HEART DISEASE PREDICTION USING DEEP LEARNING

by

ABDULSALAM M (71382006001)
AKILNANDA EK (71382006004)
NIJEETH KUMAR T (71382006302)

Report submitted in partial fulfillment of

the requirements for the Degree of

Bachelor of Technology in

Information Technology

Sri Ramakrishna Institute of Technology

Coimbatore – 641010

May 2024

ACKNOWLEDGEMENT

First, we record our grateful thanks to the almighty for his blessings to make this project a grand success.

We express our sincere thanks to our respected and honourable Principal **Dr. M. Paulraj,** for granting us permission to undergo the project.

We record our heartfelt gratitude to **Dr. M. Suresh Kumar,** Professor and Head (in - charge), Department of Computer Science and Engineering and **Dr. J. J. Adri Jovin,** Associate Professor and Head (in-charge), Department of Information Technology for rendering timely help for the successful completion of this project.

We thank our Project Coordinator **Dr. T. C. Ezhil Selvan,** Associate Professor, Department of Information Technology, Sri Ramakrishna Institute of Technology for his guidance throughout the entire period for the completion of the project.

We are greatly indebted to our project supervisor, **Mr. K. Sathyaseelan,** Assistant Professor, Department of Information Technology, Sri Ramakrishna Institute of Technology for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We thank our parents and friends for the strong support and inspiration they have provided us in bringing out this project successfully.

APPROVAL AND DECLARATION

This project report titled “Heart disease prediction using deep learning” was prepared and submitted by **Abdulsalam M (Register Number: 71382006001), Akilnanda EK (Register Number: 71382006004), Nijeeth Kumar T (Register Number: 71382006302)** and has been found satisfactory in terms of scope, quality, and presentation as partial fulfillment of the requirement for the Bachelor of Technology (Information Technology) in Sri Ramakrishna Institute of Technology, Coimbatore (SRIT).

Checked and approved by

Mr. K. Sathyaseelan

Project Supervisor

Assistant Professor

Department of Information Technology,
Sri Ramakrishna Institute of Technology,
Coimbatore -10.

Nov, 2023

BONAFIDE CERTIFICATE

Certified that this project report "**Heart Disease Prediction Using Deep Learning"** the Bonafide work of **Abdulsalam M (71382006001), Akilnanda EK (71382006001), Nijeeth Kumar T (71382006302)** who carried out the project work under my supervision.

SUPERVISOR

SIGNATURE
Mr. K. Sathyaseelan
ASSISTANT PROFESSOR
Department of Information Technology
Sri Ramakrishna Institute of Technology
Coimbatore- 10.

SIGNATURE
Dr.J. J. Adri Jovin
PROFESSOR AND HEAD
Department of Information Technology
Sri Ramakrishna Institute of Technology
Coimbatore- 10.

Submitted for viva-voice examination held on……

INTERNAL EXAMINER **EXTERNAL EXAMINER**

PREDICTION OF HEART DISEASE USING DEEP LEARNING

ABSTRACT

This research uses deep learning techniques to present a novel method for heart disease prediction. It does this by utilizing neural networks to assess a wide range of patient data. The model combines a variety of data, including clinical measures, medical history, and demographic information, to provide a comprehensive picture of cardiovascular risk factors. The deep neural network outperforms traditional models in prediction because it is trained on a vast and varied dataset, which enables it to identify complex patterns and connections in the data automatically.

The interpretability of the model is examined by feature significance analysis, in addition to its predictive capabilities. Along with providing insight into the key variables impacting cardiovascular risk, this research confirms the model's predictions. Developing clinician confidence and enabling the model's incorporation into clinical decision-making procedures depend heavily on this interpretability feature. The work highlights how deep learning may help to predict and avoid cardiac disease and creates opportunities for additional development, ongoing enhancement, and practical use in healthcare environments.

TABLE OF CONTENTS

L IST OF FIGURES

LIST OF ABBREVIATIONS

LR	Logistic Regression
ML	Machine Learning
HD	Heart Disease
KNN	K Nearest Neighbors
SVM	Support Vector Machine
DNN	Deep Neural Network
GP	Guassian Processes
QDA	Quadratic Discriminant Analysis

CHAPTER 1

INTRODUCTION

Heart disease persists as a significant concern within the realm of global public health, accounting for a large amount of the worldwide illness burden. Heart-related disorders must be managed and their effects minimized, and early detection and management are essential. Although conventional techniques for predicting cardiac disease have been beneficial, there is an increasing demand for more precise and advanced procedures.

Within this framework, deep learning has become a revolutionary technique that might completely change the area of heart disease prediction. Deep learning offers unparallel skills in complicated pattern identification by using deep neural networks to learn independently from large datasets. This capacity to identify complex links among enormous amounts of medical data is similar to that of an intelligent system that can anticipate outcomes accurately and quickly.

The application of deep learning to the prediction of heart disease has enormous potential to improve medical care. It can offer non-invasive diagnostic advancements, tailored treatment regimens, and early detection. Moreover, it provides the possibility of in-the-moment monitoring and notifications, allowing for additional preventative medical measures.

The use of deep learning for heart disease prediction is examined in this work. It looks at how this technology can improve diagnosis efficacy, timeliness, and accuracy, which will ultimately lead to better patient outcomes and heart-related condition management. It is crucial to acknowledge that utilizing deep learning in the healthcare sector carries ethical and legal implications. We can effectively use deep learning to combat heart disease and the associated health hazards by overcoming the challenges.

1.1 Background History

Ancient civilizations have long recognized the existence of heart disease. These ancient times are when the first reports of heart-related illnesses and symptoms were made. But major advances in our understanding of heart disease and in prediction techniques did not start to emerge until the 20th century.

The measuring of electrical activity in the heart was made possible by the revolutionary technique known as electrocardiography (ECG), which first appeared in the early 20th century. By making it possible for medical experts to identify anomalies and abnormalities in cardiac activity, this discovery completely changed the diagnostic process for heart problems. .

Heart disease prediction techniques and diagnostic technologies continued to advance during the course of the century. Echocardiography and angiography are two examples of cardiac imaging methods that have become indispensable for illustrating the anatomy and physiology of the heart. These developments led to a considerable improvement in the diagnosis and prognosis of cardiac disease.

Evaluating a person's chance of acquiring heart disease also required the creation of risk assessment models and scoring systems, such the Framingham Risk Score. These models estimated the risk of heart-related events by taking into account variables including age, blood pressure, cholesterol levels, and lifestyle choices.

In the latter part of the 20th century, non-invasive imaging technologies including computed tomography (CT) and magnetic resonance imaging (MRI) were developed, allowing for a better understanding of the anatomy and function of the heart.

1.2 Problem Statement

The necessity for sophisticated prediction models to enable early identification and intervention has been highlighted by the rising prevalence of cardiovascular diseases (CVDs) around the world. The existing techniques, although important, frequently fail to fully assess the complex linkages and patterns found within the many datasets related to heart disease. With its capacity to automatically extract intricate information, deep learning presents a viable remedy. Deep learning

model optimization for the complex features of cardiovascular risk prediction is still difficult, nevertheless. The goal of this work is to improve and optimize deep learning models' performance in heart disease prediction, which is an urgent issue. The research attempts to contribute to the creation of a highly successful and therapeutically useful approach by addressing challenges associated with model interpretability, generalizability across varied patient groups, and optimization of prediction accuracy.

Research focuses on reducing the current barriers that prevent a smooth integration of deep learning into clinical practice in an effort to enhance cardiac disease prediction. Because cardiovascular risk factors are complex, a model that not only makes accurate predictions but also sheds light on the underlying variables impacting those projections is necessary. One of the most important challenges is juggling the interpretability need with the intricacy of deep learning. In order to guarantee the model's efficacy in actual healthcare situations, the study also attempts to guarantee the model's generalizability across a range of clinical and demographic characteristics. By addressing these issues, this study seeks to advance the paradigm of cardiovascular disease prediction and give medical practitioners a more dependable and understandable tool for improving patient care and preventive cardiology

1.3 Applications of Heart Disease Prediction

Predicting heart disease is a critical application in medicine that has wide-ranging effects on patients, doctors, and healthcare systems. Here are some crucial applications:

1. **Early Diagnosis**: Those who are most at risk of acquiring heart disease early on can be identified using predictive models of the condition. Early detection may prevent the disease from progressing and allow for timely intervention and therapy.
2. **Risk Assessment**: Heart disease prediction tools are used to estimate an individual's risk of developing heart disease. Healthcare professionals can tailor treatment plans and preventive measures by considering a range of risk factors, such as gender, age, family history, lifestyle decisions, and past medical history.
3. **Research**: Research are aided by heart disease prediction models. These algorithms are able to detect patterns of illness development, therapeutic targets, and possible drug candidates by examining large datasets.

4. **Healthcare Resource Allocation**: Predictive models are a useful tool for healthcare systems to better allocate resources. Healthcare professionals may determine which people are at high risk and then direct resources, including equipment and cardiac specialists, to those areas.
5. **Patient Education**: Predictive models are useful resources for patient instruction. They may encourage people to modify their lifestyles in ways that lower their chance of developing heart disease by educating them about their risk factors.
6. **Clinical Decision Support**: Predictive models can be used as clinical decision support tools by medical practitioners. These models offer extra data and perspectives to support patients with cardiac disorders in their diagnosis and therapy planning.
7. **Epidemiological Studies**: Epidemiological studies employ heart disease prediction models to examine population-level trends and risk variables. Preventive strategies and public health initiatives depend heavily on this information.

Heart disease prediction is applied through a multifaceted approach that includes personalized medicine, non-invasive diagnostics, research, allocation of resources, patient education, clinical decision support, early diagnosis, risk assessment, and epidemiological studies. It is essential to raising our understanding of cardiac disorders, cutting healthcare costs, and providing better patient treatment.

1.4 Scope of the Project

The "Heart Disease Prediction" project includes a thorough examination of prediction model creation and use for the early diagnosis and management of cardiac conditions. The primary objective of the research is to create machine learning models that can accurately identify whether a patient has heart disease or not using clinical and demographic data. The goal of this prediction system is to assist healthcare professionals, researchers, and patients in many aspects of heart disease diagnosis and therapy.

Thorough data preparation is part of the project scope and is necessary to ensure the prediction models' accuracy and reliability. This covers class imbalance resolution and handling missing values. The study also involves comparing several machine learning algorithms to determine which approach is optimal. The "Heart Disease Prediction" initiative goes so far as to share findings and have conversations on how well the generated models work and how accurate they are. The project also takes the results' effects on patient treatment, public health campaigns, and healthcare practices into

account. To sum up, the project's scope includes developing predictive tools, putting them to use in real-world scenarios, and seeing how they could affect the area of managing and predicting cardiac disease.

1.5 Existing System

The current system is divided into two phases: classification models and data collection and preparation. The first stage uses datasets on cardiac illness that are obtained from a reliable machine learning library. One of the preprocessing processes is to use the appropriate attribute's mean to fill in the missing values. Features having a missing value rate greater than 50% are removed. The datasets are also subjected to standardization and normalization in order to guarantee a consistent format. The binning approach finds and replaces outliers by dividing attribute data into bins and using the mean in place of bin values. Then duplicate values are removed from the dataset, even in situations where the records are duplicate, ensuring that each patient appears only once.

Using the classification models on the pre-processed and acquired datasets is the second phase. A number of single classifiers, including a proposed Dense Neural Network, Nearest Neighbors, Gaussian Process, Linear SVM, Decision Tree, Naive Bayes, QDA, AdaBoost, Bagging, and Boosting, are employed for prediction. The models' performance is assessed using 10-Fold Cross Validation. This involves randomly dividing the datasets into 10 equal halves, nine of which are used for training and one for testing. The process is then repeated 10 times, reserving a fresh tenth part for testing each time. Scikit-Learn is a Python package that is used to implement the classification models.

Primary performance criteria including sensitivity, accuracy, and specificity are used to assess each classifier individually. By using a strict methodology, the suggested models' capacity to forecast heart disease is to be guaranteed to be robust and reliable. Cross-validation, metric-based assessment, and preprocessing methods together offer a thorough framework for evaluating the performance of the classification algorithms in the particular situation.

1.6 Disadvantage of the existing system:

Even while the previous suggested approach could have established the foundation for heart health evaluation, it might have had several drawbacks that the updated system aims to resolve:

1. **Limited Data Sources:**

The study of the breadth and depth of information accessible might be limited by the previous system's reliance on traditional databases. This restriction may lead to the exclusion of important information and subtleties found in more varied data sources, including scanned cardiac pictures of patients.

2. **Manual Model Tuning:**

The model selection procedure may have involved manual or less methodical tuning if more sophisticated methods like Grid Search, Random Search, and hyperparameter optimization were not used. In addition to being less effective and time-consuming, this method could not produce the best results possible from automated optimization procedures.

3. **Lack of Interpretability:**

Decision-making under the previous system may not have been transparent. Clinicians and end users can have trouble comprehending the logic behind the model's predictions if complex interpretability methods aren't included. In the context of clinical practice, this may impede patient trust and acceptance of the system.

4. **Limited Model Comparison:**

There may not have been a comprehensive comparison between deep learning techniques and conventional ensembles in the previous system as there was no Comparative Analysis of Ensemble Models. This lack of comparability may result in a model selection that is less informed and may miss more efficient strategies.

5. **Reduced Adaptability:**

In the absence of sophisticated model selection methods and appropriate hyperparameter tuning, the outdated system might not be able to adapt to different datasets and clinical settings and operate at its best. In some situations, this decreased flexibility could lead to less than ideal performance.

6. **Potential Lack of Robustness:**

Its resilience may be impacted by the old system's lack of some sophisticated preparation procedures and outlier identification techniques. The performance of the system as a whole may be impacted by

weak preprocessing, as strong preprocessing is essential to guaranteeing the dependability of machine learning models.

In conclusion, there are a number of potential drawbacks with the previous suggested system, including its dependence on a small number of data sources, manual model tweaking, interpretability issues, restricted model comparison, decreased flexibility, and possible lack of resilience. These shortcomings point to areas where improvements in the newly suggested approach are intended to address these issues and offer a better means of assessing heart health.

1.7 Proposed System

By adding novel approaches, the improved suggested system marks a substantial development in the evaluation of heart health. This sophisticated technology differs from others in that it uses scanned pictures of patients' hearts to get data. Three essential elements define the all-inclusive pipeline:

1. **Automated Optimal Model Selection:**

Using advanced methods including grid search, random search, cross-validation, and hyperparameter optimization, the system includes an Automated Optimal Model Selection process. By using a methodical methodology, a wide variety of neural network designs are optimized, leading to the best possible model performance. The correctness and effectiveness of the suggested system are improved by this methodological rigor.

2. **Interpretable Deep Learning for Heart Disease Diagnosis:**

Using both local interpretable model-agnostic explanations (LIME) and attention processes, our system presents an Interpretable Deep Learning method for Heart Disease Diagnosis. In order to improve model interpretability, this transparency-focused approach combines attention-focused convolutional and recurrent neural networks with XGBoost coupled with LIME. In addition to guaranteeing precise forecasts, this also offers insights into the variables affecting the model's judgment.

3. **Comparative Analysis of Ensemble Models:**

A Comparative Analysis of Ensemble Models is a component of the suggested system that compares deep learning techniques to conventional ensembles in a formal manner. Key performance indicators (KPIs) like accuracy, sensitivity, specificity, and others are evaluated using a variety of datasets. In order to choose the best model for heart disease prediction, this thorough review attempts to clarify the innate advantages and disadvantages of each strategy.

In conclusion, by utilizing scanned cardiac pictures, the sophisticated system surpasses traditional data sources. By combining interpretable deep learning, automated model selection, and a comparative study of ensemble models, the technique is improved and leads to improved performance comprehension, transparency, and accuracy.

1.8 Advantages of the Proposed System:

Numerous issues with the Existing system are addressed by the proposed system, which seeks to address and correct them:

1. **Advanced Data Source Integration:**

Scannable pictures of patients' hearts are included in the new system to overcome the constraint of limited data sources. The potential drawbacks of depending just on conventional datasets are addressed by this wider data source, which enables a more thorough assessment of heart health.

2. **Automated Model Selection and Optimization:**

To tackle the problem of manual model tuning in the previous system, an Automated Optimal Model Selection pipeline incorporating Grid Search, Random Search, cross-validation, and hyperparameter optimization has been implemented. Improved accuracy and efficiency result from the regular fine-tuning of neural network designs made possible by this automated method.

3. **Interpretability and Transparency in Predictions:**

The new system combines LIME with attention processes to present an Interpretable Deep Learning method. This improves openness and makes it easier to comprehend the variables affecting forecasts. By addressing the old system's lack of interpretability, this helps to increase confidence in the decision-making process.

4. **Rigorous Comparative Analysis for Informed Decision-Making:**

The new system's Comparative Analysis of Ensemble Models allows for a comprehensive assessment of deep learning techniques in comparison to conventional ensembles. This solves the drawback of the previous method by offering insightful information about the advantages and disadvantages of each strategy, enabling better-informed decision-making.

5. **Versatility and Adaptability:**

The new system's flexibility is increased by its capacity to adjust to various clinical circumstances and datasets. Hyperparameters that have been optimized add to the models' flexibility and effectiveness, guaranteeing top performance in a variety of scenarios. This resolves the possibility that the previous system was less flexible.

6. **Robust Data Preprocessing:**

The new system's approach implies the inclusion of sophisticated preprocessing stages and outlier identification techniques, even though it isn't stated directly. In order to ensure the dependability of machine learning models and correct any potential shortcomings in this area noted in the previous system, robust preprocessing is crucial.

To put it briefly, the new suggested system aims to provide a more sophisticated and efficient solution for heart health evaluation by deliberately addressing and correcting many of the drawbacks connected with the previous proposed system.

CHAPTER 2

LITERATURE SURVEY

1. ABDULWAHAB ALI ALMAZROI, A Clinical Decision Support System for Heart Disease Prediction Using Deep Learning
(2023)

The study underscores the critical need of early detection of heart disease, which is a leading cause of mortality globally. It leverages machine learning to overcome the challenge of navigating enormous volumes of healthcare data by using deep learning as a transformative tool to transform medical data into accessible information. The study uses a Keras-based deep learning model to build a dense neural network with three to nine hidden layer configurations. In each layer, the ReLU activation function is combined with one hundred neurons. On a range of heart disease datasets, a thorough evaluation that takes into account measures like sensitivity, specificity, accuracy, and f-measure consistently demonstrates that the proposed deep learning model outperforms single models and alternative ensembles.

2. D. CENITTA, ISCHEMIC HEART DISEASE PREDICTION USING OPTIMIZATION SQUIRREL SEARCH FECTURE SELECTION ALGORITHM(2022)

The "Ischemic Heart Disease Multiple Imputation Technique" addresses the problem of missing attributes in datasets related to Ischemic Heart Disease. It computes two values (J1 and J2) based on non-missing characteristics from comparable samples in order to estimate missing values using a closest neighbor technique. If J2 is negative, imputation is done using J1/J2; if not, the mean of the nearest neighbor's non-missing values is used. The imputation of missing data, closest neighbor approaches, multiple imputation methods, data analysis for Ischemic Heart Disease, performance metrics, applications in the medical field, and comparisons with current imputation methods should all be included in a literature review. By offering a methodical way to manage missing data in the particular context of datasets related to Ischemic Heart Disease, this technique helps with data preparation.

3. NORMA LATIF FITRIYANI, AN EFFICTIVE HEART DISEASE PREDICTION MODEL FOR A CLINICAL DECISION SUPPORT SYSTEM (2020)

In the realm of machine learning-based cardiovascular health prediction, the study's Heart Disease Prediction Model (HDPM) represents a revolutionary advancement. In terms of accuracy, precision, and sensitivity, the model outperforms six existing algorithms, demonstrating its usefulness in early diagnosis and therapy support. The HDPM has a notable accuracy rate of 98.40%, which indicates that clinical settings may find use for it. The results are contextualized in the broader literature through a detailed examination of existing machine learning algorithms, validation methods, and heart disease prediction models. The integration of statistical tests and the development of a web-based Clinical Decision Support System serve as additional evidence of the HDPM's relevance and dependability and establish the tool's viability for the proactive management of heart disease.

4. YUANYUAN PAN, ENHANCED DEEP LEARNING ASSISTED CONVOLUTIONAL NEURAL NETWORK FOR HEART DISEASE PREDICTION ON THE INTERNET OF MEDICAL THINGS PLATFORM(2020)

The article offers an innovative Enhanced Deep Learning Assisted Convolutional Neural Network (EDCNN) for heart disease prediction within the framework of the Internet of Medical Things (IoMT) architecture. The monitoring of cardiac parameters using the Internet of Medical Things (IoMT) platform is a new development in the field of healthcare. The literature review covers the existing applications of CNNs and deep learning in medical diagnostics, providing background for our investigation. Methodology and outcomes of research on IoMT platforms are subject to critical examination, particularly with regard to heart-related measures. We examine patterns of rising accuracy and efficiency when using deep learning techniques to feature extraction in heart disease prediction models. This dynamic environment is further enhanced by the proposed EDCNN model in the study, which tackles the nuances of cardiac illness prediction on the IoMT platform.

5. SYED ARSLAN ALI, AN OPTIMALLY CONFIGURED AND IMPROVED DEEP BELIEF NETWORK APPROACH FOR HEART DISEASE PREDICTION BASED ON RUZZO-TOMPA AND STACKED GENETICS ALGORITHM(2020)

The paper provides a unique OCI-DBN technique that utilizes underfitting, overfitting, and network optimization to address problems with heart disease prediction. With this approach, a Stacked Genetic

Algorithm (SGA)-configured Deep Belief Network (DBN) is combined with a feature selection process utilizing the Ruzzo-Tompa algorithm. Computing efficiency is increased using the Ruzzo-Tompa approach, which identifies significant features in the heart disease dataset. Based on a Restricted Boltzmann Machine (RBM), the DBN is trained layer by layer through the use of backpropagation. The DBN network configuration's layer count, node distribution, and hyperparameters are all modified by the SGA. This work is contextualized by the literature review, which looks at existing methods for predicting heart disease and highlights the novel use of feature selection, DBNs, and genetic algorithms for increased predictive accuracy.

PROJECT CHARTER

1. General Project Information	
Project Name:	**HEART DISEASE PREDICTION USING DEEP LEARNING**
Executive Sponsors:	-
Department Sponsor:	-
Impact of project:	Using Machine Learning to predict Heart Disease.

2. Project Team				
	Name	**Department**	**Telephone**	**E-mail**
Project Manager:	**Mr K Sathyaseelan**	IT	9629107959	sathyaseelan.it@srit.org
Team Members:	**Abdulsalam M**	IT	9360533804	abdulsalam.2006@srit.org
	Akilnanda E K	IT	9361760094	akil.2006@srit.org
	Nijeeth kumar T	IT	9677705259	nijeeth.2006@srit.org

3. Stakeholders

4. Project Scope Statement
Project Purpose / Business Justification
To predict Heart Disease in early stage. To reduce the medical treatments.
Objectives (in business terms)
Deliverables
The model that leads to predict the Heart Disease with high accuracy
Scope:
Scope of this project is to predict Heart disease as earlier and will provide a robust method of predicting HD.

Fig 2.1 Project Charter

Project Milestones

Phases	Start date (dd-mm-yyyy)	End date (dd-mm-yyyy)
Planning analysis		
Gathering the requirements		
Developing the model		
Implementation		
Testing and Documentation		

Major Known Risks (including significant Assumptions)

Risk	Risk Rating (High, Medium, Low)
We have a difficulties in learning new concept like Machine learning so this maybe lead us to schedule overruns	Medium

Constraints

Time	3 months
Budget	
Quality	The project be of high quality with assured data accuracy.
Scope	The scope shall be defined previously in this document section titled ‘scope’ and no more

5. Communication Strategy

Conduct Meetings
Email Communication
Gmeet Communication

6. Sign-off

	Name	Signature	Date (DD/MM/YYYY)
Executive Sponsor			
Department Sponsor			
Project Manager			

CHAPTER 3

REQUIREMENT SPECIFICATIONS

3.1 Software Requirements

Operating System	:	Windows 10 & above
Simulator Tool	:	Anaconda(Jupiter Notebook), Pycharm
Programming Package	:	Python

3.2 Hardware Requirements

Processor	:	Any Intel or AMD x84-64 processor
RAM	:	Minimum 4 GB and Recommended 8 GB
Hard Disk	:	24 GB to accommodate the project files, datasets, and software tools
Input Device	:	Standard Keyboard and Mouse
Output Device	:	Standard Monitor

3.3 System Tools

For Windows, Mac OS X, and Linux computers, Jupiter Notebook is a quick and effective source code editor. In addition to having built-in support for JavaScript, TypeScript, and Node.js, it also has a robust ecosystem of extensions for other languages and runtimes (such C++, C#, Java, Python, PHP, Go, and.NET). Microsoft developed the source code editor Visual Studio Code, or VS Code, for Windows, Linux, and macOS using the Electron Framework. Among the features are syntax highlighting, embedded Git, snippets, intelligent code completion, and debugging help.

CHAPTER 4

METHODOLOGY

4.1 Working of the Proposed System

Operating at the forefront of cardiac health evaluation, the suggested system introduces a revolutionary technique that uses scanned pictures of patients' hearts as the main source of data. This shift away from standard datasets is a major achievement since it gives the system's analytical processes a more thorough and complete input. The operation of the system is facilitated by three essential elements, each of which enhances the overall effectiveness of the system.

The Automated Optimal Model Selection procedure is the first important component. This stage involves the system going through a rigorous data preparation procedure, which includes scaling, normalization, and feature extraction from the scanned cardiac pictures. The use of sophisticated methods including grid search, random search, cross-validation, and hyperparameter optimization is what distinguishes this procedure. This rigorous methodology ensures the selection of optimal models by methodically fine-tuning a wide variety of neural network designs. This stage improves the system's accuracy and efficiency to a great extent, enabling it to surpass the constraints of traditional human tuning methods.

Transparency and interpretability are highlighted in the second component, Interpretable Deep Learning for Heart Disease Diagnosis. This stage takes the raw datasets and uses recurrent neural networks to extract relevant characteristics. Interpretability is further improved by the combination of Local Interpretable Model-agnostic Explanations (LIME) with the potent gradient boosting technique XGBoost. Important areas of the pictures are highlighted by the attention processes, and LIME offers information on the factors affecting the model's judgment. Because of this combination, forecasts are visible, which builds end-user and clinician trust.

An Ensemble Model Comparative Analysis is the third essential component. Across a variety of datasets, this component evaluates key performance metrics (KPIs) including accuracy, sensitivity, and specificity by methodically contrasting deep learning approaches with classical ensembles. The meticulous examination attempts to clarify the innate advantages and disadvantages of every tactic, enabling a well-informed choice-making procedure. The results provide more flexibility and adaptability to the system by guiding the choice of the best model for heart disease prediction.

To summarize, the novel usage of scanned cardiac pictures forms the basis of the proposed system's operation, which is bolstered by an interpretable deep learning method, an automated optimal model selection procedure, and a comparative analysis of ensemble models. This thorough and complex approach improves understanding, transparency, and accuracy when assessing heart health, establishing the system as a reliable and cutting-edge instrument for the prediction of cardiac illness.

4.2 Logistic Regression Algorithm

The statistical method of linear regression is used to fit a linear equation to the observed data, which depicts the relationship between a dependent variable and one or more independent variables. Selecting the best-fitting line that minimizes the total squared disparities between the expected and actual values of the dependent variable is the goal. The line's equation is Y= mx + b, where x is the independent variable, Y is the dependent variable, m is the slope of the line, and b is the y-intercept.

A statistical method called linear regression is used to characterize the relationship between a dependent variable and one or more independent variables by fitting a linear equation to the observed data. Selecting the line that minimizes the sum of squared differences between the actual and anticipated values and best matches the dependent variable is the aim. The line's equation is expressed as Y = mx + b, where x is the independent variable, m is the slope of the line, and b is the y-intercept.

4.3 KNN

A flexible machine learning method used for both regression and classification applications is K Nearest Neighbors (KNN). When predicting new data points in KNN, the majority class is determined for classification purposes, or the values of the nearest neighbors in the feature space are averaged for regression. One of the most important factors affecting the model's performance is the parameter "k" in KNN, which stands for the number of neighbors taken into account. By modifying this parameter, one may fine-tune the ratio between prediction accuracy and model complexity.

The KNN algorithm is predicated on the idea of similarity inside the feature space. The technique finds the k data points that are closest to a new data point using a selected distance metric, such the Euclidean distance. Whereas the prediction for regression is the mean of the values of the k

nearest neighbors, the predicted class for classification is the majority class among these neighbors for each new point. A greater number for k might oversmooth predictions, while a lower value could provide a noisy model that is susceptible to outliers. Therefore, choosing the right value for k is crucial. KNN's ease of use and efficiency in capturing complex patterns are highly regarded, particularly in datasets with various zones of similarity or clusters.

4.4 Gaussian Process (GP)

A Gaussian Process (GP) is a non-parametric and flexible probabilistic model used in machine learning for regression and classification tasks. It represents a distribution over functions and is defined by a mean function and a covariance function (kernel). GPs provide uncertainty estimates along with predictions, making them particularly valuable in scenarios where understanding the uncertainty in predictions is crucial. The kernel function determines the similarity between data points, influencing the smoothness and complexity of the functions generated by the GP.

In its functioning, a Gaussian Process starts with a set of observed data points. The mean and covariance functions are employed to create a distribution over functions that could explain the observed data. When predicting the value for a new data point, the GP considers the relationships and similarities between the observed and new data points, providing not only a prediction but also a confidence interval reflecting the uncertainty associated with the prediction. The flexibility of GPs makes them well-suited for applications where understanding the underlying structure of the data is essential, and they are widely used in fields such as optimization, robotics, and Bayesian optimization.

4.5 SVM

Regression and classification problems are two areas in which Support Vector Machine (SVM) excels as a supervised learning technique. To maximize the margin between two classes and successfully separate them, support vector machines (SVM) look for the best hyperplane in the feature space. Support vectors are important data points that are closest to the decision boundary. They shape the hyperplane's orientation such that it is as far away from these support vectors as feasible. Because kernel functions allow for the change of the input space and improve the capacity to extract complex relationships from the data, SVM's adaptability also extends to solving non-linear situations.

The Support Vector Machine (SVM) algorithm works by repeatedly adjusting the decision boundary to find the best hyperplane, which maximizes the margin between different classes. SVM uses the margin, which represents the separation between the hyperplane and the nearest data point for each class, as its central point. The goal is to identify the hyperplane that minimizes classification mistakes and maximizes this margin. Where it is not possible to separate data linearly, SVM uses kernel methods to convert the input space into a higher-dimensional feature space. It is in this new area that a hyperplane may successfully divide the classes. SVM is highly regarded for its ability to manage high-dimensional data, robustness against outliers, and flexibility for handling both linear and non-linear problems.

4.6 Decission Tree

The decision tree is a popular supervised machine learning technique for regression and classification problems. It works by recursively splitting the dataset into subsets, each of which is defined by the characteristic that has the greatest influence at a particular node. The ultimate goal is to build a tree structure with leaves that represent the chosen course of action or forecast. Decision Trees are widely acknowledged for their comprehensibility and simple visualization, which makes them advantageous for understanding complex decision-making procedures.

A decision tree starts to function with the whole dataset at the root node. Using parameters like information gain or Gini impurity, the algorithm carefully chooses the characteristic at each internal node that best splits the data. Until a predefined ending requirement is met—typically involving elements like a minimum number of samples in a leaf or a maximum tree depth—this recursive process continues, producing branches and nodes. The resulting tree functions as a predictive model, making predictions by following the branches according to the input features until it reaches a leaf node, which provides the classification or output. Pruning is a common approach used to improve generalization and reduce sensitivity of Decision Trees to small fluctuations in input.

4.7 Naive Bayes

One popular probabilistic machine learning method for categorization issues is Naive Bayes. It is predicated on the Bayes theorem, which calculates a hypothesis's probability in light of the available data. The assumption of conditional independence among characteristics is what gives Naive

Bayes its "naive" quality and makes the computational procedures involved simpler. Naive Bayes is computationally efficient and frequently performs well in a variety of situations, despite its simplicity.

In its operational mode, Naive Bayes estimates the probability of a class given the input characteristics by combining the class's prior probability with the chance of witnessing those features in the class. The predicted class is then determined by the algorithm to be the one with the highest probability. Naive Bayes is a popular algorithm that works well in text categorization and spam filtering applications. It is simple to use and relies on the concept of feature independence. Its strong performance is especially noticeable when the dataset is large and the characteristics show a fair amount of independence.

4.8 QDA

By providing the flexibility of unique covariance matrices for every class, Quadratic Discriminant Analysis (QDA) functions as a classification technique that expands upon Linear Discriminant Analysis (LDA). QDA estimates a distinct covariance matrix for every class, going one step farther than LDA, which assumes homogenous covariance matrices across all classes. Because of its flexibility, QDA is especially well-suited for datasets where the data distribution underlies the assumption of equal covariance matrices.

Quaternionic Discriminant Analysis (QDA) computes mean vectors and covariance matrices unique to each class using the training data that is supplied. In the prediction stage, QDA uses the class-specific mean and covariance parameters to determine how likely it is that an observation belongs in each class. Next, the class with the highest probability is identified as the anticipated class. When dealing with datasets that show significant variations in the covariance structures of different classes and non-linear decision boundaries, QDA is very helpful. It's important to remember that, even with its versatility, QDA may be computationally taxing, especially when working with large numbers of characteristics. In conclusion, because QDA allows for differences in covariance structures between classes in the dataset, it provides a flexible method to classification.

4.9 AdaBoost

AdaBoost, which stands for Adaptive Boosting, is an ensemble learning method used for both regression and classification. In order to create a strong and accurate model, it combines the outputs from weak learners, which are usually straightforward decision trees or stumps. For every data point, AdaBoost includes a weighting mechanism that repeatedly modifies to emphasize previously misclassified cases. This strategy makes sure that the algorithm prioritizes the correction of misclassifications in later rounds, allowing later weak learners to train these cases more heavily.

Adjusting weights for misclassified points to increase their impact is a crucial aspect of AdaBoost's iterative training of weak learners on the dataset. More accurate models contribute more heavily to the final prediction, and the final model is the weighted sum of these weak learners. When applied to weak learners, AdaBoost outperforms random guessing by a small margin. Its resilience is attributed to its flexibility and focus on correcting misclassifications, which reduces the possibility of overfitting. AdaBoost's susceptibility to anomalies and noisy data, however, emphasizes how crucial it is to carefully evaluate data quality in order to achieve peak performance.

4.10 Bagging Classifier

Bagging, which stands for Bootstrap Aggregating, is an ensemble learning method that aims to improve decision trees in particular while also strengthening machine learning models' stability and accuracy. By using bootstrap sampling, which includes drawing samples with replacement, the procedure creates many subsets from the original dataset. Next, every one of these subsets is used to train a base model. Typically, techniques like voting or averaging are used to integrate the predictions of several models to arrive at the final forecast. In addition to reducing overfitting, this strategy enhances the model's overall performance.

Bagging works by creating different training sets for every base model, which helps to minimize overfitting and decrease variation. A more reliable and comprehensive ensemble model is produced as a result of the slightly varied patterns that each model, which was trained on a part of the data, captures. The Random Forest method, which uses bagging to train numerous decision trees on different bootstrapped samples and aggregate their predictions, is a prime example of its application.

Bagging improves model performance and increases ensemble stability and dependability; its usefulness is especially evident when working with complicated datasets. Bagging also works well at reducing the effects of noise and outliers in the training set.

4.11 Boosting

Boosting is a unique kind of ensemble learning where weak learners perform better as a result of which a more reliable and accurate prediction model is created. Unlike bagging, which emphasizes misclassified cases more than other data points, boosting includes giving varying weights to different data items. Iteratively, the process proceeds, with each weak learner getting instruction to correct the mistakes made by its predecessor. Boosting algorithms that are well-known, such AdaBoost and Gradient Boosting, combine the results of these poor learners and give more weight to the predictions of models that show promise in handling previously misclassified data points. An effective ensemble model is a result of this adaptive and iterative process.

Boosting works by first training a weak learner on the whole dataset and then giving weights to individual data points based on how well they are classified. Higher weights are assigned to incorrectly identified points, causing later learners to give these occurrences greater weight in their instruction. This is an iterative process where each poor learner adjusts its attention according to the collective errors made up to that point. The final model is the result of adding all of the different models together, with the weights based on how accurate each model is. Boosting turns out to be a powerful method that improves both model accuracy and generalization. It is especially useful in cases when base models have relative weakness or an underfitting propensity.

4.12 Dense Neural Network

One type of artificial neural network known as a Dense Neural Network (DNN) is characterized by its highly linked layers, in which every neuron connects to every other neuron in the layers above and below. Often called a feedforward or fully connected neural network, a DNN is made up of an output layer, one or more hidden layers, and an input layer. This design uses weights to describe each connection between neurons; during training, the network dynamically modifies these weights. The DNN is able to recognize and extract complex patterns from the data because to its adaptive learning.

A Dense Neural Network (DNN) employs non-linear activation functions, applies weights to the connections, and cycles over its layers to analyze input data. Repaired linear units (ReLU) and sigmoid are two examples of activation functions that introduce important non-linearities that enable the network to understand and express complex relationships within the data. Backpropagation is the mechanism that organizes the network's learning. Prediction mistakes are sent down through the layers during backpropagation, which forces optimization procedures to be used to modify the weights. DNNs are notably efficient at a variety of tasks, including sophisticated pattern identification, natural language processing, and picture recognition. Their innate capacity to automatically extract hierarchical characteristics from data accounts for their effectiveness.

CHAPTER 5

RESULT AND DISCUSSION

5.1 Output of the proposed system:

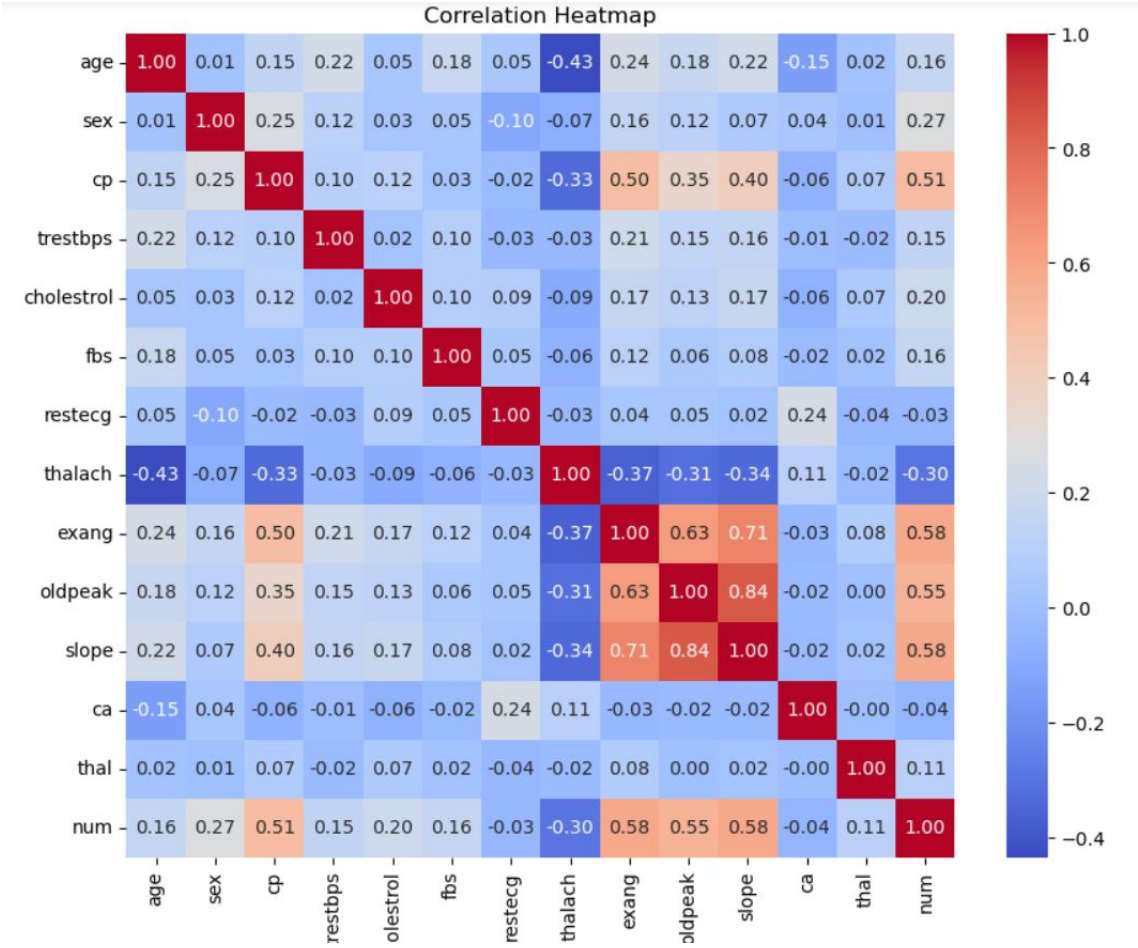


Fig 5.1.1 Heatmap

The depicted image showcases these coefficients to visualize the strength of correlation among variables.

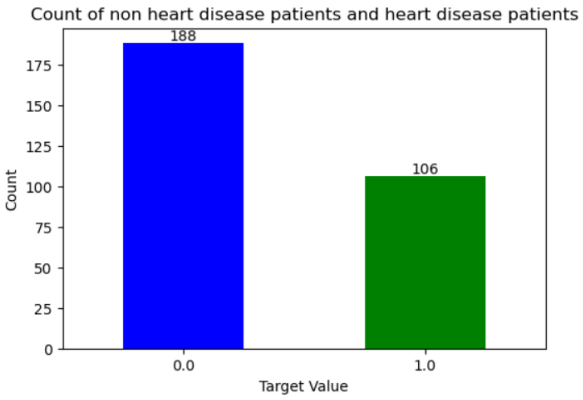


Fig 5.1.2 Distribution of status values

The above figure represents the number of Heart disease and Non-Heart disease patients .

Fig 5.1.3 Distribution of status values

The above figure represents the number of Male and Female patients

Architectural Design

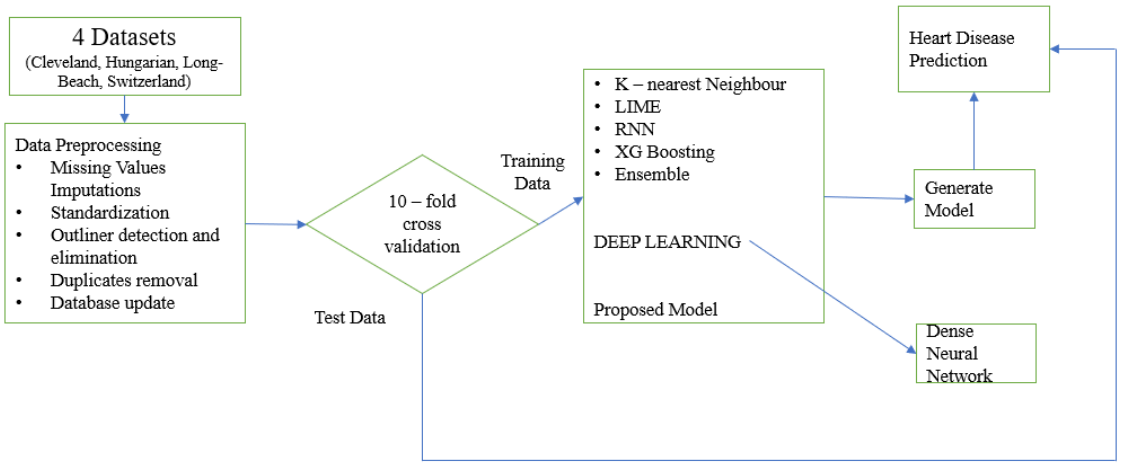


Fig 5.2

Comparision Of Algorithms

	Before Preprocessing	After Preprocessing
Logistic Regression	88.25%	85.25%
KNeighborsClassifier	65.57%	86.89%
Gaussian Process	85.25%	85.25%
Support Vector Machine	72.00%	90.16%
Decission Tree	78.69%	80.33%
Naïve Bayes	83.61%	83.61%
Quadratic Discriminant Analysis	86.89%	86.89%
Ada Boost	81.97%	81.97%
Bagging Classifier	86.89%	86.89%
Boosting	90.16%	90.16%
Deep Neural Networks	81.97%	83.61%

Fig 5.3.1

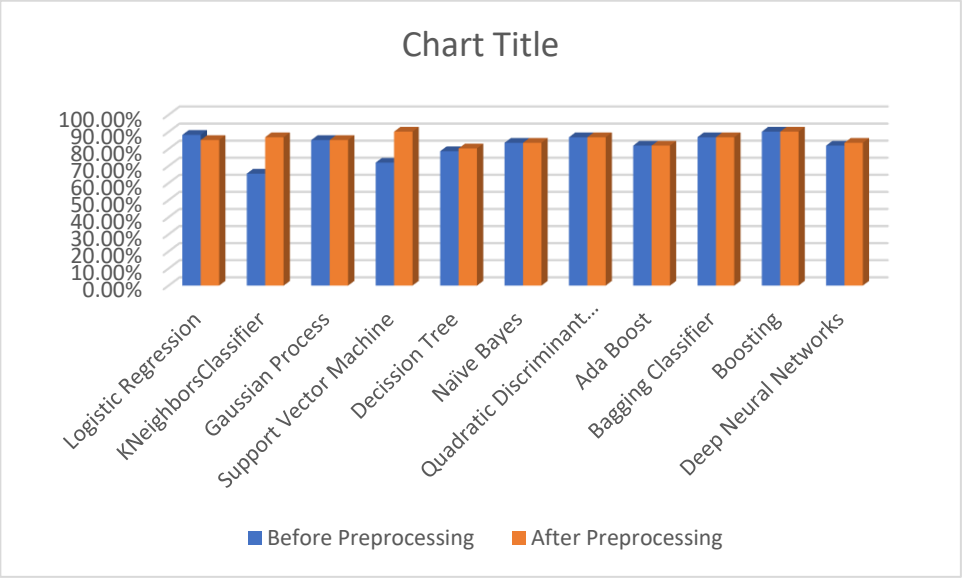


Fig 5.3.2

Cleaveland statistical Summary

	age	sex	cp	Trestbps	cholestrol	fbs	Restecg	thalach	exang	oldpeak	slope	ca	thal	num
Count	303	303	303	303	303	303	303	303	303	303	303	303	303	303
Mean	54.4	0.67	3.15	131.68	246.69	0.14	0.99	149.6	0.32	1.03	1.6	0.69	4.72	0.45
Std	9.03	0.46	0.96	17.59	51.77	0.35	0.99	22.87	0.46	1.16	0.61	0.96	1.93	0.49
Min	29	0	1	94	126	0	0	71	0	0	1	0	3	0
25 %	48	0	3	120	211	0	0	135	0	0	1	0	3	0
50 %	56	1	3	130	241	0	1	153	0	0.80	2	0	3	0
75 %	61	1	4	140	275	0	2	166	1	1.6	2	1	7	1
Max	77	1	4	200	564	1	2	202	1	6.2	3	3	7	1

Fig 5.4.1

Hungarian Statistical Summary

	age	sex	cp	Trestbps	cholestrol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
Count	294	294	294	294	294	294	294	294	294	294	294	294	294	294
Mean	47.82	0.72	2.98	132.13	231.22	0.06	0.21	138.65	0.30	0.58	0.67	0.001	0.53	0.36
Std	7.81	0.44	0.96	19.22	93.65	0.25	0.46	24.90	0.46	0.90	0.92	0.02	1.72	0.48
Min	28.0	0	1	0	0	0	0	0	0	0	0	0	0	0
25 %	42	0	2	120	198	0	0	122	0	0	0	0	0	0
50 %	49	1	3	130	237	0	0	140	0	0	0	0	0	0
75 %	54	1	4	140	277	0	0	155	1	.1	2	0	0	1
Max	66	1	4	200	603	1	2.1	190	1	5	3	0.4	7	1

Fig 5.4.2

Comparision of Algorithms (Cleaveland)

Cleveland	Accuracy	Specificity	Sensitivity	F-measure	Execution time
XG Boost	80.33%	82.76%	78.12%	80.38%	28.7102
Ada Boost	86.89%	86.21%	87.50%	86.85%	27.3439
LGBM	88.52%	93.10%	84.38%	88.52%	0.0578
GBM	83.61%	100%	12.50%	22.22%	29.0419
Random Forest	80.33%	86.21%	75%	80.21%	28.5142
Logistic	88.52%	93.10%	84.38%	88.52%	29.3103
K-nearest	83.61%	86.21%	81.25%	83.66%	28.1663
Naïve Bayes	78.69%	86.21%	71.88%	78.39%	27.6944
Decision tree	83.61%	86.21%	81.25%	83.66%	27.7524
SVM	83.61%	86.21%	81.25%	83.66%	30.2198

Fig 5.5.1

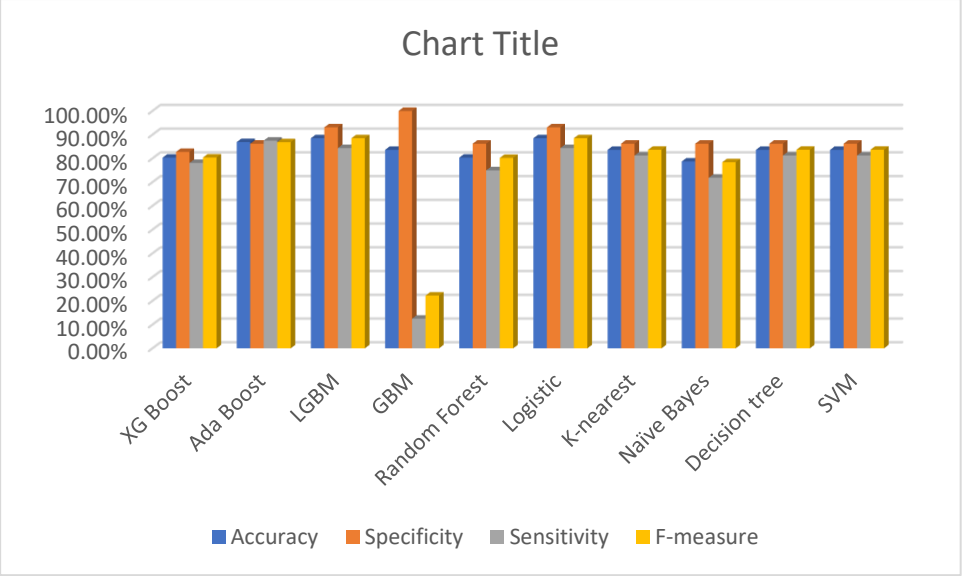


Fig 5.5.2

Comparision of Algorithms (Hungarian)

Hungarian	Accuracy	Specificity	Sensitivity	F-measure	Execution time
XG Boost	79.66%	76.32%	85.71%	80.74%	27.4996
Ada Boost	83.05%	84.21%	80.95%	82.55%	28.7462
LGBM	83,05%	84.95%	80.95%	82.55%	0.0439
GBM	74.58%	81.58%	71.43%	76.17%	27.1836
Random Forest	81.36%	78.95%	85.71%	82.19%	27.4080
Logistic	83.05%	84.21%	80.95%	82.55%	28.0320
K-nearest	83.05%	81.58%	85.71%	83.60%	28.4060
Naïve Bayes	81.36%	81.58%	80.95%	81.26%	28.6789
Decision tree	81.36%	84.21%	76.19%	80%	26.6422
SVM	77.97%	78.97%	76.19%	77.54%	26.6306

Fig 5.5.3

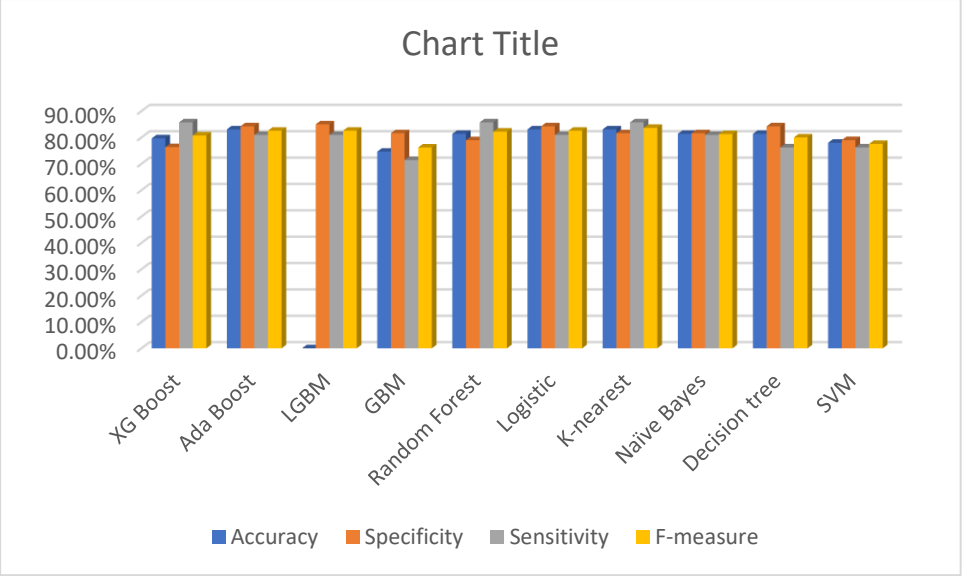


Fig 5.5.4

Fig 5.6.1

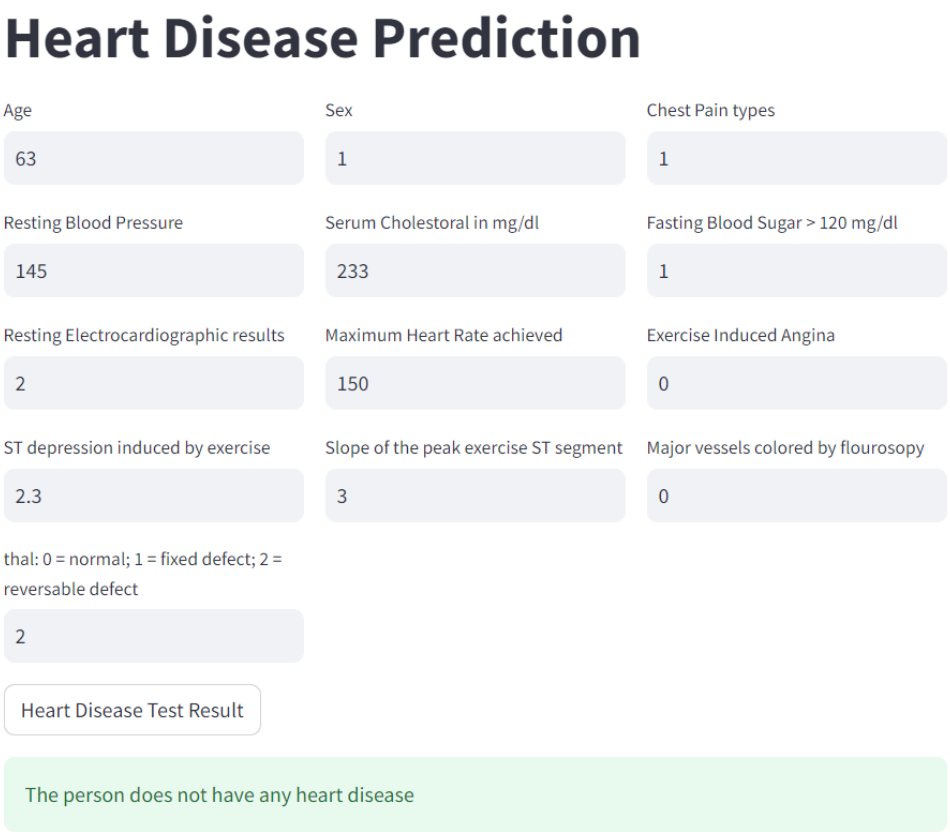


Fig 5.6.2

Appendix

Source Code :

Datavisualization

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('imputation\HungarianImputedDataset.csv')
if 'num' not in data.columns:
    print("Error: 'num' column not found in the dataset.")
else:
    value_counts = data['num'].value_counts()
    plt.figure(figsize=(6, 4))
    value_counts.plot(kind='bar', color=['blue', 'green'])
    plt.xlabel('Target Value')
    plt.ylabel('Count')
    plt.title('Count of non heart disease patients and heart disease patients')
    for i, count in enumerate(value_counts):
        plt.text(i, count, str(count), ha='center', va='bottom')
    plt.xticks(rotation=0)
    plt.show()
```

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('imputation\HungarianImputedDataset.csv')
if 'sex' not in data.columns:
    print("Error: 'sex' column not found in the dataset.")
else:
    value_counts = data['sex'].value_counts()
    plt.figure(figsize=(6, 4))
    value_counts.plot(kind='bar', color=['blue', 'green'])
    plt.xlabel('Target Value')
    plt.ylabel('Count')
    plt.title('Count of male and female.')
    for i, count in enumerate(value_counts):
        plt.text(i, count, str(count), ha='center', va='bottom')
    plt.xticks(rotation=0)
    plt.show()
```

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('imputation\HungarianImputedDataset.csv')
if 'sex' not in data.columns or 'num' not in data.columns:
    print("Error: 'sex' or 'num' columns not found in the dataset.")
else:
    gender_target_counts = data.groupby(['sex', 'num']).size().unstack(fill_value=0)
    total_males = gender_target_counts.loc[0].sum()
    total_females = gender_target_counts.loc[1].sum()
    male_heart_disease = gender_target_counts.loc[0, 1]
    male_no_heart_disease = gender_target_counts.loc[0, 0]
    female_heart_disease = gender_target_counts.loc[1, 1]
    female_no_heart_disease = gender_target_counts.loc[1, 0]
    labels = ['Male - Heart Disease', 'Male - No Heart Disease', 'Female - Heart Disease', 'Female - No Heart Disease']
    sizes = [male_heart_disease, male_no_heart_disease, female_heart_disease, female_no_heart_disease]
    colors = ['red', 'lightblue', 'green', 'lightcoral']
    explode = (0.1, 0, 0.1, 0)
    plt.figure(figsize=(8, 6))
    plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140, explode=explode)
    plt.title('Heart Disease Distribution by Gender')
    plt.axis('equal')
    plt.show()

     import pandas as pd
     import seaborn as sns
     import matplotlib.pyplot as plt
     data = pd.read_csv('imputation\HungarianImputedDataset.csv')
     plt.figure(figsize=(10, 8))
     sns.heatmap(data.corr(), annot=True, cmap='coolwarm', fmt='.2f')
     plt.title('Correlation Heatmap')
     plt.show()
```

 # Imputation

```
import pandas as pd
from sklearn.impute import KNNImputer
```

```
input_csv_file = 'processed\switzerland.csv'
df = pd.read_csv(input_csv_file, delimiter=',')
knn_imputer = KNNImputer(n_neighbors=2)
imputed_data = knn_imputer.fit_transform(df)
imputed_df = pd.DataFrame(imputed_data, columns=df.columns)
output_csv_file = 'SwitzerlandImputedDataset.csv'
imputed_df.to_csv(output_csv_file, index=False)
```

 # Standardization

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
input_csv_file = 'imputation\SwitzerlandImputedDataset.csv'
df = pd.read_csv(input_csv_file, delimiter=',')
features = df
scaler = StandardScaler()
scaler.fit(features)
standardized_data = scaler.transform(features)
standardized_df = pd.DataFrame(standardized_data, columns=features.columns)
output_csv_file = 'SwitzerlandStandardizeDataset.csv'
standardized_df.to_csv(output_csv_file, index=False)
print(f"Standardized dataset saved to {output_csv_file}")
```

 # Robust Scalar

```
import pandas as pd
from sklearn.preprocessing import RobustScaler
scaler = RobustScaler()
input_file = 'standardized\SwitzerlandStandardizeDataset.csv'
data = pd.read_csv(input_file, delimiter=',')
columns_to_scale = data.columns
scaler.fit(data[columns_to_scale])
scaled_data = scaler.transform(data[columns_to_scale])
scaled_df = pd.DataFrame(scaled_data, columns=columns_to_scale)
output_file = 'SwitzerlandScaledoutput.csv'
scaled_df.to_csv(output_file, index=False)
```

 # Logistic Regression

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
csv_file = 'imputation\ClevelandImputedDataset.csv'
data = pd.read_csv(csv_file, delimiter=',')
X = data.drop(columns=['num'])
y = data['num']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy * 100:.2f}%')
```

 # Knearest Neighbor

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import pandas as pd
import numpy as np
import warnings
csv_file = 'robust\ClevelandScaledoutput.csv'
data = pd.read_csv(csv_file, delimiter=',')
data['target'] = (data['num'] > 0).astype(int)
X = data.drop(columns=['num', 'target']) #changing to binary target value
y = data['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)
warnings.filterwarnings("ignore", category=FutureWarning)
y_pred = knn.predict(X_test)
```

```
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy* 100:.2f}%')
```

 # Gaussian Process

```
import pandas as pd
import numpy as np
from sklearn.gaussian_process import GaussianProcessClassifier
from sklearn.gaussian_process.kernels import RBF
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
data = pd.read_csv('robust\ClevelandScaledoutput.csv')
X = data.iloc[:, :-1].values
y = data.iloc[:, -1].values
numeric_features = list(range(X.shape[1]))
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
kernel = 1.0 * RBF()
classifier = GaussianProcessClassifier(kernel=kernel, random_state=0, n_jobs=-1)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Test Accuracy: {accuracy* 100:.2f}')
```

 # Support Vector Machine

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report
import pandas as pd
csv_file = 'robust\ClevelandScaledoutput.csv'
data = pd.read_csv(csv_file, delimiter=',')
data['target'] = (data['num'] > 0).astype(int)
X = data.drop(columns=['num', 'target']) #changing to binary target value
y = data['target']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
svm_classifier = SVC(kernel='linear')
svm_classifier.fit(X_train, y_train)
y_pred = svm_classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy * 100:.2f}')
```

 # Decision Tree

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
csv_file = 'imputation\ClevelandImputedDataset.csv'
data = pd.read_csv(csv_file, delimiter=',')
X = data.drop(columns=['num'])
y = data['num']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = DecisionTreeClassifier()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy * 100:.2f}%')
```

 # Naive Bayes

```
import pandas as pd
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
from sklearn.naive_bayes import GaussianNB
csv_file = 'robust\ClevelandScaledoutput.csv'
data = pd.read_csv(csv_file, delimiter=',')
data['target'] = (data['num'] > 0).astype(int)
X = data.drop(columns=['num', 'target'])
y = data['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
naive_bayes = GaussianNB()
```

```
naive_bayes.fit(X_train, y_train)
y_pred = naive_bayes.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy * 100:.2f}')
```

 # Quadratic Discriminat Analysis

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
data = pd.read_csv('robust\ClevelandScaledoutput.csv')
X = data.drop(columns=['num'])
y = data['num']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
qda_classifier = QuadraticDiscriminantAnalysis()
qda_classifier.fit(X_train, y_train)
y_pred = qda_classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100:.2f}")
```

 # Ada Boost

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
data = pd.read_csv('robust\ClevelandScaledoutput.csv')
X = data.drop(columns=['num'])
y = data['num']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
base_classifier = DecisionTreeClassifier(max_depth=1)  # You can customize the base classifier
adaboost_classifier = AdaBoostClassifier(base_classifier, n_estimators=50, random_state=42)
adaboost_classifier.fit(X_train, y_train)
y_pred = adaboost_classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
```

```
print(f"Accuracy: {accuracy * 100:.2f}")
```

 # Bagging Classifier

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
data = pd.read_csv('robust\ClevelandScaledoutput.csv')
X = data.drop(columns=['num'])
y = data['num']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
rf_classifier.fit(X_train, y_train)
y_pred = rf_classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100:.2f}")
```

 # Boosting

```
import lightgbm as lgb
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
data = pd.read_csv('robust\ClevelandScaledoutput.csv')
X = data.drop(columns=['num'])
y = data['num']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
train_data = lgb.Dataset(X_train, label=y_train)
params = {
    "objective": "binary",
    "metric": "binary_logloss",
    "boosting_type": "gbdt",
    "num_leaves": 31,
    "learning_rate": 0.05,
    "feature_fraction": 0.9,
    "bagging_fraction": 0.8,
```

```
    "bagging_freq": 5,
    "verbose": 0,
}
model = lgb.train(params, train_data, 100)
y_pred = model.predict(X_test, num_iteration=model.best_iteration)
y_pred_binary = (y_pred > 0.5).astype(int)
accuracy = accuracy_score(y_test, y_pred_binary)
print(f"Accuracy: {accuracy * 100:.2f}")
```

 # Dense Neural Networks

```
import pandas as pd
import numpy as np
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from tensorflow.keras import layers, models
data = pd.read_csv('imputation\ClevelandImputedDataset.csv')
X = data.iloc[:, :-1].values
y = data.iloc[:, -1].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
numeric_features = list(range(X.shape[1]))  # Assuming all features are numeric
preprocessor = ColumnTransformer(
    transformers=[('num', StandardScaler(), numeric_features)])
model = models.Sequential()
model.add(layers.Dense(32, activation='relu', input_shape=(X_train.shape[1],)))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))  # Assuming binary classification
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
preprocessor.fit(X_train)
X_train_transformed = preprocessor.transform(X_train)
X_test_transformed = preprocessor.transform(X_test)
model.fit(X_train_transformed, y_train, epochs=5, batch_size=64, validation_data=(X_test_transformed, y_test))
test_loss, test_acc = model.evaluate(X_test_transformed, y_test)
```

```
print(f'Test accuracy: {test_acc * 100:.2f}%')
```

Front End

```
import pickle
import streamlit as st
import warnings
warnings.filterwarnings("ignore", category=UserWarning, module="sklearn")
# Loading the saved model
heart_disease_model = pickle.load(open("C:/Users/SivaRanjan.s/Downloads/uci-heart-disease/uci-heart-disease/heart_disease_model.sav", 'rb'))
# Heart Disease Prediction Page
st.title('Heart Disease Prediction')
col1, col2, col3 = st.columns(3)
with col1:
    age = st.text_input('Age')
with col2:
    sex = st.text_input('Sex')
with col3:
    cp = st.text_input('Chest Pain types')
with col1:
    trestbps = st.text_input('Resting Blood Pressure')
with col2:
    chol = st.text_input('Serum Cholestoral in mg/dl')
with col3:
    fbs = st.text_input('Fasting Blood Sugar > 120 mg/dl')
with col1:
    restecg = st.text_input('Resting Electrocardiographic results')
with col2:
    thalach = st.text_input('Maximum Heart Rate achieved')
with col3:
    exang = st.text_input('Exercise Induced Angina')
with col1:
    oldpeak = st.text_input('ST depression induced by exercise')
with col2:
    slope = st.text_input('Slope of the peak exercise ST segment')

with col3:
    ca = st.text_input('Major vessels colored by flourosopy')
with col1:
```

```
    thal = st.text_input('thal: 0 = normal; 1 = fixed defect; 2 = reversable defect')
# Convert inputs to numeric values
try:
    # Check if the input is not empty before converting to float
    if age:
        age = float(age)
    if sex:
        sex = float(sex)
    if cp:
        cp = float(cp)
    if trestbps:
        trestbps = float(trestbps)
    if chol:
        chol = float(chol)
    if fbs:
        fbs = float(fbs)
    if restecg:
        restecg = float(restecg)
    if thalach:
        thalach = float(thalach)
    if exang:
        exang = float(exang)
    if oldpeak:
        oldpeak = float(oldpeak)
    if slope:
        slope = float(slope)
    if ca:
        ca = float(ca)
    if thal:
        thal = float(thal)
    # Code for Prediction
    heart_diagnosis = ''
    # Creating a button for Prediction with a unique key
    if st.button('Heart Disease Test Result', key='heart_diagnosis_button'):
        heart_prediction = heart_disease_model.predict([[age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal]]
        if heart_prediction[0] == 0:
            heart_diagnosis = 'The person is having heart disease'
        else:
            heart_diagnosis = 'The person does not have any heart disease'
        st.success(heart_diagnosis)
except ValueError as e:
    st.error(f"Error: {e}. Please enter valid numeric values for the input features."
```

CHAPTER 6

CONCLUSION

6.1 Summary

We have used machine learning and the 10-fold cross validation approach in this study to improve the prediction of heart disease. Our objective is to enhance medical intervention and patient outcomes by means of early diagnosis. Among the noteworthy findings and achievements are:

1. Scaling the data and removing null values during data preparation were handled well.
2. Distinguished results from a range of deep neural network-based machine learning models, including KNN, Decision Trees, SVM, and others.
3. The potential applications of our study in healthcare analytics might speed up early diagnosis, offer personalized treatment suggestions, and optimize resource use.
4. Contributions to medical research by assessing new heart disease medicines and identifying them in advance.

To sum up, our research highlights the critical significance that sophisticated preprocessing, ensemble learning, and machine learning play in the prediction of heart disease. Better patient care, advances in medical research, and ongoing improvements in healthcare analytics are all made possible by the methods and insights learned here, which offer a solid basis for researchers, data scientists, and healthcare professionals.

REFERENCES

[1]. N.-S. Tomov and S. Tomov, ‘‘On deep neural networks for detecting heart disease,’’ 2018, arXiv:1808.07168.

[2]. A. Kumar, P. Kumar, A. Srivastava, V. D. A. Kumar, K. Vengatesan, and A. Singhal, ‘‘Comparative analysis of data mining techniques to predict heart disease for diabetic patients,’’ in Proc. Int. Conf. Adv. Comput. Data Sci. Singapore: Springer, 2020.

[3]. C. Sowmiya and P. Sumitra, ‘‘Analytical study of heart disease diagnosis using classification techniques,’’ in Proc. IEEE Int. Conf. Intell. Techn. Control, Optim. Signal Process. (INCOS), Mar. 2017.

[4]. Y -J. Huang, M. Parry, Y. Zeng, Y. Luo, J. Yang, and G.-P. He, ‘‘Examination of a nurse-led community-based education and coaching intervention for coronary heart disease high-risk individuals in China,’’ Asian Nursing Res., vol. 11, no. 3, pp. 187–193, Sep. 2017.

[5]. R. Hasan, ‘‘Comparative analysis of machine learning algorithms for heart disease prediction,’’ in Proc. ITM Web Conf., vol. 40, 2021.

[6]. S. Khan and S. T. Rasool, ‘‘Current use of cardiac biomarkers in various heart conditions,’’ Endocrine, Metabolic Immune Disorders-Drug Targets, vol. 21, no. 6, pp. 980–993, Jun. 2021.

[7]. S. Bashir, A. A. Almazroi, S. Ashfaq, A. A. Almazroi, and F. H. Khan, ‘‘A knowledge-based clinical decision support system utilizing an intelligent ensemble voting scheme for improved cardiovascular disease prediction,’’ IEEE Access, vol. 9, pp. 130805–130822, 2021.

[8]. S. Chae, S. Kwon, and D. Lee, ‘‘Predicting infectious disease using deep learning and big data,’’ Int. J. Environ. Res. Public Health, vol. 15, no. 8, p. 1596, Jul. 2018.

[9]. M. Raju, V. Pagidimarri, R. Barreto, A. Kadam, V. Kasivajjala, and A. Aswath, ‘‘Development of a deep learning algorithm for automatic diagnosis of diabetic retinopathy,’’ in Proc. MEDINFO, 2017, pp. 559–563.

[10]. G. T. Reddy and N. Khare, ‘‘An efficient system for heart disease prediction using hybrid OFBAT with rule-based fuzzy logic model,’’ J. Circuits, Syst. Comput., vol. 26, no. 4, Apr. 2017, Art. no. 1750061.

[11]. N. Khateeb and M. Usman, ‘‘Efficient heart disease prediction system using K-nearest neighbor classification technique,’’ in Proc. Int. Conf. Big Data Internet Thing, Dec. 2017, pp. 21–26.

[12]. L. Verma, S. Srivastava, and P. C. Negi, ‘‘A hybrid data mining model to predict coronary artery disease cases using non-invasive clinical data,’’ J. Med. Syst., vol. 40, no. 7, p. 178, Jul. 2016.

[13]. A. H. Chen, S. Y. Huang, P. S. Hong, C. H. Cheng, and E. J. Lin, ‘‘HDPS: Heart disease prediction system,’’ in Proc. Comput. Cardiol., Sep. 2011, pp. 557–560.

[14]. M. A. Jabbar and S. Samreen, ‘‘Heart disease prediction system based on hidden naïve Bayes classifier,’’ in Proc. Int. Conf. Circuits, Controls, Commun. Comput. (I4C), Oct. 2016, pp. 1–5.

[15]. K. Srinivas, B. K. Rani, and A. Govrdhan, ‘‘Applications of data mining techniques in healthcare and prediction of heart attacks,’’ Int. J. Data Mining Techn. Appl., vol. 7, no. 1, pp. 172–176, Mar. 2018.

[16]. R. Chitra and V. Seenivasagam, ‘‘Heart attack prediction system using cascaded neural network,’’ in Proc. Int. Conf. Appl. Math. Theor. Comput. Sci., 2013, p. 223.

[17]. T. Helmy, S. M. Rahman, M. I. Hossain, and A. Abdelraheem, ‘‘Nonlinear heterogeneous ensemble model for permeability prediction of oil reservoirs,’’ Arabian J. Sci. Eng., vol. 38, no. 6, pp. 1379–1395, Jun. 2013.

[18]. G. Alfian, M. Syafrudin, and J. Rhee, ‘‘Real-time monitoring system using smartphone-based sensors and NoSQL database for perishable supply chain,’’ Sustainability, vol. 9, no. 11, p. 2073, Nov. 2017, doi: 10.3390/su9112073.

[19]. M. Syafrudin, G. Alfian, N. Fitriyani, and J. Rhee, ‘‘Performance analysis of IoT-based sensor, big data processing, and machine learning model for real-time monitoring system in automotive manufacturing,’’ Sensors, vol. 18, no. 9, p. 2946, Sep. 2018, doi: 10.3390/s18092946.

[20]. M. Syafrudin, N. Fitriyani, G. Alfian, and J. Rhee, ‘‘An affordable fast early warning system for edge computing in assembly line,’’ Appl. Sci., vol. 9, no. 1, p. 84, Dec. 2018, doi: 10.3390/app9010084.

RE-2022-222458 - Turnitin Plagiarism Report

by Akilnanda Ek

Submission date: 25-Mar-2024 06:59PM (UTC+0300)
Submission ID: 271711402282
File name: RE-2022-222458.docx (38.09K)
Word count: 5595
Character count: 33162

RE-2022-222458-plag-report

ORIGINALITY REPORT

6%	4%	3%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	fastercapital.com Internet Source	1%
2	26lnw.service-finder.eu Internet Source	<1%
3	Sushovan Khatua, Anwesha Mukherjee, Debashis De. "SoVEC: Social Vehicular Edge Computing-based Optimum Route Selection", Vehicular Communications, 2024 Publication	<1%
4	Samit Karmakar, Pralay Kumar Ghosh, Debjani Kundu, Sampad Chakraborty, Shantanu Ghosh, Soumik Kumar Kundu. "Detection of Blood Glucose Level: A Machine Learning Approach", 2023 7th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech), 2023 Publication	<1%
5	www.govinfo.gov Internet Source	<1%
6	www.geokniga.org Internet Source	<1%

ICMDPET2K24 3:51 PM (6 hours ago)

to me

Dear Author(s)

Sub: **International Conference on "Multidisciplinary Perspectives in Engineering & Technology (ICMDPET-2K24)" paper acceptance notification**

- With reference to the above, we are very happy to inform you that your **paper has been reviewed by the technical committee and selected for oral presentation** in the conference.
- Submit the **registration form along with the payment.**
- **Submit your full paper as per IEEE format on or before 26.03.2024.**

Heart Disease Prediction Using Deeplearning

Mr. K. Sathyaseelan[1], M. Abdulsalam[2], E. K. Akilnanda[3], T. NijeethKumar[4]

[1] Assistant Professor, Department of Information Technology, Sri Ramakrishna Institute of Technology, Coimbatore, Tamil Nadu, Email: sathyaseelan.it@srit.org.

[2] Student, Department of Information Technology, Sri Ramakrishna Institute of Technology, Coimbatore, Tamil Nadu, Email: abdulsalam.2006@srit.org.

[3] Student, Department of Information Technology, Sri Ramakrishna Institute of Technology, Coimbatore, Tamil Nadu, Email: akil.2006@srit.org.

[4] Student, Department of Information Technology, Sri Ramakrishna Institute of Technology, Coimbatore, Tamil Nadu, Email: nijeeth.2006@srit.org.

Abstract :

Cardiovascular diseases (CVDs) continue to be a major worldwide health concern that require prompt management and an efficient early diagnosis. The goal of this research is to predict heart disease using deep learning, a powerful technique for identifying complex patterns from a variety of datasets. The main goal is to use deep neural networks and patient data integration to improve prediction accuracy and offer insightful information for proactive healthcare decision-making. The importance of this research resides in its capacity to revolutionize risk assessment techniques, promoting customized treatment and building confidence via the creation of clear and understandable models. We aim to improve the prediction accuracy of heart disease and reveal new insights into the complex relationships found in heterogeneous datasets by utilizing the power of deep learning. It is expected that this study's findings will contribute to a paradigm shift in the medical field by empowering medical professionals to customize interventions, make well-informed decisions, and ultimately improve patient outcomes in the area of cardiovascular health.

Index Term :

Machine learning, decision support system, deep learning, ensemble classifiers, heart disease diagnosis, accuracy, performance, cross validation.

Introduction :

Cardiovascular diseases (CVDs) are a major worldwide health problem that must be effectively diagnosed at an early stage in order to minimize their detrimental effects. In order to predict cardiac disease, this study uses deep learning, a reliable technique for identifying intricate patterns in diverse datasets. By integrating patient data and applying deep neural networks, the main goal is to improve prediction accuracy and offer significant insights for proactive healthcare choices. The technology that is being examined has the potential to transform risk assessment and bring in a new age of personalized healthcare. Our technique explores the subtleties of many datasets to find patterns and connections that might be early markers of cardiovascular diseases. Our technique relies heavily on deep neural networks, which are capable of extracting complex insights from convoluted

data structures and providing a powerful predictive analytics tool. This finding is significant because it has the potential to revolutionize risk assessment techniques and encourage customized treatment regimens based on specific patient profiles. Additionally, by emphasizing interpretability and openness, the study aims to increase patient and healthcare professional confidence in prediction models. In conclusion, our study uses deep learning to make a contribution to the worldwide conversation on cardiovascular health. The project aims to shift healthcare into a proactive paradigm and improve the effectiveness of managing cardiovascular disease globally by advancing predicting accuracy, providing actionable insights, and promoting transparent models.

Literature Survey :

1.Chae et al. used big data and deep learning to forecast the incidence of infectious illnesses, including chickenpox, malaria, and scarlet fever. The paper examined how well the conventional Autoregressive Integrated Moving Average (ARIMA) model performed in comparison to the Long Short-Term Memory (LSTM) and Deep Neural Network (DNN) models. Utilizing a dataset of 576 observations obtained from several sources, including KCDC, Naver Data Lab, KMA, and Twitter, the information included temperature, query data, and social media data. While ARIMA was used to analyze non-stationary time series data and show its application to infectious disease prediction, the Deep Neural Network used a back-propagation method to forecast time series data. The research provided in-depth insights by delving into the complex variations and oscillations in the data.

2. Raju et al. focused on utilizing a deep learning system to automatically detect diabetes in their work. Their suggested architecture used a convolutional technique using deep neural networks trained on a benchmark heart disease dataset downloaded from Kaggle in order to categorize diabetic retinopathy stages based on funduscopic pictures. High sensitivity and specificity values were found for different phases of diabetic retinopathy by the classification analysis. About 35,000 photos were used in the network training to obtain a sensitivity of 80.28% and specificity of 90.29%. An additional 8,810 images were used to contribute to a model accuracy of 93.28%. In order to increase classification accuracy, the research recommended taking into account other variables including demographics and family history. It also underlined the need of removing low-quality photos in order to lower mistake rates.

3. Reddy and Khare developed a hybrid heart disease prediction system using Rule-Based Fuzzy Logic (RBFL) and Oppositional Firefly with BAT (OFBAT). By making feature selection easier, Locality Preserving Projection (LRP) helped create a classification model using a fuzzy logic system. After that, fuzzy rules were created using example data, and the relevant rules were chosen using the OFBAT algorithm. These fuzzy rules and membership functions were used to create an efficient fuzzy system for categorization. Testing using Cleveland, Hungarian, and Switzerland, three publicly accessible UCI datasets, demonstrated RBFL's 78% accuracy for UCI datasets. The research makes recommendations for possible improvements by incorporating deep learning techniques.

4. Khateeb et al. used the 303-record UCI dataset, which is available to the public, to develop a diagnostic paradigm for heart illness. They evaluated six different machine learning algorithms in their investigation. While the second case only used seven of the fourteen relevant characteristics, the first scenario assessed classifiers without feature reduction.

In the third scenario, general data such as age, sex, and resting blood pressure were removed before accuracy estimates were made. Re-sampling was done on datasets with different properties in examples four and five. Using KNN in conjunction with the Synthetic Minority Over Sampling Technique (SMOTE), the sixth case achieved the best accuracy, almost 80%. Nevertheless, the suggested method is limited to using a single dataset.

5. Verma et al. put out a thorough framework that includes the identification of important risk factors for the diagnosis of heart disease. The study successfully identified these risk variables using K-means clustering and Particle Swarm Optimization (PSO). Then, supervised learning algorithms were used to classify data from three different medical datasets: 335 cases from Indhira Gandhi Medical College, patient data from India, and Cleveland heart disease data. Multinomial Logistic Regression performed better than other classifiers in all datasets, according to the results analysis, highlighting the diagnostic framework's usefulness.

6. Chen et al. presented a decision support system in their study that is intended to help doctors identify cardiac disease. The model used existing data to predict and assess cardiac disease. It was based on the Learning Vector Quantization (LVQ) technique. 13 important qualities were chosen for the framework, and then an artificial neural network was used to classify heart disease based on these attributes. By using Receiver Operating Characteristics (ROC) curve analysis, the user-friendly system showed an impressive 80% accuracy. The study made the case that text mining might improve the model's performance, especially when it comes to predicting unstructured data in the diagnosis of heart disease.

7. The Naïve Bayes classifier was utilized by Jabbar et al. in their work to improve the predicted accuracy of early heart disease detection. To remove unnecessary and redundant features, they used discretization and genetic search. The Genetic Algorithm (GA) was then used for feature selection, which involved removing the attributes with the lowest ranking from the dataset. When comparing Naïve Bayes' performance to alternative methods, the statlog dataset showed an astounding 86.29% accuracy in diagnosing heart disease.

8. Srinivas et al. developed a clinical decision support system for the detection of cardiac disease by utilizing machine learning techniques. The technique is based on three basic algorithms for classification, with testing conducted on benchmark datasets. In order to improve the quality of the data, data preprocessing uses an expanded form of Naïve Bayes to resolve missing values using imputation techniques like mean and mode. Compared to other current frameworks, the suggested framework has the benefit of being able to handle complicated questions in heart disease diagnosis in an effective manner.

9. Chitra and Sinivasagam introduced a model for heart disease diagnosis utilizing supervised learning algorithms and medical records. Employing a Cascaded Neural Network (CNN), the authors achieved classification based on information extracted from patients' medical records. The heart disease classification utilized 13 selected attributes as input for the CNN classifier, and the model's efficiency was evaluated on records from 270 patients, demonstrating effectiveness compared to state-of-the-art methods. The proposed CNN system stands as a valuable tool to aid physicians in the accurate diagnosis of heart disease.

10. Helmy et al. presented a machine learning-based decision support system for diagnosing cardiac disease in their study. Individual classifiers are used in the approach for training, and the Bagging algorithm is then used to aggregate the outcomes. Analysis of the results showed that, in comparison to alternative ensemble approaches and single classifiers, the fusion of heterogeneous classifiers using Bagging provided more effective results. Even if the method shows encouraging results, more performance research using a variety of benchmark and real-time datasets is advised.

Proposed Methodology :

In the context of diagnosing cardiac illness, we present a thorough approach in this study that makes use of cutting-edge methods for model selection, interpretability, and comparative analysis. Our method is based on the Automated Optimal Model Selection pipeline. This pipeline is designed to optimize a wide variety of neural network topologies by methodically combining Grid Search, Random Search, cross-validation, and hyperparameter tuning. By finding the best possible configurations, this thorough investigation seeks to improve the models' performance and guarantee a reliable and effective prediction framework. Our Interpretable Deep Learning method brings transparency to the diagnosis of heart disease. To provide comprehensible and unambiguous predictions, this combines LIME (Local Interpretable Model-agnostic Explanations) with attention processes. In order to ensure model interpretability, we use XGBoost combined with LIME in conjunction with Convolutional and Recurrent Neural Networks with attention processes. The goal of this hybrid method is to give understandable insights into the model's decision-making process in addition to precise forecasts. We do a Comparative Analysis of Ensemble Models to thoroughly evaluate the effectiveness of our suggested models. In this examination, deep learning techniques and conventional ensembles are carefully compared. We assess many datasets' worth of performance indicators, including sensitivity, specificity, accuracy, and more. The aim of this study is to clarify the advantages and disadvantages of each method, offering a comprehensive comprehension of their respective benefits within the particular framework of diagnosing heart disease. Our suggested technique aims to provide a comprehensive and perceptive approach to heart disease detection by integrating automated model selection, interpretable deep learning, and a rigorous comparative examination of ensemble models. This multimodal approach seeks to provide transparency and interpretability—two essentials for practical healthcare applications—while pushing the boundaries of predictive modeling.

Data Collection :

We carefully considered each dataset, obtaining information from the UCI Data Repository as well as Data World. After first analyzing four datasets (called "Cleveland," "Hungarian," "Switzerland," and "Long Beach"), we combined their characteristics. By means of group discussions, we were able to identify and choose two datasets that had fourteen important factors. Finalized variables include age, sex, type of chest pain (cp), maximum heart rate, exercise-induced angina, oldpeak, slope, ca (number of major vessels colored by fluoroscopy), cholesterol levels, fasting blood sugar, resting electrocardiographic results, and thal (thalassemia). The adoption of the best dataset for our continuing analysis and research projects is ensured by this cooperative team effort.

Data Preprocessing :

On the chosen datasets, our team used careful preprocessing methods for maximum accuracy.

To deal with missing values, imputation was used, guaranteeing a complete dataset for analysis. To scale numerical characteristics, standardization was applied in order to encourage consistency and make robust model training easier. Furthermore, the robust scaler was utilized to reduce the impact of outliers, improving the data's resilience against extreme values. All of these methods worked together to produce a clean, well-organized dataset that would enhance the performance of the model. The meticulous amalgamation of imputation, standardization, and resilient scaling conforms to optimal methodologies in data preparation, therefore establishing a foundation for dependable and precise analytical results in our research undertaking.

Logistic Algorithm :

When there are two classes in a categorical outcome variable, a statistical technique called logistic regression is employed to solve binary classification issues. Logistic regression predicts the likelihood that an input falls into a certain category, as opposed to linear regression, which is utilized for continuous outcomes.

$$P(Y=1) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_n X_n)}}$$

K Nearest Algorithm :

Based on a data point's closeness to the closest group of points, the K-Nearest Neighbor (KNN) classification method classifies the data point. The Euclidean distance formula, which is the square root of the sum of squared differences between the corresponding coordinates of two points in an n-dimensional space, is used to calculate this closeness. KNN essentially uses the least Euclidean distance between a data point and any points in the nearest group to classify the data point. KNN is a simple yet efficient method for pattern recognition and classification applications as the classification is decided by taking into account the closest neighbors and their related classes.

$$\mathrm{d}(\mathrm{p}, \mathrm{q}) = \sqrt{\sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{qi} - \mathrm{pi})\, 2}$$

Gaussian Process :

Using properties inherited from the normal distribution, a Gaussian process is a random process defined by a set of variables indexed by time or space, which is a generalized form of the Gaussian probability distribution. Unlike the Gaussian probability distribution, which sums up random variables, a Gaussian process summarizes the properties of a function, making it particularly useful in statistical modeling. The Gaussian process is defined by a specific function that calculates its characteristics, providing a potent tool for identifying and comprehending complex dependencies in data.

$$\mathrm{f}(\mathrm{x}) = \mathrm{a}.\exp((\mathrm{x}-\mathrm{y})^2/2\mathrm{c}^2)$$

SVM :

Support Vector Machines (SVM) is the name of the classification algorithm that is discussed and is used to find classes in a given dataset. SVM creates a model that may be used to classify new data points once it has been trained using pre-classified data into two groups. SVM seeks to provide a distinct distinction by maximizing the margins between classes on a plot, going beyond simple classification. In this approach, support vectors are essential for identifying the critical spots that determine the decision limits and improve the classification model.

$$W_0^T x + b_0 = 1 or W_0^T x + b_0 = -1$$

Decision Tree :

When dealing with datasets that include a large number of attributes, some of which may be

less important, a decision tree is an invaluable tool for pattern discovery and data cleansing. A criterion for determining whether qualities are unimportant or only marginally significant is the Gini Index. When building a decision tree, characteristics with low Gini Index are picked to produce rules, and nodes are selected based on the computation of conditional probabilities. Applications for this approach may be found in bioinformatics, particularly in the diagnosis and prognosis of illness. A formula is used to construct the Gini Index, which measures a dataset's impurity.

$$G = \sum_{I=1}^{C} p\,(i) * (1 - p\,(i))$$

Naïve Bayes :

The Naïve Bayes classifier operates under the assumption that the presence of a specific feature in a class is independent of the presence of any other feature. This classifier relies solely on the class feature, considering other class features as unrelated to each other [6]. The calculation of this classifier is based on a specific formula that leverages the independence assumption to make efficient and straightforward probabilistic predictions.

$$P(C_K|X) = \frac{P(C_K) * P(X|C_K)}{P(X)}$$

QDA :

For non-linear data separation, the Quadratic Discriminant Analysis (QDA) classifier is used. It fits data to class conditional densities and establishes a quadratic decision boundary using Bayes' rule. The main difference between QDA and Linear Discriminant Analysis (LDA) is that classes in QDA do not always have the same covariance matrix. When it comes to covariance matrix requirements, QDA is more flexible than LDA and shows better data-fitting capabilities [8]. Certain formulae designed for the quadratic decision boundary method are required to calculate QDA.

$$f_K(X) = \frac{1}{2}\log\left|\sum K\right| - (X - U_K)^T \sum_K^{-1} (X - U_K) + \log \pi_K$$

Ada Boost :

Adaptive Boosting, or AdaBoost, is an ensemble learning method applied to regression and classification problems. By combining the outputs from weak learners—usually straightforward decision trees or stumps—it builds a robust model. AdaBoost uses a weighted process for every data point, and iteratively modifies the weights to emphasize cases that were previously misclassified. This iterative approach makes sure that the algorithm prioritizes fixing incorrect classifications in later iterations, which enables later weak learners to train on these difficult examples more successfully.

Bagging Classifier :

Bagging, also known as Bootstrap Aggregation, is a group machine learning technique that combines the outputs of many learners to improve model performance. Bagging is a common technique used to reduce variation in noisy data. It entails choosing random chunks of training data. After being separately trained on a variety of machine learning models, these subgroups' predictions are aggregated to get an overall forecast. Although Bagging has shown more efficacy in classification problems, it may also be used to regression methods. It is mostly utilized for classification tasks such as Decision Trees and Naïve Bayes. To calculate it, one must use a certain equation.

$$f_{bag} = f_1\,(x) + f_2\,(x) + f_3\,(x) + \ldots\ldots\ldots f_n\,(x)$$

Boosting :

Boosting is an ensemble machine learning strategy that combines many weaker models—often constructed with decision trees—to produce a robust model. It is especially well-known for its efficient dataset classification, which uses a weighted minimization method

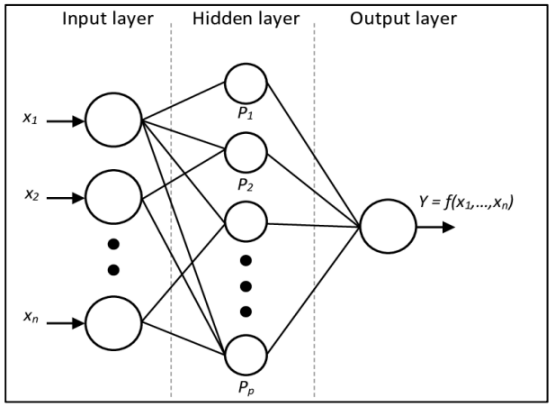
similar to Adaboost. Weighted inputs and classifiers are regenerated when weights are minimized. As the difference between real and forecasted values, loss is the main objective of boosting [10]. An exact equation must be used in order to calculate boosting.

$$F_b(x_i) = F_{b-1}(x_i) - \gamma - \times \nabla L(y_i, F(x_i))$$

Dense Neural Network :

An Artificial Neural Network (ANN) having one or more hidden layers, one input layer, and one output layer is called a Dense Neural Network (DNN). DNN components, which include neurons, weights, biases, and activation functions, simulate the workings of the human brain and are used to train machine learning algorithms . DNN neurons, which are modeled like real neurons, fire in response to particular stimuli and carry out predetermined tasks. Every neuron adds a continuous bias to a dot product of random weights and various inputs. An activation function is used to the output to determine how well the network predicts particular patterns. Weights are adjusted algorithmically if they are found to be erroneous . A generalized function is applied throughout the computation.

$$F_i = f\left(\sum_{i=1}^{n} x_i w_i + b\right)$$



Architecture of Dense Neural Network

Experimental Evolutions and Measures:

Several performance indicators are used to assess the suggested framework in order to demonstrate the efficacy of the activity.

Algorithm for Proposed Methodology

Accuracy :

The ratio of a classification model's correct evaluations to all evaluations performed on the test data is known as accuracy.

$$Accuracy = \frac{Truenegatives + Truepositives}{positives + negatives}$$

Sensitivity :

the proportion of accurate evaluations that a categorization model labels as accurate. This formula is used to compute it.

$$Sensitivity = \frac{Truepositives}{Falsepositives + Truepositives}$$

Specificity :

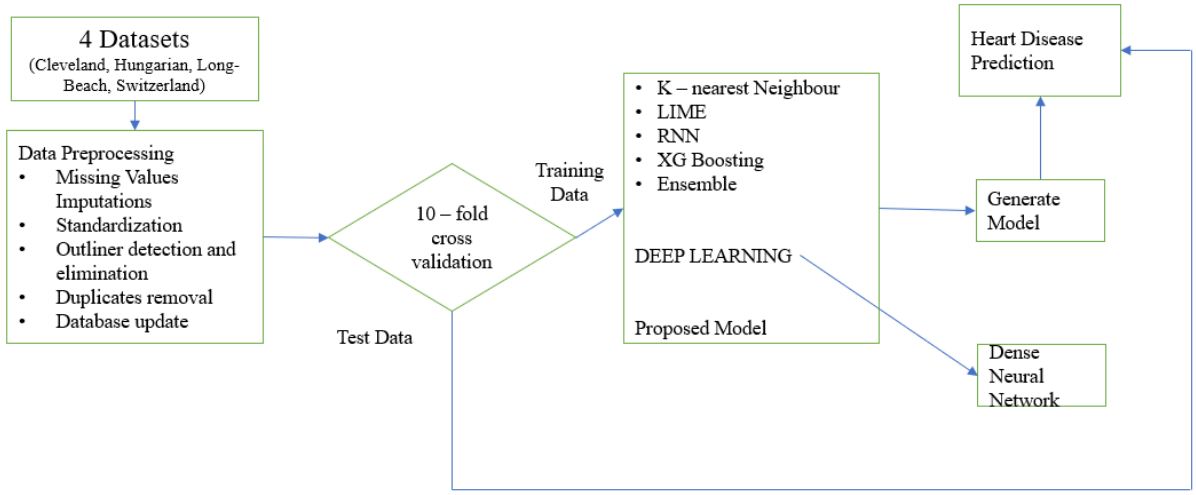
the proportion of incorrect evaluations that a categorization model flags as incorrect. This formula is used to compute it.

$$Specificity = \frac{Truenegatives}{Falsenegatives + Truenegatives}$$

F – Measure :

F-Measure is the ratio between sensitivity and specificity.

$$F - Measure = \frac{2 * Sensitivity * Specificity}{Sensitivity + Specificity}$$



Analysis and Result :

The results are obtained through the use of Jupiter Notebook, a tool that allows Python code to be run in a web browser. Users of Jupiter notebook get access to 12GB of RAM for running programming. The suggested approach starts by evaluating the output from each classifier separately. Next, the best-performing individual classifiers are selected and tested on benchmark datasets related to heart disease.

The heart disease datasets used in this work come from the reliable UCI machine learning data repository, which is highly recommended by several researchers. Cleveland and Hungarian are the two standard datasets for heart disease that are used. Every dataset is distinguished by particular characteristics that are utilized to determine whether individuals have heart disease or not. The output feature is represented by the label attribute (num), which has categorical values that indicate who is well and who is ill. In the analysis, values that are healthy are represented by 1, and those that are sick by 0.

Cleaveland Dataset :

There are fourteen features in the Cleveland dataset. The 13 characteristics are independent of one another and are input features. The final column represents the output feature, which is essentially a label attribute. It is dependent upon the input characteristics; for example, if it is 0, it indicates that the patient has heart disease; if not, it does not.

There are 303 cases in the collection.

Hungarian Dataset :

There are 294 cases and 12 characteristics in the Hungarian dataset.

While the "num" output feature is reliant on input characteristics, the 13 input features are independent of one another. There are four number values in the label property "num" (0–4). The patient has cardiac disease if the readings are zero; otherwise, they are not.

Features Name	Description
Age	Age stated in years
Sex	Gender type
CP	Type of chest pain
Trestbps	Blood pressure level at resting state
Chol	Total cholesterol in blood (mg/dl)
Fbs	Level of fasting blood sugar > 120 mg/dl
Restecg	Resting electrocardiographic results
Thalach	Max heart rate achieved
Exang	Exercise induced angina
Oldpeak	ST depression induced by exercise relative to rest
Slope	Slope of the peak exercise ST segment
Ca	Number of major vessels colored by flourosopy
Thal	6= fixed defect; 3=Normal; 7= reversible defect

PIDD - Description

Sno	Attributes	Description	Data Type	Range
1	Age	Age stated in years	Numeric	[Correct age]
2	Sex	Gender type	Numeric	[0 & 1]
3	CP	Types of chest Pain	Numeric	[1-4]
4	Trestbps	Blood pressure level at resting state	Numeric	[94-200]
5	Chol	Total cholesterol in blood (mg/dl)	Numeric	[126-564]
6	Fbs	Level of fasting blood sugar > 120 mg/dl	Numeric	[1 & 0]
7	Restecg	Resting electrocardiographic results	Numeric	[0-2]
8	Thalach	Max heart Rate achieved	Numeric	[71-202]
9	Exang	Exercise induced angina	Numeric	[0 & 1]
10	Oldpeak	ST depression induced by exercise relative to rest	Numeric	[0-6.2]
11	Slope	Slope of the peak exercise relative to rest	Numeric	[1-3]
12	Ca	Number of major vessels colored by flourosopy	Numeric	[0-3]
13	Thal	Types of defect	Numeric	3 = normal 6 = fixed 7 = reversible

Cleaveland statistical Summary

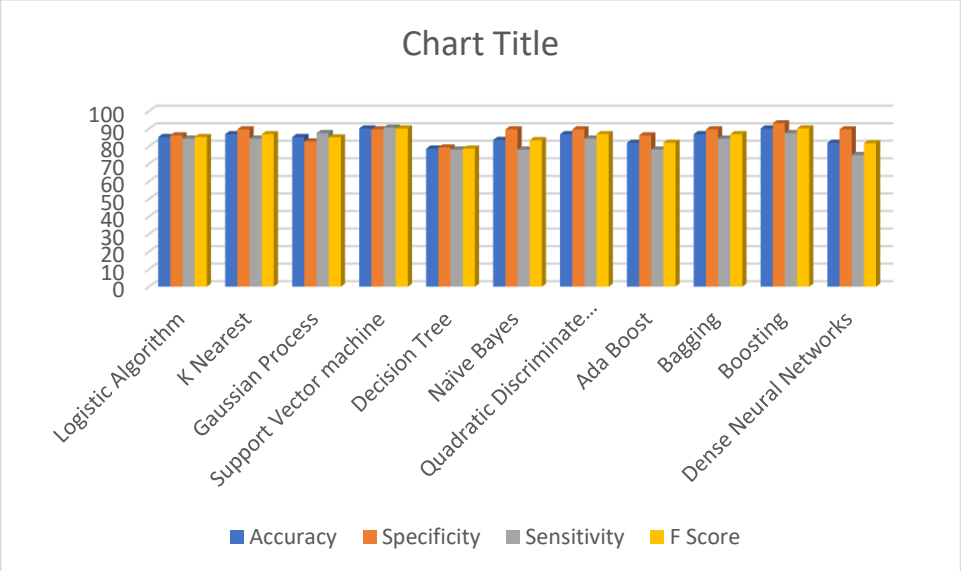
	age	sex	cp	Trestbps	cholestrol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
Count	303	303	303	303	303	303	303	303	303	303	303	303	303	303
Mean	54.4	0.67	3.15	131.68	246.69	0.14	0.99	149.6	0.32	1.03	1.6	0.69	4.72	0.45
Std	9.03	0.46	0.96	17.59	51.77	0.35	0.99	22.87	0.46	1.16	0.61	0.96	1.93	0.49
Min	29	0	1	94	126	0	0	71	0	0	1	0	3	0
25 %	48	0	3	120	211	0	0	135	0	0	1	0	3	0
50 %	56	1	3	130	241	0	1	153	0	0.80	2	0	3	0
75 %	61	1	4	140	275	0	2	166	1	1.6	2	1	7	1
Max	77	1	4	200	564	1	2	202	1	6.2	3	3	7	1

Hungarian Statistical Summary

	age	sex	cp	trestbps	cholestrol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
Count	294	294	294	294	294	294	294	294	294	294	294	294	294	294
Mean	47.82	0.72	2.98	132.13	231.22	0.06	0.21	138.65	0.30	0.58	0.67	0.001	0.53	0.36
Std	7.81	0.44	0.96	19.22	93.65	0.25	0.46	24.90	0.46	0.90	0.92	0.02	1.72	0.48
Min	28.0	0	1	0	0	0	0	0	0	0	0	0	0	0
25 %	42	0	2	120	198	0	0	122	0	0	0	0	0	0
50 %	49	1	3	130	237	0	0	140	0	0	0	0	0	0
75 %	54	1	4	140	277	0	0	155	1	.1	2	0	0	1
Max	66	1	4	200	603	1	2.1	190	1	5	3	0.4	7	1

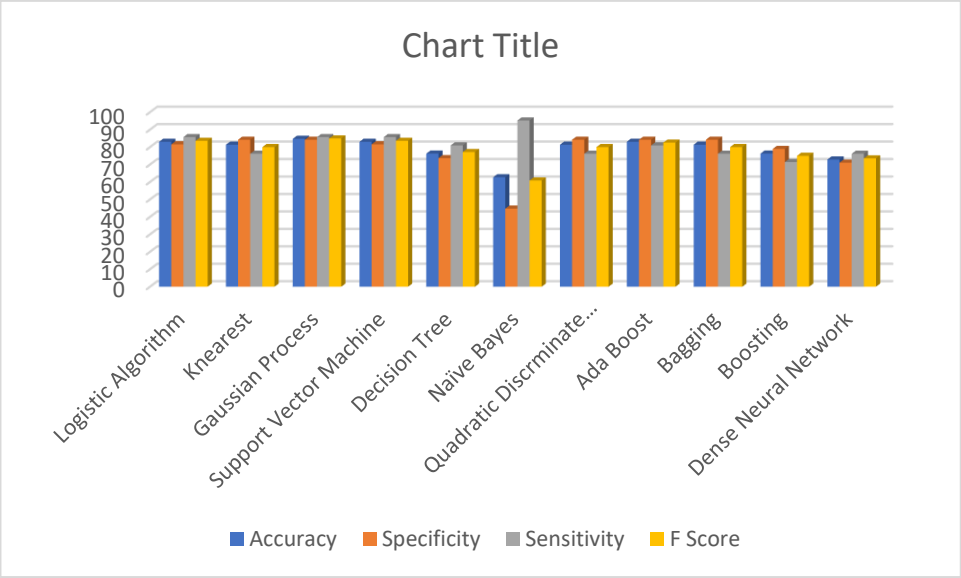
Cleaveland

	Accuracy	Specificity	Sensitivity	F Score	Exe time
Logistic Algorithm	85.25	86.21	84.38	85.28	0.0188
K Nearest	86.89	89.66	84.38	86.93	0.0220
Gaussian Process	85.25	82.76	87.50	85.06	0.7899
Support Vector machine	90.16	89.66	90.62	90.14	0.0274
Decision Tree	78.69	79.31	78.12	78.71	0.0078
Naïve Bayes	83.61	89.66	78.12	83.49	0.0106
Quadratic Discriminate Analysis	86.89	89.66	84.38	86.93	0.0302
Ada Boost	81.97	86.21	78.12	81.97	0.0625
Bagging	86.89	89.66	84.38	86.93	0.1115
Boosting	90.16	93.10	87.50	90.21	0.0560
Dense Neural Networks	81.97	89.66	75	81.68	1.1631



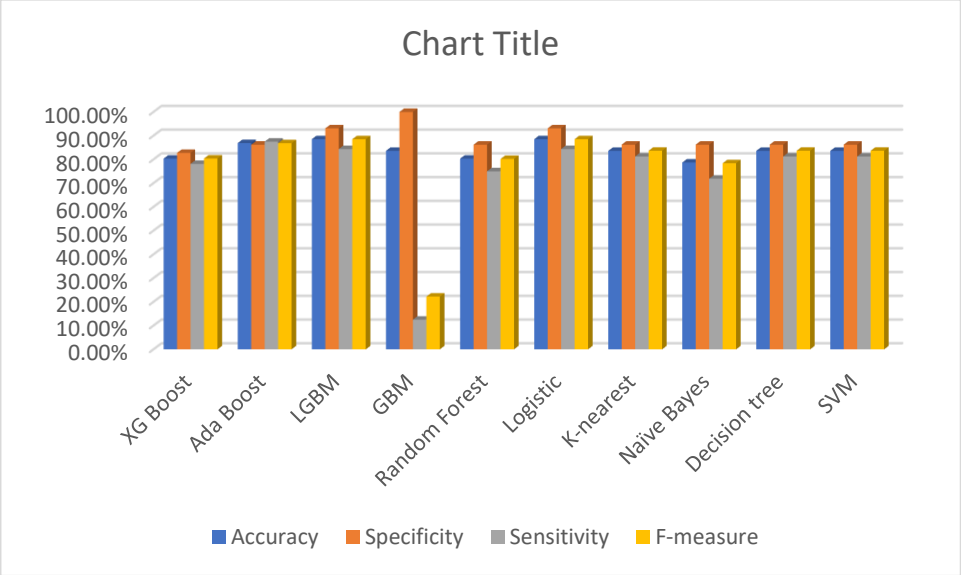
Hungarian

	Accuracy	Specificity	Sensitivity	F Score	Exe time
Logistic Algorithm	83.05	81.58	85.71	83.6	0.0198
Knearest	81.36	84.21	76.19	80	0.0309
Gaussian Process	84.75	84.21	85.71	84.96	0.4587
Support Vector Machine	83.05	81.58	85.71	83.60	0.0238
Decision Tree	76.27	73.68	80.95	77.15	0.0061
Naïve Bayes	62.71	44.74	95.24	60.88	0.0869
Quadratic Discrminate Analysis	81.36	84.21	76.19	80	0.1135
Ada Boost	83.05	84.21	80.95	82.55	0.0807
Bagging	81.36	84.21	76.19	80	0.1216
Boosting	76.27	78.95	71.43	75	0.0489
Dense Neural Network	72.88	71.05	76.19	73.53	1.1340

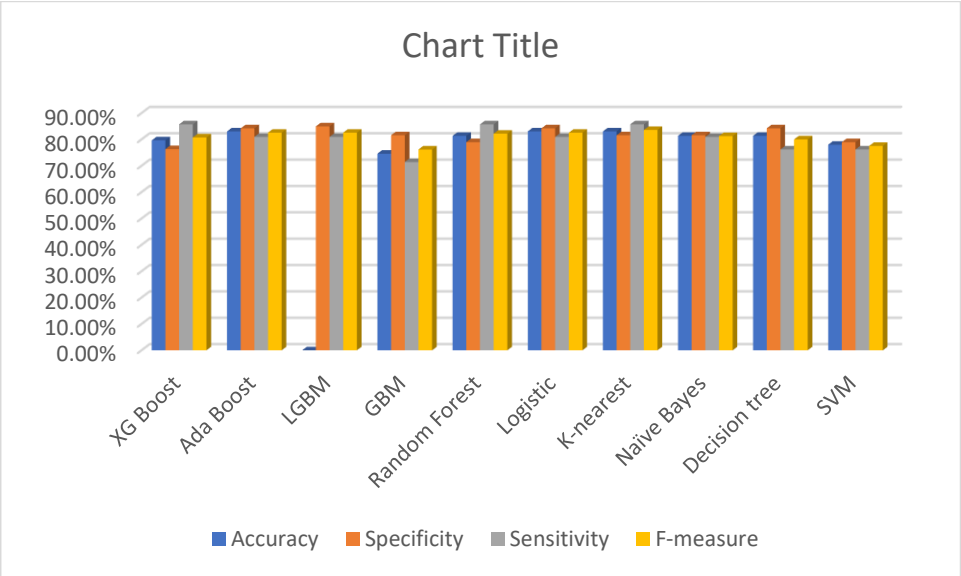


Cleveland	Accuracy	Specificity	Sensitivity	F-measure	Execution time
XG Boost	80.33%	82.76%	78.12%	80.38%	28.7102
Ada Boost	86.89%	86.21%	87.50%	86.85%	27.3439
LGBM	88.52%	93.10%	84.38%	88.52%	0.0578
GBM	83.61%	100%	12.50%	22.22%	29.0419
Random Forest	80.33%	86.21%	75%	80.21%	28.5142
Logistic	88.52%	93.10%	84.38%	88.52%	29.3103
K-nearest	83.61%	86.21%	81.25%	83.66%	28.1663

Naïve Bayes	78.69%	86.21%	71.88%	78.39%	27.6944
Decision tree	83.61%	86.21%	81.25%	83.66%	27.7524
SVM	83.61%	86.21%	81.25%	83.66%	30.2198



Hungarian	Accuracy	Specificity	Sensitivity	F-measure	Execution time
XG Boost	79.66%	76.32%	85.71%	80.74%	27.4996
Ada Boost	83.05%	84.21%	80.95%	82.55%	28.7462
LGBM	83,05%	84.95%	80.95%	82.55%	0.0439
GBM	74.58%	81.58%	71.43%	76.17%	27.1836
Random Forest	81.36%	78.95%	85.71%	82.19%	27.4080
Logistic	83.05%	84.21%	80.95%	82.55%	28.0320
K-nearest	83.05%	81.58%	85.71%	83.60%	28.4060
Naïve Bayes	81.36%	81.58%	80.95%	81.26%	28.6789
Decision tree	81.36%	84.21%	76.19%	80%	26.6422
SVM	77.97%	78.97%	76.19%	77.54%	26.6306



Heart Disease Prediction

Age

Sex

Chest Pain types

Resting Blood Pressure

Serum Cholestoral in mg/dl

Fasting Blood Sugar > 120 mg/dl

Resting Electrocardiographic results

Maximum Heart Rate achieved

Exercise Induced Angina

ST depression induced by exercise

Slope of the peak exercise ST segment

Major vessels colored by flourosopy

thal: 0 = normal; 1 = fixed defect; 2 = reversable defect

Heart Disease Test Result

Heart Disease Prediction

Age

63

Sex

1

Chest Pain types

1

Resting Blood Pressure

145

Serum Cholestoral in mg/dl

233

Fasting Blood Sugar > 120 mg/dl

1

Resting Electrocardiographic results

2

Maximum Heart Rate achieved

150

Exercise Induced Angina

0

ST depression induced by exercise

2

Slope of the peak exercise ST segment

3

Major vessels colored by flourosopy

0

thal: 0 = normal; 1 = fixed defect; 2 = reversable defect

6

Heart Disease Test Result

The person does not have any heart disease

Conclution :

In conclusion, our study leveraged machine learning and 10-fold cross-validation to significantly enhance heart disease prediction. Successful data scaling and null value handling during preparation set a strong foundation. The diverse deep neural network models explored, such as KNN, Decision Trees, and SVM, yielded distinct results. Our findings hold promise for revolutionizing healthcare analytics, enabling early diagnosis, personalized treatment recommendations, and resource optimization. Furthermore, our research contributes to medical advancements, evaluating new heart disease medicines and predicting their efficacy in advance. The significance of sophisticated preprocessing, ensemble learning, and machine learning methodologies underscore their pivotal role in improving patient care, fostering medical research, and advancing healthcare analytics for the benefit of researchers, data scientists, and healthcare professionals alike.

Reference :

[1]. N.-S. Tomov and S. Tomov, ‘‘On deep neural networks for detecting heart disease,’’ 2018, arXiv:1808.07168.

[2]. A. Kumar, P. Kumar, A. Srivastava, V. D. A. Kumar, K. Vengatesan, and A. Singhal, ‘‘Comparative analysis of data mining techniques to predict heart disease for diabetic patients,’’ in Proc. Int. Conf. Adv. Comput. Data Sci. Singapore: Springer, 2020.

[3]. C. Sowmiya and P. Sumitra, ‘‘Analytical study of heart disease diagnosis using classification techniques,’’ in Proc. IEEE Int. Conf. Intell. Techn. Control, Optim. Signal Process. (INCOS), Mar. 2017.

[4]. Y -J. Huang, M. Parry, Y. Zeng, Y. Luo, J. Yang, and G.-P. He, ‘‘Examination of a nurse-led community-based education and coaching intervention for coronary heart disease high-risk individuals in China,’’ Asian Nursing Res., vol. 11, no. 3, pp. 187–193, Sep. 2017.

[5]. R. Hasan, ‘‘Comparative analysis of machine learning algorithms for heart disease prediction,’’ in Proc. ITM Web Conf., vol. 40, 2021.

[6]. S. Khan and S. T. Rasool, ‘‘Current use of cardiac biomarkers in various heart conditions,’’ Endocrine, Metabolic Immune Disorders-Drug Targets, vol. 21, no. 6, pp. 980–993, Jun. 2021.

[7]. S. Bashir, A. A. Almazroi, S. Ashfaq, A. A. Almazroi, and F. H. Khan, ‘‘A knowledge-based clinical decision support system utilizing an intelligent ensemble voting scheme for improved cardiovascular disease prediction,’’ IEEE Access, vol. 9, pp. 130805–130822, 2021.

[8]. S. Chae, S. Kwon, and D. Lee, ‘‘Predicting infectious disease using deep learning and big data,’’ Int. J. Environ. Res. Public Health, vol. 15, no. 8, p. 1596, Jul. 2018.

[9]. M. Raju, V. Pagidimarri, R. Barreto, A. Kadam, V. Kasivajjala, and A. Aswath, ‘‘Development of a deep learning algorithm for automatic diagnosis of diabetic retinopathy,’’ in Proc. MEDINFO, 2017, pp. 559–563.

[10]. G. T. Reddy and N. Khare, ‘‘An efficient system for heart disease prediction using hybrid OFBAT with rule-based fuzzy logic model,’’ J. Circuits, Syst. Comput., vol. 26, no. 4, Apr. 2017, Art. no. 1750061.

[11]. N. Khateeb and M. Usman, ‘‘Efficient heart disease prediction system using K-nearest neighbor classification technique,’’ in Proc. Int. Conf. Big Data Internet Thing, Dec. 2017, pp. 21–26.

[12]. L. Verma, S. Srivastava, and P. C. Negi, ‘‘A hybrid data mining model to predict coronary artery disease cases using non-invasive clinical data,’’ J. Med. Syst., vol. 40, no. 7, p. 178, Jul. 2016.

[13]. A. H. Chen, S. Y. Huang, P. S. Hong, C. H. Cheng, and E. J. Lin, ‘‘HDPS: Heart disease prediction system,’’ in Proc. Comput. Cardiol., Sep. 2011, pp. 557–560.

[14]. M. A. Jabbar and S. Samreen, ‘‘Heart disease prediction system based on hidden naïve Bayes classifier,’’ in Proc. Int. Conf. Circuits, Controls, Commun. Comput. (I4C), Oct. 2016, pp. 1–5.

[15]. K. Srinivas, B. K. Rani, and A. Govrdhan, ‘‘Applications of data mining techniques in healthcare and prediction of heart attacks,’’ Int. J. Data Mining Techn. Appl., vol. 7, no. 1, pp. 172–176, Mar. 2018.

[16]. R. Chitra and V. Seenivasagam, ‘‘Heart attack prediction system using cascaded neural network,’’ in Proc. Int. Conf. Appl. Math. Theor. Comput. Sci., 2013, p. 223.

[17]. T. Helmy, S. M. Rahman, M. I. Hossain, and A. Abdelraheem, ‘‘Nonlinear heterogeneous ensemble model for permeability prediction of oil reservoirs,’’ Arabian J. Sci. Eng., vol. 38, no. 6, pp. 1379–1395, Jun. 2013.

[18]. G. Alfian, M. Syafrudin, and J. Rhee, ‘‘Real-time monitoring system using smartphone-based sensors and NoSQL database for perishable supply chain,’’ Sustainability, vol. 9, no. 11, p. 2073, Nov. 2017, doi: 10.3390/su9112073.

[19]. M. Syafrudin, G. Alfian, N. Fitriyani, and J. Rhee, ‘‘Performance analysis of IoT-based sensor, big data processing, and machine learning model for real-time monitoring system in automotive manufacturing,’’ Sensors, vol. 18, no. 9, p. 2946, Sep. 2018, doi: 10.3390/s18092946.

[20]. M. Syafrudin, N. Fitriyani, G. Alfian, and J. Rhee, ‘‘An affordable fast early warning system for edge computing in assembly line,’’ Appl. Sci., vol. 9, no. 1, p. 84, Dec. 2018, doi: 10.3390/app9010084.