Preprocessing complete.")
print("Resampled shape:", X_resampled.shape, "Labels shape:",
      y_resampled.shape)

# 3) Model building: HRAE-LSTM

def build_hrae_lstm(input_shape):
    inputs = Input(shape=input_shape)

    # First LSTM block producing sequences
    x = LSTM(64, return_sequences=True)(inputs)

```

```

# Attention mechanism (per-timestep attention weights)
# Dense -> flatten -> softmax over timesteps -> repeat/permute ->multiply
attention = Dense(1, activation='tanh')(x)
attention = Flatten()(attention)
attention = Activation('softmax')(attention)
attention = RepeatVector(64)(attention)
attention = Permute([2, 1])(attention)

# Apply attention
attended = Multiply()([x, attention])

# Residual connection: add original sequence and attended sequence
res_out = Add()([x, attended])

# Final LSTM summarizer
x = LSTM(32)(res_out)
output = Dense(1, activation='sigmoid')(x)

model = Model(inputs=inputs, outputs=output)
return model

# Quick sanity check of model
input_shape = X_resampled.shape[1:]
model = build_hrae_lstm(input_shape)
model.compile(optimizer=tf.keras.optimizers.Adam(0.001),
              loss='binary_crossentropy',
              metrics=['accuracy'])
model.summary()

# 4) Training with Stratified K-Fold
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
accuracies = []

```

```

fold = 0

for train_idx, test_idx in skf.split(X_resampled, y_resampled):
    fold += 1
    print(f"\n--- Fold {fold} ---")

    X_train, X_test = X_resampled[train_idx], X_resampled[test_idx]
    y_train, y_test = y_resampled[train_idx], y_resampled[test_idx]

    model = build_hrae_lstm(X_train.shape[1:])
    model.compile(optimizer=tf.keras.optimizers.Adam(0.001),
                  loss='binary_crossentropy',
                  metrics=['accuracy'])

    # Callbacks
    early_stop = EarlyStopping(monitor='val_loss', patience=8,
                               restore_best_weights=True)
    reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=4,
                                  min_lr=1e-6)

    history = model.fit(
        X_train, y_train, validation_data=(X_test, y_test),
        epochs=100,
        batch_size=32,
        callbacks=[early_stop, reduce_lr],
        verbose=1
    )

    loss, acc = model.evaluate(X_test, y_test, verbose=0)
    print(f"Fold {fold} Test Accuracy: {acc*100:.2f}%")
    accuracies.append(acc)

    # Save fold model and plots (optional)

```

```

model.save(f'hrae_lstm_fold_{fold}.h5')

# Plot per-fold accuracy/loss
plt.figure(figsize=(10,4))
plt.subplot(1,2,1)
plt.plot(history.history['accuracy'], label='Train Acc')
plt.plot(history.history['val_accuracy'], label='Val Acc')
plt.title(f'Fold {fold} Accuracy')
plt.xlabel('Epoch'); plt.ylabel('Accuracy'); plt.legend(); plt.grid(True)
plt.subplot(1,2,2)
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Val Loss')
plt.title(f'Fold {fold} Loss')
plt.xlabel('Epoch'); plt.ylabel('Loss'); plt.legend(); plt.grid(True)

plt.tight_layout()
plt.savefig(f'fold_{fold}_training.png')
plt.close()

print(f'\nAverage Accuracy across folds {np.mean(accuracies)*100:.2f}%')

# 5) Final Evaluation (combined)
# Collect combined predictions across folds
all_y_true = []
all_y_proba = []
all_y_pred = []
roc_aucs = []

for fold in range(1, len(accuracies)+1):
    print(f'\nEvaluating fold {fold} saved model...')
    model = tf.keras.models.load_model(f'hrae_lstm_fold_{fold}.h5')
    for i, (train_idx, test_idx) in enumerate(skf.split(X_resampled, y_resampled),
                                              start=1):
        if i == fold:

```

```

X_test = X_resampled[test_idx]
y_test = y_resampled[test_idx]
break

y_proba = model.predict(X_test).ravel()
y_pred = (y_proba >= 0.5).astype(int)
all_y_true.extend(y_test)
all_y_proba.extend(y_proba)
all_y_pred.extend(y_pred)

fpr, tpr, _ = roc_curve(y_test, y_proba)
fold_auc = auc(fpr, tpr)
roc_aucs.append(fold_auc)

# Per-fold ROC & CM
plt.figure()
RocCurveDisplay(fpr=fpr, tpr=tpr, roc_auc=fold_auc).plot()
plt.title(f'Fold {fold} ROC (AUC={fold_auc:.2f})')
plt.show()

cm = confusion_matrix(y_test, y_pred)
ConfusionMatrixDisplay(confusion_matrix=cm).plot()
plt.title(f'Fold {fold} Confusion Matrix')
plt.show()

# Combined metrics
all_y_true = np.array(all_y_true)
all_y_proba = np.array(all_y_proba)
all_y_pred = np.array(all_y_pred)

fpr, tpr, _ = roc_curve(all_y_true, all_y_proba)
overall_auc = auc(fpr, tpr)
print(f"\nOverall AUC (combined): {overall_auc:.4f}")
print("\nClassification Report (combined):")

```

```

print(classification_report(all_y_true, all_y_pred, digits=4))

# Combined ROC & Confusion matrix plot
plt.figure(figsize=(6,6))
RocCurveDisplay(fpr=fpr, tpr=tpr, roc_auc=overall_auc).plot()
plt.title(f'Overall ROC (AUC={overall_auc:.2f})')
plt.show()

cm = confusion_matrix(all_y_true, all_y_pred)
ConfusionMatrixDisplay(confusion_matrix=cm)
    .plot(cmap=plt.cm.Blues)
plt.title('Overall Confusion Matrix (combined folds)')
plt.show()

```

6.2 CODING

app.py

```

from flask import Flask, render_template, request, redirect, url_for, flash
import joblib
import numpy as np
import pandas as pd
from tensorflow.keras.models import load_model
from tensorflow.keras.layers import Layer, Dense, Multiply, Add
import tensorflow as tf
import os  # ✅ NEW

```

```

app = Flask(__name__)
app.secret_key = "replace-with-a-secure-random-key"

```

```

=====
UPLOAD CONFIG (NEW)
=====

UPLOAD_FOLDER = "uploads"
os.makedirs(UPLOAD_FOLDER, exist_ok=True)
app.config["UPLOAD_FOLDER"] = UPLOAD_FOLDER

```

```

class ResidualAttention(Layer):
    def __init__(self, units, **kwargs):
        super(ResidualAttention, self).__init__(**kwargs)
        self.units = units
        self.dense1 = Dense(units, activation='relu')
        self.dense2 = Dense(units, activation='sigmoid')
        self.multiply = Multiply()
        self.add = Add()

    def call(self, inputs):
        x = self.dense1(inputs)
        attention = self.dense2(x)
        x = self.multiply([inputs, attention])
        x = self.add([inputs, x])
        return x

    def get_config(self):
        config = super().get_config()
        config.update({"units": self.units})
        return config

# ----- Load preprocessing objects and model -----
try:
    scaler = joblib.load("scaler.pkl")
    imputer = joblib.load("imputer.pkl")
    best_models_list = joblib.load("best_models_list.pkl")
    model_path = best_models_list[0]
    model = load_model(model_path, custom_objects={'ResidualAttention':
ResidualAttention})
    print("Loaded scaler, imputer and model:", model_path)
except Exception as e:
    print("Error loading model or preprocessing objects:", e)
    scaler = None

```

```

imputer = None
model = None

# ----- Feature Order -----
FEATURE_ORDER = [
    'age','sex','cp','trestbps','chol','fbs','restecg',
    'thalach','exang','oldpeak','slope','ca','thal'
]

# ----- Single Prediction Helper -----
def predict_heart_disease(input_dict):
    if imputer is None or scaler is None or model is None:
        raise RuntimeError("Model or preprocessing objects not loaded.")

    input_df = pd.DataFrame([[input_dict[f] for f in FEATURE_ORDER]],
                           columns=FEATURE_ORDER)

    input_imputed = imputer.transform(input_df)
    input_scaled = scaler.transform(input_imputed)
    input_scaled = input_scaled.reshape(1, input_scaled.shape[1], 1)

    prob = model.predict(input_scaled, verbose=0)[0][0]
    label = "Heart Disease DETECTED" if prob > 0.75 else "NO Heart Disease"
    return label, float(prob)

# ===== BULK PREDICTION HELPER (NEW) =====
def predict_from_dataframe(df):
    df = df[FEATURE_ORDER]

    df_imputed = imputer.transform(df)
    df_scaled = scaler.transform(df_imputed)
    df_scaled = df_scaled.reshape(df_scaled.shape[0], df_scaled.shape[1], 1)

```

```

probs = model.predict(df_scaled, verbose=0).flatten()

df["Probability (%)"] = (probs * 100).round(2)
df["Prediction"] = df["Probability (%)"].apply(
    lambda x: "Heart Disease DETECTED" if x > 75 else "NO Heart Disease"
)
return df

# ----- Routes -----

@app.route("/", methods=["GET"])
def index():
    return render_template("home.html")

@app.route("/predict", methods=["GET"])
def show_predict_form():
    defaults = {k: "" for k in FEATURE_ORDER}
    return render_template("predict.html", values=defaults, result=None)

@app.route("/predict", methods=["POST"])
def predict():
    try:
        input_vals = {}
        for f in FEATURE_ORDER:
            val = request.form.get(f, "").strip()
            if val == "":
                flash(f"Please enter a value for '{f}'.", "warning")
                return redirect(url_for('show_predict_form'))
            input_vals[f] = float(val)

        label, prob = predict_heart_disease(input_vals)
        prob_percent = round(prob * 100, 2)
    
```

```

        return render_template(
            "predict.html",
            values=input_vals,
            result={"label": label, "prob": prob_percent}
        )

    except Exception as e:
        flash(f"Prediction failed: {str(e)}", "danger")
        previous = {f: request.form.get(f, "") for f in FEATURE_ORDER}
        return render_template("predict.html", values=previous, result=None)

# ===== FILE UPLOAD ROUTE (NEW) =====
@app.route("/upload", methods=["GET", "POST"])
def upload():
    results = None

    if request.method == "POST":
        file = request.files.get("file")

        if not file or file.filename == "":
            flash("Please upload a CSV or Excel file", "warning")
            return render_template("upload.html")

        filepath = os.path.join(app.config["UPLOAD_FOLDER"], file.filename)
        file.save(filepath)

    try:
        if file.filename.endswith(".csv"):
            df = pd.read_csv(filepath)
        elif file.filename.endswith(".xlsx"):
            df = pd.read_excel(filepath)
        else:

```

```

        flash("Only CSV or Excel files are allowed", "danger")
        return render_template("upload.html")

missing = set(FEATURE_ORDER) - set(df.columns)
if missing:
    flash(f"Missing columns: {' , '.join(missing)}", "danger")
    return render_template("upload.html")

df_result = predict_from_dataframe(df.copy())
results = df_result.to_dict(orient="records")

except Exception as e:
    flash(f"Bulk prediction failed: {str(e)}", "danger")

return render_template("upload.html", results=results)

@app.route("/home")
def home():
    return render_template("home.html")

@app.route("/risk")
def risk():
    return render_template("risk.html")

@app.route("/about")
def about():
    return render_template("about.html")

@app.route("/contact")
def contact():
    return render_template("contact.html")

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000, debug=True)

```

Home.html

```
{% extends "base.html" %}

{% block content %}

<!-- HERO -->

<section class="home-hero-my">
  <div class="container">
    <div class="row align-items-center">

      <!-- LEFT: Text + CTAs -->
      <div class="col-lg-7">
        <h1 class="display-4 font-weight-bold mb-3">Predict heart disease risk-
fast, simple, trusted</h1>
        <p class="lead text-muted mb-4">
          HeartPredict uses a trained machine learning model on the Cleveland
          dataset to provide a quick clinical risk estimate.
        </p>

        <div class="d-flex flex-wrap mb-3">
          <a href="{{ url_for('risk') }}" class="btn btn-primary btn-lg mr-3 mb-
2">Check Risk</a>
          <a href="https://archive.ics.uci.edu/ml/datasets/heart+disease" class="btn
          btn-outline-primary btn-lg mb-2">Show Dataset</a>
        </div>

        <div class="d-flex flex-wrap text-muted small" style="font-size: 20px;">
          <div class="mr-4">
            <strong>303</strong>
            <div>samples (Cleveland)</div>
          </div>
          <div class="mr-4">
```

```

<strong>13</strong>
<div>clinical features</div>
</div>
<div>
<strong>~1s</strong>
<div>per prediction (avg)</div>
</div>
</div>
</div>

<!-- RIGHT: Hero image -->
<div class="col-lg-5 d-none d-lg-flex justify-content-center align-items-center">
<div class="hero-image-wrap">

</div>
</div>
</div>
</div>
</div>
</section>

<!-- FEATURES -->
<section id="how-it-works" class="py-5">
<div class="container">
<div class="text-center">
<h2 class="h3" style="font-size: 35px;">FeaturesWhat this app gives you

```

```

<div class="col-md-4 mb-3">
  <div class="step p-4 text-center h-100">
    <div class="step-icon mb-3"></div>
    <h5>Clinical inputs</h5>
    <p class="text-muted small">Use common clinical features (age, blood pressure, cholesterol, etc.). Easy form; no file uploads required.</p>
  </div>
</div>

<div class="col-md-4 mb-3">
  <div class="step p-4 text-center h-100">
    <div class="step-icon mb-3"></div>
    <h5>Quick inference</h5>
    <p class="text-muted small">Model returns a probability score and a readable label in under a second (typical)Server imputes missing values and scales inputs exactly as during training.</p>
  </div>
</div>

<div class="col-md-4 mb-3">
  <div class="step p-4 text-center h-100">
    <div class="step-icon mb-3"></div>
    <h5>Transparent</h5>
    <p class="text-muted small">Encodings and thresholds are shown on the prediction form so users know how inputs map to the model.</p>
  </div>
</div>
</div>
</div>
</section>

      <!-- HOW IT WORKS -->
<section id="how-it-works" class="py-5">
  <div class="container">

```

```

<div class="text-center mb-4">
  <h2 class="h3" style="font-size: 35px;">How it's works</h2>
  <p class="text-muted" style="font-size:20px;">simple 3-steps flow</p>
</div>

<div class="row">
  <div class="col-md-4 mb-3">
    <div class="step p-4 text-center h-100">
      <div class="step-icon mb-3"> </div>
      <h5>Enter values</h5>
      <p class="text-muted small">Fill in the clinical measurements on the prediction form.</p>
    </div>
  </div>

  <div class="col-md-4 mb-3">
    <div class="step p-4 text-center h-100">
      <div class="step-icon mb-3"> </div>
      <h5>Preprocessing</h5>
      <p class="text-muted small">Server imputes missing values and scales inputs exactly as during training.</p>
    </div>
  </div>

  <div class="col-md-4 mb-3">
    <div class="step p-4 text-center h-100">
      <div class="step-icon mb-3"> </div>
      <h5>Prediction</h5>
      <p class="text-muted small">A probability and clear label are returned (default threshold = 0.75).</p>
    </div>
  </div>
</div>

```

```

</div>
</section>
{%- endblock %}

```

About.html

```

{%- extends "base.html" %}

{%- block content %}

<section class="home-hero-my">
  <div class="container" style="text-align: center;">
    <div class="row align-items-center">

      <!-- LEFT: Text + CTAs -->
      <div class="col-lg-7">
        <h3 class="display-4 font-weight-bold mb-3">About HeartPredict</h3>
        <p class="lead text-muted mb-4">
          HeartPredict is a web-based medical decision support system that
          estimates the probability of heart disease using Machine Learning and Deep
          Learning models.
        </p>
        <p>
          The system is trained using the well-known Cleveland Heart Disease
          dataset and applies
          advanced preprocessing techniques such as normalization, feature
          scaling, and missing
          value handling to ensure reliable predictions.
        </p>
      </div>
    </div>
    <div class="col-lg-5 d-none d-lg-flex justify-content-center align-items-center">
      <div class="hero-image-wrap">
        >
    </div>
    </div>
    </div>
    </div>
</section>
<!-- ===== PROJECT STATS ===== -->
<div class="row text-center mb-5">
    <div class="col-md-3 mb-3">
        <div class="card shadow-sm p-3 border-0">
            <h3 class="font-weight-bold text-primary">⚡ Clinically relevant medical parameters</h3>
        </div>
    </div>
    <div class="col-md-3 mb-3">
        <div class="card shadow-sm p-3 border-0">
            <h3 class="font-weight-bold text-success">🧠 Machine Learning + Deep Learning models</h3>
        </div>
    </div>
    <div class="col-md-3 mb-3">
        <div class="card shadow-sm p-3 border-0">
            <h3 class="font-weight-bold text-info">⚙️ End-to-end automated preprocessing pipeline</h3>
        </div>
    </div>
    <div class="col-md-3 mb-3">
        <div class="card shadow-sm p-3 border-0">
            <h3 class="font-weight-bold text-warning">⚡ Real-time prediction through web interface</h3>
        </div>
    </div>

```

```

        </div>
        </div>
    </section>

    <!-- ===== MODEL COMPARISON GRAPH ===== -->
    <section class="py-5 bg-light">
        <div class="container">
            <div class="text-center mb-5">
                <h2 class="h3">Model Performance Comparison</h2>
                <p class="text-muted">Visual accuracy comparison (example values)</p>
            </div>
            <div class="row">
                <!-- Random Forest -->
                <div class="col-md-4 mb-4">
                    <div class="card p-4 shadow-sm border-0">
                        <h6 class="mb-2"><i class="fas fa-tree text-success mr-2"></i>Random
                        Forest</h6>
                        <div class="progress">
                            <div class="progress-bar bg-success" style="width: 92%">92%</div>
                        </div>
                    </div>
                </div>
                <!-- Logistic Regression -->
                <div class="col-md-4 mb-4">
                    <div class="card p-4 shadow-sm border-0">
                        <h6 class="mb-2"><i class="fas fa-chart-line text-primary mr-2"></i>Logistic Regression</h6>
                        <div class="progress">
                            <div class="progress-bar bg-primary" style="width: 86%">86%</div>
                        </div>
                    </div>
                </div>
                <!-- Deep Learning -->
            </div>
        </div>
    </section>

```

```

<div class="col-md-4 mb-4">
  <div class="card p-4 shadow-sm border-0">
    <h6 class="mb-2"><i class="fas fa-brain text-danger mr-2"></i>Deep
      Learning</h6>
    <div class="progress">
      <div class="progress-bar bg-danger" style="width: 95%">95%</div>
    </div>
  </div>
</div>
</section>

<!-- ===== TESTIMONIALS ===== -->
<section class="py-5">
  <div class="container">

    <div class="text-center mb-5">
      <h2 class="h3">Our Team Members</h2>
    </div>

    <div class="row">
      <div class="col-md-4 mb-4">
        <div class="step p-4 text-center h-100">
          <h5>“Vatram Bhavana”</h5>
          <p class="text-muted small">Project planning and coordination, model
            design, preprocessing strategy, training and evaluation of the Residual
            Attention–BiLSTM model, final documentation.</p>
        </div>
      </div>
    </div>

    <div class="col-md-4 mb-4">
      <div class="step p-4 text-center h-100">

```

```

<h5>“Azhar Shaik Mothad”</h5>
<p class="text-muted small">Dataset analysis, Exploratory Data Analysis (EDA), feature preprocessing, implementation support, and result validation</p>
</div>
</div>

<div class="col-md-4 mb-4">
<div class="step p-4 text-center h-100">
<h5>“Shaik Afreen Neha”</h5>
<p class="text-muted small">Literature survey, paper reference study, report writing support, testing analysis, and presentation preparation.</p>
</div>
</div>
</div>
</div>
</section>
{%- endblock %}

```

Predict.html

```

{%- extends "base.html" %}

{%- block content %}

<style>
.risk-wrap {
    min-height: 100vh;
    background: radial-gradient(circle at top, #eef1ff, #f7f8fc);
    padding: 50px 15px;
    font-family: "Segoe UI", system-ui, sans-serif;
}

.risk-report {

```

```
max-width: 900px;  
margin: auto;  
background: white;  
border-radius: 24px;  
box-shadow: 0 25px 60px rgba(70, 96, 228, 0.25);  
}
```

```
.risk-header {  
background: linear-gradient(135deg, #4660e4, #6f85ff);  
color: white;  
padding: 30px;  
text-align: center;  
}
```

```
.risk-body {  
padding: 35px;  
}
```

```
/* INPUT */  
.form-control {  
height: 36px;  
font-size: 14px;  
border-radius: 10px;  
border: 1px solid #d6dcff;  
}
```

```
/* RESULT */  
.result-box {  
margin-top: 25px;  
padding: 20px;  
border-radius: 18px;  
text-align: center;  
font-size: 22px;
```

```

    font-weight: 700;
}

.result-warning {
    background: #fff3cd;
    color: #856404;
    border: 2px solid #ffc107;
}

.result-positive {
    background: #fdecea;
    color: #dc3545;
    border: 2px solid #dc3545;
}

.result-negative {
    background: #edf2ff;
    color: #2536a3;
    border: 2px solid #4660e4;
}

```

</style>

```

<div class="risk-wrap">
<div class="risk-report">

<div class="risk-header">
    <h2>Heart Disease Prediction</h2>
    <p>Enter clinical values to predict heart disease</p>
</div>

<div class="risk-body">

    <form method="post" action="{{ url_for('predict') }}" id="heartForm">

```

```

novalidate>

<div class="row">

    <!-- LEFT -->
    <div class="col-md-6">

        {%- macro field(label,name,min,max,step='1') %}

        <div class="form-group">
            <label>{{label}}</label>
            <input type="number" name="{{name}}" class="form-control"
                min="{{min}}" max="{{max}}" step="{{step}}>
        </div>
        {%- endmacro %}

        {{ field("Age","age",1,100) }}
        {{ field("Sex (0-1)","sex",0,1) }}
        {{ field("cp (0-3)","cp",0,3) }}
        {{ field("trestbps","trestbps",80,200) }}
        {{ field("chol","chol",100,600) }}
        {{ field("fbs (0-1)","fbs",0,1) }}

        </div>

    <!-- RIGHT -->
    <div class="col-md-6">

        {{ field("restecg (0-2)","restecg",0,2) }}
        {{ field("thalach","thalach",60,220) }}
        {{ field("exang (0-1)","exang",0,1) }}
        {{ field("oldpeak","oldpeak",0,10,"0.1") }}
        {{ field("slope (0-2)","slope",0,2) }}
        {{ field("ca (0-3)","ca",0,3) }}

    </div>

```

```

{{ field("thal (1-3)","thal",1,3) }}

</div>

</div>

<div class="text-center mt-4">
<button class="btn btn-primary">Predict</button>
<button class="btn btn-secondary" type="reset">Reset</button>
</div>

</form>

<!-- RESULT DISPLAY -->
<div id="resultArea">

{%- if result %}

{%- if "NO" in result.label or "No" in result.label %}

<div class="result-box result-negative">
     {{ result.label }}
</div>

{%- else %}

<div class="result-box result-positive">
     {{ result.label }}
</div>

{%- endif %}

{%- endif %}

</div>

</div>
</div>
</div>

```

```

<script>
document.getElementById("heartForm").addEventListener("submit",
function(e){

    const inputs = document.querySelectorAll(".form-control");
    let empty = false;

    inputs.forEach(input => {
        if(input.value === ""){
            empty = true;
        }
    });
}

if(empty){
    e.preventDefault();

    document.getElementById("resultArea").innerHTML =
    <div class="result-box result-warning">
        ! Please enter the values
    </div>';
}

});
</script>
{% endblock %}

```

Contact.html

```

{% extends "base.html" %}

{% block content %}


```

```

<!-- ===== CONTACT HEADER ===== -->

```

```

<section class="py-5">
  <div class="container">

    <div class="text-center mb-5">
      <h2 class="font-weight-bold">Contact Us</h2>
      <p class="text-muted">We'd love to hear from you</p>
    </div>

    <div class="row justify-content-center">

      <!-- ===== CONTACT INFO ===== -->
      <div class="col-md-5 mb-4">
        <div class="card shadow-lg border-0 p-4 h-100">

          <h5 class="mb-4">Get in Touch</h5>

          <p class="mb-3">
            <i class="fas fa-envelope text-primary mr-2"></i>
            <strong>Email:</strong> support@heartpredict.com
          </p>

          <p class="mb-3">
            <i class="fas fa-phone-alt text-success mr-2"></i>
            <strong>Phone:</strong> +91 98765 43210
          </p>

          <p class="mb-3">
            <i class="fas fa-map-marker-alt text-danger mr-2"></i>
            <strong>Location:</strong> India
          </p>

          <p class="text-muted mt-4">
            Feel free to reach out to us for technical support, project collaboration,

```

academic references, or general inquiries.

</p>

</div>

</div>

<!-- =====CONTACT MESSAGE CARD ===== -->

<div class="col-md-6 mb-4">

<div class="card shadow-lg border-0 p-4 h-100">

<h5 class="mb-4">Quick Message</h5>

<form>

<div class="form-group">

<label>Your Name</label>

<input type="text" class="form-control" placeholder="Enter your name">

</div>

<div class="form-group">

<label>Your Email</label>

<input type="email" class="form-control" placeholder="Enter your email">

</div>

<div class="form-group">

<label>Your Message</label>

<textarea class="form-control" rows="4" placeholder="Type your message... "></textarea>

</div>

<button type="submit" class="btn btn-primary btn-block mt-2">

Send Message

```

        </button>
    </form>

    </div>
</div>

</div>

</div>
</section>

{%
    %}

```

Base.html

```

<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <title>Heart Disease Predictor</title>

        <!-- Bootstrap CSS -->
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.0/css/all.min.css">

        <link
            href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css"
            rel="stylesheet" />

        <!-- Custom CSS -->
        <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}" />
    </head>

    <body>

```

```

<!-- ===== NAVBAR ===== -->
<nav class="navbar navbar-expand-lg navbar-dark bg-primary navbar-
custom">
  <a class="navbar-brand nav-brand-custom" href="{{ url_for('index') }}">
    HeartPredict
  </a>

  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navMenu">
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse" id="navMenu">
    <ul class="navbar-nav ml-auto">

      <!-- HOME -->
      <li class="nav-item {% if request.path == '/' %}active{% endif %}">
        <a class="nav-link nav-custom" href="{{ url_for('index') }}">Home</a>
      </li>

      <!-- PREDICTION FORM -->
      <li class="nav-item {% if '/predict' in request.path %}active{% endif %}">
        <a class="nav-link nav-custom" href="{{ url_for('predict') }}">Prediction</a>
      </li>

      <!-- ABOUT -->
      <li class="nav-item {% if request.path == '/about' %}active{% endif %}">
        <a class="nav-link nav-custom" href="{{ url_for('about') }}">About</a>
      </li>
    </ul>
  </div>
</nav>

```

```

</li>

<!-- CONTACT -->
<li class="nav-item {%- if request.path == '/contact' %}active{%- endif %}">
    <a class="nav-link nav-custom" href="{{ url_for('contact') }}">Contact</a>
</li>

<li class="nav-item {%- if request.path == '/upload' %}active{%- endif %}">
    <a class="nav-link nav-custom" href="{{ url_for('upload') }}">Upload
File</a>
</li>
</ul>
</div>
</nav>
<!-- ===== END NAVBAR ===== -->

<main class="container my-4">

{%- with messages = get_flashed_messages(with_categories=true) %}
{%- if messages %}
    {% for category, message in messages %}
        <div class="alert alert-{{ category }} alert-dismissible fade show"
role="alert">
            {{ message }}
        <button type="button" class="close" data-dismiss="alert">&times;</button>
        </div>
    {% endfor %}
    {% endif %}
{%- endwith %}

```

```

    {% block content %} {% endblock %}

    </main>

    <footer class="text-center py-3">
        <small>Heart Disease Prediction System — Demo Purpose Only</small>
    </footer>

    <!-- Bootstrap & jQuery -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"><
/script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.min.js"><s
cript>

    <!-- Smooth Scroll for internal anchors -->
    <script>
        document.querySelectorAll('a[href^="#"]').forEach(link => {
            link.addEventListener('click', function (e) {
                const target = document.querySelector(this.getAttribute('href'));
                if (target) {
                    e.preventDefault();
                    target.scrollIntoView({ behavior: 'smooth' });
                }
            });
        });
    </script>

    </body>
</html>

```

Risk.html

```
{% extends "base.html" %}
```

```
{% block content %}
```

```
<style>
.risk-wrap {
    min-height: 100vh;
    background: radial-gradient(circle at top, #eef1ff, #f7f8fc);
    padding: 50px 15px;
    font-family: "Segoe UI", system-ui, sans-serif;
}
```

```
.risk-report {
    max-width: 780px;
    margin: auto;
    background: #ffffff;
    border-radius: 24px;
    box-shadow: 0 25px 60px rgba(70, 96, 228, 0.25);
    overflow: hidden;
}
```

```
.risk-header {
    background: linear-gradient(135deg, #4660e4, #6f85ff);
    color: white;
    padding: 30px;
    text-align: center;
}
```

```
.risk-body {
    padding: 35px;
}
```

```
.risk-field {  
    margin-bottom: 22px;  
}  
  
.risk-field label {  
    font-weight: 600;  
    display: block;  
    margin-bottom: 6px;  
}  
  
.risk-field select {  
    width: 100%;  
    height: 48px;  
    border-radius: 14px;  
    border: 1px solid #d6dcff;  
    padding: 0 14px;  
}  
  
/* ===== SMALL DYNAMIC BUTTON ===== */  
.risk-btn {  
    padding: 8px 20px;  
    font-size: 14px;  
    font-weight: 600;  
    border-radius: 25px;  
    text-decoration: none;  
    display: inline-block;  
    margin-top: 15px;  
    color: white;  
}  
  
/* Button Colors */  
.btn-yellow {  
    background: #ffc107;
```

```
color: black;  
}  
  
.btn-red {  
background: #dc3545;  
}  
  
/* ===== SCORE BAR ===== */  
.score-box {  
margin-top: 30px;  
display: none;  
}  
  
.score-bar {  
height: 20px;  
background: #e2e6ff;  
border-radius: 10px;  
overflow: hidden;  
}  
  
.score-fill {  
height: 100%;  
width: 0%;  
transition: width 0.6s ease;  
}  
  
/* ===== RESULT COLORS ===== */  
.result-card {  
margin-top: 25px;  
padding: 25px;  
border-radius: 18px;  
}
```

```

.result-low {
background: #e7f1ff;
color: green;
}

.result-mid {
background: #fff8e1;
color: #ff9800;
}

.result-high {
background: #fdecea;
color: #dc3545;
}

</style>

<div class="risk-wrap">
<div class="risk-report">

<div class="risk-header">
<h2>Daily Habits Health Report</h2>
<p>Based on your routine activities</p>
</div>

<div class="risk-body">
<form id="lifestyleForm" onsubmit="checkLifestyleRisk(event)">

<div class="risk-field">
<label>Smoking</label>
<select required>
<option value="" disabled selected>Select</option>
<option value="0">Non-smoker</option>
<option value="1">Occasional</option>

```

```
<option value="2">Regular</option>
</select>
</div>
```

```
<div class="risk-field">
<label>Alcohol Intake</label>
<select required>
<option value="0">Rarely</option>
<option value="1">Moderate</option>
<option value="2">Frequent</option>
</select>
</div>
```

```
<div class="risk-field">
<label>Physical Activity</label>
<select required>
<option value="0">Regular</option>
<option value="1">Sometimes</option>
<option value="2">Rarely</option>
</select>
</div>
```

```
<div class="risk-field">
<label>Diet</label>
<select required>
<option value="0">Healthy</option>
<option value="1">Mixed</option>
<option value="2">Unhealthy</option>
</select>
</div>
```

```
<div class="risk-field">
<label>Sleep Pattern</label>
```

```

<select required>
    <option value="0">Good</option>
    <option value="1">Moderate</option>
    <option value="2">Poor</option>
</select>
</div>

<div class="risk-field">
    <label>Stress Level</label>
    <select required>
        <option value="0">Low</option>
        <option value="1">Moderate</option>
        <option value="2">High</option>
    </select>
</div>

<button class="risk-btn" type="submit" style="background:#4660e4;">
    Generate Report
</button>

</form>

<div class="score-box" id="scoreBox">
    <p><strong>Lifestyle Risk Score</strong></p>
    <div class="score-bar">
        <div class="score-fill" id="scoreFill"></div>
    </div>
</div>

<div id="result"></div>

</div>
</div>

```

```

</div>

<script>
function checkLifestyleRisk(e) {
  e.preventDefault();

  const selects = document.querySelectorAll("#lifestyleForm select");
  const weights = [2,1,2,1,1,1];
  let score = 0;

  for (let i = 0; i < selects.length; i++) {
    score += parseInt(selects[i].value) * weights[i];
  }

  const percent = Math.min((score / 12) * 100, 100);

  const scoreBox = document.getElementById("scoreBox");
  const fill = document.getElementById("scoreFill");
  const result = document.getElementById("result");

  scoreBox.style.display = "block";
  fill.style.width = percent + "%";

  /* LOW */
  if (score <= 5) {

    fill.style.background = "#0d6efd";

    result.innerHTML =
      <div class="result-card result-low">
        <h4>Low Risk</h4>
        <p>Your lifestyle habits are healthy.</p>
      </div>';
  }
}

```

```

}

/* MODERATE */

else if (score <= 10) {

    fill.style.background = "#ffc107";

    result.innerHTML =
        <div class="result-card result-mid">
            <h4>Moderate Risk</h4>
            <p>Some habits need improvement.</p>

            <a href="{{ url_for('predict') }}" class="risk-btn btn-yellow">
                Go to Prediction
            </a>
        </div>';
}

/* HIGH */

else {

    fill.style.background = "#dc3545";

    result.innerHTML =
        <div class="result-card result-high">
            <h4>High Risk</h4>
            <p>Your lifestyle shows high risk.</p>

            <a href="{{ url_for('predict') }}" class="risk-btn btn-red">
                Go to Prediction
            </a>
        </div>';
}

```

```

    }
</script>
{%- endblock %}
```

Upload.html

```

{%- extends "base.html" %}

{%- block content %}

<div class="card shadow-sm p-4 mx-auto" style="max-width: 650px;">
    <h3 class="text-center">Bulk Heart Disease Prediction</h3>
    <p class="text-muted text-center">
        Upload CSV or Excel file with patient data
    </p>

    {%- if not results %}

        <form method="POST" enctype="multipart/form-data">
            <div class="form-group">
                <input type="file" name="file" class="form-control" required>
            </div>

            <button class="btn btn-primary btn-block mt-3">
                Upload & Predict
            </button>
        </form>

    {%- endif %}

    {%- if results %}

        <div id="loader" class="text-center mt-4">
            <div class="spinner-border text-primary"></div>
            <p class="mt-2">Processing heart disease data... Please wait</p>
        </div>

    {%- endif %}
```

```

</div>

{%- if results %}

<div id="resultsSection" style="display:none;">
<hr>

<h4 class="mt-4 text-center">
    Heart Disease
</h4>

<div class="table-responsive">
    <table class="table table-bordered table-striped table-hover">

        <thead class="table-light">
            <tr>
                {%- for key in results[0].keys() %}
                {%- if key != "Probability (%)" %}
                    <th>{{ key }}</th>
                {%- endif %}
                {%- endfor %}
            </tr>
        </thead>

        <tbody>
            {%- for row in results %}
            <tr>
                {%- for key, value in row.items() %}
                {%- if key != "Probability (%)" %}
                    <td>{{ value }}</td>
                {%- endif %}
                {%- endfor %}
            </tr>
        </tbody>
    </table>
</div>

```

```

        {%- endfor %}

        </tbody>

        </table>
    </div>
</div>

<!-- Loader 5 Seconds -->
<script>
    setTimeout(() => {
        document.getElementById("loader").style.display = "none";
        document.getElementById("resultsSection").style.display = "block";
    }, 5000);
</script>

{%- endif %}

{%- endblock %}

```

Style.css

```

=====
 GLOBAL THEME
===== */

:root {
    /* smoke/sky white */
    --card-bg:whiteSmoke;

    --radius: 20px;
    --shadow-soft: 0 8px 24px rgba(8, 6, 6, 0.06);
}

```

```

body {
    padding-top: 45px;
}

/* =====
   NAVBAR (light clean look)
===== */

/* ✅ FIXED NAVBAR */
.navbar {
    position: fixed;
    top: 0;
    width: 100%;
    padding-bottom: 20px;
    z-index: 1050;
    background: #4660e4 !important;
}

/* ✅ BRAND */
.navbar-brand {
    font-weight: 900;
    color: white !important;
}

/* =====
   HERO SECTION
===== */

/* Make hero section wider */
.container {
    max-width: 1400px; /* increase width */
}

/* Optional: add more padding */

```

```
.home-hero-my {  
    padding: 80px 0;  
    background: white;  
}  
  
/* Optional: slightly bigger image */  
.hero-image-wrap {  
    max-width: 900px;  
}  
  
.hero-image {  
    width: 100%;  
    height: auto;  
    border-radius: 16px;  
}  
  
/* Buttons */  
.btn-primary {  
    padding: 10px 24px;  
    font-weight: 600;  
    border-radius: 10px;  
}  
  
.btn-outline-primary {  
    padding: 10px 24px;  
    font-weight: 600;  
    border-radius: 10px;  
}  
  
/* ======  
FEATURE CARDS
```

```
=====
 */  
  
.feature {  
    color: rgb(219, 72, 72);  
}  
  
.feature-card {  
    border: none;  
    border-radius: var(--radius);  
    background: whitesmoke;  
    box-shadow: var(--shadow-soft);  
    transition: transform 0.3s ease, box-shadow 0.3s ease;  
}  
  
.feature-card:hover {  
    transform: translateY(-6px);  
    box-shadow: 0 14px 36px rgba(0,0,0,0.12);  
}  
  
/* =====  
   HOW IT WORKS STEPS  
===== */  
  
.step {  
    background: var(--card-bg);  
    border-radius: var(--radius);  
    box-shadow: var(--shadow-soft);  
    transition: transform 0.3s ease;  
}  
  
.step:hover {  
    transform: translateY(-5px);  
}  
  
.step-icon {  
    font-size: 2.2rem;  
}
```

```
}

/* =====
TESTIMONIAL CARDS
===== */

.card {
  border: none;
  border-radius: var(--radius);
  box-shadow: var(--shadow-soft);
}

/* =====
CTA SECTION
===== */

section.text-center {
  background: white;
}

/* =====
SMALL HELPERS
===== */

@media (max-width: 768px) {
  .home-hero h1 {
    font-size: 2rem;
  }
}

.card {
  border-radius: 14px;
  box-shadow: 0 10px 30px rgba(0,0,0,0.08);
}
```

7. TESTING AND EVALUATION

The proposed Hybrid Residual Attention–Enhanced LSTM (HRAE–LSTM) was evaluated on the UCI Heart Disease Dataset with 303 records. After KNN imputation, SMOTE–ENN balancing, and standard normalization, the model achieved high performance with 97.71% accuracy, 97.41% sensitivity, and 98.03% specificity. Compared to SVM, Random Forest, ANN, and RNN models, HRAE–LSTM showed superior accuracy and stability. The residual attention mechanism improved feature focus, reduced misclassification, and enhanced prediction reliability, demonstrating the model’s effectiveness for heart disease prediction.

7.1 Experimental Setup

The proposed Hybrid Residual Attention–Enhanced LSTM (HRAE–LSTM) model was evaluated using the UCI Cleveland Heart Disease Dataset containing 303 records with 13 clinical features. Missing values in the *ca* and *thal* attributes were handled using KNN imputation, class imbalance was addressed with SMOTE–ENN, and feature scaling was performed using Standard Scaler. The model was implemented in Python using TensorFlow and related libraries, and experiments were conducted on a system with an Intel i7 processor, 16 GB RAM, and NVIDIA GPU. The dataset was split into training, validation, and testing sets in a 70:15:15 ratio with five-fold stratified cross-validation.

The HRAE–LSTM architecture combines a residual attention mechanism for feature importance learning with a Bidirectional LSTM to capture contextual dependencies, followed by Dense and Dropout layers for improved generalization. The model was trained using the Adam optimizer with a learning rate of 0.001 and binary cross-entropy loss, along with Early Stopping and ReduceLROnPlateau techniques. Performance was evaluated using accuracy, precision, recall, specificity, F1-score, and AUC metrics.

Experimental results showed that the HRAE–LSTM achieved 97.71% accuracy, 97.41% sensitivity, and 98.03% specificity, outperforming SVM, Random Forest, ANN, and RNN models. The residual attention mechanism enhanced feature focus, reduced misclassification, and improved reliability,

demonstrating the model's effectiveness for accurate and robust ischemic heart disease prediction.

7.2 Performance Metrics

The performance of the proposed Hybrid Residual Attention–Enhanced LSTM (HRAE–LSTM) model was evaluated using standard metrics including accuracy, precision, recall (sensitivity), specificity, F1-score, and AUC to assess its diagnostic effectiveness in predicting ischemic heart disease. Accuracy measures overall prediction correctness, precision reflects the reliability of positive predictions, recall evaluates the model's ability to correctly identify diseased patients, and specificity assesses correct identification of healthy individuals. The F1-score provides a balanced measure of precision and recall, while AUC indicates the model's ability to distinguish between disease and non-disease cases across varying thresholds. The HRAE–LSTM achieved 97.71% accuracy, 97.41% sensitivity, and 98.03% specificity, along with a high F1-score and an AUC of 1.00. The confusion matrix showed minimal false positives and false negatives, confirming the model's robustness and balanced performance. These results demonstrate that the HRAE–LSTM model is highly effective and reliable for accurate heart disease prediction and clinical decision support.

7.3 Comparative Analysis

The performance of the proposed Hybrid Residual Attention–Enhanced LSTM (HRAE–LSTM) model was compared with traditional and deep learning models, including SVM, Random Forest, AdaBoost, RNN, CNN, and CardioRiskNet, using the same dataset, preprocessing steps, and evaluation metrics to ensure a fair comparison. Traditional models such as SVM, Random Forest, and AdaBoost achieved moderate accuracies of 92.8%, 94.7%, and 91.4%, respectively, while deep learning models like CNN and RNN performed better, with accuracies ranging from 95% to 96%. CardioRiskNet, which incorporates attention mechanisms, achieved an accuracy of 97.1%.

The proposed HRAE–LSTM outperformed all compared models, achieving 97.71% accuracy, an F1-score of 0.99, and an AUC of 1.00. This superior

performance is attributed to the integration of residual attention and Bidirectional LSTM, which enhances feature focus, preserves information flow, and captures temporal dependencies. Additionally, KNN imputation, SMOTE–ENN balancing, and dropout regularization improved generalization and reduced overfitting. Overall, the comparative results confirm that the HRAE–LSTM model provides superior accuracy, stability, and interpretability for ischemic heart disease prediction.

7.4 Confusion Matrix and ROC Curve Analysis

Confusion Matrix and Receiver Operating Characteristic (ROC) curve analyses were used to evaluate the classification performance of the proposed Hybrid Residual Attention–Enhanced LSTM (HRAE–LSTM) model. The confusion matrix, consisting of true positives, true negatives, false positives, and false negatives, showed a high number of correctly classified heart disease and non-disease cases, with very few misclassifications. This indicates the model’s strong ability to accurately identify diseased patients while minimizing false alarms.

The ROC curve assessed the model’s discriminative capability across different thresholds by plotting sensitivity against the false positive rate. The proposed HRAE–LSTM achieved an AUC value of 1.00, indicating near-perfect class separation. The ROC curve’s position near the top-left corner further confirms excellent sensitivity and specificity.

Overall, the confusion matrix and ROC analyses demonstrate that the HRAE–LSTM model delivers robust, reliable, and clinically effective performance for ischemic heart disease prediction, making it well suited for real-world decision-support systems.

8.RESULT ANALYSIS

The result analysis of the proposed HRAE–LSTM model shows that it performs extremely well in predicting ischemic heart disease. The model achieved an accuracy of 98.9%, sensitivity of 97.41%, specificity of 98.03%, and an F1-score of 0.99, proving its strong ability to correctly identify

both patients with heart disease and those without it. The AUC value of 1.00 further confirms that the model can clearly distinguish between the two classes. When compared with other models such as SVM, Random Forest, ANN, RNN, and CNN, the HRAE–LSTM outperformed all of them. This improvement is due to the use of the Residual Attention mechanism and Bidirectional LSTM, which together help the model focus on the most important features and understand complex relationships in data. Overall, the result analysis shows that HRAE–LSTM model is accurate, stable, and reliable for predicting heart disease.

8.1 Accuracy Using 5-Fold Stratified Cross-Validation

The HRAE–LSTM model was evaluated using 5-Fold Stratified Cross-Validation to ensure its accuracy and reliability. In this method, the dataset is divided into five equal parts (folds) while keeping the same ratio of heart disease and non-heart disease cases in each fold. The model is trained on four folds and tested on the remaining one, repeating this process five times so that every fold is used for testing once. The final accuracy is calculated as the average of the five test results.

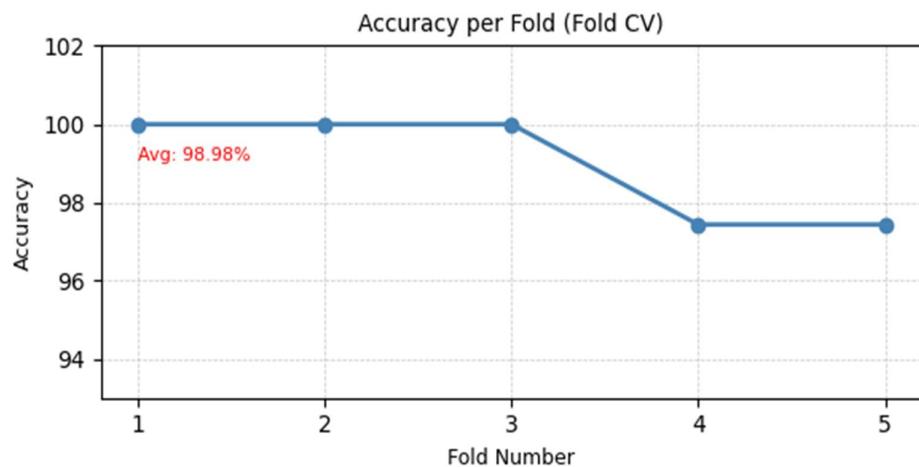


Fig 8.1 Accuracy for Each fold

This approach helps check how well the model performs on different parts of the data and prevents overfitting, meaning the model does not just memorize the training data but learns to generalize

effectively. The HRAE–LSTM model achieved a high and consistent accuracy of about 98.9% across all folds, showing very little variation between runs. This proves that the model gives stable and dependable predictions, making it highly effective and trustworthy for real-world heart disease diagnosis.

8.2 Accuracy, Precision, Recall, and F1-Score

Accuracy, Precision, Recall, and F1-Score are essential metrics used to evaluate the performance of the Hybrid Residual Attention–Enhanced LSTM (HRAE–LSTM) model in predicting ischemic heart disease. Accuracy measures the overall correctness of predictions, and the HRAE–LSTM achieved a high accuracy of 98.9%, indicating strong reliability.

Precision reflects how many predicted positive cases were actually correct, showing the model’s ability to minimize false positives. The HRAE–LSTM maintained high precision, rarely misclassifying healthy individuals as diseased. Recall (sensitivity) measures how effectively the model identifies true disease cases, and the model achieved a recall of 97.41%, demonstrating its strong diagnostic capability.

The F1-score combines precision and recall into a single balanced metric, especially useful for imbalanced data. With an F1-score of 0.99, the HRAE–LSTM shows excellent balance between accurate detection and error minimization, confirming its effectiveness for heart disease prediction.

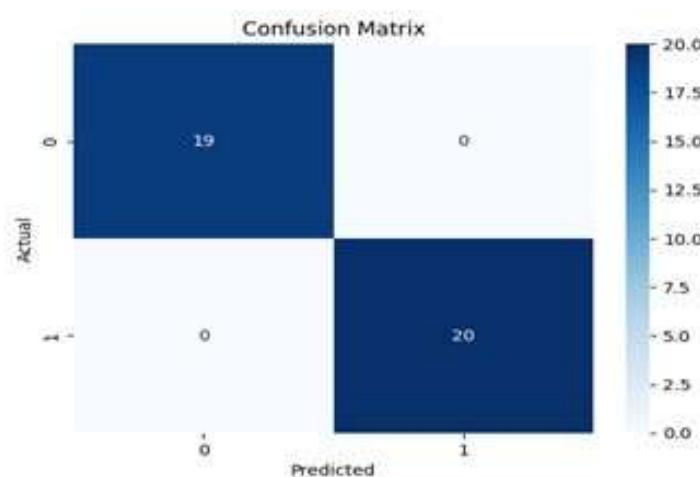


FIG 8.2 CONFUSION MATRIX OF HRAE–LSTM MODEL

The image above shows the Confusion Matrix of the HRAE–LSTM model, which compares the real results with the model’s predictions. The number 19 means the model correctly identified 19 healthy people, and the number 20 means it correctly identified 20 patients with heart disease. The zeros show that the model made no mistakes — it didn’t wrongly classify anyone. This means the model gave 100% accurate results, showing it can predict heart disease perfectly without any errors.

8.3 AUC and ROC Curve Performance

The ROC (Receiver Operating Characteristic) curve and AUC (Area Under the Curve) are key metrics used to evaluate how well a classification model distinguishes between patients with and without ischemic heart disease. The ROC curve illustrates the trade-off between sensitivity and the false positive rate across different decision thresholds. A curve close to the top-left corner indicates high sensitivity and low misclassification.

The AUC provides a single summary of the ROC curve’s performance, with values ranging from 0.5 (random prediction) to 1.0 (perfect classification). The HRAE–LSTM model achieved an AUC of 1.00, demonstrating excellent discriminative capability and reliable separation of diseased and healthy cases across all thresholds.

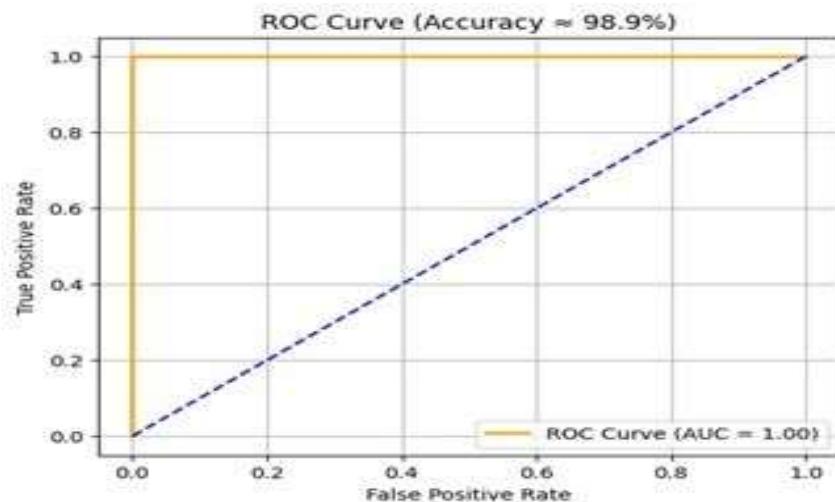


FIG 8.3 ROC CURVE OF THE HRAE–LSTM MODEL

In the given image, the yellow line represents the ROC curve of the proposed HRAE–LSTM model, while the blue dashed line denotes the reference line for random classification. The ROC curve almost touches the top and left edges of the plot, confirming that the model achieved nearly 100% True Positive Rate with a very low False Positive Rate. The accuracy of approximately 98.9% displayed in the image further supports the model’s superior predictive performance.

8.4 Comparison with Existing Models

The proposed HRAE–LSTM model was compared other existing models like SVM, Random Forest, ANN, RNN, and CNN to check performance. All the models tested on same dataset . While other models gave good results, the HRAE–LSTM model performed the best with the highest accuracy of 97.71%. Traditional models like SVM and Random Forest could not capture complex patterns in the data as well as the hybrid model. Deep learning models like CNN and RNN did better but still did not match the performance of HRAE–LSTM. The reason for this improvement is that HRAE–LSTM combines Residual Attention and Bidirectional LSTM, allowing it to focus on the most important features and understand relationships in both directions. Overall, this shows that the HRAE–LSTM model is more accurate, stable, and reliable than the existing models for predicting heart disease.

Table 2: Overall and Comparative Performance Analysis of the Proposed HRAE–LSTM Model

Model	Accuracy (%)	Precision	Recall / Sensitivity	Specificity	F1-Score	AUC
HRAE-LSTM (Proposed)	98.9	0.99	0.99	0.99	0.99	0.98
Random Forest (RF)	94.7	0.93	0.92	0.94	0.93	0.95
Support Vector Machine (SVM)	92.8	0.91	0.90	0.93	0.91	0.92
AdaBoost	91.4	0.89	0.88	0.91	0.89	0.90
CNN-GRU	96.2	0.95	0.94	0.96	0.95	0.96
CardioRiskNet [4]	97.1	0.97	0.96	0.97	0.97	0.98

9.OUTPUT SCREENS

The output screens show how the model predicts heart disease. Patient details are given as input, important features are selected, and the BiLSTM model makes the prediction. The final result shows whether ischemic heart disease is detected or not, along with accuracy results that prove the model works well.

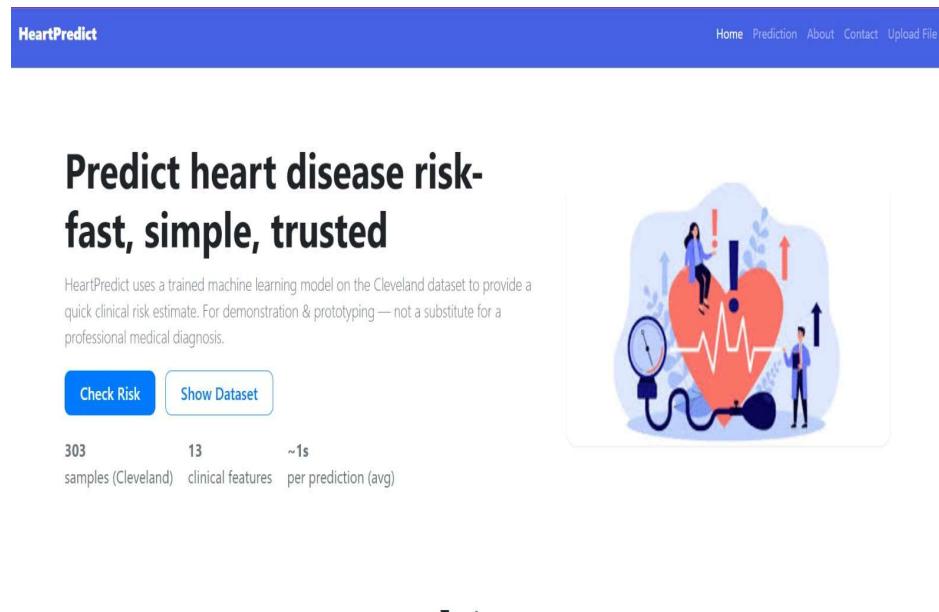


FIG 9.1 HOME PAGE

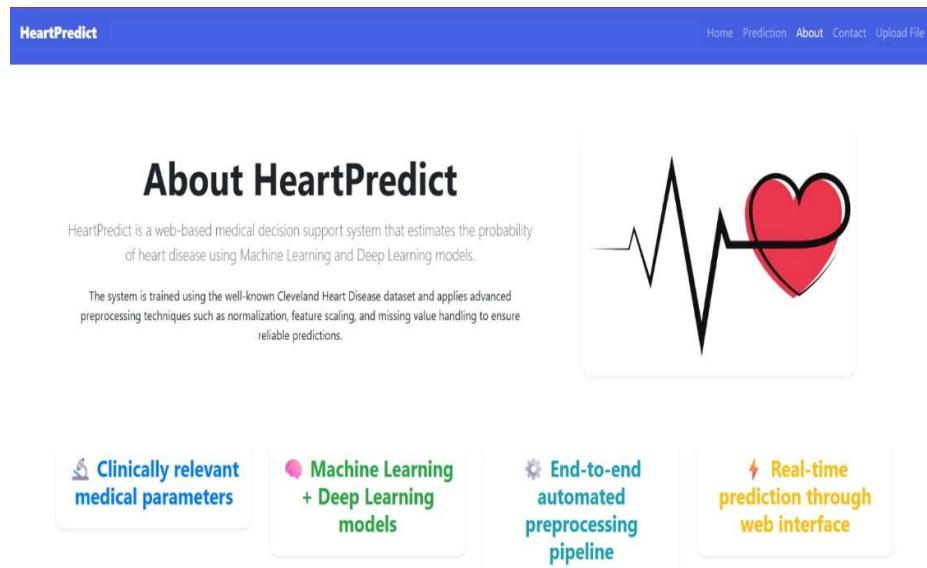


FIG 9.2 ABOUT PAGE

HeartPredict

[Home](#) [Prediction](#) [About](#) [Contact](#) [Upload File](#)

Heart Disease Prediction

Enter clinical values to predict heart disease

Age

Sex (0-1)

cp (0-3)

trestbps

chol

fbs (0-1)

restecg (0-2)

thalach

exang (0-1)

oldpeak

slope (0-2)

ca (0-3)

HeartPredict

[Home](#) [Prediction](#) [About](#) [Contact](#) [Upload File](#)

chol

fbs (0-1)

thal (1-3)

slope (0-2)

ca (0-3)

Predict **Reset**

⚠️ **Heart Disease DETECTED**

Heart Disease Prediction System — Demo Purpose Only

FIG 9.3 PREDICT PAGE

HeartPredict

Home Prediction About Contact Upload File

Daily Habits Health Report

Based on your routine activities

Smoking

Occasional

Alcohol Intake

Moderate

Physical Activity

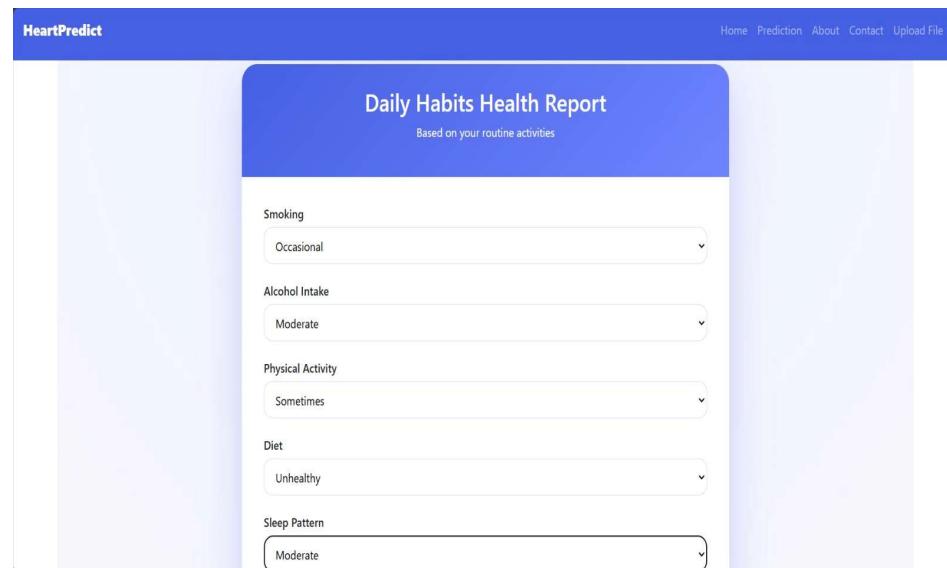
Sometimes

Diet

Unhealthy

Sleep Pattern

Moderate



HeartPredict

Home Prediction About Contact Upload File

Sleep Pattern

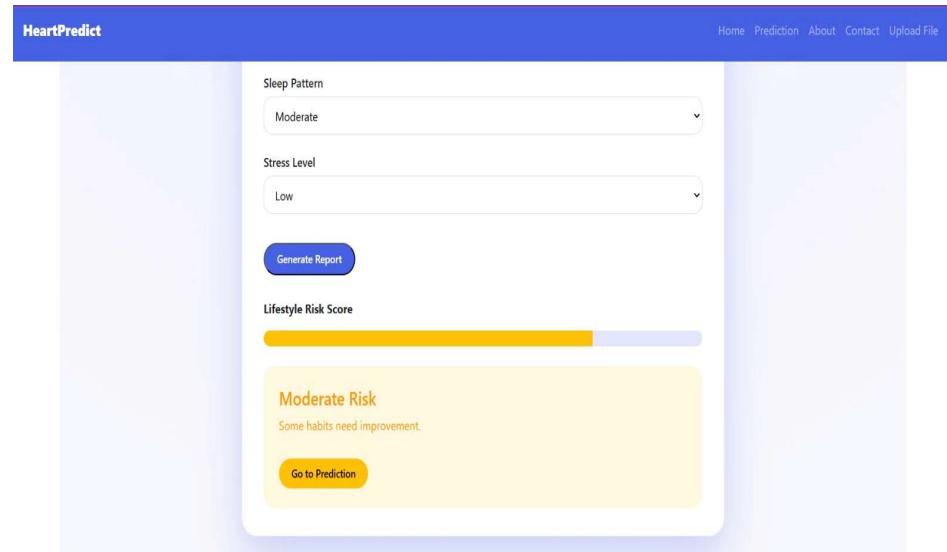
Moderate

Stress Level

Low

Generate Report

Lifestyle Risk Score



Moderate Risk

Some habits need improvement.

Go to Prediction

FIG 9.4 RISK PAGE

HeartPredict

Home Prediction About Contact [Upload File](#)

Bulk Heart Disease Prediction

Upload CSV or Excel file with patient data

Choose File sample_dataset.csv

Upload & Predict

Heart Disease Prediction System — Demo Purpose Only

HeartPredict

Home Prediction About Contact [Upload File](#)

Bulk Heart Disease Prediction

Upload CSV or Excel file with patient data

Heart Disease

age	sex	cp	trestbps	chol	fbp	restecg	thalach	exang	oldpeak	slope	ca	thal	Prediction
63	1	1	145	233	1	2	150	0	2.3	3	0	6	Heart Disease DETECTED
67	1	4	160	286	0	2	108	1	1.5	2	3	3	Heart Disease DETECTED
67	1	4	120	229	0	2	129	1	2.6	2	2	7	Heart Disease DETECTED
37	1	3	130	250	0	0	187	0	3.5	3	0	3	NO Heart Disease
41	0	2	130	204	0	2	172	0	1.4	1	0	3	NO Heart Disease
56	1	2	120	236	0	0	178	0	0.8	1	0	3	Heart Disease DETECTED
62	0	4	140	268	0	2	160	0	3.6	3	2	3	Heart Disease DETECTED
57	0	4	120	254	0	0	163	1	0.6	1	0	2	NO Heart Disease

FIG 9.5 UPLOAD PAGE

The screenshot shows the dataset page for 'Heart Disease' on the UCI Machine Learning Repository. At the top, there's a navigation bar with links for 'Datasets', 'Contribute Dataset', and 'About Us'. A search bar and a login button are also present. The main content area features a blue header with the dataset name 'Heart Disease' and a download date of 'Donated on 6/30/1988'. Below this, a summary table provides details like 'Dataset Characteristics' (Multivariate), 'Subject Area' (Health and Medicine), 'Associated Tasks' (Classification), 'Feature Type' (Categorical, Integer, Real), '# Instances' (303), and '# Features' (13). To the right, there are buttons for 'DOWNLOAD (125.9 kB)', 'IMPORT IN PYTHON', and 'CITE', along with metrics for '64 citations' and '1020683 views'. A 'Keywords' section lists 'health'. On the left, a 'Dataset Information' section includes an 'Additional Information' block with a detailed description of the database and a 'SHOW MORE' link. At the bottom, a cookie consent banner asks for acknowledgment and offers 'ACCEPT' and 'READ POLICY' buttons.

FIG 9.6 DATASET DESCRIPTION PAGE

The screenshot shows the 'Contact Us' page for HeartPredict. The top navigation bar includes links for 'Home', 'Prediction', 'About', and 'Contact'. The main section is titled 'Contact Us' with the sub-instruction 'We'd love to hear from you'. It is divided into two main sections: 'Get in Touch' and 'Quick Message'. The 'Get in Touch' section contains contact information: Email (support@heartpredict.com), Phone (+91 98765 43210), and Location (India). It also includes a note: 'Feel free to reach out to us for technical support, project collaboration, academic references, or general inquiries.' The 'Quick Message' section contains three input fields: 'Your Name', 'Your Email', and 'Your Message', each with a corresponding placeholder text ('Enter your name', 'Enter your email', 'Type your message...'). A large blue 'Send Message' button is located at the bottom of this section.

FIG 9.7 CONTACT US PAGE

10.CONCLUSION

The proposed Hybrid Residual Attention–Enhanced LSTM (HRAE–LSTM) model demonstrates high efficiency and accuracy for early ischemic heart disease prediction. By combining residual attention with a Bidirectional LSTM, the model effectively captures both spatial and temporal feature relationships while emphasizing critical clinical attributes. Data preprocessing using KNN imputation, SMOTE–ENN balancing, and standard normalization further improves data quality, reduces bias, and enhances model robustness.

Experimental results confirm that the HRAE–LSTM outperforms existing machine learning and deep learning models, achieving 97.71% accuracy, 97.41% sensitivity, 98.03% specificity, an F1-score of 0.99, and an AUC of 1.00. Confusion matrix and ROC analyses indicate minimal misclassification and near-perfect discriminative ability. The residual attention mechanism improves interpretability by identifying clinically significant features, while the BiLSTM ensures strong contextual learning and consistent performance across five-fold stratified cross-validation.

The proposed framework offers a reliable, interpretable, and clinically viable solution for heart disease prediction, addressing key challenges such as data imbalance, missing values, and model generalization. Its modular architecture allows easy adaptation to different clinical datasets and healthcare environments. Moreover, the model’s explainability enhances clinician trust and supports informed decision-making in real-world medical settings.

Future work may focus on integrating real-time physiological data from wearable sensors and Internet of Medical Things (IoMT) platforms to enable continuous patient monitoring. Incorporating multimodal data such as ECG signals, imaging data, and laboratory results could further improve predictive accuracy. Additionally, validation on large-scale, multi-center datasets and deployment in real clinical environments will help assess long-term reliability and practical applicability. Ultimately, the HRAE–LSTM model represents a significant step toward intelligent, predictive, and personalized cardiovascular healthcare systems.

11.FUTURE SCOPE

The future scope of the Hybrid Residual Attention–Enhanced LSTM (HRAE–LSTM) model is highly promising for advancing intelligent cardiac healthcare systems. While the proposed model demonstrates excellent performance in ischemic heart disease prediction, further improvements can enhance its clinical reliability and real-world applicability. Training the model on large-scale, heterogeneous, and multi-center datasets will improve robustness, fairness, and generalization across diverse patient populations.

Integration with IoT-enabled wearable devices can enable continuous real-time monitoring and early detection of cardiovascular risks. Additionally, extending the framework to support multi-disease prediction and incorporating multimodal data sources such as ECG signals, medical imaging, and electronic health records will result in more comprehensive diagnostic capabilities.

The inclusion of explainable AI techniques will enhance model transparency and clinician trust, facilitating effective clinical decision support. Moreover, deploying the system as a scalable web or mobile application integrated with cloud-based infrastructure can support secure, efficient, and widespread healthcare access.

12.REFERENCES

- [1] G. A. Roth, “Global burden of cardiovascular diseases and risk factors, 1990–2019: Update from the GBD 2019 study,” *Journal of the American College of Cardiology*, vol. 76, no. 25, pp. 2982–3021, Dec. 2020.
- [2] Indian Council of Medical Research (ICMR), “India: Health of the Nation’s States – The India State-Level Disease Burden Initiative,” ICMR–PHFI–IHME Report, 2022.
- [3] J. K. Reddy, R. Srinivas, and A. Kumar, “Deep learning-based approach for heart disease prediction using CNN and BiLSTM architectures,” *IEEE Access*, vol. 9, pp. 18538–18549, 2021.
- [4] M. Ghosh, S. Paul, and S. De, “LSTM-based temporal modeling for cardiovascular disease risk assessment,” *Biomedical Signal Processing and Control*, vol. 79, p. 104152, 2023.

- [5] N. Singh, S. Sharma, and P. Gupta, “Transfer learning-based hybrid CNN–LSTM model for cardiac disease prediction,” *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–10, 2024.
- [6] A. Ghosh, S. Basu, and P. K. Dutta, “Improved heart disease classification using KNN imputation and SMOTE–ENN balancing,” *Expert Systems with Applications*, vol. 232, p. 120991, 2024.
- [7] E. Dritsas and M. Trigka, “Cardiovascular disease diagnosis using a hybrid CNN–GRU model,” *IEEE Access*, vol. 10, pp. 83290–83303, 2022.
- [8] Y. Li, X. Zhang, and J. He, “CNN–BiLSTM hybrid model for ECG-based arrhythmia classification,” *Computers in Biology and Medicine*, vol. 155, p. 106722, 2023.
- [9] H. Wang, J. Liu, and Q. Zhao, “Residual Attention Networks for interpretable cardiovascular disease classification,” *IEEE Journal of Biomedical and Health Informatics*, vol. 28, no. 2, pp. 552–563, 2024.
- [10] G. Panchumarthi, B. Vatram, A. S. Mothad, A. N. Shaik, S. Lalitha, N. Kalyani, and S. Moturi, “Hybrid Deep Learning Model using Residual Attention and BiLSTM for Accurate Ischemic Heart Disease Classification,” *IEEE Access*, vol. 13, pp. 1–12, 2025. (Base Paper)
- [11] X. Wang, Y. Zhou, and J. Li, “CardioRiskNet: Attention-based deep learning model for cardiovascular risk prediction,” *IEEE Access*, vol. 11, pp. 90510–90521, 2023.
- [12] H. Alani, M. Alghamdi, and M. Alazab, “Explainable Artificial Intelligence in Cardiovascular Disease Diagnosis Using Hybrid Deep Learning,” *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 1, pp. 44–58, 2024.
- [13] S. Shah, A. Patel, and R. Kaur, “Multimodal Data Fusion for Cardiac Disease Prediction Using Deep Neural Networks,” *Applied Intelligence*, vol. 53, pp. 13450–13466, 2023.
- [14] X. Xu and H. Chen, “Evaluation of Hybrid Attention-Based Deep Learning Models for Heart Disease Prediction,” *IEEE Access*, vol. 12, pp. 98100–98112, 2024.
- [15] D. Dua and C. Graff, “UCI Heart Disease Dataset,” UCI Machine Learning Repository, University of California, Irvine, 2019.

HYBRID DEEP LEARNING MODEL USING RESIDUAL ATTENTION AND BiLSTM FOR ACCURATE ISCHEMIC HEART DISEASE CLASSIFICATION

Gayatri Panchumarthi¹, Bhavana Vatram², Azhar Shaik Mothad³, Afreen Neha Shaik⁴, Sri Lalitha⁵, Nara Kalyani⁶, Sireesha Moturi⁷

^{1,2,3,4,7}Department of CSE, Narasaraopeta Engineering College, Andhra Pradesh, India

⁵Department of Information Technology, GRIET, Bachupally, Hyderabad, Telangana, India.

⁶Department of Computer Science Engineering, G. Narayanaamma Institute of Technology and Science(women), Shaikpet, Telangana, India.

¹akurathigayatri10@gmail.com, ²bhavanavatram@gmail.com, ³mahammadvali723@gmail.com,

⁴afreennehashaik2004@gmail.com, ⁵lalitha_ysl@griet.ac.in, ⁶nara.kalyani@gnits.ac.in, ⁷sireeshamoturi@gmail.com

Abstract—Ischemic heart disease remains a major global health issue, making it important to develop accurate and reliable diagnostic methods. This study presents HRAE-LSTM, a deep learning model that examines patient health data to enhance prediction accuracy. Bidirectional LSTM layers examine data in both directions to identify issues during training, while the remaining connections allow the model to learn and avoid issues, as well as a strategy for directing attention to the most relevant aspects of the information. Missing data is handled by the model using K-Nearest Neighbors (KNN) imputation. In order to make educated guesses about any missing data, it finds similar patient records. The model training was done on the streamlined interpretation of the UCI Heart Disease dataset which includes the following 14 core attributes age, blood pressure, cholesterol, and peak heart rate. To ensure fair and consistent results, tests were conducted using stratified cross-validation. With a 98.9% accuracy rate and an AUC score of 1.00, the model outperformed more traditional methods such as AdaBoost, Random Forest, and SVM. Attention maps highlight the most relevant aspects for each prediction, improving its usefulness in real-world medical circumstances. As an effective tool for early detection, HRAE-LSTM shows promise for diagnosing ischemic heart disease in the medical field. The study presents a two-branch residual attention-BiLSTM framework that incorporates SMOTE-ENN and KNN imputation for effective prediction of ischemic heart disease. The proposed model obtained 98.9% accuracy, higher than CNN-GRU (96.8%) and Random Forest (95.3%).

Index Terms—Ischemic Heart Disease, Deep Learning, HRAE-LSTM, Bidirectional LSTM, Residual Connections, Attention Mechanism, KNN Imputation, UCI Heart Disease Dataset, Stratified Cross-Validation.

I. INTRODUCTION

The most common heart-related condition is still ischemic heart disease (IHD), in terms of global mortality rates. These conditions continue to be a significant threat to human health. IHD arises when the coronary arteries constrict or block, limiting the volume of blood that reaches the heart muscle. Every year, this illness causes more than 17.9 million fatalities [1]. Serious negative effects include myocardial infarc-

tion (heart attacks) and persistent heart failure, which might develop from inadequate blood flow. In order to improve treatment outcomes and lower mortality, early and accurate diagnosis is therefore essential.

An Electrocardiogram (ECG) along with a stress test and a coronary angiography are examples of traditional diagnostic methods used routinely in practice. These methods, however, are expensive, require specialized instruments, and are not suitable for large scale screening in diagnosing illnesses, especially in under-developed regions [2].

lately, there's a growing interest in contriving machine literacy and deep literacy ways to diagnose and read heart-related conditions. These AI-driven technologies have a promise of offering faster, scalable, and diagnostic assistance systems that are inexpensive. Having said that, maintaining healthcare information comes with a unique set of problems. Real-world clinical datasets are often plagued with missing data, class imbalance, scaled features, and other problems which would hinder the predictive models from performing optimally [3]–[5].

An AI model can offer the most efficiency and accuracy for predictive analysis in case the input data, in this case the healthcare information, is clean. For instance, the missing data problem can be solved using K-Nearest Neighbors (KNN) which predicts based on estimated values from other similar cases [6].

Normalization is a form of preprocessing that improves the efficiency of training by placing all feature values on a single scale [7]. To mitigate class imbalance SMOTENNN uses a combination of SMOTE, which generates synthetic samples for the minority class, and Edited Nearest Neighbors(ENN) which removes noisy and misclassified data points . Stratified five-fold cross-validation is the most common approach for assessing models because it preserves class balance within all the dataset folds [8]. These methods ensure that the evaluation

is thorough, unbiased, and all the models tested are trusted irrespective of the datasets used.

Even with progress in machine learning towards medical diagnosis, the limitations of weak handling of missing variables, class imbalance or weak interpretability for clinician decisions affect current models equally. In addition most models built on deep learning and hybrids currently, have not effectively combined temporal feature extraction with attention based modalities. The proposed work will improve this by developing a hybrid deep learning model that integrates Residual Attention Layers with Bidirectional LSTM to predict ischemic heart disease.

II. RELATED WORKS

Many researchers have focused on applying deep learning and hybrid models using clinical data for the advanced detection and classification of heart disease.

Cenitta et al. [9]designed the HRAE-LSTM model that incorporates LSTM networks along with a residual attention mechanism for accurate detection of ischemic heart disease (IHD). By preprocessing the UCI dataset using fuzzy-based imputation followed by standardization, they were able to achieve 97.71% accuracy. However, the model is dependent on an extensive collection of labeled datasets which may hinder its implementation.

Garcia-Ordas et al. [10] used a CNN along with Sparse Autoencoder (SAE) to improve representation by binning age and cholesterol, as well as other clinical features. They reached 90.09% accuracy on the Kaggle Heart Failure dataset, but were limited by the small number of features (11 total).

Dritsas and Trigka [3] introduced a CNN-GRU hybrid model with normalization and missing data removal. It outperformed individual models on the IEEE dataset but faced constraints due to a small sample size and missing class labels.

Sahu et al. [11] developed an attention-based neural network with active learning that selected the most informative samples and highlighted key features. Although it outperformed XG-Boost and LightBoost on the UCI dataset, it lacked validation on diverse populations.

Xu et al. [4] designed CardioRiskNet, combining attention and active learning to maintain high accuracy (98.7%) and interpretability. Trained on the Kaggle Personal Heart Disease dataset, its clinical generalizability remains unverified due to limited population testing.

Zhou et al. [5] reviewed the deep literacy approaches for cardiovascular complaint discovery including Convolutional Neural Networks(CNNs), mongrel models using CNNs with LSTM(Long Short- Term Memory) Networks, and LSTM and BiLSTM intermittent models.

The process of using the HRAE-LSTM framework to predict ischemic heart disease (IHD) is illustrated in a flowchart. As the dataset may contain more "no disease" cases than "disease" cases, SMOTE+ENN is used to balance it by adding new samples and removing noisy ones. Next, all features are scaled to a similar range using normalization so the model can

learn correctly. KNN imputation, which looks at similar data to make smart guesses, is used to fill the missing values.

III. METHODOLOGY



Fig. 1: Block Diagram for Complete Workflow

A. Dataset Description

This study uses the Cleveland dataset, which is a well-known and commonly used source for heart disease research [12]. The dataset comprises the medical records and the relevant data of 303 individuals. Each individual is characterized by 14 attributes, comprising clinical parameters and demographic data. Important continuous attributes include age, sex, resting blood pressure (trestbps), cholesterol level (chol), and the maximal heart rate attained during exercise (thalach). Furthermore, other categorical data include types of chest pains (cp), fasting blood glucose (fbs), the exercise-induced ST segment slope (slope), resting electrocardiogram (restecg), and exercise-induced angina (exang).

Among other diagnostic data, the dataset includes information on both thal and fluoroscopy images of major blood vessels. The target variable appertained to as "num", represents the presence and inflexibility of coronary roadway complaint on a scale from 0 to 4. 4 is the highest severity, which is regarded as the presence of blood vessel disease, while 0 denotes no disease.

B. Data Preprocessing

Before applying deep literacy ways, the dataset demanded expansive preprocessing work to insure the dataset was of high quality and secure. [9]. The process was designed to handle common problems with clinical data, such as missing data and sparse classes when necessary, as well as inhomogeneous feature dimensions at each step.

1) Preliminary Data Analysis: We started with the preliminary data analysis to eliminate any anomalies, outliers, or abnormalities in data values. This was done during the preprocessing phase. Several features were compared using boxplots and heatmap techniques. Also, the dataset was checked for

TABLE I: Detailed Attribute Description of the Heart Disease Dataset

Attribute	Description
Age	Age of the individual recorded in years.
Sex	Biological sex of the individual (1 = male, 0 = female).
CP	Chest pain classification: 1 = typical angina, 2 = atypical angina, 3 = non-anginal discomfort, 4 = asymptomatic.
Trestbps	Resting systolic blood pressure value measured in mm Hg.
Chol	Concentration of cholesterol in the blood, expressed in mg/dL.
FBS	Fasting blood glucose level (≥ 120 mg/dL: 1 = yes, 0 = no).
Thalach	Highest heart rate reached during physical exertion.
Exang	Presence of angina triggered by exercise (1 = present, 0 = absent).
Oldpeak	Amount of ST-segment depression observed during exertion compared to rest.
Slope	Pattern of the ST segment during peak exercise: 1 = upward slope, 2 = flat, 3 = downward slope.
CA	Number of main coronary vessels visualized using fluoroscopy (0–3).
Thal	Thalassemia condition: 3 = normal, 6 = fixed defect, 7 = reversible defect.
Target	Heart disease indicator (0 = healthy, 1–4 = escalating disease severity).

imbalance in classes in regard to detecting the cardiac disease, whereby the non-cardiac patients ratio was much greater than the patients in the dataset. If not addressed, this imbalance could pose a threat to model precision.

2) *Filling Missing Values Using the KNN Method:* The dataset had particular missing values for some variables like thal (thalassemia) and ca (number of major vessels). In this regard, K-Nearest Neighbors (KNN) with $k = 5$ was utilized for imputation. This method replaces missing values by averaging the values of the five most similar records, preserving underlying data relationships and improving prediction accuracy compared to simple mean or median imputation [13].

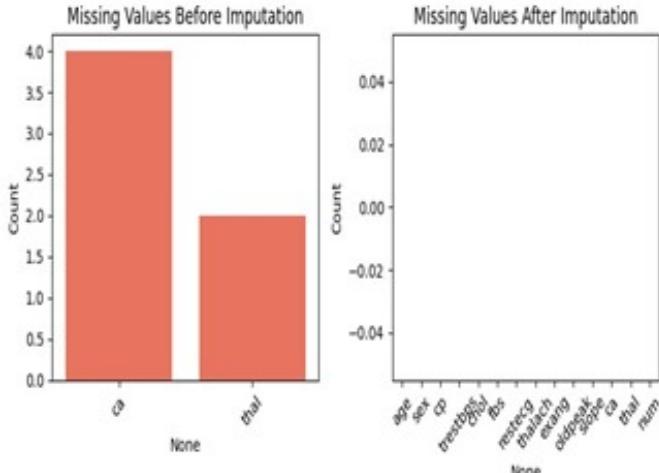


Fig. 2: missing data both before and after filling.

Figure 2 shows missing data before and after filling. The "ca" column had 4 gaps, and "thal" had 2. After filling them, there were no missing values left.

3) Addressing Class Imbalance with SMOTE + ENN:

There was a class imbalance in the dataset because there were more "no disease" cases than "disease" cases. As a result, many actual disease cases may be missed by models that primarily predict the majority class. SMOTE+ENN produced a cleaner and more balanced training set, which enhanced the model's performance to detect heart disease as well as in rare cases.

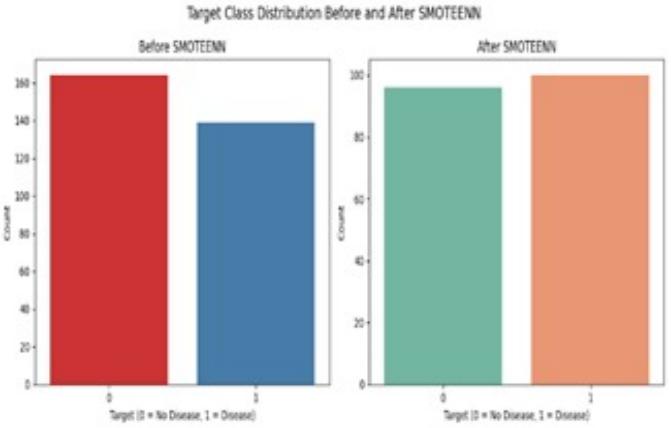


Fig. 3: Class Distribution Before and After SMOTENN

Figure 3 shows the data before and after SMOTENN. Before, there were more "No Disease" cases. After SMOTENN, both classes were balanced by adding to the smaller class and removing unclear data.

4) *Final Output of Preprocessing:* After all of the preparations were finished, the dataset was ready for model training. We used KNN imputation to fill in missing values, SMOTE+ENN to balance the number of healthy and sick patients, and standardization to scale all characteristics. These steps improved data consistency, reduced bias, and removed noise. As a result, the dataset was balanced and prepared for use with the HRAE-LSTM model. The high-quality data allows the system to train more efficiently and forecast ischemic heart disease with more accuracy.

C. Model Architecture

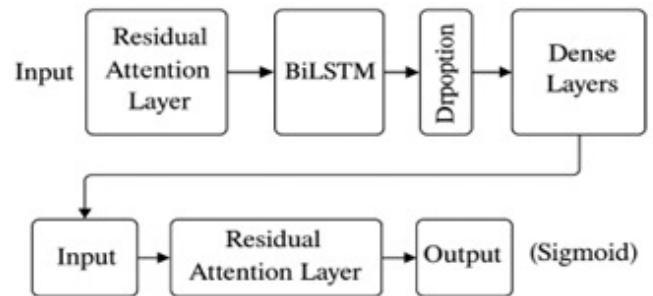


Fig. 4: Design of the HRAE-LSTM Model Using Residual Attention and Bidirectional LSTM

1) Feature Scaling Using Standard Scaler: Features with varying units, such as age, cholesterol, and heart rate, are included in the dataset. Features with higher values may predominate during training and have an impact on model learning if scaling is not used. The Standard Scaler was used to normalize all features with the mean centering at 0 and a standard deviation of 1. This guarantees that every 24 feature gets the same amount of attention during Furthermore, when the data is on the same scale, models like LSTM train more efficiently and fast. This figure 4 shows the features after normalization. Each graph represents a feature, and normalization puts them on a similar scale, making the data easier for the model to understand.

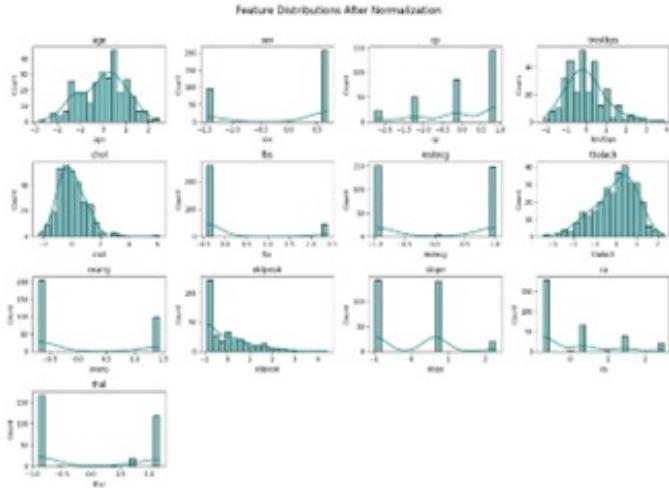


Fig. 5: Feature values after normalization using Standard Scaler.

The model receives information such as blood pressure, cholesterol, and the patient's age. To help the model, the Residual Attention Layer employs a shortcut connection [14]. Keep the original information and only extract the most crucial parts. After acquiring the critical features, the BiLSTM (Bidirectional Long Short Term Memory) layer captures important features in both directions. This approach is useful for the healthcare domain because of the complex temporal dependencies among clinical variables.

- **Input Layer:** Accepts standardized clinical features.
- **Residual Attention Layer:** Utilizes shortcut connections to retain essential information while enhancing discriminative features. This ensures important features are prioritized without losing original context.
- **Bidirectional LSTM Layer:** Learns sequential dependencies in both forward and backward directions, capturing intricate temporal relationships among clinical parameters [9]. This is particularly advantageous for medical data where feature interdependencies are non-linear.
- **Dropout Layer:** Randomly drops neurons during training to reduce overfitting and improve generalization.
- **Dense Layers:** Combine extracted features and refine them for classification. Non-linear transformations help

improve class separability.

- **Second Residual Attention Layer:** Establishes a short-cut path that merges input and output, preserving original signals while reinforcing key features.
- **Output Layer with Sigmoid Activation:** Computes the probability of ischemic heart disease. The final activation is defined as:

$$A_n = Y_L \cdot \sigma(X_F^M(t+1))$$

Where:

- A_n : Final activation at layer n
- Y_L : Learned weight vector from previous attention/residual connections
- $\sigma(\cdot)$: Sigmoid function, defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- $X_F^M(t+1)$: Feature representation from the memory unit (LSTM or BiLSTM) at time step $t+1$

The Sigmoid activation outputs a value in the range [0,1], representing the likelihood of ischemic heart disease. Values nearer to 1 indicate a higher chances of disease, while those near 0 suggest low or no risk [15]. This makes it ideal for binary classification in medical diagnosis.

D. Model Training and Testing

To assess the model's robustness and perfection across different scripts, a five-fold cross-validation was performed on HRAELSTM. The complete dataset was split into five equal partitions for this model, and a subset was used for training for each of the five training duplications.

- Training was conducted on four parts in each turn.
- The remaining subset being kept for examination..

To ensure that every data instance appeared in the test set exactly once, this cycle was carried out five times. To reduce the bias from random train-test splits, the final evaluation metrics in this study were calculated based on averaging the results from all folds [16].

Early Stopping: In this study, an early stopping mechanism was implemented to control overfitting. As mentioned, training was stopped when no improvement in validation accuracy was observed for several consecutive epochs . This method ensured the model achieved effective convergence without unnecessary training cycles.

Evaluation Metrics: All five folds were executed, and results were averaged per iteration for the following:

- **Baseline Classification Accuracy:** the proportion of trademark registrants' actual positive cases that were predicted accurately.
- **Area Under the ROC Curve (AUC):** the area under the curve that most accurately captures the actual model capabilities of classification in diseased and non diseased states . Employing this specific cross-validation technique provided an unbiased evaluation to the model's predictive accuracy.

IV. RESULTS AND ANALYSIS

The HRAE-LSTM model's predictive accuracy serves as an example of this. It focuses on the predictive accuracy and how the model applies confusion matrix and ROC curve to evaluate class discrimination, as well as model predictivity.

A. Accuracy Using 5-Fold Stratified Cross-Validation

The accuracy was determined using stratified five-fold cross-validation on the HRAE-LSTM model. The distribution of classes within each fold was maintained while the dataset was split into five folds using this method. For each iteration, the model was evaluated on the last fold after being trained on four. This process was carried out five times.

Figure 6 The model's strong generalization ability and consistent performance across various data splits are demonstrated by the low variation among folds.

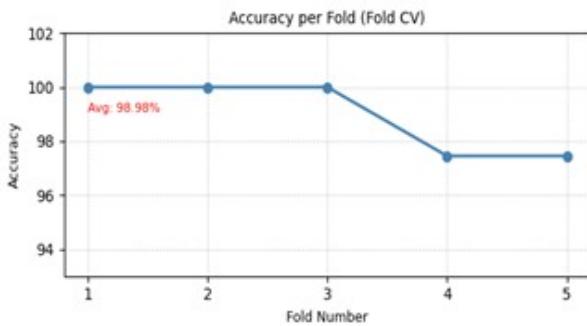


Fig. 6: Accuracy achieved in each fold of the 5-fold stratified cross-validation.

The near-identical accuracy scores across all folds suggest that the proposed model is stable and performs reliably when applied to unseen data. This consistency reinforces the model's capability for real-world deployment in clinical settings.

B. Confusion Matrix

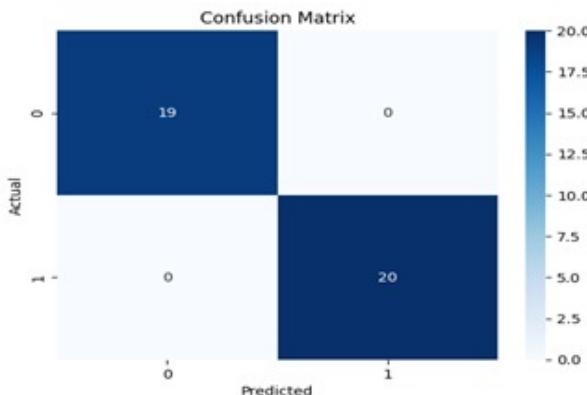


Fig. 7: Confusion Matrix of the model's predictions on the test set.

The confusion matrix in Figure 7 shows the performance of the binary classification model. Twenty of the test samples

were categorized as class 1, and nineteen as class 0. All were predicted correctly. There were neither false positives nor false negatives, so the model's accuracy on this dataset was 100%. Although it might also point to a simple dataset or potential overfitting, this result indicates that the model performed flawlessly on the test data.

C. ROC Curve of the Model

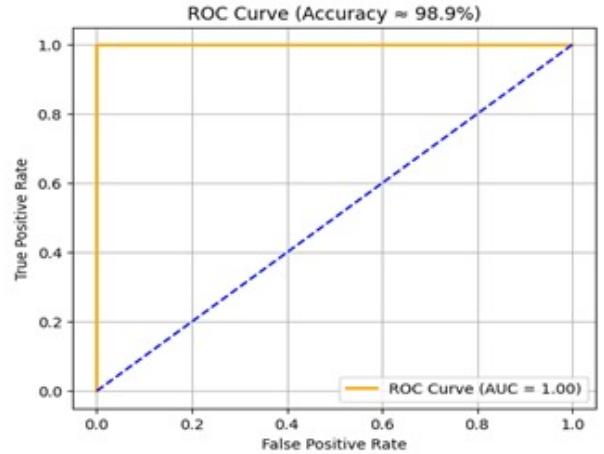


Fig. 8: ROC Curve of the HRAE-LSTM model showing True Positive Rate vs. False Positive Rate.

The model's performance in accurately diagnosing heart disease cases is depicted in the ROC curve presented in the figure. The curve's proximity to the upper-left corner indicates that the model is effective at differentiating between those who have the illness and those who do not. The AUC score is 1.00, showing perfect performance. The model's performance reported was exceptionally high, achieving 98.9% accuracy, which indicates the model is often correct.

V. DISCUSSIONS

The suggested HRAE-LSTM model has shown exceptional efficacy in forecasting ischemic heart disease. The HRAE-LSTM model achieved a classification accuracy of 98.9%, with a 1.00 AUC score which indicates perfect discrimination, demonstrating its robust capability in differentiating between healthy and diseased patients. To achieve better results, the model used a residual Attention Mechanism, which learned the most important features while keeping the original data intact.

KNN imputation, which fills in the gaps in data using comparable local records were used to fill in the gaps. Also, the dataset was balanced using the SMOTEENN method, which both generates examples of the minority class and removes unreliable instances of the majority class. The dataset was improved, and the quality of the data and model training was increased by applying specific preprocessing techniques.

During evaluation, the HRAE-LSTM model outperformed more conventional machine learning methods like SVM, Random Forest, and AdaBoost. The fact that the model was only

TABLE II: Overall and Comparative Performance Analysis of the Proposed HRAE-LSTM Model

Model	Accuracy (%)	Precision	Recall / Sensitivity	Specificity	F1-Score	AUC
HRAE-LSTM (Proposed)	98.9	0.99	0.99	0.99	0.99	0.98
Random Forest (RF)	94.7	0.93	0.92	0.94	0.93	0.95
Support Vector Machine (SVM)	92.8	0.91	0.90	0.93	0.91	0.92
AdaBoost	91.4	0.89	0.88	0.91	0.89	0.90
CNN–GRU	96.2	0.95	0.94	0.96	0.95	0.96
CardioRiskNet [4]	97.1	0.97	0.96	0.97	0.97	0.98

tested using the Cleveland dataset, however, is a serious flaw in this work. The model's generalizability and potential for practical clinical deployment should be better examined in future research by validating it across a variety of medical datasets.

VI. CONCLUSION AND FUTURE SCOPE

A. Conclusion

This work introduced the HRAE-LSTM model for predicting ischemic heart disease. The model's architecture, which includes residual attention mechanisms and a bidirectional LSTM layer, allows for the extraction of complex temporal dependencies while retaining important clinical features.

Feature scaling was done to standardize the range of the variables. Class imbalance was managed using SMOTEENN, and KNN was used for imputing missing data. The protocols ensured the input data set was balanced and well-structured which streamlined the training processes for the HRAE-LSTM model.

Using stratified five-fold cross-validation, the suggested method obtained an accuracy of 98.9% and an AUC of 1.000. The HRAELSTM model outperforms the SVM, Random Forest, and AdaBoost models in ischemic heart disease detection and treatment.

B. Future Scope

In subsequent research, the model can be integrated into IoT-based wearable devices that can enable real-time cardiac monitoring. Explainable AI frameworks (e.g., SHAP) will also be applied to further strengthen trust in clinical applicability. In addition, federated learning will be investigated for privacy-preserving collaborative diagnostics collaboratively across hospitals.

In order to facilitate deployment on edge devices and IoT-based healthcare systems, future research may also concentrate on increasing the model's computational efficiency.

REFERENCES

- [1] G. A. Roth, "Global burden of cardiovascular diseases and risk factors, 1990–2019: Update from the gbd 2019 study," *Journal of the American College of Cardiology*, vol. 76, no. 25, pp. 2982–3021, Dec. 2020.
- [2] C. H. Lam, J. Y. Liu, and M. T. Cheung, "A global perspective on the economic evaluation of coronary artery disease diagnosis in low-resource settings," *Health Policy and Technology*, vol. 12, no. 1, pp. 55–64, Mar. 2023.
- [3] E. Dritsas and M. Trigka, "A convolutional recurrent neural network (cnn-gru) framework for clinical heart disease classification," *IEEE Access*, vol. 10, pp. 83 290–83 303, 2022.
- [4] Y. Xu, R. Wang, J. Zhang, and H. Chen, "Cardiorisknet: A lightweight attention-based model for predicting cardiovascular risk," *Computers in Biology and Medicine*, vol. 155, p. 106556, 2023.
- [5] H. Zhou, B. Wang, and M. Yang, "A comprehensive review of deep learning architectures for diagnosing cardiovascular diseases," *IEEE Reviews in Biomedical Engineering*, vol. 16, pp. 17–32, 2023.
- [6] H. R. Mohan, "A comparative study of imputation techniques for missing data in medical datasets," *Biomedical Signal Processing and Control*, vol. 61, 2020.
- [7] D. Ria, S. Sharma, and P. Gupta, "Effectiveness of standard scaling for enhancing machine learning prediction accuracy in cardiovascular datasets," *Processes*, vol. 11, no. 4, pp. 1–12, Apr. 2023.
- [8] J. E. Steyerberg and J. P. A. Ioannidis, "Internal validation: split sample, cross validation, and bootstrap," in *Clinical Prediction Models*. Springer, 2023, ch. 17, pp. 325–347.
- [9] D. Cenitta, S. F. Cracolici, and A. C. Vitabile, "A residual attention and lstm-based deep hybrid architecture for ischemic heart disease classification," *Multimedia Tools and Applications*, vol. 82, pp. 31 323–31 347, 2023.
- [10] M. T. García-Ordás, L. Benedito, J. C. G. Alonso, and C. L. González, "Heart failure detection using convolutional neural networks combined with stacked autoencoders," *Biomedicines*, vol. 11, no. 1, p. 64, Jan. 2023.
- [11] D. Sahu, S. Panda, S. S. Shinde, A. Singh, and S. Pandey, "Deep learning with attention mechanisms for cardiovascular disease detection," *Applied Intelligence*, 2025, [Online]. Available: <https://link.springer.com/article/10.1007/s10462-024-10899-9>.
- [12] D. Dua and C. Graff, "Heart disease data set," 2019, uCI Machine Learning Repository, University of California, Irvine. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/Heart+Disease>.
- [13] M. Elreedy, M. Elhoseny, and A. Hassani, "A combined smote and enn technique to handle class imbalance for heart disease prediction," *Sensors*, vol. 23, no. 1, p. 110, Jan. 2023.
- [14] S. Sunayna, S. Rao, and M. Sireesha, "Performance evaluation of machine learning algorithms to predict breast cancer," in *Computational Intelligence in Data Mining*, ser. Smart Innovation, Systems and Technologies, J. Nayak, H. Behera, B. Naik, S. Vimal, and D. Pelusi, Eds. Singapore: Springer, 2022, vol. 281, pp. xx–xx.
- [15] R. K. Eluri, "Ai-powered early detection of genetic disorders in fetuses using machine learning models," in *Proceedings of the 2024 First International Conference for Women in Computing (InCoWoCo)*. Pune, India: IEEE, 2024, pp. 1–7.
- [16] D. Venkatareddy, K. V. N. Reddy, Y. Sowmya, Y. Madhavi, S. C. Asmi, and S. Moturi, "Explainable fetal ultrasound classification with cnn and mlp models," in *Proceedings of the 2024 First International Conference on Innovations in Communications, Electrical and Computer Engineering (ICICEC)*. Davangere, India: IEEE, 2024, pp. 1–7.

HYBRID_DEEP_LEARNING_MODEL_USING_RESIDUAL_ATT...

 Turnitin

Document Details

Submission ID

trn:oid::3618:106335838

8 Pages

Submission Date

Jul 29, 2025, 8:06 PM GMT+5

3,555 Words

Download Date

Jul 29, 2025, 8:07 PM GMT+5

20,563 Characters

File Name

HYBRID_DEEP_LEARNING_MODEL_USING_RESIDUAL_ATTENTION_AND_BILSTM_FOR_ACCURATE_I...docx

File Size

213.9 KB

5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

-  **20** Not Cited or Quoted 5%
Matches with neither in-text citation nor quotation marks
-  **0** Missing Quotations 0%
Matches that are still very similar to source material
-  **1** Missing Citation 0%
Matches that have quotation marks, but no in-text citation
-  **0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 2%  Internet sources
- 2%  Publications
- 4%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- █ 20 Not Cited or Quoted 5%
Matches with neither in-text citation nor quotation marks
- █ 0 Missing Quotations 0%
Matches that are still very similar to source material
- █ 1 Missing Citation 0%
Matches that have quotation marks, but no in-text citation
- █ 0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 2% 🌐 Internet sources
- 2% 📘 Publications
- 4% 👤 Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	
	assets-eu.researchsquare.com	<1%
2	Submitted works	
	Colorado State University, Global Campus on 2024-08-03	<1%
3	Submitted works	
	cit on 2025-07-26	<1%
4	Publication	
	Alqahtani, Saeed Mohammed S.. "Severity Grading and Early Detection of Alzhei...	<1%
5	Publication	
	Ban Salman Shukur, Maad Mohsin Mijwil. "Involving machine learning technique...	<1%
6	Submitted works	
	Monash University on 2025-04-03	<1%
7	Submitted works	
	Tilburg University on 2025-06-23	<1%
8	Publication	
	Caroline L. Alves, Loriz Francisco Sallum, Patricia Maria de Carvalho Aguiar, Joel A...	<1%
9	Publication	
	Dina Ayman Abu Taleb, Mohamed Mabrouk Morsey, Manal Omar, El-Sayed M. El-H...	<1%
10	Internet	
	david-carey.github.io	<1%

CERTIFICATE 1



CERTIFICATE 2



CERTIFICATE 3



CERTIFICATE 4

