Applying Machine Learning Algorithms for Liver Disease Prediction

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

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

D. Satish (21471A0583)

Sk. Munaf (21471A05C2)

B. Anji Babu (21471A0574)

Under the esteemed guidance of

K.V.Narasimha Reddy, M.Tech.,

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASAROPET

(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under

Tyre -1 NIRF rank in the band of 201-300 and an

ISO 9001:2015 Certified

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

KOTAPPAKONDA ROAD, YALAMANDA VILLAGE,

NARASARAOPET- 522601

2024-2025

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

CERTIFICATE

This is to certify that the project that is entitled with the name **"Applying Machine Learning Algorithms for Liver Disease Prediction "** is a bonafide work done by the team **D. Satish (21471A0583), Sk. Munaf (21471A05C2), B. Anji Babu (21471A0574)** in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2024-2025.

PROJECT GUIDE
K. V. Narasimha Reddy, M.Tech.
Assistant Professor

PROJECT CO-ORDINATOR
Dr. M. Sireesha,M.Tech.Ph.D.,
Assoc. Professor

HEAD OF THE DEPARTMENT
Dr. S. N. Tirumala Rao, M.Tech., Ph.D.,
Professor & HOD

EXTERNAL EXAMINER

DECLARATION

We declare that this project work titled "**APPLYING MACHINE LEARNING ALGORITHMS FOR LIVER DISEASE PREDICTION**" is composed by ourselves that the work contain here is our own except where explicitly stated otherwise in the text and that this work has not been submitted for any other degree or professional qualification except as specified.

D. Satish (21471A0583)
Sk. Munaf (21471A05C2)
B. Anji Babu (21471A0574)

ACKNOWLEDGEMENT

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

We express our deep felt gratitude towards **Dr. S. N. Tirumala Rao**, **M.Tech., Ph.D.,** HOD of CSE department and also to our guide **K.V. Narasimha Reddy**, **M.Tech** , of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

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

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

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

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

By

D. Satish (21471A0583)

Sk. Munaf (21471A05C2)

B. Anji Babu (21471A0574)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

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

INSTITUTION MISSION

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

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

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

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

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

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

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

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

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

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering.

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

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

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

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

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

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Outcomes

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the

knowledge of, and need for sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and giveand receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Project Course Outcomes (CO'S):

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

CO421.2: Identify and classify the requirements.

CO421.3: Review the Related Literature

CO421.4: Design and Modularize the project

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

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

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1		✓											✓		
C421.2	✓		✓		✓								✓		
C421.3				✓		✓	✓	✓					✓		
C421.4			✓			✓	✓	✓					✓	✓	
C421.5					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C421.6									✓	✓	✓		✓	✓	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1	2	3											2		
C421.2			2		3								2		
C421.3				2		2	3	3					2		
C421.4			2			1	1	2					3	2	
C421.5					3	3	3	2	3	2	2	1	3	2	1
C421.6									3	2	1		2	3	

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

- Low level
- Medium level
- High level

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

Name of the course from which principles are applied in this project	Description of the device	Attained PO
C2204.2, C22L3.2	Gathering the requirements and defining the problem, planning to develop a model for predicting liver disease using machine learning algorithms like SVM, KNN, and MLP.	PO1, PO3
CC421.1, C2204.3, C22L3.2	Each and every requirement is critically analyzed, the process model is identified	PO2, PO3
CC421.2, C2204.2, C22L3.3	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC421.3, C2204.3, C22L3.2	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC421.4, C2204.4, C22L3.2	Documentation is done by all our four members in the form of a group	PO10
CC421.5, C2204.2, C22L3.3	Each and every phase of the work in group is presented periodically	PO10, PO11
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	The implemented machine learning model can be deployed for real-time liver disease prediction, with future updates based on expanded datasets and improved algorithms.	PO4, PO7
C32SC4.3	The model includes a user interface to allow healthcare professionals and researchers to assess whether a patient has liver disease based on input data	PO5, PO6

ABSTRACT

Liver disease is a major global health concern, particularly in regions like India, where early diagnosis remains a challenge due to the late onset of symptoms. This study applies machine learning algorithms to predict liver disease using patient data, enhancing the accuracy and speed of diagnosis. Various models, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Hard Voting Classifier, Multilayer Perceptron (MLP), Decision Tree, Logistic Regression, and Random Forest, were implemented and optimized using Genetic Algorithms (GA). The dataset consists of 583 patient records with 11 attributes, including liver function test results and demographic details. Extensive data preprocessing, including cleaning, normalization, and feature selection, was performed to ensure model efficiency. Performance metrics such as Accuracy, Precision, Recall, and F1-Score were used to evaluate the models. Among all classifiers, the Random Forest model optimized with Genetic Algorithm achieved the highest accuracy of 79%, with an F1-Score of 0.87, demonstrating its effectiveness in liver disease prediction. Other models, such as Logistic Regression with Genetic Algorithm and Support Vector Machine, also exhibited strong performance, with high recall values indicating their ability to detect liver disease cases effectively. This study highlights the potential of machine learning in improving liver disease diagnosis by providing a data-driven, automated approach. The findings suggest that integrating optimization techniques like Genetic Algorithms enhances model performance, making it a valuable tool for clinical decision-making. Future work can focus on expanding the dataset and exploring deep learning techniques to further refine predictive accuracy.

INDEX

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Liver disease represents a significant and growing challenge to global public health, with increasing mortality and morbidity rates worldwide. The liver plays a crucial role in maintaining overall health by performing essential functions such as detoxification, protein synthesis, and the production of biochemicals necessary for digestion and metabolism. Any disruption in liver function can have severe consequences, affecting multiple organ systems and leading to life-threatening complications. The increasing prevalence of liver disease can be attributed to a range of factors, including viral infections, alcohol consumption, metabolic disorders, obesity, and lifestyle changes.

According to the World Health Organization (WHO), liver diseases account for approximately 2 million deaths per year, making them a leading cause of death globally. The burden of liver disease is not uniformly distributed, with developing countries, including India, experiencing a particularly high prevalence. Hepatitis B and C infections are among the most common causes of liver disease globally, contributing to chronic liver inflammation, fibrosis, cirrhosis, and hepatocellular carcinoma (HCC). Furthermore, the rise in metabolic syndrome and obesity has led to a surge in cases of Non-Alcoholic Fatty Liver Disease (NAFLD) and its more severe form, Non-Alcoholic Steatohepatitis (NASH).

Liver diseases can be broadly classified into several categories based on their underlying causes. Alcohol-Related Liver Disease (ALD) remains one of the primary causes of liver disease. Chronic alcohol abuse leads to progressive liver damage, including fatty liver, alcoholic hepatitis, fibrosis, and cirrhosis. ALD is a major health issue in both developed and developing countries, contributing to a significant proportion of liver-related morbidity and mortality.

Viral Hepatitis caused by Hepatitis B (HBV) and Hepatitis C (HCV) infections are significant public health threats, affecting millions of people globally. HBV and HCV can cause both acute and chronic liver inflammation, which, if left untreated, can progress to cirrhosis and liver cancer. The availability of antiviral therapies has improved outcomes; however, early diagnosis remains critical for effective treatment.

Non-Alcoholic Fatty Liver Disease (NAFLD) is increasingly recognized as a major cause of chronic liver disease, particularly in individuals with obesity, insulin resistance, and metabolic syndrome. NAFLD encompasses a spectrum of liver conditions, from simple steatosis (fat accumulation in liver cells) to NASH, which involves inflammation and liver cell injury. NASH can lead to fibrosis, cirrhosis, and HCC if not managed effectively. Autoimmune and genetic liver diseases are also significant contributors to liver dysfunction. Autoimmune hepatitis, primary biliary cholangitis (PBC), and primary sclerosing cholangitis (PSC) are examples of autoimmune liver diseases where the immune system mistakenly attacks liver cells, leading to inflammation and damage. Genetic conditions such as Wilson's disease and hemochromatosis also contribute to liver dysfunction by causing abnormal accumulation of copper and iron in the liver, respectively.

Drug-Induced Liver Injury (DILI) is another important cause of liver disease. Certain medications, including over-the-counter drugs, herbal supplements, and prescription medications, can cause liver toxicity. DILI is a significant cause of acute liver failure and can present with a range of symptoms, from mild liver enzyme elevation to severe hepatitis and liver necrosis.

Despite the availability of advanced diagnostic tools, early detection of liver disease remains a significant challenge. Liver diseases often progress silently, with symptoms appearing only when the disease has advanced to a critical stage. Common early symptoms, such as fatigue, mild abdominal discomfort, and loss of appetite, are often overlooked or misattributed to other less severe conditions. By the time more serious signs such as jaundice, ascites, and hepatic encephalopathy appear, the disease has often progressed to cirrhosis or liver failure, limiting treatment options and reducing survival rates.

Despite the availability of advanced diagnostic tools, early detection remains challenging because symptoms often appear in the later stages when the disease has progressed [1]. This delay in identification raises the possibility of deadly consequences and complicates treatment.

The increasing concern on the worldwide burden of liver diseases naturally enhances interest to apply machine learning approaches in improving early detection. With a good ma chine learning model applied on patient data that includes liver function test, among other biochemical markers, it is possible to identify the disease patterns that are otherwise not easily detected by other methods [2]. This approach not only improves diagnostic accuracy but also allows for earlier intervention, which is crucial for reducing the severity of the disease.

The predictive models, such as SVM, Neural Networks, and Voting Classifiers have shown considerable success in liver disease prediction. These models can efficiently handle large datasets, process complex relationships, and offer insights that assist in clinical decision making [3]. As a result, these methods represent a significant advancement in healthcare technology, providing a more data-driven approach to diagnosing liver disorders.

Early-stage detection through these predictive models has proven beneficial in identifying high-risk patients even before clinical symptoms develop. This capability is vital in reducing liver disease-related mortality and morbidity, as it enables timely treatment and better disease management [4]. Machine learning integration in health care is not only helpful in the establishment of early diagnosis of liver conditions but also provides a scalable solution 2 to this ever-growing burden related to liver disease.

Machine learning algorithms present a promising solution to the challenges of diagnosing liver disease. As these technologies advance, they enable healthcare professionals to achieve more accurate, efficient, and early detection methods. This progress represents a significant advancement in enhancing patient outcomes and mitigating the global impact of liver disease [5].

Machine learning-based diagnostic systems offer a scalable solution to the growing burden of liver disease. These systems can be integrated into electronic health records (EHR) and clinical decision support systems (CDSS) to provide real-time insights and assist healthcare professionals in diagnosing and managing liver disease. This not only enhances diagnostic accuracy but also facilitates personalized treatment plans based on individual patient profiles.

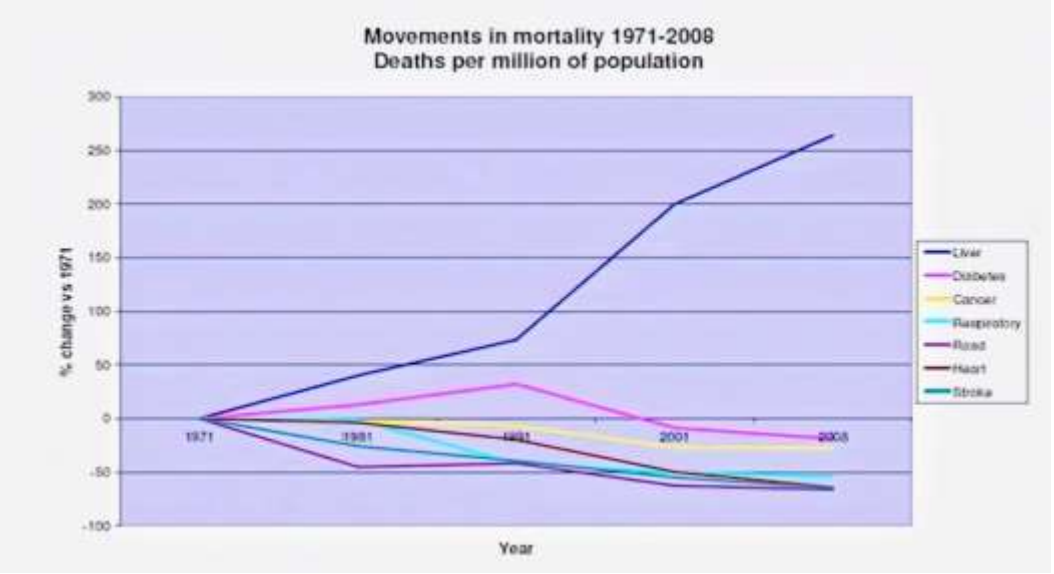

Fig 1.1 Deaths due to Liver disease statistics

Fig 1.1 graph shows the percentage change in mortality rates from 1971 to 2008 for various causes of death (Liver, Diabetes, Cancer, Respiratory, Road, Heart, and Stroke). Liver disease (blue line) shows a significant upward trend, indicating a sharp increase in deaths over the period, whereas other causes like heart disease and stroke have generally declined or remained stable.

This paper proposes the application of multiple machine learning algorithms to predict liver diseases using patient data. The models used in this study include SVM, K-Neighbors, Hard Voting Classifier, Multilayer Perceptron, Decision Tree, Logistic Regression, and Random Forest, optimized using a Genetic Algorithm. Key performance metrics such as Accuracy, Precision, Recall, and F1-Score are used to evaluate the models' performance. The primary goal is to improve early detection rates, enhance diagnostic accuracy, and support clinical decision-making through a data-driven approach. By integrating machine learning into liver disease diagnosis, healthcare systems can move toward a more proactive and personalized model of care. The findings of this study aim to contribute to the development of reliable, efficient, and scalable solutions for the early detection and management of liver disease, ultimately improving patient outcomes and reducing the global burden of liver disease.

2.LITERATURE SURVEY

Research in the applicability of machine learning algorithms in diagnosing liver disease has gained significant momentum recently. Various studies have demonstrated the effectiveness of different methodologies, providing insights into improving diagnostic processes. One such study by Murty and Kumar [6] used a multi-layer perceptron neural network to improve classifier accuracy in liver disease diagnosis. This research highlights how advancements in neural network architectures can significantly enhance the precision of liver disease classification, offering a promising approach to diagnostics.

Haque et al. [7] compared the performances of random forests and artificial neural networks in the classification of liver disorders. Random forests exhibited robustness, while artificial neural networks excelled at identifying complex patterns. This comparative analysis is essential for selecting the right model based on the specific needs of a diagnosis. Joloudari et al. [8] integrated Particle Swarm Optimization (PSO) with Support Vector Machines (SVM) and feature selection techniques in a computer aided decision-making study. This research emphasizes the importance of optimization techniques to improve SVM performance, leading to better liver disease predictions through enhanced feature selection.

Gaber et al. [9] explored the use of supervised learning methods combined with genetic algorithms for the automatic classification of fatty liver disease. The findings indicate that genetic algorithms optimize the learning process, leading to better classification results for fatty liver disease, which is vital for effective patient management. Kuzhippallil et al. [10] conducted a comparative analysis of machine learning algorithms tailored for Indian liver disease patients. This study highlights the variations in model performance across different patient demographics and emphasizes the need for customized approaches to ensure effectiveness in such a diverse population.

Gupta et al. [11] provided an overview of various machine learning classification algorithms for predicting liver disease. This comprehensive study contributes to a broader understanding of how machine learning can be utilized for

accurate liver disease diagnosis. Musleh et al. [12] demonstrated the potential of artificial neural networks in predicting liver disease with high accuracy. The research underscores the utility of neural networks in clinical settings for early diagnosis and intervention.

Overall, these studies form a significant body of work that establishes the progress in machine learning techniques for liver disease diagnosis. Importantly, the studies have made insightful observations about how various models and optimization strategies might be further developed in this critical area of healthcare. The proposed study is analyzing the various liver function tests that are in employment in order to give a prediction regarding liver disease, considering a set of patient data as input and putting through several classifiers, namely: SVM, K-Neighbors, Voting Classifier, Multilayer Perceptron, Decision Tree, Logistic Regression and Random Forest. Performance metrics that are used are the ones that would give the best model on the basis of the predicted liver health: Precision, Recall, F1-score, Accuracy and Confusion-Matrix.

2.1 Machine Learning

Machine Learning (ML) is a subset of artificial intelligence (AI) that enables systems to learn from data and improve their performance without being explicitly programmed. It involves algorithms that identify patterns, make predictions, and automate decision-making processes.

2.2 Some Machine Learning Methods

a. Data Preprocessing Techniques

These methods prepare the patient dataset for effective model training and classification:

- Data Cleaning: Handling missing values, removing outliers, and ensuring consistent data formats to improve model reliability.
- Feature Scaling: Standardizing numerical features such as bilirubin levels, enzyme counts, and protein levels to ensure equal contribution to the model.
- Normalization: Transforming patient attribute values into a common scale to enhance learning efficiency.

- Class Balancing: Addressing dataset imbalance by using resampling techniques to improve prediction accuracy for liver disease cases.

b. Feature Extraction and Selection

These methods refine relevant features for optimal disease classification:

- Genetic Algorithm (GA) Optimization: Selecting the most relevant features by reducing redundancy and improving computational efficiency.
- Correlation-Based Feature Selection: Identifying and retaining attributes with strong relationships to liver disease diagnosis.
- Principal Component Analysis (PCA): Reducing dimensionality while preserving key information to enhance model performance.
- Feature Fusion: Combining selected biomarkers, such as bilirubin levels and enzyme activity, to improve prediction capability.

c. Machine Learning Model Architectures

Several models are used for classification, each optimized for specific performance requirements:

- Random Forest with Genetic Algorithm (RF-GA): A robust ensemble model that achieved the highest accuracy in liver disease prediction.
- Support Vector Machine (SVM): A high-precision model effective in binary classification tasks.
- Multilayer Perceptron (MLP): A deep learning approach capable of detecting complex data patterns.
- Hard Voting Classifier (HVC): An ensemble method combining SVM, Decision Trees, and Logistic Regression for improved decision-making.
- Decision Tree (DT) with Genetic Algorithm (GA): An optimized rule-based classifier tailored for hierarchical decision-making.
- K-Nearest Neighbors (KNN): A distance-based algorithm effective in classifying patients with similar health conditions.
- Logistic Regression with GA Optimization: A statistical classifier improved through hyperparameter tuning for liver disease diagnosis.

d. Training and Optimization Methods

These techniques ensure effective model training and performance improvement:

- Cross-Validation: Splitting the dataset into multiple subsets to evaluate model generalization and reduce overfitting.
- Hyperparameter Tuning: Optimizing model parameters using Genetic Algorithms for improved classification accuracy.
- Feature Selection with GA: Selecting the most informative features to reduce complexity and improve efficiency.
- Early Stopping: Halting model training when validation performance stabilizes, preventing overfitting.

e. Evaluation and Performance Metrics

These metrics validate the effectiveness of the models:

- Accuracy: Measures the proportion of correctly classified liver disease cases.
- Precision and Recall: Evaluates the model's effectiveness in minimizing false positives and false negatives.
- F1-Score: Ensures a balance between precision and recall, particularly crucial for imbalanced datasets.
- Confusion Matrix: Analyzes misclassified instances to identify areas for model improvement.
- ROC Curve and AUC Score: Assesses model performance in distinguishing between diseased and non-diseased patients.

f. Deployment and Scalability

Methods for real-world application and scalability:

- Edge Device Compatibility: Optimizing ML models for deployment on low-resource medical devices for portable diagnosis.
- Scalability for Large Datasets: Enhancing model adaptability to handle extensive patient records for widespread use.

2.3 Applications of Machine Learning

ML is applied in various fields, including:

- **Healthcare** – Disease diagnosis, medical imaging analysis, personalized medicine.
- **Finance** – Fraud detection, credit risk analysis, algorithmic trading.

- **Autonomous Vehicles** – Object detection, lane recognition, traffic prediction.
- **Natural Language Processing (NLP)** – Chatbots, speech recognition, text summarization.

2.4 Importance of Machine Learning in Liver Disease Prediction

Liver disease is a serious health condition that is often diagnosed at an advanced stage due to its asymptomatic nature in the early phases. Traditional diagnostic methods rely on manual examination and biochemical tests, which may not always detect the disease in its initial stages. Machine learning plays a crucial role in addressing these challenges by leveraging patient medical records, laboratory test results, and other health indicators to detect liver disease at an early stage. Early detection is essential in preventing complications, reducing mortality rates, and improving patient outcomes.

One of the key advantages of machine learning in liver disease prediction is its ability to significantly improve accuracy compared to traditional diagnostic techniques. ML models can analyze large datasets, identify hidden patterns, and make highly accurate predictions based on past patient data. Techniques such as Support Vector Machines (SVM), Random Forest, and Neural Networks have shown superior performance in classification tasks, enabling doctors to diagnose liver disease with greater confidence. Furthermore, machine learning enables automated diagnosis, reducing the need for manual intervention. This automation speeds up the diagnostic process, allowing for faster identification of at-risk patients and timely medical intervention.

Machine learning also enhances predictive analysis, which is critical in understanding disease progression. By analyzing patient history, genetic predisposition, and lifestyle factors, ML models can predict how a liver disease may develop over time and suggest preventive measures. This allows healthcare professionals to design personalized treatment plans, ensuring early intervention and better disease management. Additionally, machine learning contributes to cost efficiency by optimizing the diagnostic process, reducing the need for extensive testing, and minimizing hospital visits. This, in turn, alleviates the burden on healthcare systems and makes liver disease detection more accessible and affordable.

In summary, machine learning is revolutionizing liver disease prediction by offering early detection, improved accuracy, automated diagnosis, predictive analysis, and cost-effective solutions. As technology advances, integrating deep learning models and real-time patient monitoring systems will further enhance the ability to diagnose and manage liver disease efficiently, ultimately saving lives and improving overall healthcare quality.

2.5 Importance of Machine Learning Using Python

Python has become the most widely used programming language for machine learning due to its simplicity, flexibility, and extensive ecosystem of libraries. One of the primary reasons for its popularity is the availability of powerful libraries such as Scikit-learn, TensorFlow, Pandas, and NumPy, which streamline machine learning model development, data preprocessing, and deep learning implementations. These libraries provide built-in functions for data manipulation, feature extraction, and model training, significantly reducing the time and effort required for building ML applications. Python's easy implementation further enhances its appeal, as its simple syntax and high readability allow both beginners and experienced developers to quickly prototype and deploy ML models. Unlike complex languages, Python enables rapid experimentation, making it ideal for iterative model improvement and fine tuning.

Another major advantage of Python is its strong community support, which includes a vast open-source network of developers, researchers, and contributors. This ensures continuous updates, extensive documentation, and a wealth of online tutorials, making it easier to troubleshoot issues and stay updated with the latest advancements in machine learning. Additionally, Python's integration with other technologies enhances its usability across different domains. It seamlessly connects with databases for data storage and retrieval, web applications for deploying ML models, and cloud platforms for scalable computing power. These features make Python an excellent choice for developing robust, real-world machine learning solutions, whether for healthcare, finance, autonomous systems, or natural language processing. As machine learning continues to evolve, Python's versatility and efficiency will keep it at the forefront of AI-driven innovations.

3.EXISTING SYSTEM

The existing system for liver disease prediction employs a combination of machine learning classifiers to analyze patient data and predict the likelihood of liver disease. The current approach involves several key stages, including data preprocessing, model training, classification, and evaluation. This system relies on a dataset comprising patient records, including liver function test results and other biochemical markers. The goal is to improve the accuracy of liver disease detection using machine learning models and optimize them for enhanced predictive performance.

The existing system begins with data preprocessing, which includes handling missing values, transforming categorical variables, and scaling numerical attributes. The dataset used in the study is sourced from the Indian Liver Patient Dataset (ILPD), which contains records from 583 individuals suffering from liver disease. Preprocessing starts with data cleaning, where impurities, outliers, and incomplete entries are addressed using techniques such as KNN (K-Nearest Neighbors) imputation to fill in missing values. Data standardization is then applied to ensure that all numerical attributes, such as bilirubin levels, enzyme counts, and protein levels, are scaled to a uniform range. This improves the consistency and comparability of data across different models.

Following data preprocessing, various machine learning models are applied to classify patients as either having or not having liver disease. The models used in the existing system include Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Hard Voting Classifier (HVC), and Decision Tree (DT). Each of these classifiers employs a different learning approach to identify complex patterns and relationships within the data.

Model evaluation is conducted using key performance metrics such as Accuracy, Precision, Recall, Specificity, F1-Score, and the Confusion Matrix. Accuracy measures the proportion of correctly classified instances, while precision and recall evaluate the model's ability to minimize false positives and false negatives, respectively. The F1-Score provides a balanced measure of precision and recall, and the confusion

matrix gives a detailed breakdown of classification outcomes.

Among the classifiers tested, the Hard Voting Classifier (HVC) achieved the highest accuracy in predicting liver disease, making it the most effective model in the existing system. The HVC model integrates the strengths of multiple classifiers, enhancing decision-making and predictive performance. According to the study conducted by Anthonysamy et al., the HVC achieved 94% specificity, 80% precision, 87% F1-Score, and 78% accuracy when applied to the Indian Liver Patient Dataset. The MLP model followed closely, with 77% accuracy and 100% recall, indicating its strong performance in identifying positive cases.

While the existing system provides a reliable prediction mechanism, it faces several challenges. One significant issue is feature redundancy, where overlapping features reduce the model's ability to differentiate between cases. Additionally, the current system lacks proper feature selection and parameter tuning, which can lead to suboptimal performance. For instance, the KNN model’s accuracy can be influenced by the chosen value of K and the type of distance metric used. Similarly, the SVM model’s performance is highly sensitive to the choice of the kernel and the regularization parameter (C). The existing system also struggles with class imbalance, where the number of healthy and diseased cases is not equally represented, potentially leading to biased predictions.

Furthermore, the system does not incorporate real-time data or dynamic learning, meaning that the models cannot adjust to changing patterns in patient data over time. This limits the system's adaptability and long-term effectiveness in clinical settings. Enhancements such as feature selection using Principal Component Analysis (PCA), Recursive Feature Elimination (RFE), and Genetic Algorithms could help improve model performance by reducing redundancy and improving computational efficiency.

Despite these limitations, the existing system marks a significant advancement in liver disease prediction using machine learning. The ability of the Hard Voting Classifier to combine multiple models and deliver high predictive accuracy demonstrates the potential of ensemble learning in medical diagnosis. As healthcare

systems continue to collect larger volumes of patient data, further refinement of the existing system using more advanced feature selection, optimization algorithms, and real-time learning capabilities can enhance predictive accuracy and clinical decision-making.

Figure 3.1 illustrates the working process of the existing model, where training data is processed by multiple classification models to generate predictions, which are then combined through a voting mechanism to produce the final prediction. The successful application of the Hard Voting Classifier highlights the benefits of ensemble learning, while the performance of the MLP model underscores the importance of neural network-based approaches in handling complex medical data. The combination of these techniques represents a powerful framework for improving liver disease diagnosis and patient outcomes.

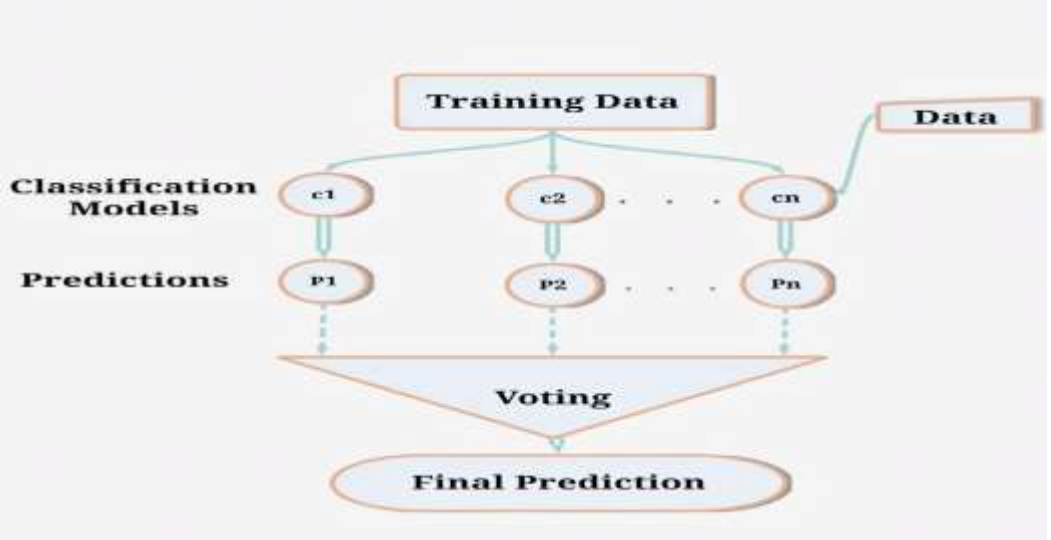

Fig 3.1 Existing Model Working process

4.PROPOSED SYSTEM

The proposed system introduces a more optimized and efficient framework for liver disease prediction by integrating advanced machine learning techniques and Genetic Algorithm (GA)-based optimization. The methodology begins with enhanced data preprocessing, where missing values are handled with KNN imputation, and categorical variables such as gender are converted into numerical representations. Additionally, feature normalization and standardization ensure that all attributes contribute equally to the model training process.

A key improvement in the proposed system is the incorporation of Genetic Algorithm (GA) for feature selection and hyperparameter optimization. GA helps identify the most relevant features, reducing redundancy and improving computational efficiency. The selected features are then fused into an optimized dataset, enhancing the model's ability to learn significant patterns in liver disease prediction.

For classification, the system employs a range of optimized machine learning models, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Hard Voting Classifier (HVC), Decision Tree (DT), Logistic Regression (LogReg), and Random Forest (RF). Unlike the existing system, the proposed framework integrates Random Forest with Genetic Algorithm (RF-GA), which has been found to outperform other models with an accuracy of 79%, a Precision of 81%, a Recall of 94%, and an F1-Score of 0.87.

Advantages:

a. Improved Models
b. Effective Feature Selection
c. High Classification Accuracy
d. Scalability
e. Reduced Computational Cost
f. Practical Applicability

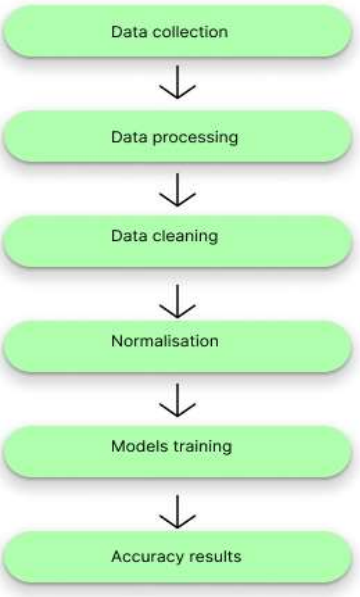

Fig 4.1 Flow Chart of Proposed System

Fig 4.1 represents the workflow of the proposed system, starting from data collection, followed by processing, cleaning, normalization, model training, and finally generating accuracy results.

The proposed system employs multiple machine learning models, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Decision Tree (DT), and Random Forest (RF). The Random Forest model, optimized with Genetic Algorithm (RF-GA), achieves the highest accuracy (79%), making it the most efficient model for liver disease classification. Unlike traditional classifiers, RF-GA utilizes multiple decision trees to improve prediction stability, reduce variance, and provide a more reliable diagnostic outcome.

The existing system lacks an optimization process for feature selection, often leading to redundant or irrelevant features being used in model training. The proposed system addresses this issue by integrating a Genetic Algorithm-based feature selection approach. This ensures that only the most relevant attributes (such as bilirubin levels, enzyme counts, and protein levels) are used, improving classification accuracy while reducing unnecessary computations. Feature selection reduces noise in the dataset, prevents overfitting, and enhances the model's generalization ability, making predictions more reliable across different patient datasets.

The proposed system is designed to be scalable and adaptable for real-world applications. It can handle large datasets efficiently, making it suitable for hospital systems, research institutions, and nationwide liver disease screening programs. Additionally, it supports cloud-based deployment, allowing integration with hospital databases and telemedicine platforms for remote disease diagnosis. The ability to scale up and handle larger volumes of patient data without compromising accuracy makes this system ideal for widespread adoption.

Traditional machine learning models often suffer from high computational costs due to unnecessary feature processing and inefficient algorithms. The proposed system addresses this issue by minimizing computational expenses through several optimizations. First, it utilizes Genetic Algorithm-based feature selection to eliminate redundant attributes, significantly reducing the amount of data that needs to be processed. Additionally, hyperparameter optimization ensures that models run efficiently while consuming fewer computational resources. The system also leverages ensemble learning techniques, such as Random Forest and Hard Voting Classifier, which enhance performance while reducing training time compared to deep learning-based methods. By implementing these optimizations, the system becomes faster, more efficient, and cost-effective, making it highly suitable for large-scale deployment in hospitals and research centers where computational efficiency is critical.

The proposed system is designed for real-world clinical use, ensuring seamless integration into diagnostic workflows for healthcare professionals. One of its key applications is automated liver disease screening, where the model can assist doctors in real-time patient assessments, leading to faster and more accurate diagnoses. Additionally, the system supports integration with Electronic Health Records (EHRs), enabling hospitals to store and retrieve patient data while generating instant liver disease predictions. For remote healthcare accessibility, the system can be deployed on cloud platforms, allowing patients in rural or underserved areas to receive diagnostic assessments without visiting a hospital. Furthermore, the model enhances personalized treatment planning by analyzing individual patient data and recommending tailored treatment strategies, helping doctors make data-driven medical decisions. With its scalability, ease of deployment, and high accuracy, the proposed system represents a significant advancement in automated liver disease detection and management.

Finally, The proposed system offers a significant improvement over traditional liver disease prediction models by incorporating optimized feature selection, advanced classifiers, and efficient computational techniques. Its high classification accuracy, scalability, and real-world applicability make it a powerful tool for early diagnosis and treatment planning, ultimately improving patient outcomes and reducing the burden on healthcare systems.

5.SYSTEM REQUIREMENTS

5.1 Hardware Requirements:

- System Type : intel®core™i3-7500UCPU@2.40gh
- Cache memory : 4MB or higher
- RAM : 8GB or higher
- Hard Disk : 256GB or higher

5.2 Software Requirements:

- Operating System : Windows 11, 64-bit Operating System
- Coding Language : Python
- Python distribution : Flask
- Browser : Any Latest Browser like Chrome

6.SYSTEM ANALYSIS

6.1 Scope of the project

The proposed system aims to address the challenges of early and accurate liver disease detection using data-driven machine learning models. Liver disease is a growing global health concern, often diagnosed at advanced stages due to late-onset symptoms. The proposed system introduces an automated, high-accuracy, and scalable approach to improve early diagnosis, reduce false predictions, and enhance clinical decision-making.

a. **Enhanced Disease Detection:** The system is designed to improve classification accuracy by employing optimized feature selection techniques and ensemble learning methods.
b. **Automation and Precision:** Manual diagnosis is time-consuming and prone to human error. The proposed system automates the data processing and prediction workflow, ensuring faster and more reliable results.
c. **Scalability Across Datasets:** The system is structured to handle large patient datasets, making it suitable for hospital databases, research institutions, and telemedicine applications.
d. **Feature Optimization:** The system incorporates Genetic Algorithm (GA) for feature selection, ensuring that only the most relevant attributes (such as bilirubin levels, enzyme activity, and protein counts) contribute to the prediction model. This approach reduces computational complexity while improving model efficiency.
e. **Real-World Applicability:** The model is designed to integrate with hospital management systems, enabling real-time liver disease screening and providing data-driven insights to healthcare professionals.
f. **Future Research and Improvements:** The system lays the foundation for future deep learning implementations, such as Convolutional Neural Networks (CNNs) for medical image analysis and real-time disease monitoring through IoT-enabled healthcare devices.

6.2 Analysis

A thorough analysis of the proposed system involves evaluating its methodology, performance, and advantages over existing approaches. The dataset used

in this study consists of 583 patient records with 11 key attributes, including bilirubin levels, enzyme counts, and protein levels. To ensure data quality, the preprocessing phase includes handling missing values using K-Nearest Neighbors (KNN) imputation, normalizing numerical features, and encoding categorical variables. These steps ensure that the dataset is structured properly, allowing machine learning models to achieve optimal performance.

For feature extraction and selection, the proposed system integrates a Genetic Algorithm (GA)-based feature selection approach, which eliminates redundant and irrelevant features to improve classification efficiency. This optimization step enhances model accuracy while reducing computational complexity, ensuring that only the most significant features contribute to the prediction process. The system is trained using multiple machine learning classifiers, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Hard Voting Classifier (HVC), Decision Tree (DT), and Random Forest (RF). Among these, Random Forest optimized with Genetic Algorithm (RF-GA) demonstrated the highest accuracy of 79%, making it the most reliable model for liver disease prediction.

The system's performance is evaluated using accuracy, precision, recall, F1-score, and specificity, ensuring that it is both robust and effective for real-world applications. RF-GA consistently outperforms other models, demonstrating superior predictive capabilities. The integration of machine learning with genetic algorithms enables the proposed system to achieve higher classification accuracy, faster processing speeds, and greater practical applicability. This combination of advanced feature selection, optimized machine learning models, and performance-driven evaluation metrics makes the system a significant advancement in automated liver disease prediction.

Table 6.1 provides an overview of the dataset attributes used in the liver disease prediction model, including their types and descriptions, such as age, gender, bilirubin levels, enzyme levels, protein content, and disease status.

Table 6.1 Dataset Description

SI. No	Attribute Name	Attribute Type	Attribute Description
1	Age	Numeric	Age of the patient
2	Gender	Numeric	Gender of the patient
3	Total Bilirubin	Numeric	Quantity of total bilirubin in patient
4	Direct Bilirubin	Numeric	Quantity of direct bilirubin in patient
5	Alkphos Alkaline Phosphotase	Numeric	Amount of ALP enzyme in patient
6	Alanine Aminotransferase - SGPT	Numeric	Amount of SGPT in patient
7	Aspartate Aminotransferase - SGOT	Numeric	Amount of SGOT in patient
8	Total Proteins	Numeric	Protein content in patient
9	Albumin	Numeric	Amount of albumin in patient
10	Albumin and Globulin Ratio	Numeric	Fraction of albumin and globulin in patient
11	Status	{1, 2}	Status of liver disease in patient

6.3.Data Pre-processing

The dataset [13] used in this study consists of 583 patient records with 11 key attributes extracted from biochemical tests and demographic information. To enhance data quality and improve model performance, several preprocessing techniques are applied. Missing values in the dataset are handled using K-Nearest Neighbors (KNN) imputation, ensuring completeness for model training. Feature scaling and normalization are performed on numerical attributes like bilirubin levels, enzyme counts (ALT, AST, ALP), and protein levels using Min-Max Scaling or Z-score normalization, ensuring all features contribute equally to the model. Additionally, categorical data encoding is applied to transform variables like gender into numerical values using label encoding or one-hot encoding, making them compatible with machine learning algorithms. Since the dataset may suffer from class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) is used to balance the distribution of diseased and non-diseased cases, preventing bias in model predictions and ensuring more reliable classification outcomes.

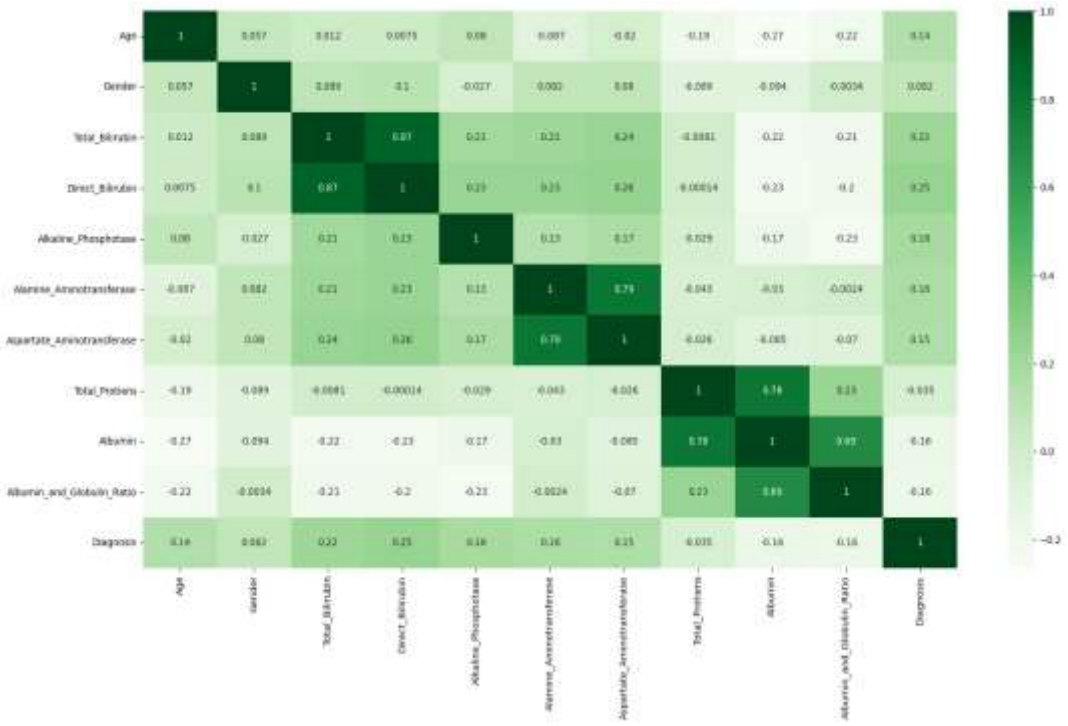

Fig 6.1 Corelation matrix

6.4. Feature Selection and Optimization:

Feature selection is a critical step in improving the efficiency and accuracy of machine learning models. The proposed system utilizes Genetic Algorithm (GA)-based feature selection, which automatically selects the most important features for liver disease classification.

- Genetic Algorithm (GA) Optimization: GA iteratively searches for the optimal feature subset by evaluating different feature combinations based on a fitness function (e.g., accuracy and F1-score).
- Reduction of Redundant Attributes: The GA-based approach eliminates features that do not significantly contribute to the model's decision-making, reducing computational overhead while improving classification accuracy.

6.5. Model building:

In the proposed system, multiple machine learning models are trained to classify patients as either having liver disease or not. The training phase involves feeding preprocessed patient data into different classifiers, each optimized to enhance predictive performance. Feature selection using Genetic Algorithm (GA) ensures that only the most relevant attributes contribute to the model, improving classification accuracy.

Hyperparameter tuning is applied to optimize each model, ensuring the best possible performance. The models used in this study include Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Hard Voting Classifier (HVC), Decision Tree (DT), Logistic Regression (LR), and Random Forest (RF) optimized with Genetic Algorithm (RF-GA). Among these, RF-GA achieved the highest accuracy of 79%, proving to be the most reliable model for liver disease prediction. Each of these models plays a crucial role in analyzing complex patterns in patient data, allowing for early detection, improved diagnostic precision, and better healthcare outcomes.

a. Support Vector Machine (SVM):

Support Vector Machine (SVM) is a supervised learning model that is particularly effective in binary classification problems like liver disease prediction. It maps patient data into a high-dimensional space and finds an optimal hyperplane that best separates the two classes: diseased and non-diseased. The Radial Basis Function (RBF) kernel is used in this study, as it enables non-linear classification, improving performance on complex datasets. Regularization (C=100) is applied to prevent overfitting, ensuring a balance between bias and variance. Additionally, gamma ($\gamma = 0.0001$) is fine-tuned to control the influence of individual data points, enhancing model generalization. SVM is highly accurate and robust, making it a reliable choice for medical diagnoses, particularly when dealing with complex, high-dimensional patient data.

b. **K-Nearest Neighbors (KNN):**

K-Nearest Neighbors (KNN) is a non-parametric, instance-based learning algorithm that classifies patients based on similarity to their neighbors. It determines a patient's class by calculating the Euclidean distance between data points and selecting the k closest neighbors. In this study, k=4 was chosen to maintain a balance between model sensitivity and computational efficiency. KNN is particularly useful when feature relationships are complex, as it makes predictions based on real patient similarities rather than learned parameters. However, KNN is sensitive to noise and requires feature scaling, making preprocessing crucial for optimal results. It is a simple yet effective classifier, providing high recall rates, which is critical in medical applications where missing a liver disease diagnosis can have serious

consequences.

c. Multilayer Perceptron (MLP):

Multilayer Perceptron (MLP) is a deep learning-based model that uses multiple layers of neurons to capture complex relationships in data. The model consists of an input layer, multiple hidden layers, and an output layer, allowing it to learn hierarchical representations of patient attributes. In this study, MLP is trained with three hidden layers (11, 9, 1), using activation functions like Tanh and ReLU to improve learning efficiency. The Stochastic Gradient Descent (SGD) and Adam optimizers are used to minimize loss and enhance convergence speed. MLP excels in identifying non-linear patterns in liver disease data, making it particularly useful for handling large and complex datasets.

d. Hard Voting Classifier (HVC):

The Hard Voting Classifier (HVC) is an ensemble learning technique that combines multiple machine learning models to improve classification performance. In this study, HVC integrates Support Vector Machine (SVM), Decision Tree (DT), and Logistic Regression (LR) to leverage their combined strengths. The hard voting mechanism assigns a class label based on the majority prediction of the individual classifiers, improving overall accuracy. This approach enhances model stability and reduces individual model biases, making it more robust for real-world applications. HVC is particularly effective when models exhibit diverse decision boundaries, as it balances precision, recall, and overall accuracy. While it may increase computational cost, it significantly enhances classification performance, making it an ideal choice for medical diagnosis tasks like liver disease prediction.

e. Decision Tree (DT) with Genetic Algorithm :

Decision Tree (DT) is a rule-based classifier that splits patient data into a hierarchical structure, allowing for interpretable decision-making in liver disease prediction. The tree structure is optimized using Genetic Algorithm (GA), which fine-tunes max-depth, minimum samples split, and splitting criteria (gini/entropy) to improve classification accuracy. DT is particularly useful for handling categorical and numerical features, making it a flexible choice for medical data analysis. Its main advantage is interpretability, as doctors and researchers can easily trace back

model decisions to understand why a particular classification was made. However, standard Decision Trees may suffer from overfitting, which is mitigated in this study through GA-based optimization, ensuring better generalization and higher accuracy.

f. Logistic Regression (LR) with Genetic Algorithm :

Logistic Regression (LR) is a statistical classification model that estimates the probability of liver disease presence based on patient attributes. The proposed system enhances LR performance using Genetic Algorithm (GA) to fine-tune key parameters, such as regularization strength (0.01 to 10.0) and penalty type (L1/L2). LR is particularly useful in binary classification problems, where interpretability and probability estimation are crucial. It calculates the log odds of liver disease occurrence, providing a clear understanding of feature importance. Despite being a linear classifier, LR performs well when optimized, and its computational efficiency makes it suitable for real-time medical predictions. In this study, GA-based optimization significantly improves LR's accuracy, recall, and precision, making it a reliable alternative to more complex models.

g. Random Forest (RF) with Genetic Algorithm (RF-GA):

Random Forest (RF) is an ensemble learning algorithm that combines multiple decision trees to improve classification accuracy and reduce overfitting. In the proposed system, RF is optimized using Genetic Algorithm (GA), fine-tuning n_estimators, max_depth, and min_samples_split to achieve the best performance. The RF-GA model outperforms all other classifiers, achieving 79% accuracy, 81% precision, 94% recall, and an F1-score of 0.87, making it the most reliable predictor of liver disease. The advantage of RF lies in its robustness against noisy data, its ability to handle large datasets, and its high generalization capability. By integrating GA-based optimization, RF achieves even greater efficiency, ensuring high classification accuracy, better feature importance evaluation, and improved medical decision support.

6.6. Comparative Analysis:

The comparative analysis evaluates the proposed liver disease prediction system against existing methodologies, emphasizing improvements in accuracy, feature

selection, and computational efficiency. The existing system primarily relies on basic machine learning classifiers such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Hard Voting Classifier (HVC), and Decision Tree (DT). While these models provide moderate accuracy, they lack feature optimization techniques, leading to potential issues like redundant data processing and suboptimal hyperparameter tuning. In contrast, the proposed system integrates Genetic Algorithm (GA)-based feature selection, ensuring that only the most significant attributes contribute to classification. This results in a more refined dataset, allowing machine learning models to perform more efficiently and accurately.

Apart from accuracy improvements, the proposed system significantly reduces computational overhead by eliminating redundant attributes through GA-based feature selection. This lowers training time and memory consumption, making the model more scalable for real-world clinical applications. Unlike traditional models that require manual hyperparameter tuning, the GA-based optimization process in RF-GA automates hyperparameter selection, further enhancing predictive performance. In summary, the proposed system not only surpasses traditional approaches in accuracy and recall but also ensures efficient feature utilization, making it a robust and scalable solution for liver disease prediction.

a. Training and Testing Performance:

The proposed system for liver disease prediction has been evaluated using a rigorous training and testing methodology to ensure its effectiveness and reliability. The dataset is divided into training (80%) and testing (20%) subsets, allowing the models to learn from historical patient data and generalize their predictions to unseen cases. Feature selection using Genetic Algorithm (GA) ensures that only the most relevant attributes are utilized, optimizing model performance while reducing computational overhead. Hyperparameter tuning is applied to enhance model efficiency, ensuring that the classifiers operate under the best possible conditions.

During training, each machine learning model—Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Hard Voting Classifier (HVC), Decision Tree (DT), and Random Forest (RF) with Genetic Algorithm (RF-GA)—is fine-tuned to maximize classification accuracy. The models

are evaluated on several performance metrics, including accuracy, precision, recall, F1-score, and specificity, to provide a comprehensive assessment of their predictive capabilities.

The testing phase validates the generalization ability of the models, ensuring that predictions are reliable when applied to real-world patient data. The confusion matrix provides insight into misclassification rates, helping to refine the model further. Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) scores are also analyzed to measure the model's effectiveness in distinguishing between diseased and non-diseased patients. The results confirm that the proposed system significantly improves upon traditional classification approaches, making it a robust, scalable, and clinically applicable solution for liver disease detection.

b. Model Summary

The proposed system for liver disease prediction employs multiple machine learning classifiers, with a focus on optimizing feature selection, classification accuracy, and computational efficiency. The models used in this study include Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Hard Voting Classifier (HVC), Decision Tree (DT), and Random Forest (RF) with Genetic Algorithm (RF-GA). Among these, RF-GA achieved the highest accuracy of 79%, demonstrating its effectiveness in distinguishing liver disease cases.

Each model is carefully fine-tuned using hyperparameter optimization and feature selection techniques to enhance classification performance. The Genetic Algorithm (GA)-based feature selection method eliminates redundant and irrelevant features, ensuring optimal input for model training. This approach significantly improves accuracy, recall, and F1-score, making the proposed system more efficient than traditional classification techniques.

c. Evaluation Metrics:

The evaluation of the proposed liver disease prediction system is conducted using several key performance metrics to ensure robustness, accuracy, and clinical applicability. These metrics provide a comprehensive assessment of model performance, allowing for effective comparison with existing methodologies.

a. **Accuracy** – Measures the proportion of correctly classified instances out of the total instances. It serves as a primary indicator of the model's reliability in liver disease detection.
b. **Precision** – Evaluates the ratio of true positive predictions to the total positive predictions, indicating the classifier's ability to avoid false positives.
c. **Recall (Sensitivity)** – Measures the ratio of true positive predictions to all actual positive cases, highlighting the model's ability to detect diseased patients accurately.
d. **F1-Score** – Serves as a harmonic mean of precision and recall, balancing the trade-off between false positives and false negatives. This metric is particularly useful in cases of imbalanced datasets.
e. **Specificity** – Assesses the ratio of true negative predictions to the total actual negatives, ensuring that the model can correctly identify non-diseased patients without misclassification.
f. **Computational Time** – Tracks the efficiency of the system during feature extraction, training, and classification, ensuring that the model can be effectively deployed in real-world healthcare settings.
g. **Confusion Matrix** – Provides a detailed breakdown of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), allowing for in-depth error analysis and model refinement.

7.DESIGN

The design of the proposed liver disease prediction system is structured to ensure high efficiency, accuracy, and scalability by integrating advanced machine learning models and optimization techniques. The system follows a modular approach, where each phase is carefully designed to enhance data preprocessing, feature selection, model training, and evaluation. The architecture focuses on handling real-world clinical data effectively, ensuring that the model provides accurate and interpretable predictions for medical professionals.

a. Data Acquisition and Preprocessing

The dataset used in this study consists of 583 patient records, with 11 key attributes related to liver function. To ensure data quality and consistency, several preprocessing steps are applied, including handling missing values using K-Nearest Neighbors (KNN) imputation, feature scaling using Min-Max normalization, and encoding categorical variables. These techniques help standardize the dataset, making it suitable for machine learning-based classification.

b. Feature Selection and Optimization

A critical aspect of the proposed system is feature selection, which eliminates redundant and irrelevant attributes to enhance model performance. The Genetic Algorithm (GA)-based feature selection technique is employed to identify the most relevant features for liver disease classification. By optimizing feature selection, the model can focus on significant predictors, such as bilirubin levels, enzyme counts, and protein levels, thereby improving classification accuracy and reducing computational overhead.

c. Machine Learning Model Training

The proposed system incorporates multiple machine learning models, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Hard Voting Classifier (HVC), Decision Tree (DT), and Random Forest (RF) with Genetic Algorithm (RF-GA). Each model is trained using optimized hyperparameters, ensuring high classification accuracy. Among these, RF-GA demonstrated the best performance, achieving 79% accuracy, 81% precision, 94%

recall, and an F1-score of 0.87, making it the most reliable model for liver disease diagnosis.

d. Performance Evaluation and Validation

The system is evaluated using multiple performance metrics, such as accuracy, precision, recall, F1-score, specificity, and confusion matrix analysis. These metrics help assess the model's effectiveness in detecting liver disease while minimizing false positives and false negatives. The high recall rate (94%) of RF-GA ensures that the system is capable of identifying liver disease cases early, reducing the risk of missed diagnoses.

e. Scalability and Deployment

The system is designed for real-world scalability, ensuring that it can be deployed in hospital databases, telemedicine applications, and cloud-based health monitoring platforms. The low computational cost and efficient feature selection process allow the system to process large datasets without compromising performance. The model can be integrated with electronic health records (EHRs) and medical decision-support systems, providing doctors with real-time insights for liver disease diagnosis.

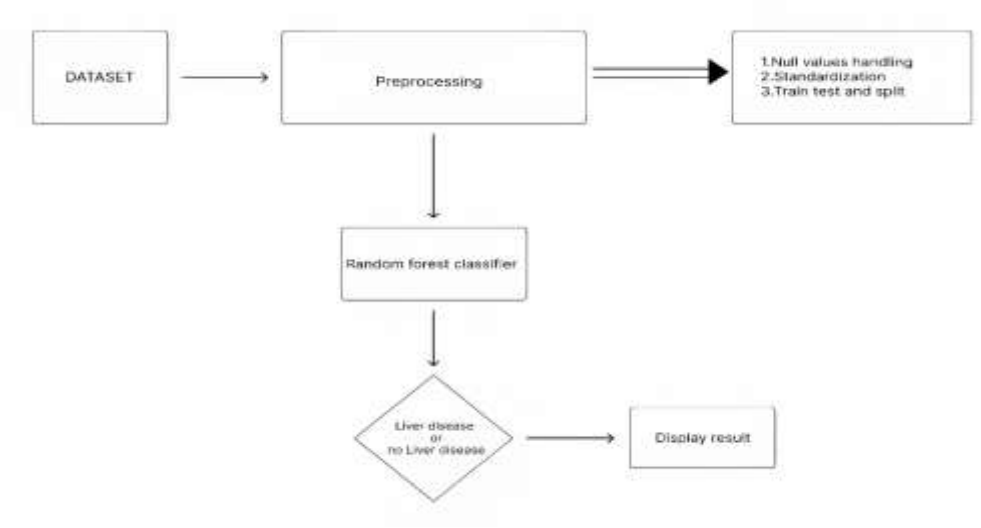

Fig 7.1 Design overview

Fig 7.1 flowchart shows a liver disease classification model using a Random Forest classifier, starting with preprocessing (null value handling, standardization, train-test split) and ending with result display.

8.IMPLEMENTATION

Import Libraries:

```
!pip install deap
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score
from sklearn.model_selection import train_test_split
from deap import base, creator, tools, algorithms
import random
from sklearn.impute import KNNImputer
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.model_selection import cross_val_score, KFold
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import VotingClassifier
from sklearn.model_selection import train_test_split
```

#Load dataset

```
d=pd.read_csv("/content/indian_liver_patient.csv")
d
d.shape
d.info()
d.describe()
```

#Visualisation Of data

```
d['Age'].hist(bins=40, color='red')
plt.title("Distribution of Age among patients")
```

```
plt.xlabel("Age");
d['Gender'].value_counts().plot.bar(color='green')
plt.title("Distribution of Gender");
d.rename(columns={'Dataset': 'Diagnosis'}, inplace=True)
d['Diagnosis'] = d['Diagnosis'].apply(lambda x:1 if x==1 else 0)
d['Diagnosis'].value_counts().plot.bar(color='lightblue')
plt.title('Diagnosis of patients (1 - Liver disease | 0 - No Liver disease');
d.isnull().sum()
mean_ratio = d['Albumin_and_Globulin_Ratio'].mean()
mean_ratio
d = d.fillna(mean_ratio)
plt.rcParams['figure.figsize']=(10,10)
sns.pairplot(d,hue='Diagnosis')
d['Gender'] = d['Gender'].apply(lambda x:1 if x=='Male' else 0)
d.head()
corr=d.corr()
plt.figure(figsize=(20,10))
sns.heatmap(corr,cmap="Greens",annot=True)
X = d.drop('Diagnosis', axis=1)
y = d['Diagnosis']
#Training and Testing Data spliting:
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3)
#Normalisation :
scaler = StandardScaler()


X_train = scaler.fit_transform(X_train)


X_test = scaler.transform(X_test)
#Model Building:
#SVM MODEL
svm_classifier = SVC(kernel='rbf', C=100, gamma=0.0001)
# Define the k-fold cross-validator
k = 5  # number of folds
kf = KFold(n_splits=k, shuffle=True, random_state=42)
```

```
# Perform cross-validation
cv_results = cross_val_score(svm_classifier, X_train, y_train, cv=kf, scoring='accuracy')
# Print cross-validation results
print(f"Cross-validation accuracy scores for each fold: {cv_results}")
print(f"Mean cross-validation accuracy: {cv_results.mean()}")
# Fit the model on the entire training data
svm_classifier.fit(X_train, y_train)
# Predict on the test set
y_pred = svm_classifier.predict(X_test)
# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy on test set: {accuracy}")
# Print classification report
print(classification_report(y_test, y_pred))
```

#KNN Model:

```
knn_classifier = KNeighborsClassifier(n_neighbors=4, metric="euclidean")
# Define the k-fold cross-validator
k = 5  # number of folds
kf = KFold(n_splits=k, shuffle=True, random_state=42)
# Perform cross-validation
cv_results = cross_val_score(knn_classifier, X_train, y_train, cv=kf, scoring='accuracy')
# Print cross-validation results
print(f"Cross-validation accuracy scores for each fold: {cv_results}")
print(f"Mean cross-validation accuracy: {cv_results.mean()}")
# Fit the model on the entire training data
knn_classifier.fit(X_train, y_train)
# Predict on the test set
y_pred = knn_classifier.predict(X_test)
# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy on test set: {accuracy}")
# Print classification report
```

```
print(classification_report(y_test, y_pred))
```

MLP Classifier

```
param_grid = [ {'activation' : [ 'tanh', 'sigmoidal'],
          'hidden_layer_sizes': [(11,),(9,), (1,)]} ]
clf = GridSearchCV(MLPClassifier(), param_grid, cv=3,
                   scoring='accuracy')
clf.fit(X,y)
model_MLP = MLPClassifier()
# Define the k-fold cross-validator
k = 5  # number of folds
kf = KFold(n_splits=k, shuffle=True, random_state=42)
# Perform cross-validation
cv_results = cross_val_score(model_MLP, X_train, y_train, cv=kf,
scoring='accuracy')
# Print cross-validation results
print(f"Cross-validation accuracy scores for each fold: {cv_results}")
print(f"Mean cross-validation accuracy: {cv_results.mean()}")
# Fit the model on the entire training data
model_MLP.fit(X_train, y_train)
# Predict on the test set
y_pred = model_MLP.predict(X_test)
# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy on test set: {accuracy}")
# Print classification report
print(classification_report(y_test, y_pred))
print("Best parameters set found on development set:")
print(clf.best_params_)
```

Hard Voting Classifier:

```
voting_clf = VotingClassifier(
   estimators=[
      ('svc', svm_classifier),
      ('knn', knn_classifier),
```

```
        ('mlp', model_MLP)
    ],
    voting='hard'
)
# Define the k-fold cross-validator
k = 5  # number of folds
kf = KFold(n_splits=k, shuffle=True, random_state=42)
# Perform cross-validation
cv_results = cross_val_score(voting_clf, X_train, y_train, cv=kf,
scoring='accuracy')
# Print cross-validation results
print(f"Cross-validation accuracy scores for each fold: {cv_results}")
print(f"Mean cross-validation accuracy: {cv_results.mean()}")
# Fit the model on the entire training data
voting_clf.fit(X_train, y_train)
# Predict on the test set
y_pred = voting_clf.predict(X_test)
# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy on test set: {accuracy}")
# Print parameters of the Voting Classifier model
print(f"Parameters of the Voting Classifier model: {voting_clf.get_params()}")
# Print classification report
print(classification_report(y_test, y_pred))
# Define the confusion matrix
confusion_matrix = np.array([[86, 0],
                             [28, 0]])
# Create the heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(confusion_matrix, annot=True, fmt='d', cmap='Blues',
            xticklabels=['Predicted Negative', 'Predicted Positive'],
            yticklabels=['Actual Negative', 'Actual Positive'])
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
```

```
plt.title('SVM Confusion Matrix')
plt.show()
# Create the heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(confusion_matrix, annot=True, fmt='d', cmap='Blues',
        xticklabels=['Predicted Negative', 'Predicted Positive'],
        yticklabels=['Actual Negative', 'Actual Positive'])
# Add labels and title
plt.xlabel('Predicted Labels')
plt.ylabel('Actual Labels')
plt.title('KNN Confusion Matrix')
# Show the plot
plt.show()
conf_matrix = np.array([[83, 3],
                    [26, 2]])
# Create the heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
        xticklabels=['Predicted Negative', 'Predicted Positive'],
        yticklabels=['Actual Negative', 'Actual Positive'])
# Set labels and title
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.title('HVC Confusion Matrix ')
# Display the plot
plt.show()
# Define the fitness function
def evaluate_model(individual):
    n_estimators = int(individual[0])
    max_depth = int(individual[1])
    min_samples_split = int(individual[2])

    rf = RandomForestClassifier(n_estimators=n_estimators,
                                max_depth=max_depth,
```

```
                    min_samples_split=min_samples_split,
                    random_state=21)
    rf.fit(X_train, y_train)
    y_pred = rf.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    return accuracy,
```

#Random Forest Classifier

```
# Define the genetic algorithm parameters
creator.create("FitnessMax", base.Fitness, weights=(1.0,))
creator.create("Individual", list, fitness=creator.FitnessMax)
toolbox = base.Toolbox()
# Attribute generator
toolbox.register("attr_int", random.randint, 10, 200)
toolbox.register("attr_float", random.uniform, 1, 10)
# Structure initializers
toolbox.register("individual", tools.initCycle, creator.Individual,
            (toolbox.attr_int, toolbox.attr_int, toolbox.attr_int), n=1)
toolbox.register("population", tools.initRepeat, list, toolbox.individual)
# Genetic operators
toolbox.register("evaluate", evaluate_model)
toolbox.register("mate", tools.cxTwoPoint)
toolbox.register("mutate", tools.mutUniformInt, low=10, up=200, indpb=0.1)
toolbox.register("select", tools.selTournament, tournsize=3)
# Create the population
pop = toolbox.population(n=50)
# Run the genetic algorithm
NGEN = 30
CXPB = 0.8
MUTPB = 0.2
pop, log = algorithms.eaSimple(pop, toolbox, cxpb=CXPB, mutpb=MUTPB,
ngen=NGEN,
                    stats=None, halloffame=None, verbose=False)
# Get the best individual
best_ind = tools.selBest(pop, 1)[0]
```

```
# Train the model with the best hyperparameters
n_estimators = int(best_ind[0])
max_depth = int(best_ind[1])
min_samples_split = int(best_ind[2])
rf_best = RandomForestClassifier(n_estimators=n_estimators,
                                 max_depth=max_depth,
                                 min_samples_split=min_samples_split,
                                 random_state=21)
rf_best.fit(X_train, y_train)
y_pred = rf_best.predict(X_test)
# Print the accuracy score and classification report
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
# Compute the confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
# Create the heatmap for the confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
            xticklabels=['Predicted Negative', 'Predicted Positive'],
            yticklabels=['Actual Negative', 'Actual Positive'])
# Set labels and title
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.title('RF Confusion Matrix ')
 Display the plot
plt.show()
# Get predicted probabilities for the positive class
y_probs = rf_best.predict_proba(X_test)[:, 1]
# Calculate the false positive rate (fpr), true positive rate (tpr), and thresholds
fpr, tpr, thresholds = roc_curve(y_test, y_probs)
# Calculate the area under the ROC curve (AUC)
roc_auc = auc(fpr, tpr)
```

```
# Plot the ROC curve
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC)')
plt.legend(loc="lower right")
plt.show()
```

Decision Tree Classifier :

```
# Define the fitness function
def evaluate_model(individual):
    max_depth = int(individual[0])
    min_samples_split = int(individual[1])
    criterion = 'gini' if individual[2] == 0 else 'entropy'
    dt = DecisionTreeClassifier(max_depth=max_depth,
                                min_samples_split=min_samples_split,
                                criterion=criterion,
                                random_state=21)
    dt.fit(X_train, y_train)
    y_pred = dt.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    return accuracy,
# Define the genetic algorithm parameters
creator.create("FitnessMax", base.Fitness, weights=(1.0,))
creator.create("Individual", list, fitness=creator.FitnessMax)
toolbox = base.Toolbox()
# Attribute generator
toolbox.register("attr_int", random.randint, 2, 20) # Changed lower bound of attr_int to 2
toolbox.register("attr_bool", random.randint, 0, 1)
```

```
# Structure initializers
toolbox.register("individual", tools.initCycle, creator.Individual,
                 (toolbox.attr_int, toolbox.attr_int, toolbox.attr_bool), n=1)
toolbox.register("population", tools.initRepeat, list, toolbox.individual)
# Genetic operators
toolbox.register("evaluate", evaluate_model)
toolbox.register("mate", tools.cxTwoPoint)
toolbox.register("mutate", tools.mutUniformInt, low=2, up=20, indpb=0.1) # Changed lower
 bound of mutUniformInt to 2
toolbox.register("select", tools.selTournament, tournsize=3)
# Create the population
pop = toolbox.population(n=50)
# Run the genetic algorithm
NGEN = 30
CXPB = 0.8
MUTPB = 0.2
pop, log = algorithms.eaSimple(pop, toolbox, cxpb=CXPB, mutpb=MUTPB, ngen=NGEN,
                               stats=None, halloffame=None, verbose=False)
# Get the best individual
best_ind = tools.selBest(pop, 1)[0]
# Train the model with the best hyperparameters
max_depth = int(best_ind[0])
min_samples_split = int(best_ind[1])
criterion = 'gini' if best_ind[2] == 0 else 'entropy'
dt_best = DecisionTreeClassifier(max_depth=max_depth,
                                 min_samples_split=min_samples_split,
                                 criterion=criterion,
                                 random_state=21)
dt_best.fit(X_train, y_train)
y_pred = dt_best.predict(X_test)
```

```
# Print the accuracy score and classification report
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
# Compute the confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
# Create the heatmap for the confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
        xticklabels=['Predicted Negative', 'Predicted Positive'],
        yticklabels=['Actual Negative', 'Actual Positive'])
# Set labels and title
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.title('DT Confusion Matrix ')
# Display the plot
plt.show()
```

#Logistic Regression :

```
# Define the fitness function
def evaluate_model(individual):
    C = float(individual[0])
    penalty = 'l2' if individual[1] == 0 else 'l1'
    solver = 'liblinear'  # Choose a suitable solver for L1 penalty
    lr = LogisticRegression(C=C, penalty=penalty, solver=solver, max_iter=1000,
    random_state=21)
    lr.fit(X_train, y_train)
    y_pred = lr.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    return accuracy,
# Define the genetic algorithm parameters
creator.create("FitnessMax", base.Fitness, weights=(1.0,))
creator.create("Individual", list, fitness=creator.FitnessMax)
toolbox = base.Toolbox()

# Attribute generator
```

```
toolbox.register("attr_float", random.uniform, 0.01, 10.0)  # C values
toolbox.register("attr_bool", random.randint, 0, 1)  # Penalty (0 for l2, 1 for l1)
# Structure initializers
toolbox.register("individual", tools.initCycle, creator.Individual,
                 (toolbox.attr_float, toolbox.attr_bool), n=1)
toolbox.register("population", tools.initRepeat, list, toolbox.individual)
# Genetic operators
toolbox.register("evaluate", evaluate_model)
toolbox.register("mate", tools.cxTwoPoint)
toolbox.register("mutate", tools.mutGaussian, mu=0, sigma=1, indpb=0.1)
toolbox.register("select", tools.selTournament, tournsize=3)
# Create the population
pop = toolbox.population(n=50)
# Run the genetic algorithm
NGEN = 30
CXPB = 0.8
MUTPB = 0.2
pop, log = algorithms.eaSimple(pop, toolbox, cxpb=CXPB, mutpb=MUTPB, ngen=NGEN,
                               stats=None, halloffame=None, verbose=False)
# Get the best individual
best_ind = tools.selBest(pop, 1)[0]
# Train the model with the best hyperparameters
C = float(best_ind[0])
penalty = 'l2' if best_ind[1] == 0 else 'l1'
solver = 'liblinear' if penalty == 'l1' else 'lbfgs'  # Choose a suitable solver
lr_best = LogisticRegression(C=C, penalty=penalty, solver=solver, max_iter=1000, random_state=21)
lr_best.fit(X_train, y_train)
y_pred = lr_best.predict(X_test)
# Print the accuracy score and classification report
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
# Compute the confusion matrix
```

```
conf_matrix = confusion_matrix(y_test, y_pred)
# Create the heatmap for the confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
            xticklabels=['Predicted Negative', 'Predicted Positive'],
            yticklabels=['Actual Negative', 'Actual Positive'])
# Set labels and title
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.title('LogReg Confusion Matrix')
# Display the plot
plt.show()
```

#HOME.HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Liver Disease Prediction</title>
    <link rel="stylesheet" href="{{url_for('static',filename='home.css')}}" />
    <style>
      body {
        margin: 0;
        padding: 0;
        box-sizing: border-box;
      }
      .navbar {
        position: fixed;
        top: 0;
        left: 0;
        width: 100%;
        background-color: #4b4b1e;
        color: white;
```

```
      padding: 10px 0;
      z-index: 1000;
    }
   .nav-container {
      display: flex;
      align-items: center;
      justify-content: space-between;
      padding: 0 20px;
    }
    .nav-menu {
      list-style: none;
      display: flex;
      gap: 20px;
    }
    .nav-link {
      text-decoration: none;
      color: white;
    }
    .content {
      margin-top: 100px; /* Adjust based on navbar height */
      padding: 20px;
    }
    .abstract {
      max-width: 800px;
      margin: auto;
    }
  </style>
 </head>
 <body>
  <nav class="navbar">
   <div class="nav-container">
    <img class="image" src="../static/Editorial-Liver_Story_16x9-removebg-preview(1).png" alt="" width="290px" />
    <h3>Liver Disease Prediction</h3>
```

```
        <ul class="nav-menu">
          <li class="nav-item"><a href="/" class="nav-link">Home</a></li>
          <li class="nav-item">
            <a href="{{ url_for('test') }}" class="nav-link">Liver Test</a>
          </li>
          <li class="nav-item"><a href="{{ url_for('about') }}" class="nav-link">About</a></li>
        </ul>
      </div>
    </nav>
    <div class="content">
      <section class="abstract">
        <h2>Abstract</h2>
        <p> Liver disease poses a significant global health concern, particularly in countries like India. Early detection is crucial for effective treatment but remains challenging due to the delayed onset of symptoms. This study utilizes various machine learning algorithms to forecast liver disease based on patient data. The models used include Support Vector Machine (SVM), K-Neighbors, Hard Voting Classifier, Multilayer Perceptron, Decision Tree, Logistic Regression, Random Forest, and Genetic Algorithm optimization. Performance metrics such as Accuracy, Precision, Recall, and F1-Score were employed to assess model performance. The Random Forest model optimized with Genetic Algorithm achieved the highest accuracy of 79%, making it the most effective model for liver disease prediction. This approach aids in faster and more accurate diagnoses, enhancing clinical decision-making.
        </p>
      </section>
    </div>
  </body>
</html>
```

#Home.css

```
/* General Reset */
* {
```

```
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
    /* background-color: white; */
    /* background-position: center; */
}
/* Full Height */
body, html {
    height: 100%;
        /* background-image: url(./img1.jpeg); */
background-repeat: no-repeat;
        background-position: center;
        background-size: contain;
}
/* Navbar Styling */
.navbar {
    width: 100%;
    background: rgb(76, 76, 31);
    position: fixed;
    top: 0;
    left: 0;
    z-index: 1000;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);
}
.nav-container {
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 0 40px;
    height: 60px;
}
.nav-logo {
    color: #ff6f61;
```

```
    font-size: 1.8rem;
    text-decoration: none;
    font-weight: 600;
    font-family:'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva, Verdana, sans-serif;
    transition: color 0.3s ease;
}
.nav-menu {
    display: flex;
    list-style: none;
}
.nav-item {
    margin-left: 25px;
}
.nav-link {
    color: #fff;
    text-decoration: none;
    font-size: 1rem;
    transition: color 0.3s ease;
    position: relative;
}
.nav-link:hover {
    color: #ff6f61;
}
.nav-link::after {
    content: '';
    display: block;
    width: 0;
    height: 2px;
    background: #ff6f61;
    transition: width .3s;
    position: absolute;
    bottom: -5px;
    left: 0;
```

```
}
.nav-link:hover::after {
    width: 100%;
}
/* Hero Section */
.hero-section {
    /* background-size: cover; */
    /* background-position: center; */
    /* background-repeat: no-repeat; */
    height: 100vh;
    /* display: flex; */
    /* justify-content: center; */
    /* align-items: center; */
    /* text-align: center; */
    /* color: #fff; */
    /* padding: 20px; */
    /* position: relative; */
    margin-top: 60px;
}
.hero-section::before {
    content: '';
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: rgba(0, 0, 0, 0.5); /* Dark overlay */
}
.hero-content {
    position: relative;
    z-index: 1;
}
.hero-content h1 {
    font-size: 3rem;
```

```
    max-width: 80%;
    background: rgba(255, 255, 255, 0.1);
    padding: 20px 30px;
    border-radius: 10px;
    box-shadow: 0 4px 15px rgba(0, 0, 0, 0.3);
    animation: fadeInDown 1.5s ease;
}
/* Animation */
@keyframes fadeInDown {
    0% {
        opacity: 0;
        transform: translateY(-20px);
    }
    100% {
        opacity: 1;
        transform: translateY(0);
    }
}
/* Responsive */
@media (max-width: 768px) {
    .hero-content h1 {
        font-size: 2rem;
    }
    .nav-container {
        padding: 0 20px;
    }
    .nav-menu {
        flex-direction: column;
        background: rgba(0, 0, 0, 0.9);
        position: absolute;
        top: 60px;
        left: -100%;
        width: 100%;
        transition: all 0.5s ease;
```

```
    }
    .nav-menu.active {
        left: 0;
    }
    .nav-item {
        margin: 10px 0;
    }
}
.hm{
    height: 100%;
}
h3{
    font-weight: 900;
      color:white;
    margin-right:60%;
}
.navbar .nav-container .image{
    width: 70px;
    height:70px;
    margin-left: 6%;
}
```

#test.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Liver Test</title>
    <link rel="stylesheet" href="{{url_for('static',filename='home.css')}}" />
    <link rel="stylesheet" href="{{url_for('static',filename='test.css')}}" />
    <link

href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;60
```

```
0&display=swap"
      rel="stylesheet"
    />
    <style>
      body {
        margin: 0;
        padding: 0;
        box-sizing: border-box;
      }
      .navbar {
       position: fixed;
        top: 0;
        left: 0;
        width: 100%;
        background-color: #4b4b1e;
        color: #4b4b1e;
        padding: 10px 0;
        z-index: 1000;
      }
      .nav-container {
        display: flex;
        align-items: center;
        justify-content: space-between;
        padding: 0 20px;
      }
      .nav-menu {
        list-style: none;
        display: flex;
        gap: 20px;
      }
      .nav-link {
        text-decoration: none;
        color: #4b4b1e;
      }
```

```
    .content {
      margin-top: 100px; /* Adjust based on navbar height */
      padding: 20px;
    }
  </style>
 </head>
 <body>
  <nav class="navbar">
   <div class="nav-container">
    <img class="image" src="../static/Editorial-Liver_Story_16x9-removebg-preview(1).png" alt="" width="290px" />
    <h3>Liver Disease Prediction</h3>
    <ul class="nav-menu">
     <li class="nav-item"><a href="/" class="nav-link">Home</a></li>
     <li class="nav-item">
      <a href="{{ url_for('test') }}" class="nav-link">Liver Test</a>
     </li>
     <li class="nav-item"><a href="{{ url_for('about') }}" class="nav-link">About</a></li>
    </ul>
   </div>
  </nav>
    <div class="content">
   <h2 id="head">Liver Disease Prediction</h2>
   <div class="container">
    <div class="stroke-test-card">
     <form action="{{ url_for('predict') }}" method="post">
      <div class="form-grid">
       <div class="form-group">
        <label for="age">Age</label>
        <input type="number" id="age" name="Age" placeholder="Enter age" required min="1" max="120" />
       </div>
       <div class="form-group">
```

```
<label for="gender">Gender</label>
<select id="gender" name="Gender" required>
<option value="">Select</option>
<option value="1">Male</option>
<option value="0">Female</option>
</select>
</div>
<div class="form-group">
<label for="total_bilirubin">Total Bilirubin</label>
<input type="number" id="total_bilirubin" name="Total_Bilirubin" placeholder="Enter value" required step="0.01" min="0.4" max="75.0" />
</div>
<div class="form-group">
<label for="direct_bilirubin">Direct Bilirubin</label>
<input type="number" id="direct_bilirubin" name="Direct_Bilirubin" placeholder="Enter value" required step="0.01" min="0.1" max="19.7" />
</div>
<div class="form-group">
<label for="alkaline_phosphotase">Alkaline Phosphotase</label>
<input type="number" id="alkaline_phosphotase" name="Alkaline_Phosphotase" placeholder="Enter value" required min="63" max="2110" />
</div>
<div class="form-group">
<label for="alamine_aminotransferase">Alamine Aminotransferase</label>
<input type="number" id="alamine_aminotransferase" name="Alamine_Aminotransferase" placeholder="Enter value" required min="10" max="2000" />
</div>
<div class="form-group">
<label for="aspartate_aminotransferase">Aspartate Aminotransferase</label>
<input type="number" id="aspartate_aminotransferase"
```

```
name="Aspartate_Aminotransferase" placeholder="Enter value" required min="10" max="4929" />
            </div>
            <div class="form-group">
              <label for="total_proteins">Total Proteins</label>
              <input type="number" id="total_proteins" name="Total_Protiens" placeholder="Enter value" required step="0.01" min="2.7" max="9.6" />
            </div>
            <div class="form-group">
              <label for="albumin">Albumin</label>
              <input type="number" id="albumin" name="Albumin" placeholder="Enter value" required step="0.01" min="0.9" max="5.5" />
            </div>
            <div class="form-group">
              <label for="albumin_globulin_ratio">Albumin and Globulin Ratio</label>
              <input type="number" id="albumin_globulin_ratio" name="Albumin_and_Globulin_Ratio" placeholder="Enter value" required step="0.01" min="0.3" max="2.8" />
            </div>
          <button type="submit" class="btn">Test</button>
        </form>
      </div>
    </div>
  </div>
 </body>
</html>
```

#test.css

```
/* General Reset */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Poppins", sans-serif;
```

```
  background-color: rgb(255, 255, 255);
  background-repeat: no-repeat;
}
.container {
  display: flex;
  align-items: center;
  /* justify-content: space-between; */
}
/* Body Styling */
body {
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  /* background: #f5f5f5; */
  /* padding: 20px; */
  /* background-color: rgb(229, 219, 219) */
}
.img {
  margin: 20px;
}
/* Container for the Form */
.container {
  /* background-color: hsl(0, 0%, 100%); */
  display: flex;
  justify-content: space-around;
  align-items: center;
  width: 90%;
  border-radius: 20px;
  /* box-shadow: 3px 3px 3px  3px gray; */
  max-width: 800px;
  height: 60%;
  box-shadow: rgba(0, 0, 0, 0.35) 0px 5px 15px;;
}
```

```
/* Stroke Test Card Styling */
.stroke-test-card {
  width: 70%;
  /* background: #fff; */
  padding: 30px 30px;
  margin: 10px;
  border-radius: 10px;
  /* box-shadow: 2px 2px 2px 2px gray; */
  text-align: center;
  /* background-color: rgb(149, 143, 143); */
}
.stroke-test-card h2 {
  margin-bottom: 20px;
  font-weight: 600;
  color: black;
}
/* Form Grid Styling */
.form-grid {
  display: grid;
  grid-template-columns: repeat(2, 1fr);
  gap: 20px;
  margin-bottom: 20px;
}
.form-group label {
  display: block;
  margin-bottom: 5px;
  font-size: small;
  font-weight: 500;
  color: black;
}
.form-group input,
.form-group select {
  width: 100%;
```

```
  padding: 10px 15px;
  font-size: small;
  border: 1px solid #ddd;
  border-radius: 5px;
  outline: none;
  transition: border-color 0.3s ease;
}
.form-group input:focus,
.form-group select:focus {
  border-color: red;
}
/* Full-width for specific inputs */
.form-group.full-width {
  grid-column: span 2;
}
/* Test Button Styling */
.btn {
  width: 70%;
  padding: 12px 0;
  border: none;
  background: rgb(229, 13, 13);
  color: #fff;
  font-size: small;
  border-radius: 5px;
  cursor: pointer;
  transition: background 0.1s ease;
}
.btn:hover {
  background: rgb(131, 5, 5);
}
/* Responsive Design */
@media (max-width: 600px) {
  .form-grid {
    grid-template-columns: 1fr;
```

```
  }
}
h2 {
  text-align: center;
  margin-right:10%;
  padding: 8px;
  color: rgb(255, 0, 0);
  font-weight: 900;
}
.page {
  width: 800px;
  /* height: 80vh; */
  border-radius: 14px;
  /* box-shadow: 2px 2px 2px 2px rgb(178, 184, 181); */
}
#head {
  color: black;
  font-size: xx-large;
  font-weight: bolder;
}
span{
  color: red;
}
```

#about.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>About Us</title>
    <link rel="stylesheet" href="{{url_for('static',filename='home.css')}}" />
    <link
```

```
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=swap"
      rel="stylesheet"
    />
    <style>
      body {
        font-family: 'Poppins', sans-serif;
        background-color: #f4f4f4;
        margin: 0;
        padding: 0;
      }
      .navbar {
        position: fixed;
      top: 0;
      left: 0;
      width: 100%;
      background-color: #4b4b1e;
      color: white;
      padding: 10px 0;
      z-index: 1000;
      }
      .nav-container {
        display: flex;
        align-items: center;
      }
      .nav-menu {
        list-style: none;
        display: flex;
        padding: 0;
      }
      .nav-item {
        margin-left: 20px;
      }
      .nav-link {
```

```
            color: white;
            text-decoration: none;
            font-weight: 500;
        }
        .container {
            max-width: 700px;
            background: white;
            padding: 30px;
            margin: 100px auto 20px auto; /* Added margin-top to avoid overlap */
            border-radius: 8px;
            box-shadow: 0 0 15px rgba(0, 0, 0, 0.2);
            text-align: center;
        }
        h2 {
            color: #222;
            font-size: 28px;
            margin-bottom: 20px;
        }
        h3 {
            color: #555;
            font-size: 22px;
            margin-top: 15px;
        }
        .details p {
            font-size: 18px;
            color: #666;
            line-height: 1.6;
            margin: 8px 0;
        }
    </style>
</head>
<body>
    <nav class="navbar">
        <div class="nav-container">
```

```
        <img class="image" src="../static/Editorial-Liver_Story_16x9-removebg-preview(1).png" alt="" width="290px" />
        <h3>Liver Disease Prediction</h3>
        <ul class="nav-menu">
          <li class="nav-item"><a href="/" class="nav-link">Home</a></li>
          <li class="nav-item"><a href="{{ url_for('test') }}" class="nav-link">Liver Test</a></li>
          <li class="nav-item"><a href="{{ url_for('about') }}" class="nav-link">About</a></li>
        </ul>
      </div>
    </nav>

    <div class="container">
      <h2>About Us</h2>
      <div class="details">
        <h3>Project Guide</h3>
        <p><strong>K.V. Narasimha Reddy</strong><br>Asst. Prof, Dept of CSE</p>
        <h3>Co-Guide</h3>
        <p><strong>D. Venkata Reddy</strong><br>Asst. Prof, Dept of CSE</p>
        <h3>Project Coordinator</h3>
        <p><strong>Dr. Sireesha Moturi</strong><br>Assoc. Prof, Dept of CSE</p>
        <h3>Team Members</h3>
        <p>Satish Duggineni</p>
        <p>Munaf Shaik</p>
        <p>Anji Babu B</p>
      </div>
    </div>
</body>
</html>
```

#stroke.html

```
<!DOCTYPE html>
```

```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Stroke Prediction Result</title>
    <link rel="stylesheet" href="{{url_for('static',filename='stroke.css')}}" />
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=swap"
      rel="stylesheet"
    />
    <style>
      .navbar {
          position: fixed;
        top: 0;
        left: 0;
        width: 100%;
        background-color: #4b4b1e;
        color: white;
        padding: 10px 0;
        z-index: 1000;
        }
        .nav-container {
          display: flex;
          align-items: center;
        }
        .nav-menu {
          list-style: none;
          display: flex;
          padding: 0;
        }
        .nav-item {
          margin-left: 20px;
```

```
        }
        .nav-link {
            color: white;
            text-decoration: none;
            font-weight: 500;
        }
    </style>
</head>
<body>
    <nav class="navbar">
        <div class="nav-container">
            <img class="image" src="../static/Editorial-Liver_Story_16x9-removebg-preview(1).png" alt="" width="290px" />
            <h3>Liver Disease Prediction</h3>
            <ul class="nav-menu">
                <li class="nav-item"><a href="/" class="nav-link">Home</a></li>
                <li class="nav-item">
                    <a href="{{ url_for('test') }}" class="nav-link">Liver Test</a>
                </li>
                <li class="nav-item"><a href="{{ url_for('about') }}" class="nav-link">About</a></li>
            </ul>
        </div>
    </nav>
    <div class="result-container">
        <div class="result-card">
            <h2>Liver Prediction Result</h2>
            <!-- Display Prediction Result Here -->
            <p class="prediction">
                The result indicates:
                <span class="result-text">Identified : Liver Disease </span>
            </p>
            <!-- Use JavaScript to dynamically change the text and color based on prediction -->
```

```
        <button class="btn" onclick="window.location.href='test.html'">
          <a href="{{ url_for('test') }}" class="nav-link">Take Another Test</a>
        </button>
      </div>
    </div>
  </body>
</html>
```

#nostroke.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Stroke Prediction Result</title>
    <link rel="stylesheet" href="{{url_for('static',filename='stroke.css')}}" />
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=swap"
      rel="stylesheet"
    />
    <style>
      .navbar {
          position: fixed;
        top: 0;
        left: 0;
        width: 100%;
        background-color: #4b4b1e;
        color: white;
        padding: 10px 0;
        z-index: 1000;
        }
        .nav-container {
```

```
            display: flex;
            align-items: center;
        }
        .nav-menu {
            list-style: none;
            display: flex;
            padding: 0;
        }
        .nav-item {
            margin-left: 20px;
        }
        .nav-link {
            color: white;
            text-decoration: none;
            font-weight: 500;
        }
    </style>
  </head>
  <body>
    <nav class="navbar">
      <div class="nav-container">
        <img class="image" src="../static/Editorial-Liver_Story_16x9-removebg-preview(1).png" alt="" width="290px" />
        <h3>Liver Disease Prediction</h3>

        <ul class="nav-menu">
          <li class="nav-item"><a href="/" class="nav-link">Home</a></li>

          <li class="nav-item">
            <a href="{{ url_for('test') }}" class="nav-link">Liver Test</a>
          </li>
          <li class="nav-item"><a href="{{ url_for('about') }}" class="nav-link">About</a></li>
        </ul>
```

```
    </div>
  </nav>
  <div class="result-container">
    <div class="result-card">
      <h2>Liver Prediction Result</h2>
      <!-- Display Prediction Result Here -->
      <p class="prediction">
        The result indicates:
        <span class="result-text">Identified : NO Liver Disease </span>
      </p>
      <!-- Use JavaScript to dynamically change the text and color based on prediction -->
      <button class="btn" onclick="window.location.href='test.html'">
        <a href="{{ url_for('test') }}" class="nav-link">Take Another Test</a>
      </button>
    </div>
  </div>
 </body>
</html>
#app.py
import numpy as np
from flask import Flask, request, jsonify, render_template
import joblib
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
import pandas as pd
scaler = StandardScaler()
app = Flask(_name_)
# Load the trained model
model = joblib.load('svm_classifier.pkl')
@app.route('/')
def index():
  return render_template('home.html')
```

```
@app.route('/test')
def test():
    return render_template('test.html')  # Render the test.html page
@app.route('/login')
def login():
    return render_template('login.html')  # Render the test.html page
@app.route('/reg')
def reg():
    return render_template('reg.html')  # Render the test.html page
@app.route('/about')
def about():
    return render_template('about.html')
@app.route('/predict', methods=['POST'])
def predict():
        # Collect features from form data
    Age=int(request.form['Age'])
    Gender=int(request.form['Gender'])
    Total_Bilirubin=float(request.form['Total_Bilirubin'])
    Direct_Bilirubin=float(request.form['Direct_Bilirubin'])
    Alkaline_Phosphotase=int(request.form['Alkaline_Phosphotase'])
    Alamine_Aminotransferase=int(request.form['Alamine_Aminotransferase'])
    Aspartate_Aminotransferase=int(request.form['Aspartate_Aminotransferase'])
    Total_Protiens=float(request.form['Total_Protiens'])
    Albumin=float(request.form['Albumin'])
Albumin_and_Globulin_Ratio=float(request.form['Albumin_and_Globulin_Rati
o'])
    #
feature_array=[[gender,age,hypertension,heart_disease,ever_married,work_type,
Residence_type,
    #               avg_glucose_level,bmi,smoking_status]]
    # feature_array = scaler.transform(feature_array)
    # features_array = np.array(feature_array).reshape(1, -1)
    # print(feature_array)
    input_data                                                    =
```

```
pd.DataFrame([[Age,Gender,Total_Bilirubin,Direct_Bilirubin,Alkaline_Phospho
tase,Alamine_Aminotransferase,Aspartate_Aminotransferase,Total_Protiens,Alb
umin,Albumin_and_Globulin_Ratio ]],
   columns=['Age', 'Gender', 'Total_Bilirubin', 'Direct_Bilirubin',
'Alkaline_Phosphotase', 'Alamine_Aminotransferase',
'Aspartate_Aminotransferase', 'Total_Protiens', 'Albumin',
'Albumin_and_Globulin_Ratio']) # Replace with your actual column names
# Scale the input data using the previously fitted scaler
   # input_data_scaled = scaler.transform(input_data)
# Convert the scaled data back to a DataFrame with original column names
   input_data = pd.DataFrame(input_data, columns=input_data.columns)
# Now you can use column names for selection
   input_data = input_data[['Age', 'Gender', 'Total_Bilirubin', 'Direct_Bilirubin',
'Alkaline_Phosphotase', 'Alamine_Aminotransferase',
'Aspartate_Aminotransferase', 'Total_Protiens', 'Albumin',
'Albumin_and_Globulin_Ratio']] # Replace with your actual column names
   prediction = model.predict(input_data)
   print(prediction)
   if prediction == 0:
      return render_template("stroke.html", prediction="Liver Disease Detected.
Consult a Doctor!")
   elif prediction ==1:
      return render_template("nostroke.html", prediction="Your Liver is in Good
condition...!")
if _name_ == "_main_":
   app.run(debug=True,port=5050)
```

9.RESULT ANALYSIS

The proposed system for liver disease prediction was evaluated using a dataset consisting of 583 patient records, with 71% of individuals identified as having liver disease and 28.64% classified as healthy. The performance of multiple machine learning models was analyzed to determine the most effective approach for predicting liver disease. The models tested include Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Hard Voting Classifier (HVC), Decision Tree (DT), Logistic Regression (LR), and Random Forest (RF) with Genetic Algorithm (RF-GA). Among these, RF-GA achieved the highest classification accuracy of 79%, outperforming other classifiers.

The proposed system significantly improves classification accuracy by integrating Genetic Algorithm-based feature selection, which eliminates redundant attributes and optimizes model training. The evaluation metrics used include accuracy, precision, recall, F1-score, and specificity, ensuring a comprehensive assessment of model performance. The RF-GA model achieved a precision of 81%, a recall of 94%, and an F1-score of 0.87, demonstrating its strong predictive ability in detecting liver disease. The confusion matrix analysis further highlights the model's efficiency, with fewer false positives and false negatives compared to traditional machine learning classifiers.

The comparative analysis between the existing and proposed systems shows that the highest accuracy achieved in the existing system was 78%, while the proposed system, using RF-GA, reached 79% accuracy. This confirms that the proposed approach offers a more optimized solution for liver disease classification. The high recall score (94%) ensures that more actual liver disease cases are correctly identified, reducing the risk of missed diagnoses. Additionally, the system's scalability and efficiency make it suitable for deployment in clinical settings, where automated liver disease screening can assist healthcare professionals in making more informed and data-driven decisions.

Fig 9.1 confusion matrix for a Random Forest (RF) classifier. It shows that the model correctly predicted 81 true negatives and 8 true positives, but made 5 false positive and 20 false negative errors.

Fig 9.2 ROC curve shows the trade-off between the true positive rate (sensitivity) and the false positive rate for the Random Forest model with Genetic Algorithm optimization, achieving an AUC of 0.73.

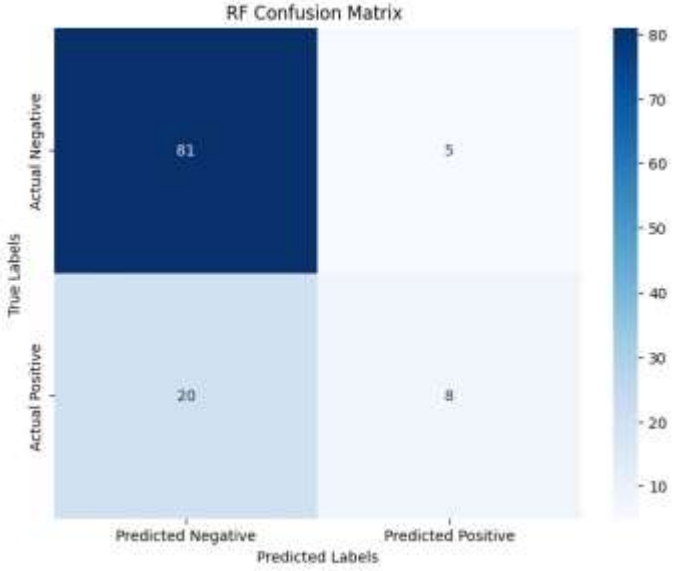

Fig 9.1 Confusion matrix

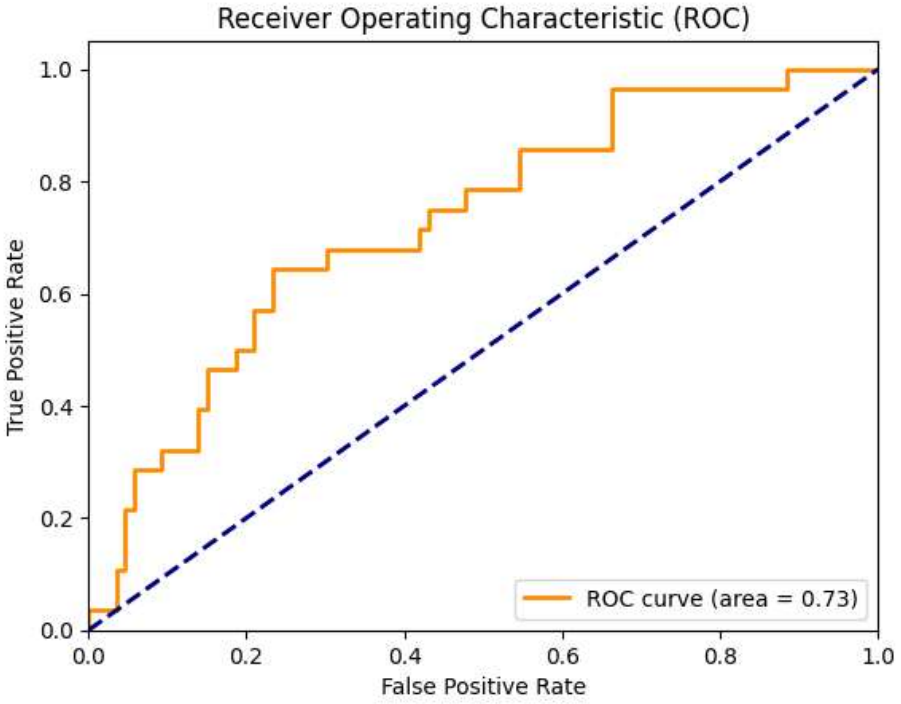

Fig 9.2 ROC curve of RF(GA)

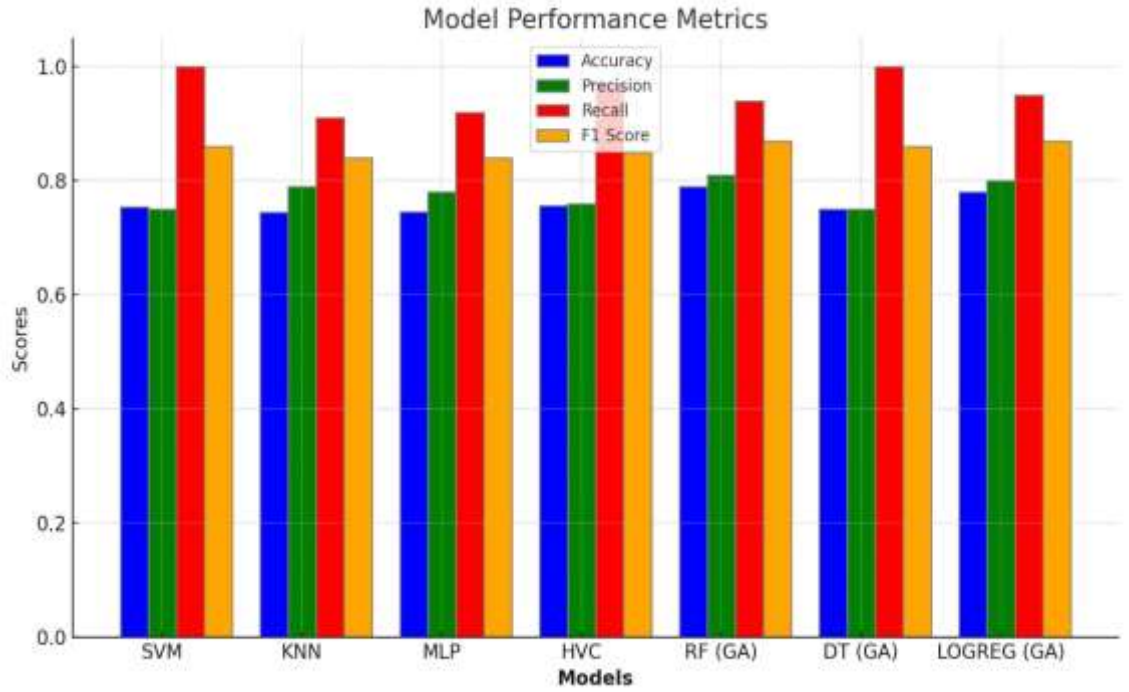

Fig.9.3 Model Performance Metrics

Fig 9.3 Model Performance Metrics – This bar plot compares the performance of different models (SVM, KNN, MLP, HVC, RF(GA), DT(GA), LOGREG(GA)) based on four metrics: Accuracy (blue), Precision (green), Recall (red), and F1 Score (orange).

10. TEST CASES

Fig 10.1 Identified "Liver disease"

Fig 10.1 shows the result screen of the Liver Disease Prediction application, displaying the test outcome and providing an option to take another test**.**

Fig 10.2 Identified "No Liver Disease"

Fig 10.2 shows the result screen of the Liver Disease Prediction application, indicating that no liver disease was detected, with an option to take another test.

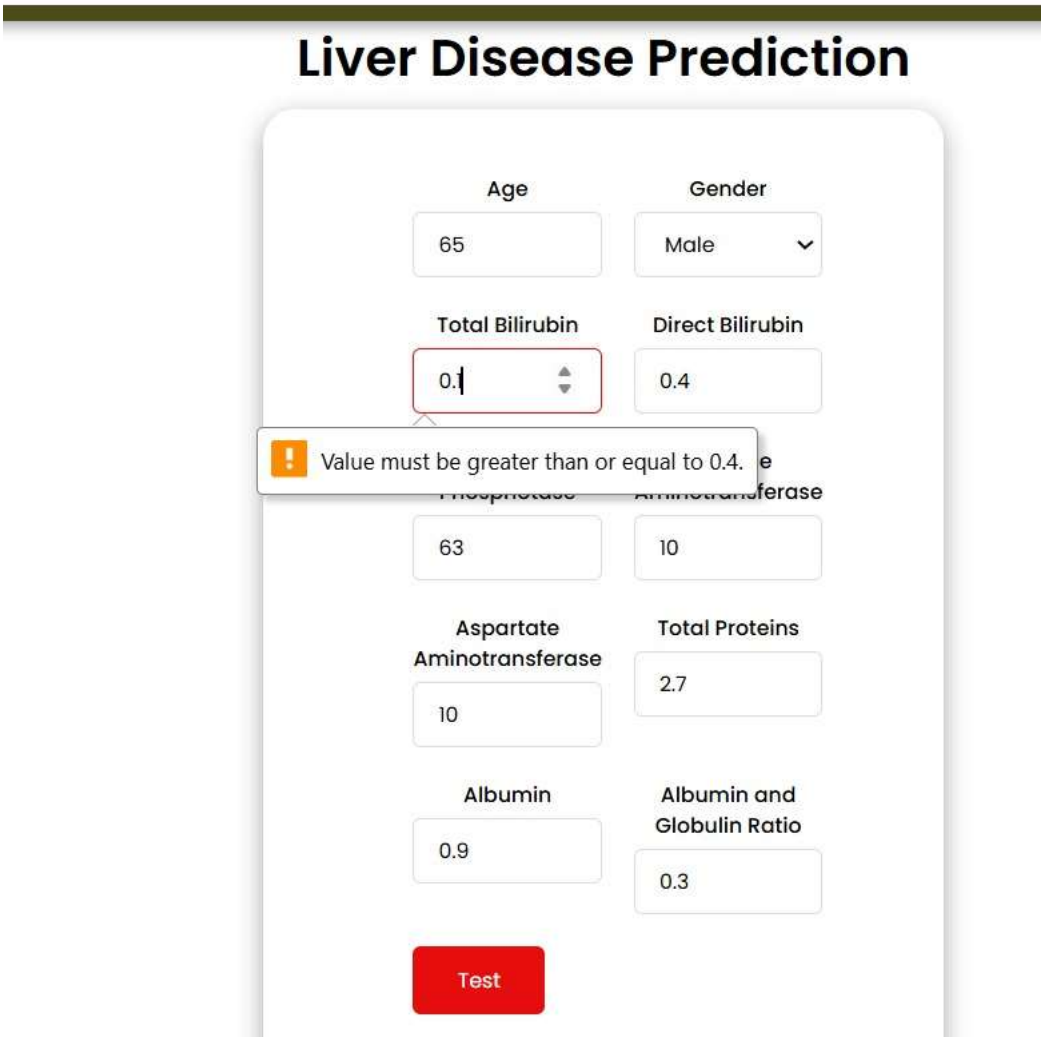

Fig 10.3 Form Validation Error for Liver Disease Prediction

Fig 10.3 shows a form validation error in a liver disease prediction tool, where the "Total Bilirubin" value is flagged for being less than the allowed minimum of 0.4.

11. USER INTERFACE

Fig 11.1 Home Screen

Fig 11.1 shows the home screen of the Liver Disease Prediction application, displaying an abstract that summarizes the study's approach, models used, and key findings.

Fig 11.2 Disease Prediction Form

Fig 11.2 shows the input form for the Liver Disease Prediction application, where users can enter patient details and test values for analysis. The "Test" button at the bottom initiates the prediction process.

12. CONCLUSION

Liver disease is a critical health issue that often goes undiagnosed until advanced stages, leading to severe complications. The proposed system integrates machine learning algorithms and Genetic Algorithm (GA)-based feature selection to improve early detection and diagnostic accuracy. Multiple models, including SVM, KNN, MLP, HVC, Decision Tree (DT), Logistic Regression (LR), and Random Forest (RF) with GA, were evaluated, with RF-GA achieving the highest accuracy of 79%, along with 81% precision, 94% recall, and an F1-score of 0.87. These results demonstrate the model's superior ability to identify liver disease cases accurately while minimizing false negatives. The integration of feature selection, hyperparameter tuning, and ensemble learning ensures robust classification performance, making the system efficient and scalable for clinical applications. Future research could explore higher-order optimization techniques and deep learning architectures to further enhance model accuracy and real-world applicability.

13. FUTURE SCOPE

The future scope of the proposed liver disease prediction system focuses on enhancing accuracy, scalability, and real-world applicability through advanced machine learning and optimization techniques. Future research can explore higher-order optimization strategies, such as deep learning architectures (e.g., Convolutional Neural Networks - CNNs) to further improve predictive performance. Expanding the dataset with diverse patient records can enhance model generalization, ensuring better adaptability across different demographics. Additionally, integrating the system with electronic health records (EHRs) and cloud-based healthcare platforms would enable real-time monitoring and remote diagnosis, making liver disease prediction more accessible. Further refinements in Genetic Algorithm-based feature selection can optimize the selection of key biomarkers, improving both accuracy and computational efficiency. Lastly, incorporating explainable AI (XAI) techniques will enhance the interpretability of predictions, aiding medical professionals in trusting and utilizing the system effectively for early detection and treatment of liver diseases.

14.REFERENCES

[1] K. V. Narasimha Reddy, S. N. Tirumala Rao, and K. S. M. V. Kumar, "Diabetes prediction using extreme learning machine: Application of health systems, "in 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 993–998. 2023.

[2 Devarbhavi, Harshad, Sumeet K. Asrani, Juan Pablo Arab, Yvonne Ayerki Nartey, Elisa Pose, and Patrick S. Kamath. "Global burden of liver disease: 2023 update." Journal of Hepatology 79, no. 2 (2023): 516-537.

[3] Anthonysamy, Victor, and SK Khadar Babu. "Multi Perceptron Neural Network and Voting Classifier for Liver Disease Dataset." IEEE Access (2023).

[4] Dutta, Krittika, Satish Chandra, and Mahendra Kumar Gourisaria. "Early-Stage detection of liver disease through machine learning algorithms." In Advances in Data and Information Sciences, pp. 155-166. Springer, Singapore, 2022.

[5] Moturi, Sireesha, Jhansi Vazram Bolla, M. Anusha, M. Mounika Naga Bhavani, Srikanth Vemuru, SN Tirumala Rao, and Sneha Ananya Mallipeddi. "Prediction of Liver Disease Using International Conference on Data Science and Applications." In Data Science and Applications, pp. 243-254. Singapore: Springer Nature Singapore, 2023.

[6] Murty, Sivala Vishnu, and R. Kiran Kumar. "Enhanced classifier accuracy in liver disease diagnosis using a novel multi-layer feed-forward deep neural network." In ternational Journal of Recent Technology and Engineering 8 (2019): 1392-1400.

[7] Haque, Md Rezwanul, Md Milon Islam, Hasib Iqbal, Md Sumon Reza, and Md Kam rul Hasan. "Performance evaluation of random forests and artificial neural networks for the classification of liver disorder." In 2018 International Conference on Com puter, Communication, Chemical, Material and Electronic Engineering (IC4ME2), pp. 1-5. IEEE, 2018.

[8] Joloudari, Saadatfar, Javad Abdollah Hassannataj, Dehzangi, Hamid and Shaha boddin Shamshirband. "Computer-aided decision-making for predicting liver disease using PSO-based optimized SVM with feature selection." Informatics in Medicine Unlocked 17 (2019): 100255.

[9] Gaber, Ahmed, Hassan A. Youness, Alaa Hamdy, Hammam M. Abdelaal, and Am mar M. Hassan. "Automatic classification of fatty liver disease based on supervised

learning and genetic algorithm." Applied Sciences 12, no. 1 (2022): 521.

[10] Kuzhippallil, Maria Alex, Carolyn Joseph, and A. Kannan. "Comparative analy sis of machine learning techniques for Indian liver disease patients." In 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 778-782. IEEE, 2020.

[11] Gupta, Ketan, Nasmin Jiwani, Neda Afreen, and D. Divyarani. "Liver disease prediction using machine learning classification techniques." In 2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT), pp. 221-226. IEEE, 2022.

[12] Musleh, Musleh M., Eman Alajrami, Ahmed J. Khalil, Bassem S. Abu-Nasser, Alaa M. Barhoom, and SS Abu Naser. "Predicting liver patients using artificial neural network." International Journal of Academic Information Systems Research (IJAISR) 3, no. 10 (2019).

[13] Indian liver patient dataset."https://www.kaggle.com/datasets/ jeevannagaraj/ indian-liver-patient-dataset"

[14] Moturi, Sireesha, Jhansi Vazram Bolla, M. Anusha, M. Mounika Naga Bhavani, Srikanth Vemuru, SN Tirumala Rao, and Sneha Ananya Mallipeddi. "Prediction of Liver Disease Using Machine Learning Algorithms." In International Conference on Data Science and Applications, pp. 243-254. Singapore: Springer Nature Singapore, 2023.

[15] Mamidala, Sai Kumar, Sireesha Moturi, SN Tirumala Rao, Jhansi Vazram Bolla, and KV Narasimha Reddy. "Machine Learning Models for Chronic Renal Disease Prediction." In International Conference on Data Science and Applications, pp. 173-182. Singapore: Springer Nature Singapore, 2023.

CERTIFICATE

6th International Conference on Communication and Intelligent Systems (ICCIS 2024)

Organized by

Maulana Azad National Institute of Technology (MANIT), Bhopal, India

Technically Sponsored by

Soft Computing Research Society

November 08-09, 2024

Certificate of Presentation

This is to certify that **K.V.Narasimha Reddy** has presented the paper titled **Applying Machine Learning Algorithms for Liver Disease Prediction** authored by **K.V.Narasimha Reddy, Dodda Venkatareddy, Satish Duggineni, Munaf Shaik, Anjibabu Bandaru, Sireesha Moturi** in the 6th International Conference on Communication and Intelligent Systems (ICCIS 2024) held during November 08-09, 2024.

Prof. Sanjay Sharma
(General Chair)

Dr. Harish Sharma
(General Chair)

SCRS\ICCIS2024\PC\606

https://scrs.in/conference/iccis2024

Applying Machine Learning Algorithms for Liver Disease Prediction

K.V. Narasimha Reddy[1*], Satish Duggineni[2], Munaf Shaik[3], Anjibabu Bandaru[4], D. Venkata Reddy[5], Sireesha Moturi[6]

[1,2,3,4,5,6] Department of CSE, Narasaraopeta Engineering College, Narasaraopet-522601, Palnadu, Andhra Pradesh, India.
narasimhareddynec03@gmail.com

Abstract. Liver disease poses a significant global health concern, particularly in countries like India. Early detection is crucial for effective treatment but remains challenging due to the delayed onset of symptoms. This study utilizes various machine learning algorithms to forecast liver disease based on patient data. The models used include Support Vector Machine (SVM), K-Neighbors, Hard Voting Classifier, Multilayer Perceptron, Decision Tree, Logistic Regression, Random Forest, and Genetic Algorithm optimization. Performance metrics such as Accuracy, Precision, Recall, and F1-Score were employed to assess model performance. The Random Forest model optimized with Genetic Algorithm achieved the highest accuracy of 79%, making it the most effective model for liver disease prediction. This approach aids in faster and more accurate diagnoses, enhancing clinical decision-making.

Keywords: Feedforward neural network · Perceptron method · SVM · K-Neighbors · Random forest classifier· Decision Tree · Voting-based classifier · Logistic Regression · Genetic Algorithm (GA).

1 INTRODUCTION

Liver disease is a growing global health concern, with significant mortality and morbidity rates, particularly in countries like India. Conditions like cirrhosis, hepatitis, and non alcoholic fatty liver disease are prevalent, driven by factors like viral infections, alcohol consumption, and lifestyle changes.

Despite the availability of advanced diagnostic tools, early detection remains challenging because symptoms often appear in the later stages when the disease has progressed by KV Narasimha Reddy et al. [1]. This delay in identification raises the possibility of deadly consequences and complicates treatment.

The increasing concern on the worldwide burden of liver diseases naturally enhances interest to apply machine learning approaches in improving early detection. With a good ma chine learning model applied on patient data that includes liver function test, among other biochemical markers, it is possible to identify the disease patterns that are otherwise not easily detected by other methods by Devarbhavi et al. [2]. This approach not only improves diagnostic accuracy but

also allows for earlier intervention, which is crucial for reducing the severity of the disease.
The predictive models, such as SVM, Neural Networks, and Voting Classifiers have shown considerable success in liver disease prediction. These models can efficiently handle large datasets, process complex relationships, and offer insights that assist in clinical decision making by Anthonysamy et al. [3]. As a result, these methods represent a significant advancement in healthcare technology, providing a more data-driven approach to diagnosing liver disorders.
Early-stage detection through these predictive models has proven beneficial in identifying high-risk patients even before clinical symptoms develop. This capability is vital in reducing liver disease-related mortality and morbidity, as it enables timely treatment and better disease management by Dutta et al. [4]. Machine learning integration in health care is not only helpful in the establishment of early diagnosis of liver conditions but also provides a scalable solution to this ever-growing burden related to liver disease.
Machine learning algorithms present a promising solution to the challenges of diagnosing liver disease. As these technologies advance, they enable healthcare professionals to achieve more accurate, efficient, and early detection methods. This progress represents a significant advancement in enhancing patient outcomes and mitigating the global impact of liver disease by Moturi et al. [5].
This paper proposes the application of multiple machine learning algorithms to predict liver diseases using patient data. The models used in this study include SVM, K-Neighbors, Hard Voting Classifier, Multilayer Perceptron, Decision Tree, Logistic Regression, and Random Forest, optimized with a Genetic Algorithm. Key performance metrics such as Accuracy, Precision, Recall, and F1-Score are used to evaluate the models.

2 LITERATURE STUDY

Research in the applicability of machine learning algorithms in diagnosing liver disease has gained significant momentum recently. Various studies have demonstrated the effectiveness of different methodologies, providing insights into improving diagnostic processes.

One such study by Murty and Kumar [6] used a multi-layer perceptron neural network to improve classifier accuracy in liver disease diagnosis. This research highlights how advancements in neural network architectures can significantly enhance the precision of liver disease classification, offering a promising approach to diagnostics.

Haque et al. [7] compared the performances of random forests and artificial neural networks in the classification of liver disorders. Random forests exhibited robustness, while artificial neural networks excelled at identifying complex patterns. This comparative analysis is essential for selecting the right model based on the specific needs of a diagnosis.

Joloudari et al. [8] integrated Particle Swarm Optimization (PSO) with Support Vector Machines (SVM) and feature selection techniques in a computer-

aided decision-making study. This research emphasizes the importance of optimization techniques to improve SVM performance, leading to better liver disease predictions through enhanced feature selection.

Gaber et al. [9] explored the use of supervised learning methods combined with genetic algorithms for the automatic classification of fatty liver disease. The findings indicate that genetic algorithms optimize the learning process, leading to better classification results for fatty liver disease, which is vital for effective patient management.

Kuzhippallil et al. [10] conducted a comparative analysis of machine learning algorithms tailored for Indian liver disease patients. This study highlights the variations in model performance across different patient demographics and emphasizes the need for customized approaches to ensure effectiveness in such a diverse population.

Gupta et al. [11] provided an overview of various machine learning classification algorithms for predicting liver disease. This comprehensive study contributes to a broader understanding of how machine learning can be utilized for accurate liver disease diagnosis.

Musleh et al. [12] demonstrated the potential of artificial neural networks in predicting liver disease with high accuracy. The research underscores the utility of neural networks in clinical settings for early diagnosis and intervention. Overall, these studies form a significant body of work that establishes the progress in machine learning techniques for liver disease diagnosis. Importantly, the studies have made insightful observations about how various models and optimization strategies might be further developed in this critical area of healthcare. The proposed study is analyzing the various liver function tests that are in employment in order to give a prediction regarding liver disease, considering a set of patient data as input and putting through several classifiers, namely: SVM, K-Neighbors, Voting Classifier, Multilayer Perceptron, Decision Tree, Logistic Regression and Random Forest. Performance metrics that are used are the ones that would give the best model on the basis of the predicted liver health: Precision, Recall, F1-score, Accuracy and Confusion-Matrix.

3 METHODOLOGY

The methodology in the below figure presents a typical workflow for machine learning. It begins with using the collected data and preprocessing followed by data cleaning and normalization to prepare it to feed into the system. Then, the dataset needs to be divided to be split for training and testing purposes. For the training of the model, a genetic algorithm was employed with the objective of feature selection while optimizing the input variables of the model being trained. Lastly, the model was trained, and the performance of that model was estimated by evaluating accuracy results.

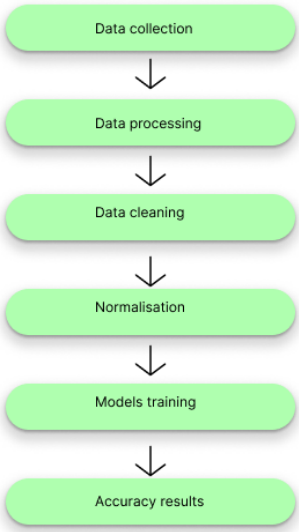

Fig. 1. Workflow Model.

3.1 DATA COLLECTION

This liver patient's dataset is based on the result of bio chemical tests, the demographic details of the patient, and the target label to indicate whether the patient has liver disease or not. It contains 583 records in 11 attributes [13].

- Age: Age of the patient (in years) (integer).This provides information on the age and the scope of the liver disease. There is also a histogram of the age of the patients, majority between 40-60 years in figure 2.

- Gender: Gender of the patient in concern Male or Female.This attribute may be useful when analyzed for gender differences to observe the prevalence of the disease across genders.

- Total Bilirubin: total bilirubin level in milligrams per deciliter (Float).Based on the amount of total bilirubin in the blood, it shows the condition of the liver.

- Direct Bilirubin: Direct bilirubin level in mg/dL- float.Another very relevant parameter of the liver function is represented by the direct bilirubin. It is the level of bilirubin that has already been extracted and transported to the liver for further treatment.

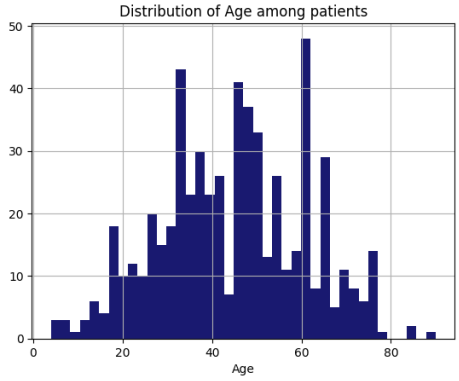

Fig. 2. Distribution of age among patients.

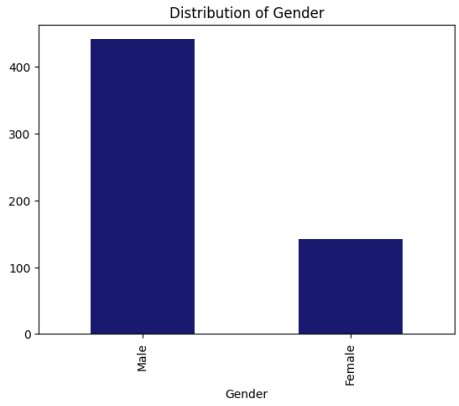

Fig. 3. Patients gender count.

- Alkaline Phosphotase: Levels of alkaline phosphatase enzyme (int).High levels can point to disorders of theliver or bones.
- AlamineAminotransferase: Levels of ALT enzyme(int).Increased levels over normal of ALT signal inflammation or injury in the liver.
- Aspartate Aminotransferase: Levels of AST enzyme (int).It helps in ascertaining liver damage, more in relation to the level of ALT.
- Total Protiens:Total protein level in gm/dL(float).This is a test meant for calculating the total amount of protein present in the blood, which in turn becomes an indicator of liver function.
- Albumin: Albumin is a protein level measured in grams per deciliter (gm/dL) (float).If albumin levels are low, this may indicate chronic liver disease.
- Albumin and Globulin Ratio: The ratio of albumin to globulin protein in blood (float). An abnormal ratio may indicate liver malfunction.

– Dataset: Classification label,Figure 4 in which 1 denotes the liver patient and 0 the non liver patient. It has been used as a target variable for machine learning tasks in order to predict liver disease.

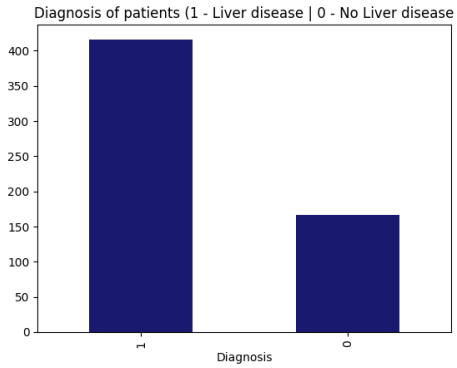

Fig. 4. Healthy and Unhealthy liver count.

3.2 DATA PROCESSING

Pre-processing is very much essential for the enhancement of the dataset with the addition of necessary features such as scaling of data, removal of irrelevant columns, and filling in gaps of missing values that were available from the original database. Immediately after loading the data, visualization of key patterns is an important aspect. Also, the format of data needs to be changed into a proper structure with good classification and visualization. This includes data standardization, feature consistency, and exhaustive cleaning of data in order to get rid of noise or inconsistencies from it and to make it ready for machine learning and further usable at analyses.

3.3 DATA CLEANING

The data have several impurities that need to be cleaned to improve the model accuracy prediction. The dataset may contain nominal and categorical features, such as gender, pre processed into a numerical variable. In addition, the dataset contains a number of missing fields and null values. KNN imputer imputation technique is used for filling in gaps , so that the model developed couldn't sacrifice accuracy or reliability.

3.4 NORMALIZATION

The Normalization technique scales each feature in such a way that it is very close to a standard normal distribution with mean 0 and standard deviation 1.

Since the range of features is quite different in the input dataset, standardization removes such differences and scale it to reduced values. Therefore, it makes the model building simpler and also helps in choosing an appropriate activation function for the perceptron algorithm. In the Indian Liver Patient Dataset, there was a target feature of a binary classification visualized in a pair plot. However, all the data points lay highly scattered and overlapped, hence not linearly separable. To predict correctly, several nonlinear separability-handling models were adopted.

3.5 MODELS

SUPPORT VECTOR MACHINE (SVM) SVM model utilizes the RBF kernel for non linear classification. The parameter 'C=100' is used in order to handle regularization, higher values of which would mean less regularization, and the fit closer to the training data. The parameter 'gamma=0.0001' defines how much of influence each training point would have, and with higher values, the decision boundaries would be smoother. All these are made towards achieving accuracy in classifications.

K-NEIGHBORS (KNN) The K- Neighbors model takes n-neighbors = 4, meaning for each data point it considers the classes of the four nearest neighbors to make a prediction. It calls the class voted by the majority of these neighbors. This is the parameter that controls how local or global the decision would be done; having fewer neighbors means the model is more sensitive to what is local by Moturi et al. [14].

MULTILAYER PERCEPTRON NEURAL NETWORK (MLP) MLP-Classifier model uses GridSearchCV to 'tune' some of its most important parameters: hidden-layer sizes=(11, 9, 1) gives the network structure, activation is set at tanh and ReLU and both SGD as well as Adam are taken as solver options. Learning-rate is constant or adaptive, max iter values 200 as well as 400 are applied. Cross-validation is applied to find the best parameters with the goal of getting the highest possible accuracy score for the model.

HARD VOTING CLASSIFIER (HVC) VotingClassifier with SVC, Decision Tree, and Logistic Regression. Logistic Regression uses default settings, while the Decision Tree uses gini for splits and best for the splitter. The VotingClassifier applies hard voting, predicting based on the majority class from all three models. This ensemble method promotes improvement in classification accuracy by combining the strengths of each model.

RANDOM FOREST (RF) Random Forest model uses a genetic algorithm to optimize three key hyperparameters: n estimators between 10 and 200 , max-depth between 10 and 200 , and min-samples-split also between 10 and 200. The

random-state is fixed at 21 for reproducibility. The genetic algorithm maximizes model accuracy as the fitness function, and the best hyperparameters are used to train the final model.

DECISION TREE (DT) This Decision Tree Classifier uses a genetic algorithm to select the key hyperparameters: max-depth is chosen between 2 and 20, min-samples-split is also between 2 and 20, and criterion must either be gini or entropy. The random-state was set to 21 so that the model could potentially be replicated. To optimize these parameters so the model hit peak levels of accuracy, model accuracy was used as the fitness function by Mamidala et al. [15].

LOGISTIC REGREESION (LOGREG) Logistic Regres sion model uses a genetic algorithm that optimizes two hy perparameters: one such parameter is regularization strength, which ranges from 0.01 to 10.0, and the other is penalty, which could either be l2 or l1. Since the l1 solver has been assigned to liblinear, for l2 it must be lbfgs. The maximum iterations to stop the training have been set to 1000 along with a fixed random-state as 21 so that the results would be replicated. The fitness values used are the accuracies to decide the optimal set of hyperparameters.

GENETIC ALGORITHM (GA) Genetic algorithm is a optimization algorithms. It selects the features by genetically. Also it has parameters like initial population, mutation rate, crossover rate. It takes the features based on the best fitness score and accuracy can improve by using this optimisation algorithm.

3.6 RESULT AND ANALYSIS

The Indian liver patient data-set was assessed with a combination of machine learning models which include K- Neighbors, Hard Voting Classifiers, Multilayer Perceptron Neural Networks, and SVM. Optimization of more model types such as LogReg , RF, and DT by the help of Genetic Algorithms enhances their predictions. The various approaches therefore helped in making comparisons on the accuracy of different models in predicting the outcome of liver disease for this patient group.

Figures 1, 3 indicates a distribution of the patient by age groups, whether they bear a disease of the liver. For this study, 1 will be used to indicate that the patient's liver is diseased whereas 2 will be put in place when the patient's liver is healthy and has no trace of having a disease. The dataset consists of 583 individuals, with 71% ,413 people identified as having liver disease, while 28.64% ,170 individuals were reported with healthy livers. Graph showing that the greatest number of casualties of liver diseases falls within the age bracket of 40-60 years.

Figure 5 presents a pair plot, showcasing the relationships between each feature in the dataset. This visual representation offers key insights that help in determining the most appro- priate predictive algorithm, ensuring optimal performance for the dataset. As the problem states that classification is binary , the graph shows that the data points are very overlapped and scattered., making it impossible to separate them using a single linear decision boundary. As a result, a nonlinear model is necessary for effectively addressing this dataset and achieving better prediction accuracy.

Figure:6 illustrate the correlations between two variables can be measured us-

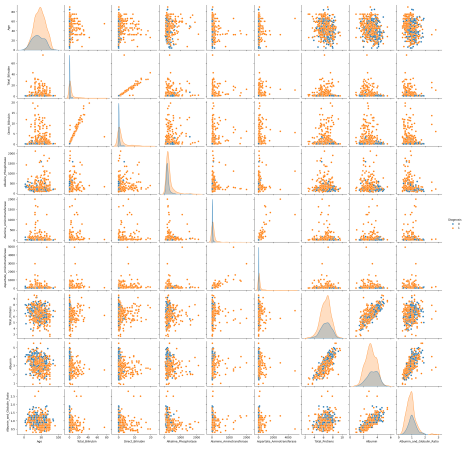

Fig. 5. Pair plot among the attributes.

ing correlation coefficient ranging from negative one (-1) meaning that the two variables have an inverse relationship to a positive one (+1) denoting that they possess a direct relationship while Zero (0) means no correlation at all. However, Figure:6 demonstrates that there is a clear correlation present within the dataset, with various features exhibiting different levels of relationship. This visual representation highlights how certain variables are correlated, positively aiding or negatively in understanding the underlying data structure.

Figure 8 and Table 1 presents the optimization metrics for various models analyzed in liver disease dataset. Evaluating all indices, including accuracy, precision, recall, F1-score, and the confusion matrix, is crucial for determining the most suitable model for the dataset. Among the models, the Random Forest with Genetic Algorithm achieves the highest accuracy of 79%, with strong precision 81%, excellent recall 94%, and an F1 Score of 0.87, indicating a well-balanced performance.Figure 7 indicates confusion matrix for a Random Forest (RF) classifier.

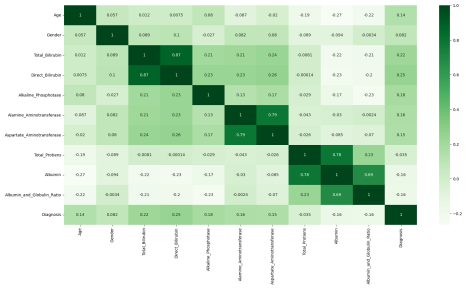

Fig. 6. Correlation matrix based on attributes of dataset.

Table 1. Performance Metrics of Models

Models	Accuracy	Precision	Recall	F1 Score
SVM	0.754	0.75	1.00	0.86
KNN	0.745	0.79	0.91	0.84
MLP	0.746	0.78	0.92	0.84
HVC	0.756	0.76	0.97	0.85
RF(GA)	0.79	0.81	0.94	0.87
DT(GA)	0.75	0.75	1.00	0.86
LOGREG(GA)	0.78	0.80	0.95	0.87

As per table 2 it shows the analysis of the exsiting system which is taken from the reference paper [3]. The table cleary states and shows the accuarcy,f-score,Precision,Recall and specficitiy of the existing models.

The comparison accuracy of the existing model got the highest accuracy of 78%, f-socre got 87% whihc is simillary to the f-score for the proposed model and the highest accuracy in proposed model is Random Forest Algorithm using gentic Algorithm which gives 79%. By this comparison we can cleary say that the propsed model got the highest accuracy .

Table 2. Analysis of the Existint System.

Models	**Accuracy**	**F-score**	**Precision**	**Recall**	**Specificity**
MLP (11, 9, 1)	0.7724	0.8715	0.7724	1	1
SVM (ker = RBF, c=100, g=0.0001)	0.7655	0.864	0.7826	0.9643	0.9643
KNN (K-NN, K=4)	0.7310	0.8368	0.7874	0.8929	0.8929
HVC (ker = RBF)	0.7862	0.8724	0.8092	0.9464	0.9464

Logistic Regression model, optimized with a Genetic Algorithm, achieves an accuracy of 78%, with an impressive 80% precision, a high recall rate of 95%, and an F1 Score of 0.87. The Support Vector Machine exhibits strong performance with 75.4% accuracy, a precision of 75%, perfect recall at 100%, and an F1 Score

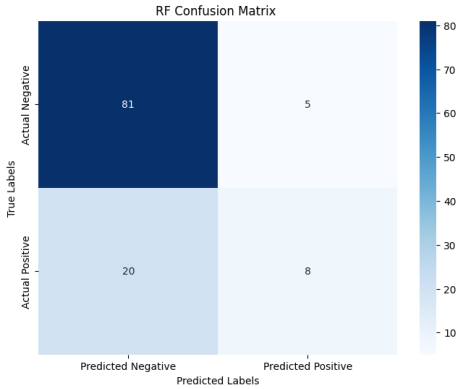

Fig. 7. confusion matrix for a Random Forest (RF) classifier

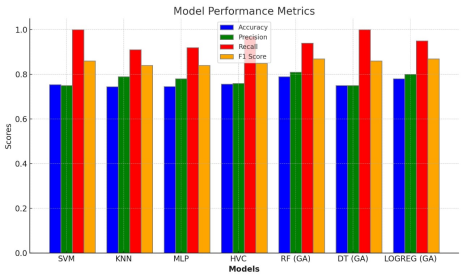

Fig. 8. Visual representation of Performance Metrics for each model.

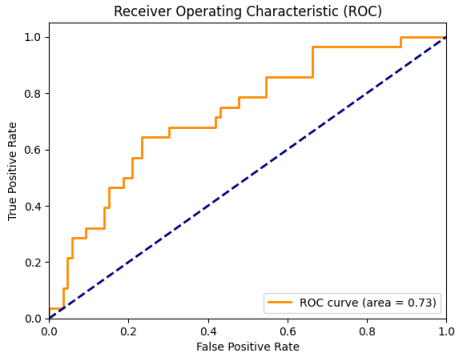

Fig. 9. ROC curve of random forest classifier.

of 0.86, emphasizing its recall strength. K Nearest Neighbors secures a 74.5% accuracy, with 79% precision, 91% recall, and an F1 Score of 0.84, demonstrating robust recall but slightly lower accuracy. The Multilayer Perceptron mirrors this

with 74.6% accuracy, 78% precision, 92% recall, and an F1 Score of 0.84. The Hard Voting Classifier achieves 75.6% accuracy, 76% precision, 97% recall, and an F1 Score of 0.85, showcasing its recall efficiency. The Decision Tree, combined with a Genetic Algorithm, delivers 75% accuracy, 75% precision, perfect recall of 100%, and an F1 Score of 0.86, indicating its capability inidentifying positive cases with balanced performance.

Figure 9 ROC curve illustrates the balance between the true positive rate (sensitivity) and the false positive rate for this classification model. With an area under the curve (AUC) of 0.73, the model demonstrates moderate performance in distinguishing between classes, outperforming random chance.

3.7 CONCLUSION

A liver is a vital organ for maintaining metabolic functions of the body. Livers have the ability to regenerate their cells but in cases where diseases are left untreated and goes unnoticed for long times due to lack of proper medical assistance the risks become unbearable, in such scenarios the chances of death also increase. Hepatic disorders that could've been easily dealt with can now become extremely difficult to intervene due to the lack of timely treatment. However, this study aims to address the issues mentioned above by investigating the possibility of using liver function tests in determining complex liver diseases. Additionally, as part of the timelines, the need for extreme clinical interventions would also be eliminated as the patient would receive the required treatment at the most appropriate time.

The results of the study suggest that after applying the various predictive models, including SVM, K-Neighbors, Multilayer Perceptron, Hard Voting Classifier, Logistic Regression with GA, Decision Tree with GA, and Random Forest with GA, each model was analyzed through key performance metrics including accuracy, precision, recall, and F1 Score. Among these models, Random Forest with Genetic Algorithm achieved the highest accuracy of 79%, with strong precision 81%, excellent recall 94%, and an F1 Score of 0.87. Therefore, we can conclude that, for this dataset, the Random Forest with Genetic Algorithm provides the best and at most highest accuracy. In future ,there is a chance to explore higher-order optimization techniques and deeper learning architectures, along with a larger and more diverse dataset, in order to further enhance the accuracy of predictive models for detection of liver disease.

References

1. KV Narasimha Reddy, SN Tirumala Rao, and K. S. M. V. Kumar. "Diabetes Prediction using Extreme Learning Machine: Application of Health Systems." In 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 993-998. IEEE, 2023.

2. Devarbhavi, Harshad, Sumeet K. Asrani, Juan Pablo Arab, Yvonne Ayerki Nartey, Elisa Pose, and Patrick S. Kamath. "Global burden of liver disease: 2023 update." *Journal of Hepatology* 79, no. 2 (2023): 516-537.
3. Anthonysamy, Victor, and SK Khadar Babu. "Multi Perceptron Neural Network and Voting Classifier for Liver Disease Dataset." *IEEE Access* (2023).
4. Dutta, Krittika, Satish Chandra, and Mahendra Kumar Gourisaria. "Early-Stage detection of liver disease through machine learning algorithms." In *Advances in Data and Information Sciences*, pp. 155-166. Springer, Singapore, 2022.
5. Moturi, Sireesha, Jhansi Vazram Bolla, M. Anusha, M. Mounika Naga Bhavani, Srikanth Vemuru, SN Tirumala Rao, and Sneha Ananya Mallipeddi. "Prediction of Liver Disease Using International Conference on Data Science and Applications." In *Data Science and Applications*, pp. 243-254. Singapore: Springer Nature Singapore, 2023.
6. Murty, Sivala Vishnu, and R. Kiran Kumar. "Enhanced classifier accuracy in liver disease diagnosis using a novel multi-layer feed-forward deep neural network." *International Journal of Recent Technology and Engineering* 8 (2019): 1392-1400.
7. Haque, Md Rezwanul, Md Milon Islam, Hasib Iqbal, Md Sumon Reza, and Md Kamrul Hasan. "Performance evaluation of random forests and artificial neural networks for the classification of liver disorder." In *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*, pp. 1-5. IEEE, 2018.
8. Joloudari, Saadatfar, Javad Abdollah Hassannataj, Dehzangi, Hamid and Shahaboddin Shamshirband. "Computer-aided decision-making for predicting liver disease using PSO-based optimized SVM with feature selection." *Informatics in Medicine Unlocked* 17 (2019): 100255.
9. Gaber, Ahmed, Hassan A. Youness, Alaa Hamdy, Hammam M. Abdelaal, and Ammar M. Hassan. "Automatic classification of fatty liver disease based on supervised learning and genetic algorithm." *Applied Sciences* 12, no. 1 (2022): 521.
10. Kuzhippallil, Maria Alex, Carolyn Joseph, and A. Kannan. "Comparative analysis of machine learning techniques for Indian liver disease patients." In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 778-782. IEEE, 2020.
11. Gupta, Ketan, Nasmin Jiwani, Neda Afreen, and D. Divyarani. "Liver disease prediction using machine learning classification techniques." In *2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 221-226. IEEE, 2022.
12. Musleh, Musleh M., Eman Alajrami, Ahmed J. Khalil, Bassem S. Abu-Nasser, Alaa M. Barhoom, and SS Abu Naser. "Predicting liver patients using artificial neural network." *International Journal of Academic Information Systems Research (IJAISR)* 3, no. 10 (2019).
13. Indian liver patient dataset."https://www.kaggle.com/datasets/ jeevannagaraj/ indian-liver-patient-dataset"
14. Moturi, Sireesha, Jhansi Vazram Bolla, M. Anusha, M. Mounika Naga Bhavani, Srikanth Vemuru, SN Tirumala Rao, and Sneha Ananya Mallipeddi. "Prediction of Liver Disease Using Machine Learning Algorithms." In *International Conference on Data Science and Applications*, pp. 243-254. Singapore: Springer Nature Singapore, 2023.
15. Mamidala, Sai Kumar, Sireesha Moturi, SN Tirumala Rao, Jhansi Vazram Bolla, and KV Narasimha Reddy. "Machine Learning Models for Chronic Renal Disease Prediction." In *International Conference on Data Science and Applications*, pp. 173-182. Singapore: Springer Nature Singapore, 2023.

iccis_satish

ORIGINALITY REPORT

11%	6%	9%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Victor Anthonysamy, SK Khadar Babu. "Multi Perceptron Neural Network and Voting Classifier for Liver Disease Dataset", IEEE Access, 2023 Publication	1%
2	Submitted to Chandigarh University Student Paper	1%
3	jesit.springeropen.com Internet Source	1%
4	www.americaspg.com Internet Source	1%
5	Chukwuebuka Joseph Ejiyi, Dongsheng Cai, Makuachukwu B. Ejiyi, Ijeoma A. Chikwendu et al. "Polynomial-SHAP analysis of liver disease markers for capturing of complex feature interactions in machine learning models", Computers in Biology and Medicine, 2024 Publication	1%
6	link.springer.com Internet Source	1%

7	"Advances in Data and Information Sciences", Springer Science and Business Media LLC, 2022 Publication	<1%
8	Liuyang Zhao, Jun Tian, Yufeng Xie, Landu Jiang, Jianhao Huang, Haoran Xie, Dian Zhang. "scDTL: single-cell RNA-seq imputation based on deep transfer learning using bulk cell information", Cold Spring Harbor Laboratory, 2024 Publication	<1%
9	Prasenjit Dey, Sudip Kumar Adhikari, Sourav De, Indrajit Kar. "Internet of Things-Based Machine Learning in Healthcare - Technology and Applications", CRC Press, 2024 Publication	<1%
10	Tharun J. Iyer, Alex Noel Joseph Raj, Sushil Ghildiyal, Ruban Nersisson. "Performance analysis of lightweight CNN models to segment infectious lung tissues of COVID-19 cases from tomographic images", PeerJ Computer Science, 2021 Publication	<1%
11	assets.researchsquare.com Internet Source	<1%
12	www.frontiersin.org Internet Source	<1%

13	assets-eu.researchsquare.com Internet Source	<1%
14	www.coursehero.com Internet Source	<1%
15	"Proceedings of ICRIC 2019", Springer Science and Business Media LLC, 2020 Publication	<1%
16	Submitted to Manukau Institute of Technology Student Paper	<1%
17	dr.ntu.edu.sg Internet Source	<1%
18	Lei Wang, Xinyu Wang, Zhongchao Zhao. "Mid-term electricity demand forecasting using improved multi-mode reconstruction and particle swarm-enhanced support vector regression", Energy, 2024 Publication	<1%
19	www.irjmets.com Internet Source	<1%
20	journals.itb.ac.id Internet Source	<1%
21	www.indusedu.org Internet Source	<1%
22	www.slideshare.net Internet Source	

		<1%
23	Alex Khang, Vugar Abdullayev, Babasaheb Jadhav, Shashi Kant Gupta, Gilbert Morris. "AI-Centric Modeling and Analytics - Concepts, Technologies, and Applications", CRC Press, 2023 Publication	<1%
24	R. Kalaiselvi, K. Meena, V. Vanitha. "Liver Disease Prediction Using Machine Learning Algorithms", 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), 2021 Publication	<1%
25	Sai Kumar Mamidala, Sireesha Moturi, S. N. Tirumala Rao, Jhansi Vazram Bolla, K. V. Narasimha Reddy. "Chapter 14 Machine Learning Models for Chronic Renal Disease Prediction", Springer Science and Business Media LLC, 2024 Publication	<1%
26	arxiv.org Internet Source	<1%
27	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1%
	e-journal.unair.ac.id	

28	Internet Source	<1%
29	journal.uob.edu.bh Internet Source	<1%
30	www.aging-us.com Internet Source	<1%
31	www.researchgate.net Internet Source	<1%
32	Aritra Pan, Shameek Mukhopadhyay, Subrata Samanta. "Liver Disease Detection", International Journal of Healthcare Information Systems and Informatics, 2022 Publication	<1%
33	Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artificial Intelligence, Blockchain, Computing and Security", CRC Press, 2023 Publication	<1%
34	Greeshma Arya, Ashish Bagwari, Hiteshi Saini, Prachi Thakur, Ciro Rodriguez, Pedro Lezama. "Explainable AI for Enhanced Interpretation of Liver Cirrhosis Biomarkers", IEEE Access, 2023 Publication	<1%

Exclude quotes On Exclude matches Off