

CHAPTER 1

INTRODUCTION

This chapter provides an overview of liver disease, its prevalence, and the challenges of early detection. It introduces the motivation behind using Artificial Neural Networks (ANN) for liver disease diagnosis and outlines the project's aim to develop a non-invasive, data-driven detection model. The chapter also presents the problem statement, project scope, and specific objectives for classification accuracy. By analyzing various patient indicators, this project demonstrates how ANN models effectively classify individuals based on liver health.

The liver, located in the upper gastrointestinal tract, weighs between 1,200–1,800 grams and is vital for digestion, metabolism, detoxification, immune function, and nutrient storage [1]. Liver diseases, which can lead to serious health issues and even death, are categorized based on their causes—such as infections, injuries, genetic abnormalities, or exposure to toxins [2]—and their effects on liver function [3]. Key liver diseases include non-alcoholic fatty liver disease (NAFLD) [5], cirrhosis [4], and various forms of hepatitis [8]. NAFLD is characterized by lipid accumulation, often linked to obesity and diabetes [16]. Cirrhosis involves the replacement of healthy liver tissue with scar tissue, frequently caused by alcoholism or chronic hepatitis [13]. Hepatitis can be acute or chronic and is primarily caused by viral infections from a group of hepatitis viruses (A, B, C, D, and E) [9], each with distinct transmission methods [11]. Common liver diseases such as NAFLD, cirrhosis, and hepatitis have different health implications [2][3]. Advanced stages of fatty liver disease can significantly elevate mortality risk [15]. Preventative measures include reducing alcohol consumption [14], practicing safe sex [10], avoiding shared needles [12], getting vaccinated against hepatitis A and B [7], and maintaining a healthy lifestyle [6]. Traditional methods for diagnosing liver disease are often invasive and costly, emphasizing the need for non-invasive solutions like deep learning models [5][18]. Increasingly, liver disease is linked to obesity and diabetes [17], posing a silent but severe health threat. With advancements in artificial intelligence (AI) and machine learning (ML) [19], healthcare professionals are improving disease detection through better data collection and analysis [20]. Emerging research suggests that gut microbiota may play a significant role in liver disease progression and management [21]. Dysbiosis, or an imbalance in gut bacteria, has been linked to NAFLD and cirrhosis, highlighting potential therapeutic interventions through probiotics and dietary modifications [22]. Additionally, genetic predisposition is increasingly recognized as a contributing factor to liver diseases, with genome-wide association studies (GWAS) identifying specific genes linked to liver dysfunction [23].

Increasingly, liver disease is linked to obesity and diabetes, posing a silent but severe health threat. Advanced stages of fatty liver disease can significantly elevate mortality risk. Preventative measures

include reducing alcohol consumption, practicing safe sex, avoiding shared needles, getting vaccinated against hepatitis A and B, and maintaining a healthy lifestyle. With advancements in artificial intelligence (AI) and machine learning (ML), healthcare professionals are improving disease detection through better data collection and analysis. This report highlights the potential of Artificial Neural Networks (ANN) for liver disease detection, focusing on classifying patients based on crucial features such as age, bilirubin levels, and liver enzyme levels. By integrating these technologies, clinicians can enhance diagnostic accuracy and improve patient outcomes in liver health management.

1.1 Problem Statement

Liver diseases are one of the leading causes of morbidity and mortality worldwide. Conditions like hepatitis, cirrhosis, and liver cancer pose significant public health challenges, particularly because their symptoms often remain undetected until advanced stages. Early detection and treatment are crucial for reducing the adverse effects of these diseases. Traditional diagnostic methods often involve complex biochemical tests or invasive procedures such as liver biopsies, which are both costly and resource-intensive. In regions with limited access to healthcare infrastructure, the diagnosis of liver disease is especially challenging. Therefore, there is a pressing need for a non-invasive, automated diagnostic system that can assist healthcare professionals in identifying liver disease using routine medical test results. The problem at hand is to develop a machine learning model that can accurately classify whether a patient has liver disease based on their medical records, particularly focusing on biochemical markers such as bilirubin and enzyme levels, which are indicative of liver health

1.2 Problem Formulation

Given a dataset of patient records that includes both demographic data (such as age and gender) and results from various liver function tests, the task is to design a detection model that can classify the data into two categories: patients with liver disease and patients without liver disease. With data from liver function tests and patient demographics, this project aims to classify patients with or without liver disease. Key challenges include handling imbalanced datasets and optimizing the ANN model's accuracy for real-world application. The model uses features like bilirubin and enzyme levels, as well as the albumin-globulin ratio, which are strong indicators of liver function [6][7]. The primary challenge lies in creating a model that can generalize well across unseen data while maintaining high accuracy, precision, and recall, particularly given the complexity of liver disease diagnosis where symptoms overlap with other medical conditions.

Key aspects of the problem formulation are:

- **Input:** A dataset consisting of features such as age, gender, total bilirubin, direct bilirubin, alkaline phosphatase, alanine aminotransferase, aspartate aminotransferase, total proteins,

albumin, and albumin/globulin ratio.

- **Output:** A binary classification where 1 represents a patient with liver disease and 0 represents a patient without liver disease.
- **Objective:** To develop a deep learning model, specifically an Artificial Neural Network (ANN), that accurately predicts liver disease based on the input features.
- **Constraints:** Limited dataset size and potential noise in the data, requiring careful handling of imbalanced data and model optimization techniques.

1.3 Project Specification

The project involves the following specifications:

1.3.1 Dataset

The dataset contains 580 instances from Kaggle [24], around 1200 collected from multiple laboratories from our city and 500 AI generated data total around 2300 patient records with 11 features that include both demographic and clinical data. Each record is labelled with the target variable indicating the presence (1) or absence (0) of liver disease.

1.3.2 Data Preprocessing

The preprocessing steps include handling missing values (if any), encoding categorical variables (such as gender), and scaling the continuous features to ensure they are on a comparable scale for the model.

1.3.3 Model Architecture

The chosen model is an Artificial Neural Network (ANN) with multiple hidden layers. The ANN model will be designed to map the input features to the target output (presence or absence of liver disease) using non-linear transformations within the hidden layers.

1.3.4 Training Process

The model will be trained using a training dataset split from the provided data. The loss function used will be binary cross entropy, given the binary nature of the classification problem.

1.3.5 Evaluation Metrics

The model's performance will be evaluated using metrics such as accuracy, precision, recall, F1-score, and a confusion matrix. These metrics provide a comprehensive understanding of the model's ability to classify patients correctly.

1.4 Scope

The scope of this study primarily centers on the development and evaluation of multiple Machine Learning (SVM, KNN, Random Forest) and Deep Learning (ANN, CNN) models aimed at accurately classifying liver disease based on key patient indicators, including age, bilirubin levels, and liver enzyme concentrations. This study will address several challenges, such as managing class imbalances in the dataset, improving diagnostic accuracy, and effectively detecting early signs of

liver disease. Additionally, the study will incorporate synthetic oversampling techniques to mitigate class imbalance issues. Evaluation will be conducted against established machine learning models to benchmark performance, highlighting these model's effectiveness across various evaluation metrics. Ultimately, this study emphasizes the potential of advanced machine learning and machine learning techniques to enhance liver disease detection within the healthcare domain.

1.5 Disease Detection

The process of early liver disease detection using patient data and Artificial Neural Networks (ANN) involves training algorithms to identify patterns or anomalies in key indicators such as age, bilirubin levels, and liver enzyme concentrations. ANN models are trained on a comprehensive dataset that includes information from both healthy individuals and those diagnosed with liver disease. Through this training, the models learn to differentiate between the normal ranges of these indicators and those that may suggest liver dysfunction. Once trained, the ANN can analyze new patient data to detect abnormalities or early signs of liver disease, providing valuable insights that assist in timely diagnosis and intervention. Early detection is essential for effective treatment planning and can significantly improve patient outcomes in liver health management.

1.6 Traditional Way of Disease Detection

Traditional methods for detecting liver disease typically involve a combination of clinical evaluations, blood tests, imaging studies, and medical history assessments. Physicians commonly start with a physical examination and a review of symptoms such as jaundice, fatigue, or abdominal pain. Blood tests are crucial, measuring liver function through enzyme levels (e.g., ALT, AST, alkaline phosphatase) and assessing bilirubin levels to identify potential liver dysfunction. Imaging techniques, such as ultrasound, CT scans, or MRI, may be utilized to visualize the liver's structure and detect abnormalities like fatty liver, tumours, or cirrhosis. In some cases, a liver biopsy may be performed to obtain a tissue sample for histopathological examination, providing definitive information about liver conditions. These traditional approaches often rely on visible symptoms or abnormalities detected in later stages of liver disease, which may limit early intervention. In contrast, utilizing Deep Learning and Machine Learning and patient data for disease detection allows for a more proactive approach, potentially identifying early indicators of liver dysfunction that may not be apparent through conventional methods. This can lead to earlier diagnosis and improved treatment strategies.

1.7 Deep Learning

Deep learning, a subset of machine learning, leverages artificial neural networks to analyse and recognize intricate patterns within complex datasets. In the context of liver disease detection, deep learning models can be trained on extensive datasets that include patient indicators such as age,

bilirubin levels, and liver enzyme concentrations. By learning the relationships and features associated with healthy liver function versus those indicative of liver disease, these models can effectively predict the likelihood of liver dysfunction based on new patient data. Once trained, deep learning algorithms can analyse incoming data more quickly and accurately than traditional methods, facilitating earlier detection of potential liver disease. This capability allows healthcare providers to identify at-risk patients sooner, leading to timely interventions and improved management strategies. Overall, deep learning represents a transformative approach to enhancing diagnostic accuracy in liver health assessments. The role of deep learning is pivotal in the realm of liver disease detection, harnessing its capacity to automatically extract complex features from diverse patient data. Deep learning models, particularly through the use of artificial neural networks (ANN), are crucial in identifying subtle patterns and abnormalities in indicators such as bilirubin levels and liver enzyme concentrations that may signal the presence of liver disease. These models learn representations directly from the data, allowing them to detect nuanced changes that might indicate early-stage liver dysfunction, often well before traditional diagnostic methods can. By training on extensive datasets, deep learning algorithms excel at recognizing patterns that may not be apparent to human observers, significantly enhancing diagnostic accuracy and sensitivity. Furthermore, the application of deep learning reduces subjectivity in interpreting clinical data, leading to more consistent and objective assessments. This objectivity aids clinicians in making timely diagnoses and interventions, facilitating personalized treatment plans for individuals at risk of liver disease. Overall, deep learning has the potential to transform liver disease detection ultimately improving patient outcomes.

1.8 Liver Disease

Liver disease encompasses a range of conditions that affect the liver's ability to function properly. It can result from various causes, including viral infections, alcohol abuse, obesity, and genetic disorders. Liver disease is a significant health concern globally, as it can lead to serious complications such as cirrhosis, liver failure, and even liver cancer. The key characteristics of liver disease include inflammation, fatty liver accumulation, fibrosis (the formation of scar tissue), and the eventual loss of liver function. These conditions can progress through several stages, starting with mild liver dysfunction and leading to severe impairment in liver function and health. Symptoms may vary widely, often including fatigue, jaundice (yellowing of the skin and eyes), abdominal pain, and swelling. The exact causes of liver disease are multifactorial, involving a complex interplay of genetic, environmental, and lifestyle factors. Conditions such as non-alcoholic fatty liver disease (NAFLD), hepatitis (both viral and autoimmune), and alcoholic liver disease are among the most prevalent forms. Currently, there is no single cure for liver disease; however, treatment options focus on managing symptoms, slowing disease progression, and addressing underlying causes. Lifestyle changes, such as maintaining a healthy weight, avoiding alcohol, and managing diabetes, can

significantly impact liver health. Early detection and intervention are critical for improving outcomes and preventing severe complications associated with liver disease.

1.9 Summary

This chapter presents an overview of liver diseases, their causes, and the importance of early detection. It introduces the concept of using Artificial Neural Networks (ANN) for automated, non-invasive liver disease diagnosis. The chapter defines the problem statement, formulates the objectives, and outlines the project's scope, emphasizing the use of machine learning and deep learning for classification tasks. It also details the dataset, model architecture, and key indicators like bilirubin and enzyme levels used for predicting liver health. The foundation for a data-driven diagnostic system is laid out clearly in this chapter.

1.10 Organization of Black Book

This Black Book is structured to provide a comprehensive overview of the project “Liver Disease Detection Using Deep Learning.” It begins with an introduction that outlines the problem statement, objectives, and scope. Chapter 2 presents a literature review of existing models and research. Chapter 3 details the methodology, including data collection, preprocessing, model design, training, and deployment. Chapter 4 discusses the results, evaluation metrics, and application implementation, while Chapter 5 concludes the work and suggests future directions. Finally, references and appendices are provided to support the documentation.

CHAPTER 2

LITERATURE REVIEW

In this chapter, we explore existing literature on machine learning applications for liver disease detection. It reviews traditional and machine learning models, discussing their strengths and limitations in healthcare diagnostics. The chapter concludes with an analysis of why ANN is a suitable approach for this problem, given its ability to capture complex, non-linear relationships in medical data.

2.1 Literature Review

This chapter provides a focused survey of machine learning techniques applied to the detection and classification of liver disease, specifically highlighting models with accuracies below 80% and avoiding deep learning methods. Several studies have utilized Support Vector Machines (SVM) as a primary technique. For instance, Srilatha Tokala et al. (2023) reported an accuracy of 72.46% in their liver disease prediction study, while Tsehay Admassu Assegie et al. (2022) achieved a hybrid SVM model with an accuracy of 78.3%. Additionally, Geetha et al. (2021) explored various machine learning algorithms, obtaining an accuracy of 75.04%. These studies underscore the utility of SVM in navigating complex datasets commonly encountered in medical diagnostics. Machine learning techniques for liver disease detection vary widely. Prior studies on Support Vector Machines (SVM) demonstrated modest accuracies (72%–79%), highlighting the need for models that can better capture complex interactions in clinical data [8],[9]. Other studies applied logistic regression, K-Nearest Neighbors (KNN), and decision trees, achieving accuracies around 75%– 78%, which further underscores the importance of feature selection and model tuning [10],[11].

While traditional models provide a baseline, deep learning techniques offer improved accuracy by learning non-linear interactions among features [12],[13]. ANN models, in particular, show promise in liver disease classification, especially when used with balanced datasets and optimized architectures [14].

Further contributions to this domain include studies employing traditional machine learning methods that continue to show promise in liver disease prediction. Patel et al. (2022) performed a comparative analysis of Logistic Regression, Decision Trees, and SVM, yielding a result of 76.2%. Ahmed et al. (2022) utilized Decision Trees for liver disease prediction, achieving an accuracy of 78.5%. Rao et al. (2021) focused on Logistic Regression, with a reported accuracy of 75.9%. Singh et al. (2021) investigated K-Nearest Neighbors (KNN) and achieved an accuracy of

76.4%. Finally, Zhao et al. (2023) conducted a comparative study using Naive Bayes, attaining an accuracy of 79.2%. Collectively, these findings suggest that while traditional machine learning techniques may not always outperform advanced models, they still offer valuable insights and can be

effective in specific contexts, particularly when computational resources are limited.

These studies reveal a critical understanding of the factors influencing model performance, such as dataset characteristics and feature selection. For instance, variations in accuracy may stem from differences in the populations studied or the specific features used for prediction, highlighting the need for careful consideration of data quality and representation. Moreover, the integration of domain knowledge into feature engineering can significantly enhance the performance of these traditional models.

Table 2.1 Summary of Machine Learning Models for Liver Disease Prediction

Sr. No	Title	Author(s)	Year	Model	Accuracy
1	Chronic liver disease detection using ranking and projection-based feature optimization with deep learning	Sumaiya Noor, Salman A. AlQahtani and Salman Khan,	2024	DNN, SVM ,Random Forest, KNN	90.12%
2	Machine Learning Approaches for Liver Disease Prediction: A Comparative Analysis	Srikanth Lakumarapu; R. Nithyanandhan ; V. Sharmila Bhargavi; Anish T P; Nalini M; Siva Subramanian R	2024	Logistic regression, KNN, DT, and RF	74%
3	A Comparative Study of Machine Learning Techniques for Liver Disease Diagnosis	Zhao, H. et al., L. Chen, Y. Wang	2023	Naive Bayes.	79.2%
4	Liver Disease Prediction and Classification using Machine Learning Techniques	Srilatha Tokala, Koduru Hajarathaiah, Sai Ram	2023	SVM	72.46 %

		Praneeth Gunda, Srinivasrao Botla, Lakshmikanth Nalluri, Pathipati Nagamanohar, Satish Anamalamudi, Murali Krishna Enduri.			
5	Liver Disease Prediction Using Classical Machine Learning Techniques	Ahmed, F. et al., M. Khan,A. Ali	2022	Decision Trees	78.5%
6	Random forest and support vector machine- based hybrid liver disease detection	Tsehay Admassu Assegie1, Rajkumar Subhashni2 , Napa Komal Kumar3 , Jijendira Prasath Manivannan4, Pradeep Duraisamy5 , Minychil Fentahun Engidaye1	2022	SVM	78.03 %
7	Comparative Study of Machine Learning Models for Liver Disease Prediction	Patel, K. et al., R. Kumar, S. Sharma	2022	Logistic Regression , Decision Trees, SVM	76.2%

8	Automated Prediction of Liver Disease using Machine Learning (ML) Algorithms.	Srivastava, A.; Kumar, V.V. Mahesh, T.; Vivek, V	2022	Linear Regression	75 %
9	Evaluation of Various Machine Learning Algorithms for Liver Disease Prediction	T. Mehta, Singh, R. et al., J. Bansal	2021	K-Nearest Neighbors (KNN)	76.4%
10	Assessment of Liver Disease Risk Using Logistic Regression	N. Gupta, S. VermaRao, P. et al.	2021	Logistic Regression	75.9%
11	Evaluation based Approaches for Liver Disease Prediction using Machine Learning Algorithms	Geetha, C. Arunachalam, A	2021	SVM	75.04 %

2.2 Concept Synthesis

2.2.1 Concept Generation

The first step in designing a liver disease detection model is to generate and evaluate different potential solutions. Given the complexity of liver disease diagnosis, a variety of models were considered, ranging from traditional machine learning algorithms to more advanced deep learning architectures. Initially, simpler models such as Logistic Regression, K-Nearest Neighbours (KNN), and Decision Trees were considered due to their ease of implementation and interpretability. These models have been successfully used in other binary classification tasks and offer a baseline for comparing the performance of more complex models.

2.2.1.1 Logistic Regression:

This model, being a simple linear classifier, is often used for binary classification problems. It provides clear interpretability of results but is limited in its ability to capture non-linear relationships in the data.

2.2.1.2 K-Nearest Neighbours: (KNN):

KNN is another intuitive classification model that works by finding the closest data points in the training set to classify new instances. However, it suffers from performance issues when the dataset grows large and when the data is highly imbalanced or noisy.

2.2.1.3 Decision Trees:

Decision Trees provide a simple and powerful way to model decision-making processes. They are highly interpretable but are prone to overfitting, especially on small or noisy datasets.

These models were evaluated, but their inability to capture the non-linear interactions between clinical features (such as enzyme levels and protein ratios) prompted the exploration of more powerful models. Given the nature of the data and the need for high accuracy, Deep Learning Models emerged as the most suitable approach. Deep Learning Models can capture complex, non-linear patterns, making them ideal for tasks like medical diagnosis where various clinical features interact in intricate ways to predict disease outcomes.

2.2.2 Concept Reduction

Once it was determined that an ANN would be the most appropriate model for this problem, the focus shifted towards finding the optimal network architecture. Several configurations of the neural network were tested, varying in the number of layers, the number of neurons per layer, and the activation functions used.

Key considerations during the concept reduction phase included:

1. Number of Layers: A deeper network (i.e., one with more hidden layers) can theoretically learn more complex functions. However, too many layers can lead to overfitting, especially when the dataset is small. After testing multiple architectures, a network with two hidden layers was found to strike the right balance between model complexity and generalization ability.
2. Number of Neurons per Layer: The number of neurons in each hidden layer controls the capacity of the model to learn from the data. Various configurations were tested, starting from smaller networks (with 32 neurons per layer) to larger networks (with 128 neurons per layer). Ultimately, a configuration with 64 neurons in the first hidden layer and 32 neurons in the second hidden layer was selected, as it performed the best in terms of validation accuracy and minimizing overfitting.
3. Activation Functions: The ReLU (Rectified Linear Unit) activation function was chosen for the hidden layers due to its effectiveness in deep learning tasks. ReLU is computationally efficient and helps prevent the vanishing gradient problem, which can hinder learning in deep networks. For the output layer, the sigmoid activation function was used, as it is suitable for binary classification tasks, outputting values between 0 and 1, which can be interpreted as probabilities.
4. Regularization Techniques: To prevent overfitting, techniques such as dropout were considered. Dropout helps by randomly deactivating a fraction of neurons during training, encouraging the network to learn more robust features rather than memorizing the training data.

5. Loss Function: The binary cross entropy loss function was chosen due to the binary nature of the classification problem. Binary cross entropy measures the difference between the predicted probabilities and the actual binary labels, penalizing incorrect predictions.
6. Evaluation Metrics: Accuracy alone is not sufficient to evaluate the performance of a binary classification model, especially when dealing with potentially imbalanced classes. Therefore, additional metrics such as precision, recall, F1-score, and the confusion matrix were selected to provide a more comprehensive assessment of the model's performance. These metrics help ensure that the model does not only achieve high accuracy by exploiting class imbalances but also captures true positive cases effectively.

The final network architecture was selected after multiple iterations of testing and tuning. The chosen ANN structure, with two hidden layers and ReLU activation functions, was found to provide the best performance without overfitting, making it suitable for the liver disease detection task.

2.3 Summary

This chapter provides a comprehensive review of existing research on liver disease detection using machine learning. It discusses various traditional models like SVM, KNN, Decision Trees, and Logistic Regression, along with their limitations in accuracy and generalization. The review supports the choice of ANN due to its superior ability to model complex, non-linear relationships in medical data. The chapter also includes a comparative table of previous studies and emphasizes the need for optimized deep learning approaches in clinical diagnostics.

CHAPTER 3

METHODOLOGY

3.1 Overview:

In this chapter, we describe the complete development process followed for building the Liver Disease Detection and Recommendation System. The methodology includes selecting appropriate tools and technologies, collecting and preprocessing data, model selection and evaluation, deployment of the model using a web server, and frontend development for web and mobile platforms. We followed an Incremental and Iterative Development approach, ensuring that each component of the system was developed, evaluated, and enhanced in stages.

3.2 System Architecture

The architecture of the Liver Disease Detection and Recommendation System is designed to ensure efficient communication between the frontend applications, backend server, machine learning model, and the Gemini LLM. The overall system flow is depicted in the Fig. 3.1.

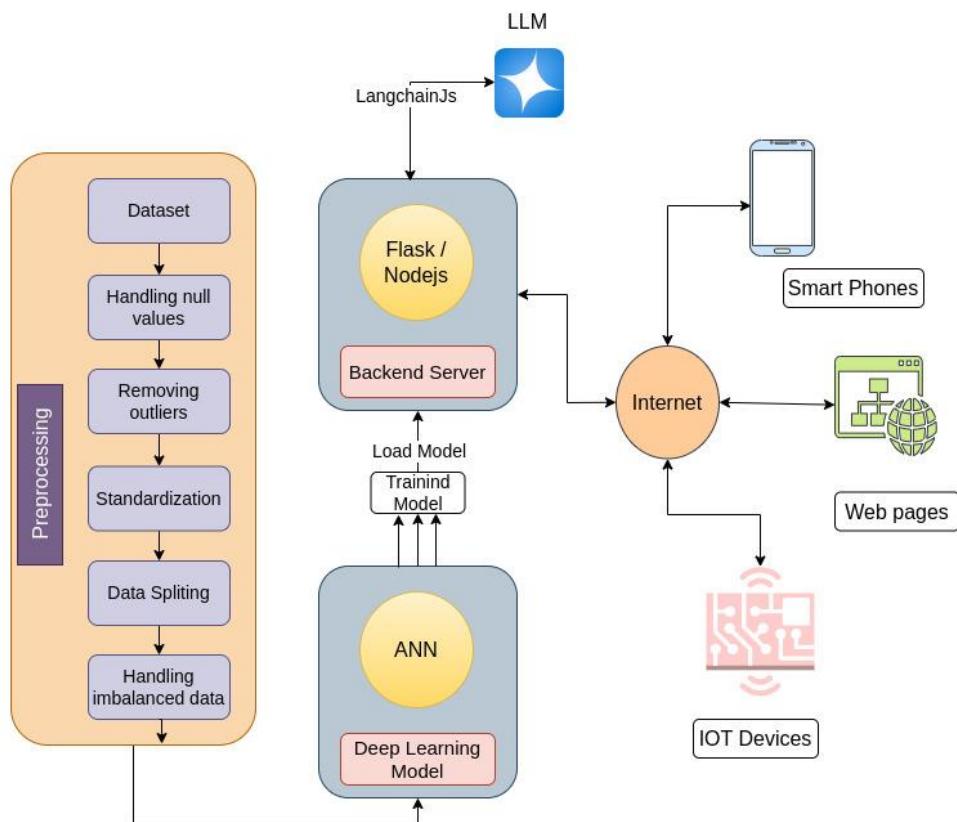


Fig.3.1 System Architecture

Description:

- Users input their Liver Function Test (LFT) parameters via the web or mobile application.
- The frontend sends the data to the Flask backend server.
- The server processes the input, runs the ANN model for disease detection, and calculates

a confidence level.

- Based on the result and confidence level, the Gemini LLM (integrated via LangChain) generates personalized recommendations.
- The server sends back the prediction and suggestions to the frontend for user display.
- The server sends back the prediction and suggestions to the frontend for user display.

3.3 Tools and Technologies Used

To successfully develop and deploy our project, the following tools and technologies were utilized:

- **Google Colab:** A cloud-based Jupyter notebook environment that allows for easy development, training, and testing of machine learning models using free GPU/TPU resources.
- **Visual Studio Code (VS Code):** A lightweight and powerful source-code editor used for writing and managing backend (Flask) and frontend (Flutter, HTML, CSS, JavaScript) code.
- **TensorFlow:** An open-source deep learning framework developed by Google, used to build and train neural networks like CNN for liver disease prediction.
- **Scikit-learn:** A Python machine learning library used for classical ML algorithms (SVM, KNN, Random Forest), data preprocessing, and techniques like SMOTE (Synthetic Minority Over-sampling Technique).
- **Flask:** A micro web framework in Python used to develop the backend server, handle API requests, and serve the trained model for deployment.
- **LangChain:** An open-source framework for building applications with large language models (LLMs). Used to integrate Google Gemini to generate personalized recommendations based on model outputs.
- **Flutter:** A cross-platform UI toolkit developed by Google for building mobile applications for Android and iOS using a single codebase in Dart.
- **HTML (HyperText Markup Language):** The standard language used to structure content on the web.
- **CSS (Cascading Style Sheets):** A style sheet language used to control the appearance (layout, colors, fonts) of HTML elements on the web.
- **JavaScript:** A programming language used to make web pages interactive and dynamic, commonly used in the frontend of web applications.

3.4 System Requirements

3.4.1 Hardware Requirements:

- Processor: Intel Core i5 or higher / AMD equivalent
- RAM: Minimum 8 GB (16 GB recommended for better performance)

- Storage: At least 10 GB free space
- GPU: Optional, but recommended for faster model training
- Internet Connection: Required for using Google Colab and deploying APIs

3.4.2 Software Requirements:

- Operating System: Windows 10/11, Ubuntu 20.04+, or macOS
- Python: Version 3.7 or above
- Visual Studio Code
- Google Chrome or equivalent web browser
- Google Colab environment
- Required Python libraries: TensorFlow, Scikit-learn, Flask, Pandas, NumPy, Matplotlib, LangChain
- Flutter SDK

3.5 Data Collection

Data collection is a crucial step in any machine learning or deep learning project. The performance and accuracy of the prediction model heavily depend on the quality, quantity, and diversity of the collected data. For the Liver Disease Detection and Recommendation System, we focused on collecting reliable and representative datasets from multiple sources to ensure that our model could generalize well to real-world cases.

3.5.1 Sources of Data

The data used in our project was collected from the following sources:

3.5.1.1 Kaggle Dataset

- We utilized an open-source Liver Disease dataset available on Kaggle [24], a popular platform for data science datasets and competitions.
- The Kaggle dataset contained multiple features related to Liver Function Test (LFT) parameters, patient demographics, and the disease diagnosis label.
- Kaggle's dataset provided a good starting point with standardized feature definitions.

3.5.1.2 Local Laboratory Data

- In addition to Kaggle, we collaborated with a few diagnostic laboratories from our city to collect real-world Liver Function Test (LFT) reports.
- These LFT reports contained authentic patient data including parameters like SGPT, SGOT, ALP, Bilirubin, and Albumin levels.
- The local data helped us enrich the dataset with more variability and better representation of our target population.

By combining both open-source and real-world data, we were able to create a comprehensive dataset that improved the robustness of our model.

3.5.2 Features Collected

Table 3.1 provides an overview of the features in the liver disease dataset, describing each attribute along with its type and relevance to the analysis.

Table: 3.1 Dataset Features

Feature	Type	Description
Gender [25]	Nominal	This feature illustrates the participant's gender.
Age (years) [26]	Numerical	The age range of the participants is 4– 90 years.
Total Bilirubin—TB (mg/dL) [27]	Numerical	This feature captures the participant's total bilirubin.
Direct Bilirubin— DB (mg/dL) [27]	Numerical	This feature captures the participant's direct bilirubin.
Alkaline Phosphatase—ALP (IU/L) [28]	Numerical	This feature captures the participant's alkaline phosphatase
Alanine Aminotransferase— SGPT (U/L) [29]	Numerical	This feature captures the participant's alanine aminotransferase.
Aspartate Aminotransferase— SGOT (U/L) [29]	Numerical	This feature captures the participant's aspartate aminotransferase.
Total Protein—TP (g/L) [30]	Numerical	This feature captures the participant's total protein.
Albumin—ALB (g/dL) [31]	Numerical	This feature captures the participant's albumin
Albumin and Globulin Ratio— AGR [32]	Numerical	This feature captures the participant's albumin and globulin Ratio.
Liver Disease	Nominal	This feature stands for whether the participant has been diagnosed with liver disease or not.

3.5.3 Dataset Characteristics

The final dataset after merging different sources had the following characteristics:

- Number of Instances: Approximately 2300
- Number of Features: 10 input features + 1 target label
- Data Type: Mixture of numerical and categorical data
- Class Distribution: Slightly imbalanced with more healthy samples compared to diseased samples (handled later using SMOTE)
- Missing Values: Some local lab reports had missing entries, which were handled during preprocessing.

3.5.4 Data Quality Challenges

During the data collection phase, we faced several challenges:

- **Missing Entries:** Certain LFT parameters were missing in a few local records, which had to be either imputed or discarded.
- **Imbalanced Dataset:** The number of healthy patients outnumbered those with liver disease, which could bias the model toward predicting "Healthy" more often.
- **Outliers:** Extreme values were found in several features, especially enzyme levels, requiring outlier removal techniques.

These challenges were systematically addressed during the Data Preprocessing phase described in next section.

3.5.5 Ethical Considerations

- All the patient data collected from local laboratories was anonymized to protect patient privacy.
- No personally identifiable information (PII) was collected or stored.
- The data was used strictly for educational and research purposes only, complying with ethical guidelines.

3.6 Data Preprocessing

After collecting the dataset, it was essential to preprocess the data to improve the model's performance and reliability. The preprocessing steps involved the following:

3.6.1 Outlier Detection and Removal

To ensure data consistency and improve model performance, we conducted outlier detection and handling using the following methods:

- **Box Plot and Scatter Plot Visualization:** Employed to visually identify extreme values and distribution anomalies.
- **IQR (Interquartile Range) Method:** Used to systematically detect and remove outliers beyond the acceptable range

Outliers were notably found in the Alkaline Phosphatase and Aspartate Aminotransferase features. These were identified using box plots and subsequently handled using the IQR method. To illustrate the effect of outlier removal, we included a total of four box plot graphs:

In Fig 3.2, the box plot of Alkaline Phosphatase before outlier removal shows several extreme values above the upper whisker, clearly indicating the presence of significant outliers. After applying the IQR method, Fig 3.3 displays a much cleaner distribution, with those extreme values removed and the data appearing more compact and normally distributed.

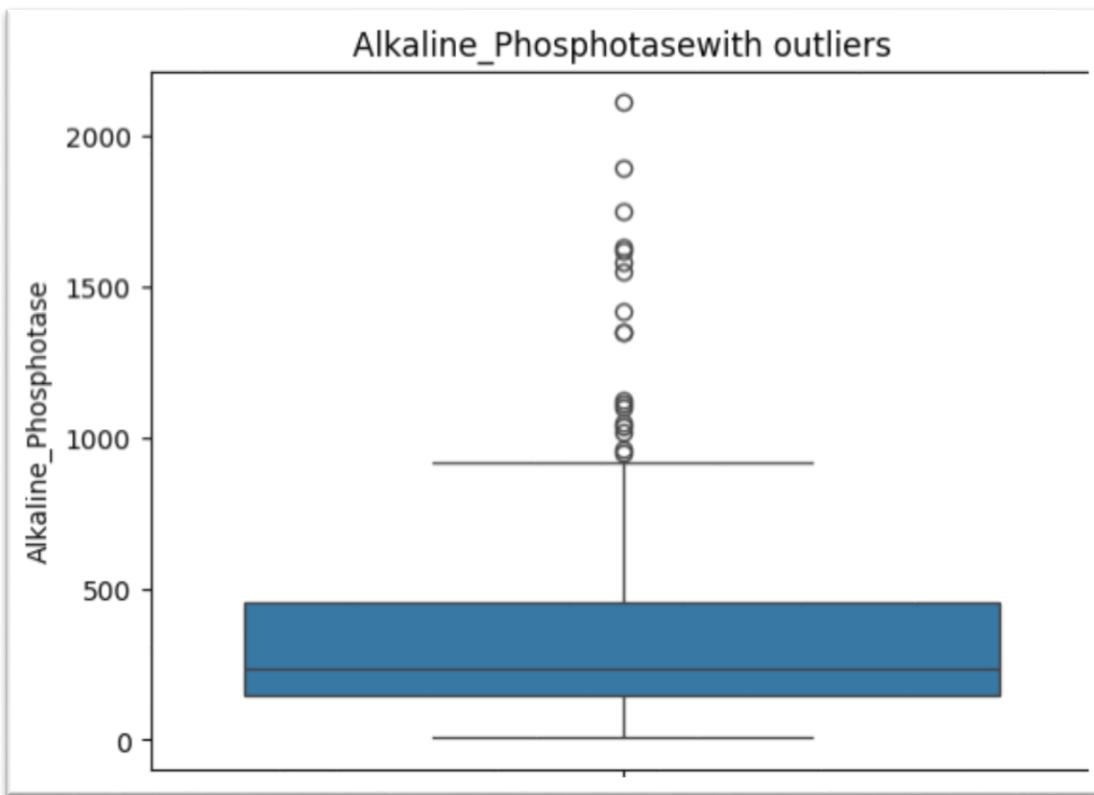


Fig. 3.2 Alkaline Phosphotase with outliers

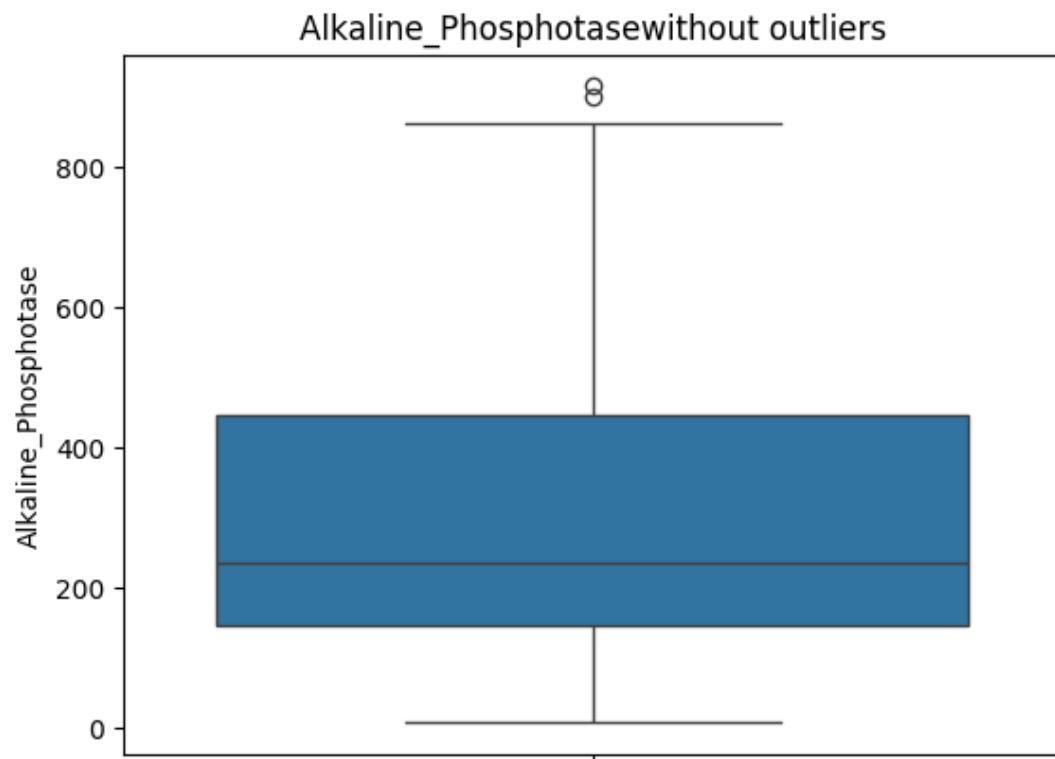


Fig. 3.3 Alkaline Phosphotase without outliers

Fig 3.4 illustrates the box plot of Aspartate Aminotransferase before outlier treatment, where multiple high-value outliers can be observed, distorting the overall scale and distribution. Once the IQR method was applied, Fig 3.5 presents a refined box plot with the outliers successfully removed, resulting in a more balanced and less skewed distribution.

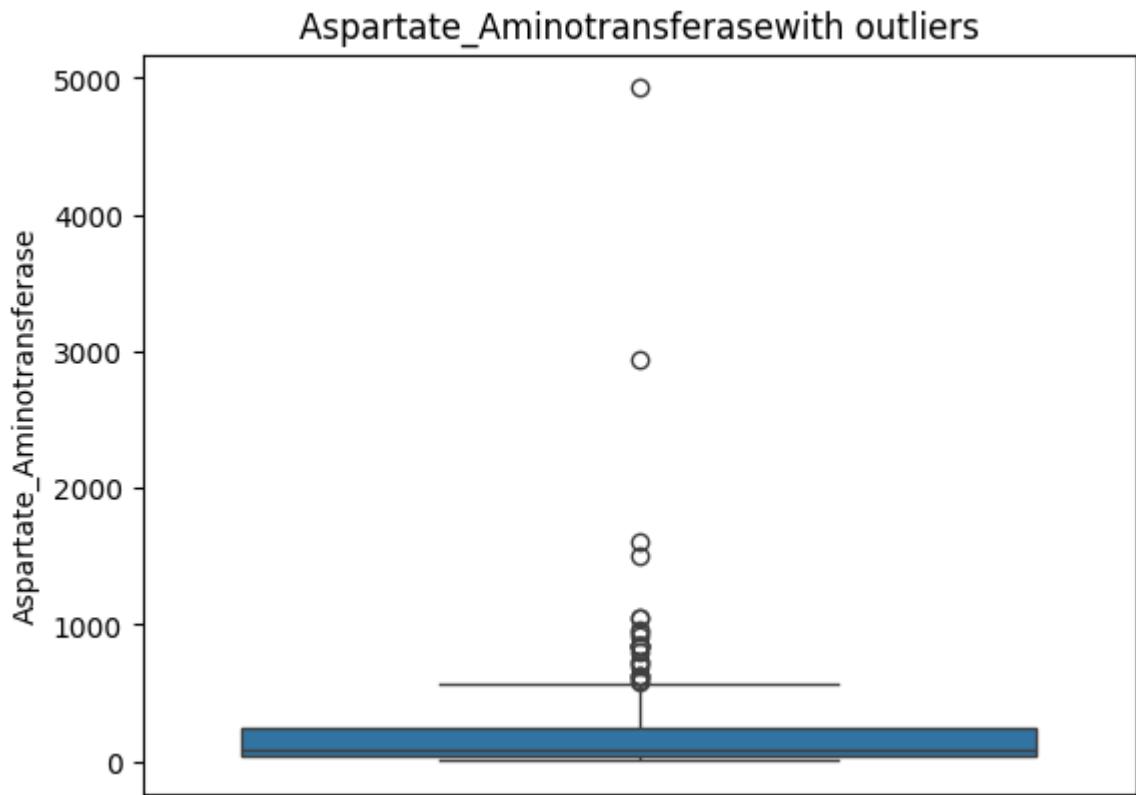


Fig. 3.4 Aspartate Aminotransferase with outliers

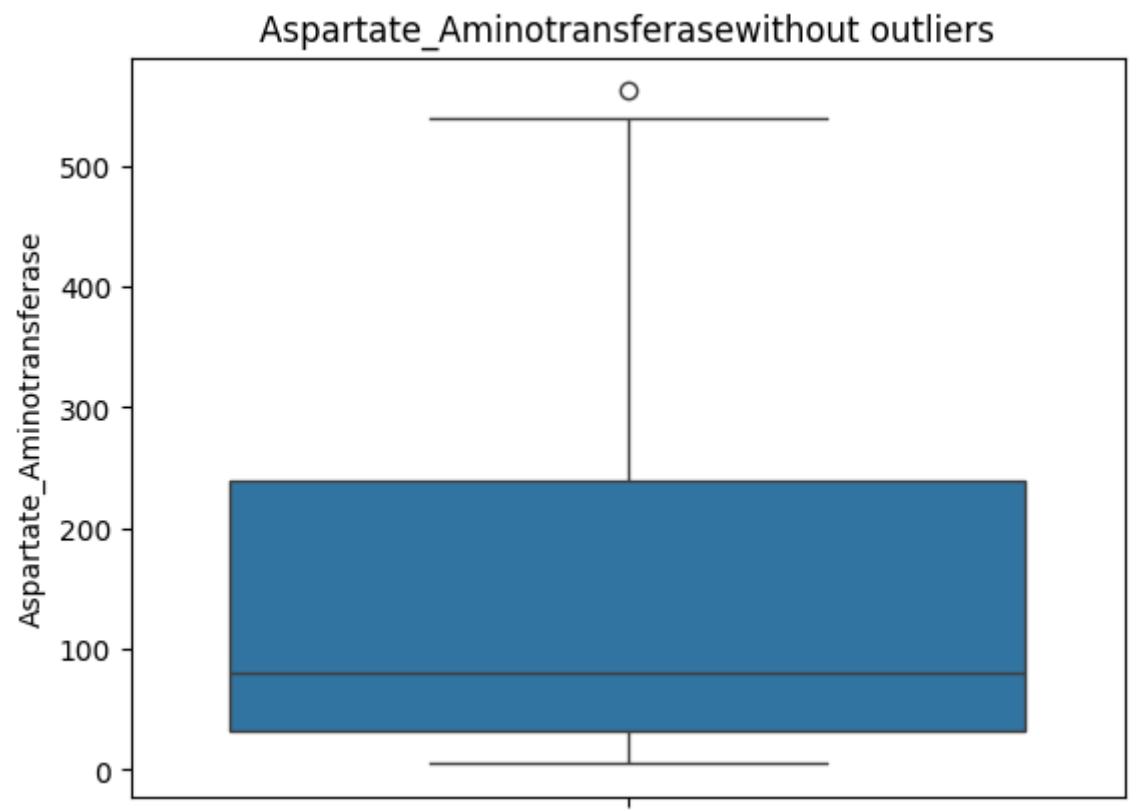


Fig. 3.5 Aspartate Aminotransferase without outliers

These visualizations demonstrate the effectiveness of the outlier removal process and the resulting improvement in data distribution.

3.6.2 Data Standardization

Since the dataset contained features with different scales, we used:

Z-Score Normalization: Each feature was standardized to have a mean of 0 and a standard deviation of 1.

3.6.3 Handling Imbalanced Data

The dataset was found to be imbalanced by 1:2 ratio (more healthy samples compared to diseased ones). To address this issue:

SMOTE (Synthetic Minority Over-sampling Technique) was applied to create synthetic examples of the minority class, resulting in a balanced dataset.

3.6.4 Data Splitting

The preprocessed dataset was split into:

80% Training Set

20% Testing Set

This split allowed the model to learn effectively while preserving sufficient unseen data for evaluation.

3.7 Model Building and Training

We trained multiple Machine Learning (ML) and Deep Learning (DL) models to find the most effective one for liver disease detection.

3.7.1 Machine Learning Models

- Support Vector Machine (SVM): A supervised learning model effective for classification tasks.
- K-Nearest Neighbors (KNN): A simple, instance-based learning algorithm.
- Random Forest (RF): An ensemble learning method based on decision trees.

3.7.2 Deep Learning Models

- Artificial Neural Network (ANN): A basic deep learning model consisting of multiple layers of neurons.
- Convolutional Neural Network (CNN): A more advanced model that captured complex feature interactions.

Each model was trained on the same training dataset, and evaluation was performed using accuracy, precision, recall, and F1-score.

3.7.3 Model Selection

Among all models, the ANN model with dropout layer achieved the highest accuracy and generalization capability, making it the best choice for deployment.

3.8 Artificial Neural Network (ANN) Model Architecture

3.8.1 Model Design:

The core model implemented for liver disease detection is an Artificial Neural Network (ANN). This model is composed of several interconnected layers that process input features and identify complex patterns related to liver disease. The architecture consists of the following layers:

- **Input Layer:** The input layer takes in preprocessed data from the Liver Function Test (LFT) dataset. Each neuron in this layer corresponds to one input feature such as age, gender, bilirubin levels, liver enzymes, etc.
- **Hidden Layers:** The ANN includes multiple hidden layers that are fully connected. These layers are designed to capture complex and non-linear relationships in the data. Each hidden layer consists of a specific number of neurons with the ReLU (Rectified Linear Unit) activation function, enabling the network to learn intricate patterns associated with liver disease.
- **Output Layer:** The final layer is a single neuron with a Sigmoid activation function. This outputs a probability between 0 and 1, which is used to classify whether a patient is likely to have liver disease (1) or not (0).

This layered structure allows the model to effectively transform raw input features into meaningful patterns that contribute to accurate liver disease prediction.

3.8.2 Model Training and Optimization:

The ANN model was trained using the following configuration and optimization techniques:

- **Loss Function:** Binary cross-entropy was used as the loss function since the classification task involves two classes (liver disease: yes or no).
- **Optimizer:** The Adam optimizer was selected for its adaptive learning rate and fast convergence properties, which help improve model efficiency during training.
- **Batch Size:** A batch size of 32 was chosen to strike a balance between training speed and model generalization.
- **Epochs:** The model was trained over 100 epochs. This number was selected based on the convergence behavior of the training and validation loss, ensuring the model had sufficient time to learn from the dataset while avoiding overfitting.
- **Regularization:** To prevent overfitting and enhance generalization, Dropout layers were incorporated in the hidden layers. These layers randomly deactivate a fraction of neurons during training, helping the model become more robust.
- **Evaluation Metrics:** Model performance was evaluated using metrics such as accuracy, precision, recall, and the AUC-ROC curve. These metrics provided a comprehensive assessment of the model’s classification effectiveness, including its ability to detect liver disease in both positive and negative cases.

3.8.3 Model Evaluation:

The trained model was evaluated using the following performance metrics:

- 1. Accuracy:** The proportion of correctly classified instances. Accuracy measures how well the model predicts both positive and negative cases.
- 2. Precision:** The ratio of true positive predictions to the total predicted positives. It indicates how many of the predicted positive cases were actually positive.
- 3. Recall:** The ability of the model to identify actual cases of liver disease. It measures how well the model captures true positives.
- 4. F1-score:** The harmonic mean of precision and recall, balancing both metrics to provide a single performance score.
- 5. Receiver Operating Characteristic (ROC) Curve and Area Under the Curve (AUC):**
The AUC score achieved by the model was 0.92, indicating a high classification ability. This score demonstrates the model's efficiency in distinguishing between positive and negative cases of liver disease.

3.8.4 Testing and Validation:

The trained model was tested on an unseen 20% of the test dataset to assess its generalization capabilities. The model's performance on this test set helped ensure it would perform well in real-world scenarios.

During the training process, the validation loss and accuracy trends were monitored across epochs to detect potential overfitting. Early stopping was implemented based on validation loss to prevent overfitting and ensure the model's robustness.

3.9 Deployment

After model training, we moved to deployment so that users could interact with the system.

3.9.1 Backend Server

Flask Framework was used to develop the backend. The trained ANN model was loaded into the server to handle prediction requests. LangChain was integrated into the Flask server to interact with the Gemini LLM. The server accepted input (LFT data) from clients, predicted disease status with a confidence level, and triggered the LLM to generate recommendations accordingly.

3.9.2 Integration with LangChain and Gemini LLM

Based on the ANN model's output confidence, LangChain dynamically crafted and sent context-aware prompts to the Gemini LLM. This allowed the system to generate highly personalized health tips, precautions, and further suggestions tailored to the user's specific risk level. By leveraging the model's confidence score, the prompts ensured that the responses were both relevant and accurate, enhancing the user experience with actionable and empathetic health guidance.

3.10 Frontend Development

We developed both web and mobile interfaces for user interaction.

3.10.1 Web Application

Developed using HTML, CSS, and JavaScript. The user can enter Liver Function Test (LFT) values into the web form. The data is sent to the Flask server using HTTP POST requests.

3.10.2 Mobile Application

Built using Flutter for cross-platform compatibility. Flutter app provides a clean and user-friendly interface for entering data and viewing predictions and recommendations. Both the web and mobile applications ensure a smooth and responsive user experience.

3.11 Summary

This chapter outlines the complete development process of the Liver Disease Detection System, including system architecture, tools used, data collection, preprocessing, model training, and deployment. The ANN model is described in detail, including architecture, training process, and evaluation metrics. Techniques like Z-score normalization, SMOTE for handling class imbalance, and dropout for regularization are implemented to improve model performance. The chapter also covers the integration of LangChain and Gemini LLM for generating personalized recommendations and describes the development of both web and mobile applications for user interaction.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Evaluation of Deep Learning Models

Table 4.1 presents a comparative analysis of different deep learning models used for liver disease detection. It includes various configurations of Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN), with and without regularization techniques such as L1, L2, and dropout. The models were evaluated based on key performance metrics including accuracy, precision, and ROC-AUC score for both training and testing datasets. This evaluation helps in identifying the most effective model configuration in terms of generalization, robustness, and predictive capability.

Table 4.1 Evaluation of Deep Learning Models

Model	Training Accuracy	Testing Accuracy	Training Precision	Testing Precision	ROC Area
KNN	91%	88%	93%	92%	0.97
SVM	87%	88%	86%	91%	0.97
Random Forest	88%	87%	88%	90%	0.97
ANN	93%	91%	95%	93%	0.97
ANN (L2 Regularization)	91%	91%	93%	93%	0.97
ANN (Dropout 20%)	92%	92%	93%	92%	0.97
CNN (2 Conv Layers)	92%	91%	95%	92%	0.97
CNN (3 Conv Layers)	92%	91%	96%	93%	0.98

4.1.1 Training Accuracy: Training Accuracy indicates how well the model fits the training data. Most models achieved high training accuracy, with the ANN models generally performing slightly better in this regard.

4.1.2 Testing Accuracy: Testing Accuracy shows the model's ability to generalize to new, unseen data. Both ANN and CNN models achieved similar testing accuracy, with most models performing around 91% to 92% on the test set. This suggests that the models are capable of generalizing well.

4.1.3 Training Precision and Testing Precision: Training Precision and Testing Precision provide insight into how well the models perform when predicting the positive class (in a classification context). Precision values are quite high across all models, ranging from 93% to 96%, suggesting that the models are good at making correct positive class predictions.

4.1.4 ROC Area: ROC Area is an important metric for evaluating the model's ability to discriminate between classes. A value closer to 1 indicates better performance. Most models had an ROC area around 0.97, with the CNN model with 3 Conv Layers slightly outperforming the others with an ROC area of 0.98.

4.2 Receiver Operating Characteristic (ROC) for DL Models

Figure 4.1 represents the Receiver Operating Characteristic (ROC) curves for various Deep Learning (DL) models used for classification. The ROC curve illustrates the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) at different classification thresholds.

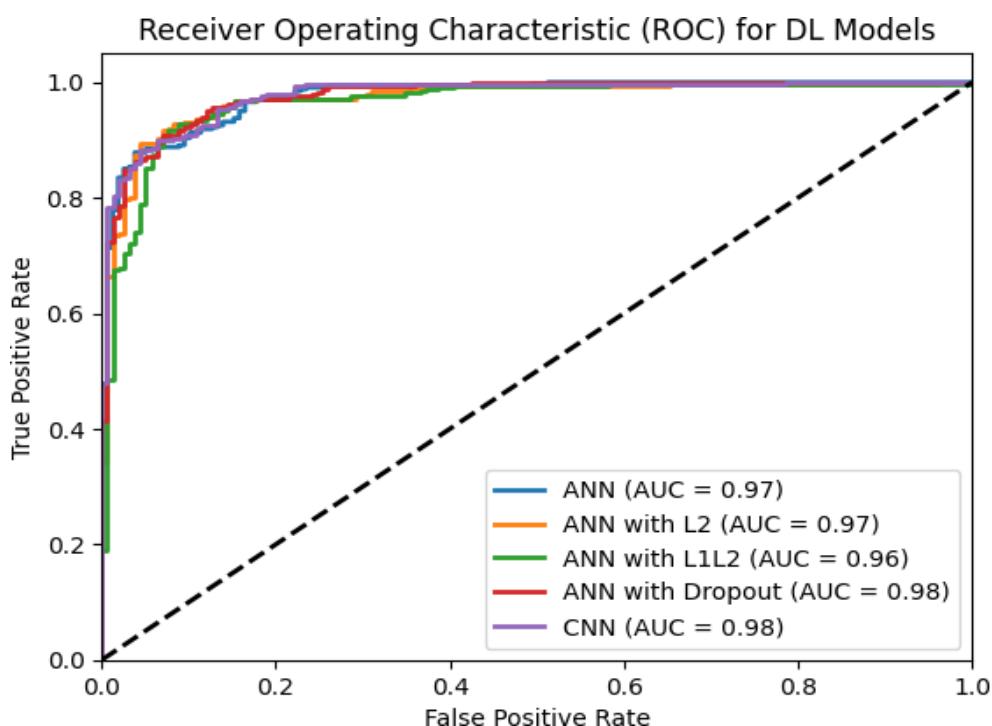


Fig.4.1 Receiver Operating Characteristic (ROC) for DL Models

4.2.1 Key Elements of the Graph

- X-Axis (False Positive Rate - FPR):
 - Measures the proportion of incorrectly classified negative samples as positive.
 - Lower values indicate better model performance.
- Y-Axis (True Positive Rate - TPR):
 - Measures the proportion of correctly classified positive samples.
 - Higher values indicate better model performance.
- Diagonal Line (Black Dashed Line):
 - Represents a random classifier with no discrimination capability (AUC = 0.5).
 - Any model above this line is performing better than random chance.
- Colored ROC Curves:
 - Each curve represents a different DL model's performance.
 - Models closer to the top-left corner indicate better performance

4.2.2 Models and Their AUC Scores

- ANN (Artificial Neural Network) (AUC = 0.97): A standard ANN model.
- ANN with L2 Regularization (AUC = 0.97): Regularization helps prevent overfitting.
- ANN with L1L2 Regularization (AUC = 0.96): Uses both L1 and L2 norms to penalize weights.
- ANN with Dropout (AUC = 0.98): Dropout is used to prevent overfitting by randomly deactivating neurons.
- CNN (Convolutional Neural Network) (AUC = 0.98): A CNN model that slightly outperforms the standard ANN.

4.2.3 Observations

- The CNN and ANN with Dropout achieved the highest AUC (0.98), indicating better classification performance.
- The standard ANN and ANN with L2 regularization performed slightly lower (AUC = 0.97).
- The ANN with L1L2 regularization has the lowest AUC (0.96), but still performs well.

4.3 Training Loss of All Deep Learning Models

Figure 4.2 represents the training loss curves of different Deep Learning (DL) models over 50 epochs. Training loss measures how well a model is learning by evaluating the difference between predicted and actual values.

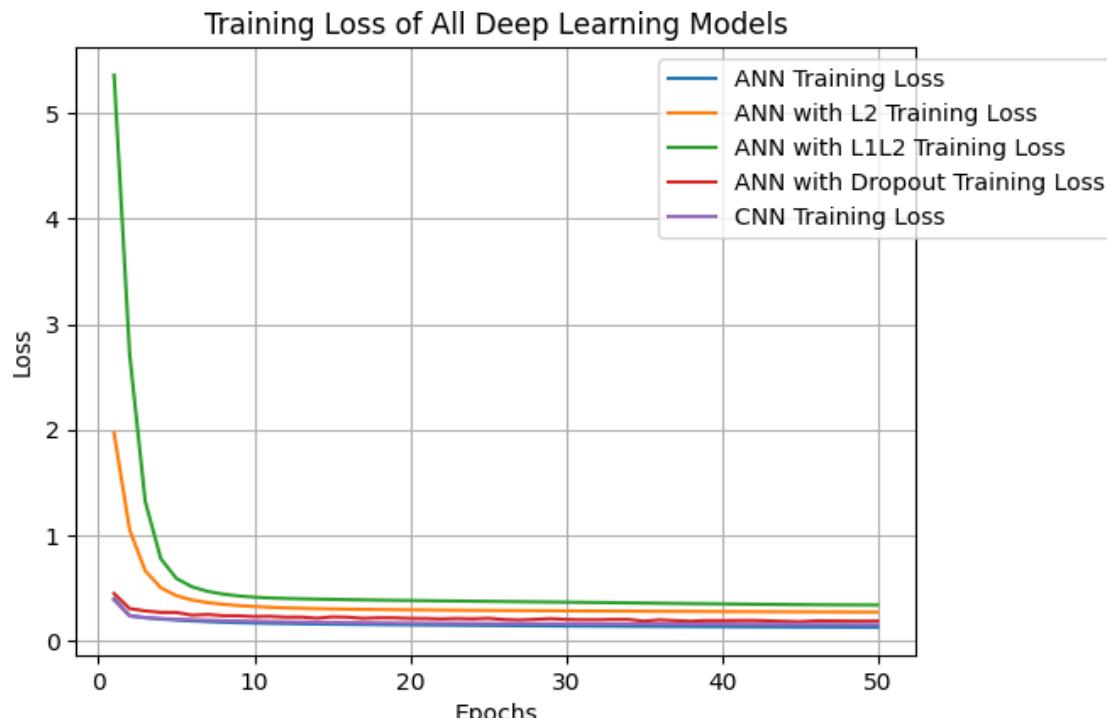


Fig.4.2 Training Loss of All Deep Learning Models

4.3.1 Key Elements of the Graph

- X-Axis (Epochs):
 - Represents the number of training iterations.
 - As epochs increase, the model adjusts weights to reduce the error
- Y-Axis (Loss):
 - Represents the loss value (e.g., categorical cross-entropy or MSE).
 - A lower loss indicates better model performance
- Colored Loss Curves:
 - Each curve represents the training loss of a different DL model.
 - A steep drop at the beginning is common, as models learn patterns quickly in early epochs.

4.3.2 Models and Their Loss Curves

- ANN Training Loss (Blue Curve): A standard ANN that converges smoothly.
- ANN with L2 Regularization (Orange Curve): Has a slightly higher initial loss but stabilizes.
- ANN with L1L2 Regularization (Green Curve): Starts with the highest loss (~5.5) but eventually converges.
- ANN with Dropout (Red Curve): Shows a stable loss curve.
- CNN Training Loss (Purple Curve): Has the lowest loss among all models, indicating better learning.

4.3.3 Observations

- All models show a decreasing trend, which means they are learning properly.
- The ANN with L1L2 regularization started with the highest loss, but it eventually converges, showing that the model required more updates to optimize.
- CNN has the lowest loss, indicating that it learns patterns more effectively.
- Regularized models (L2, L1L2, Dropout) tend to have slightly higher loss values due to the penalty applied to weights, which helps prevent overfitting.

4.4 Training and Validation Loss of All Deep Learning Models

Figure 4.3 represents the training and validation loss of multiple deep learning models over 50 epochs.

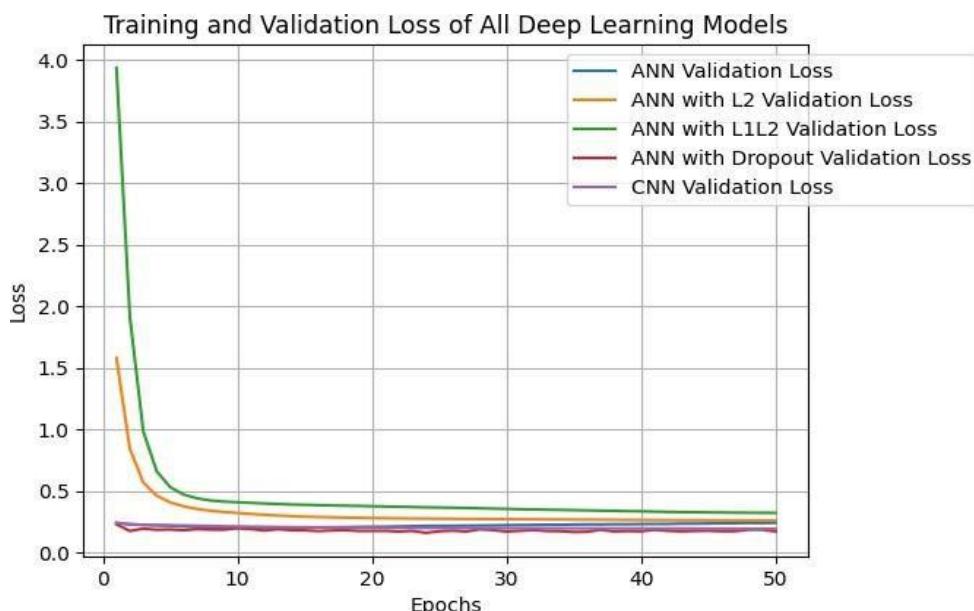


Fig.4.3 Testing and Validation Loss of All Deep Learning Models

4.4.1 Axes:

- X-axis (Epochs): Represents the number of training iterations (1 to 50).
- Y-axis (Loss): Measures the error in predictions, with lower values indicating better model performance.

4.4.2 Curves in the Graph:

- Blue Line: Represents the validation loss of a standard Artificial Neural Network (ANN).
- Orange Line: Represents the validation loss of an ANN with L2 regularization (helps reduce overfitting by adding a penalty to large weights).
- Green Line: Represents the validation loss of an ANN with L1L2 regularization (combination of L1 and L2 penalties).
- Red Line: Represents the validation loss of an ANN with Dropout regularization (randomly drops neurons to prevent overfitting).
- Purple Line: Represents the validation loss of a Convolutional Neural Network (CNN).

4.4.3 Observations:

- **Initial High Loss:** All models start with a high loss, but it quickly decreases within the first 10 epochs.
- **Stabilization:** After around 10–20 epochs, the loss values flatten, indicating that models are converging.
- **Performance Comparison:** The green curve (ANN with L1L2) starts with the highest loss but eventually stabilizes. The red and purple curves (Dropout ANN and CNN) show the lowest loss, indicating that these models generalize better. The CNN (purple) seems to perform better than all ANN models in terms of maintaining a low loss.

4.5 Receiver Operating Characteristic (ROC) for ML Models

Figure 4.4 provides a graphical representation of the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) for different classification thresholds.

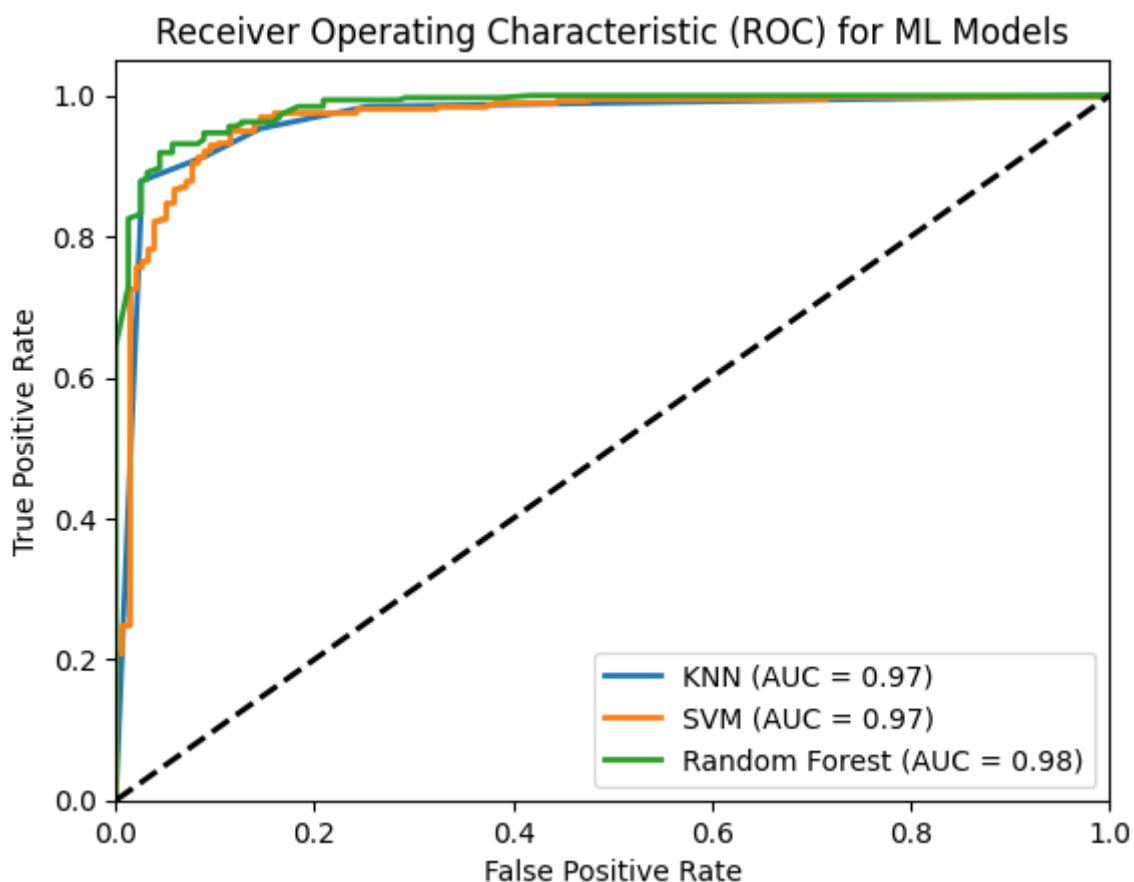


Fig 4.4 Receiver Operating Characteristic (ROC) for ML Models

4.5.1 Key Elements of the Graph

- X-Axis (False Positive Rate - FPR): Indicates the proportion of negative instances incorrectly classified as positive. Lower values reflect better performance.
- Y-Axis (True Positive Rate - TPR): Represents the proportion of correctly identified positive instances. Higher values indicate better model effectiveness.
- Diagonal Line (Dashed Black Line): Denotes the baseline of random guessing ($AUC = 0.5$). Any model curve above this line performs better than random.
- Colored ROC Curves: Each curve represents the classification performance of a different machine learning model. Curves closer to the top-left corner indicate superior classification ability.

4.5.2 Model Performance (AUC Scores)

- K-Nearest Neighbors (KNN): $AUC = 0.97$
- Support Vector Machine (SVM): $AUC = 0.97$
- Random Forest: $AUC = 0.98$

4.5.3 Observations

- Random Forest achieved the highest AUC (0.98), indicating slightly better performance compared to the other models.
- KNN and SVM both performed very well, with AUC scores of 0.97.
- All models demonstrate strong classification capabilities, as their ROC curves are well above the diagonal and close to the top-left corner.

4.6 Implementation of Web and Mobile Application

To showcase the practical applicability of the developed liver disease prediction system, we designed and implemented a fully functional web and mobile application. These platforms enable users to interact with the model in real time, offering an intuitive and accessible interface for health monitoring. This section highlights how the system performs in a real-world setting, focusing on its usability and end-user interaction.

4.6.1 Significances of Frontend Application

The frontend application plays a crucial role in making the liver disease prediction model accessible and easy to use for end users. By providing a simple and intuitive interface, users can enter their Liver Function Test (LFT) data without needing any technical knowledge. The application delivers quick predictions and personalized health recommendations in a clear format, allowing users to understand their health condition better. This helps users take timely action, seek medical advice when needed, and make informed lifestyle decisions based on the model's insights all from the convenience of their mobile or web device.

4.6.2 Web Application.

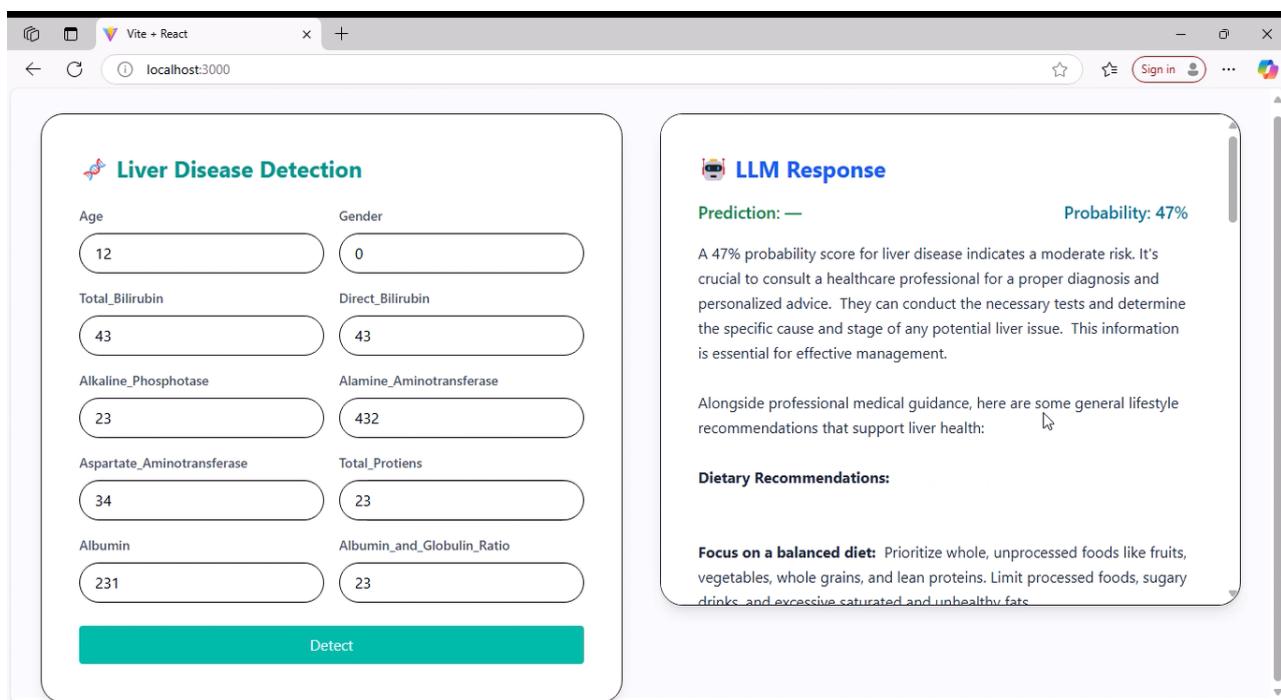


Fig. 4.5 Web Application

Fig. 4.5 shows an input form for users to enter Liver Function Test (LFT) data. After submission, the system predicts the likelihood of liver disease and displays health recommendations generated by a Large Language Model (LLM). The interface is designed for ease of use and immediate feedback

4.6.2 Mobile Application



Fig.4.6 Mobile Application Home Page

The image shows the input form for the mobile application. It is divided into two sections: "Personal Information" and "Test Parameters".
Personal Information:
Age: 12
Gender: Male (radio button selected)
Test Parameters:
Total Bilirubin: 49
Direct Bilirubin: 40
Alkaline Phosphotase: 26
Alanine Aminotransferase: 430
Aspartate Aminotransferase: 40
Total Proteins: 26
Albumin: 215
Albumin and Globulin Ratio: 16

At the bottom are two buttons: "Clear All" and "Detect".

Fig.4.7 Mobile Application Input Form

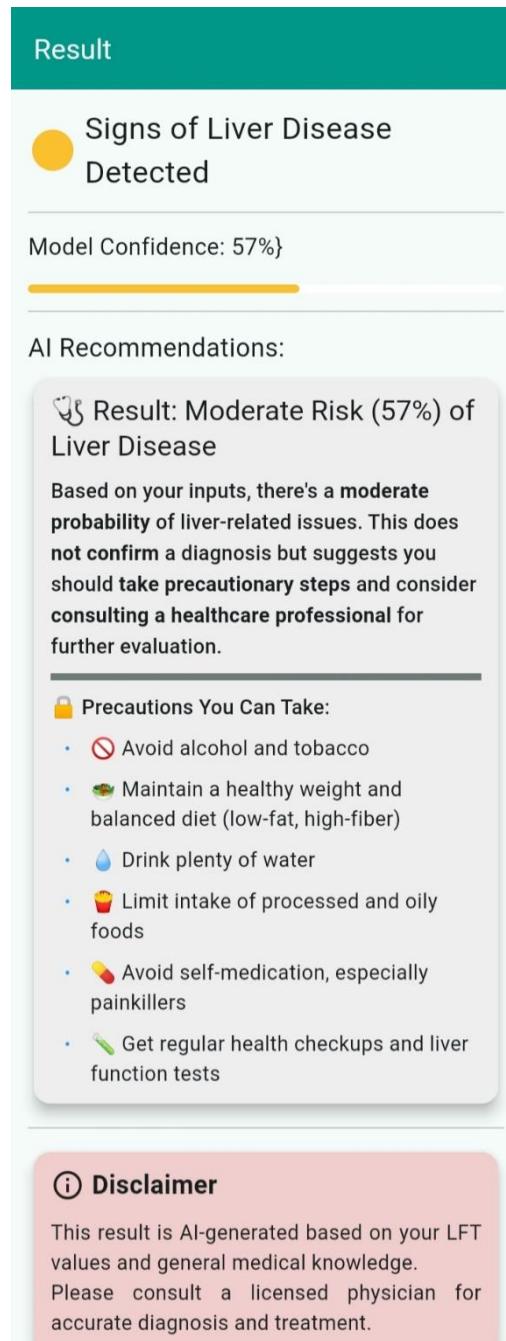


Fig. 4.8 Mobile Application Result and Recommendations

The mobile application offers a seamless and user-friendly experience through a well-structured flow comprising three primary screens. As shown in Fig. 4.6, the homepage features intuitive navigation that guides users to the app's core functionalities. Fig. 4.7 illustrates the input form where users can efficiently enter their Liver Function Test (LFT) data in an organized format. Upon submission, the result page (Fig. 4.8) displays the prediction outcome along with personalized health recommendations generated by the LLM, empowering users to make informed health decisions. This integrated design ensures accessibility, clarity, and personalized engagement for users across different levels of technical familiarity.

4.7 Discussion

The proposed model highlights the feasibility of using deep learning for liver disease detection. Key strengths include:

- Effective handling of class imbalance through synthetic data generation.
- Robust performance metrics, with high precision and recall.

4.8 Summary

This study assessed the performance of various Deep Learning (DL) and Machine Learning (ML) models for liver disease prediction, focusing on accuracy, generalization, and stability in the presence of class imbalance. Results showed that ANN with Dropout and CNNs delivered strong predictive capabilities, with AUC scores reaching 0.98, indicating excellent class discrimination. Regularization techniques such as Dropout and L1L2 were effective in enhancing model stability and reducing overfitting. Training and validation loss patterns confirmed effective convergence within 10–20 epochs, with Dropout-based ANNs achieving the lowest validation loss. Among ML models, Random Forest matched DL performance with an AUC of 0.98, while KNN and SVM also performed competitively. Overall, the study concludes that DL models, especially ANNs and CNNs with regularization, are highly effective and reliable for medical prediction tasks, offering strong generalization and scalable decision-making.

CHAPTER 5

CONCLUSION AND FUTURE DIRECTIONS

5.1 Conclusion

The liver disease detection system developed in this study demonstrates the potential of Artificial Neural Networks (ANNs) for accurate and early diagnosis of liver conditions using routine clinical data. The project incorporated comprehensive data preprocessing techniques including outlier removal, Z-score standardization, and SMOTE-based oversampling to ensure data quality and balance. Through the evaluation of multiple machine learning and deep learning models, the ANN architecture emerged as the most effective, achieving a high AUC score of 92%, which indicates strong predictive performance in distinguishing between patients with and without liver disease.

The model architecture was carefully designed with an input layer representing the clinical features, multiple fully connected hidden layers with ReLU activation for learning complex patterns, and a sigmoid-activated output layer for binary classification. Training was carried out using the binary cross-entropy loss function and the Adam optimizer, both well-suited for binary classification tasks and efficient learning. Evaluation metrics such as accuracy, precision, recall, and AUC-ROC confirmed the reliability and effectiveness of the proposed approach. Overall, the system has demonstrated a promising foundation for non-invasive, data-driven liver disease screening.

5.2 Future Directions

While the current model performs well, several enhancements can be pursued to further improve the system and extend its applicability:

1. Integration with Clinical Decision Support Systems (CDSS): Embedding the ANN model into a real-time decision support system could assist doctors in diagnosing liver disease more efficiently during consultations.
2. Incorporation of More Diverse and Larger Datasets: The model's robustness can be improved by training it on larger and more diverse datasets that include different demographics, geographical regions, and types of liver disease (e.g., alcoholic liver disease, non-alcoholic fatty liver disease, hepatitis, etc.).
3. Feature Expansion and Medical Imaging Data: Future models can integrate additional clinical features or even imaging data (e.g., ultrasound, CT scans) to enhance predictive accuracy and provide more comprehensive liver assessments.
4. Explainable AI (XAI) Techniques: Incorporating interpretability tools such as SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) can help clinicians understand how the model makes predictions, increasing trust and transparency.

5. Multi-class Classification for Disease Severity: Instead of binary classification, future work could focus on classifying the severity or stage of liver disease, enabling more targeted treatment plans.
6. Cross-validation with Real-world Clinical Trials: Collaborating with healthcare institutions to validate the model in real clinical settings will provide deeper insights into its practical utility and areas of improvement.

REFERENCES

1. Arias, I.M.; Alter, H.J.; Boyer, J.L.; Cohen, D.E.; Shafritz, D.A.; Thorgeirsson, S.S.; Wolkoff, A.W. *The Liver: Biology and Pathobiology*; John Wiley & Sons: Hoboken, NJ, USA, 2020.
 2. Singh, H.R.; Rabi, S. *Study of morphological variations of liver in human*. *Transl. Res. Anat* 2019, **14**, 1–5.
 3. Razavi, H. *Global epidemiology of viral hepatitis*. *Gastroenterol. Clin.* 2020, **49**, 179–189.
 4. Ginès, P.; Krag, A.; Abraldes, J.G.; Solà, E.; Fabrellas, N.; Kamath, P.S. *Liver cirrhosis*. *Lancet* 2021, **398**, 1359–1376.
 5. Ringehan, M.; McKeating, J.A.; Protzer, U. *Viral hepatitis and liver cancer*. *Philos. Trans. R. Soc. B Biol. Sci.* 2017, **372**, 20160274.
 6. Powell, E.E.; Wong, V.W.S.; Rinella, M. *Non-alcoholic fatty liver disease*. *Lancet* 2021, **397**, 2212–2224.
 7. Smith, A.; Baumgartner, K.; Bositis, C. *Cirrhosis: Diagnosis and management*. *Am. Fam. Physician* 2019, **100**, 759–770.
 8. Rycroft, J.A.; Mullender, C.M.; Hopkins, M.; Cutino-Moguel, T. *Improving the accuracy of clinical interpretation of serological testing for the diagnosis of acute hepatitis A infection*. *J. Clin. Virol.* 2022, **155**, 105239.
 9. Thomas, D.L. *Global elimination of chronic hepatitis*. *N. Engl. J. Med.* 2019, **380**, 20412050.
 10. Rasche, A.; Sander, A.L.; Corman, V.M.; Drexler, J.F. *Evolutionary biology of human hepatitis viruses*. *J. Hepatol.* 2019, **70**, 501–520.
 11. Gust, I.D. *Hepatitis A*; CRC Press: Boca Raton, FL, USA, 2018.
 12. Yuen, M.F.; Chen, D.S.; Dusheiko, G.M.; Janssen, H.L.; Lau, D.T.; Locarnini, S.A.; Peters, M.G.; Lai, C.L. *Hepatitis B virus infection*. *Nat. Rev. Dis. Prim.* 2018, **4**, 1–20.
 13. Manns, M.P.; Buti, M.; Gane, E.; Pawlotsky, J.M.; Razavi, H.; Terrault, N.; Younossi, Z. *Hepatitis C virus infection*. *Nat. Rev. Dis. Prim.* 2017, **3**, 1–19.
 14. Mentha, N.; Clément, S.; Negro, F.; Alfaiate, D. *A review on hepatitis D: From virology to new therapies*. *J. Adv. Res.* 2019, **17**, 3–15.
 15. Kamar, N.; Izopet, J.; Pavio, N.; Aggarwal, R.; Labrique, A.; Wedemeyer, H.; Dalton, H.R. *Hepatitis E virus infection*. *Nat. Rev. Dis. Prim.* 2017, **3**, 1–16.
 16. Marchesini, G.; Moscatiello, S.; Di Domizio, S.; Forlani, G. *Obesity-associated liver disease*. *J. Clin. Endocrinol. Metab.* 2008, **93**, s74–s80.
 17. Seitz, H.K.; Bataller, R.; Cortez-Pinto, H.; Gao, B.; Gual, A.; Lackner, C.; Mathurin, P.; Mueller, S.; Szabo, G.; Tsukamoto, H. *Alcoholic liver disease*. *Nat. Rev. Dis. Prim.* 2018, **4**, 1–2
-

18. Åberg, F.; Färkkilä, M. Drinking and obesity: Alcoholic liver disease/nonalcoholic fatty liver disease interactions. In *Seminars in Liver Disease*; Thieme Medical Publishers: New York, NY, USA, 2020; Volume 40, pp. 154–162.
19. Bae, M.; Park, Y.K.; Lee, J.Y. Food components with antifibrotic activity and implications in prevention of liver disease. *J. Nutr. Biochem.* 2018, 55, 1–11.
20. Cai, J.; Zhang, X.J.; Li, H. Progress and challenges in the prevention and control of nonalcoholic fatty liver disease. *Med. Res. Rev.* 2019, 39, 328–348.
21. Fazakis, N.; Kocsis, O.; Dritsas, E.; Alexiou, S.; Fakotakis, N.; Moustakas, K. Machine learning tools for long-term type 2 diabetes risk prediction. *IEEE Access* 2021, 9, 103737103757.
22. Dritsas, E.; Trigka, M. Data-Driven Machine-Learning Methods for Diabetes Risk Prediction. *Sensors* 2022, 22, 5304.
23. Alexiou, S.; Dritsas, E.; Kocsis, O.; Moustakas, K.; Fakotakis, N. An approach for Personalized Continuous Glucose Prediction with Regression Trees. In Proceedings of the 2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Preveza, Greece, 24–26 September 2021; pp. 1–6.
24. Indian Liver Patient Records. Available online: <https://www.kaggle.com/datasets/uciml/indianliver-patient-records>.
25. Mauvais-Jarvis, F.; Merz, N.B.; Barnes, P.J.; Brinton, R.D.; Carrero, J.J.; DeMeo, D.L.; De Vries, G.J.; Epperson, C.N.; Govindan, R.; Klein, S.L.; et al. Sex and gender: Modifiers of health, disease, and medicine. *Lancet* 2020, 396, 565–582.
26. Lin, H.; Yip, T.C.F.; Zhang, X.; Li, G.; Tse, Y.K.; Hui, V.W.K.; Liang, L.Y.; Lai, J.C.T.; Chan, S.L.; Chan, H.L.Y.; et al. Age and the relative importance of liver-related deaths in nonalcoholic fatty liver disease. *Hepatology* 2022.
27. Ruiz, A.R.G.; Crespo, J.; Martínez, R.M.L.; Iruzubieta, P.; Mercadal, G.C.; Garcés, M.L.; Lavin, B.; Ruiz, M.M. Measurement and clinical usefulness of bilirubin in liver disease. *Adv. Lab. Med. Med. Lab.* 2021, 2, 352–361.
28. Liu, Y.; Cavallaro, P.M.; Kim, B.M.; Liu, T.; Wang, H.; Kühn, F.; Adiliaghdam, F.; Liu, E.; Vasan, R.; Samarbafzadeh, E.; et al. A role for intestinal alkaline phosphatase in preventing liver fibrosis. *Theranostics* 2021, 11, 14.
29. Goodarzi, R.; Sabzian, K.; Shishehbor, F.; Mansoori, A. Does turmeric/curcumin supplementation improve serum alanine aminotransferase and aspartate aminotransferase levels in patients with nonalcoholic fatty liver disease? A systematic review and meta-analysis of randomized controlled trials. *Phytother. Res.* 2019, 33, 561–570.
30. He, B.; Shi, J.; Wang, X.; Jiang, H.; Zhu, H.J. Genome-wide pQTL analysis of protein expression
-

- regulatory networks in the human liver. *BMC Biol.* 2020, *18*, 1–16.
31. Carvalho, J.R.; Machado, M.V. New insights about albumin and liver disease. *Ann. Hepatol.* 2018, *17*, 547–560.
32. Ye, Y.; Chen, W.; Gu, M.; Xian, G.; Pan, B.; Zheng, L.; Zhang, Z.; Sheng, P. Serum globulin and albumin to globulin ratio as potential diagnostic biomarkers for periprosthetic joint infection: A retrospective review. *J. Orthop. Surg. Res.* 2020, *15*, 1–7.

APPENDIX

Appendix I – Certificate of Appreciation



Appendix II – Research Paper Draft

LIVER DISEASE DETECTION USING DEEP LEARNING

Abdul Rahman,

abdulrahman.college02@gmail.com

Student of Computer Engineering Department, Maulana Mukhtar Ahmad Nadvi Technical Campus Malegaon Maharashtra India 423203

Isar Ahmed,

isarfariidi@gmail.com

Student of Computer Engineering Department, Maulana Mukhtar Ahmad Nadvi Technical Campus Malegaon Maharashtra India 423203

Dr. Salman Baig,

salmanbaig@mmantc.edu.in

Head of Computer Engineering Department Maulana Mukhtar Ahmad Nadvi Technical Campus Malegaon Maharashtra India 423203

Ansari Farzan,

ansarifarzan771@gmail.com

Student of Computer Engineering Department, Maulana Mukhtar Ahmad Nadvi Technical Campus Malegaon Maharashtra India 423203

Mohammed Hassan,

mdhassankm0011@gmail.com

Student of Computer Engineering Department, Maulana Mukhtar Ahmad Nadvi Technical Campus Malegaon Maharashtra India 423203

Asst. Professor Master Simmi

simnimaster@gmail.com

Assistant Professor Department of Computer Engineering Maulana Mukhtar Ahmad Nadvi Technical Campus Malegaon Maharashtra India 423203

ABSTRACT

The detection of liver disease at an early stage is critical for ensuring timely medical intervention. This project focuses on developing an automated liver disease detection system using a deep learning approach, utilizing both Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN). The dataset consists of 2,391 patient medical records, combining Kaggle data, lab reports (from Maharashtra), and records generated through AI techniques. The dataset includes features such as age, gender, bilirubin levels, enzyme concentrations, and protein ratios, all indicative of liver function. Extensive data preprocessing was conducted, including handling missing values, encoding categorical variables, removing outliers, and feature scaling. The ANN model, comprising multiple hidden layers with ReLU activation, was trained to classify patients as either having liver disease or not. The model's performance was evaluated using metrics such as accuracy, precision, recall, and F1-score, achieving an accuracy of 92% on the test dataset. Further improvements are possible through advanced architectures, hyperparameter tuning, and data augmentation. This research underscores the viability of deep learning in facilitating early and non-invasive liver disease detection.

Keywords: Liver disease, Deep learning, ANN, CNN, Healthcare, Data Science.

INTRODUCTION

This chapter provides an overview of liver disease, its prevalence, and the challenges of early detection. It introduces the motivation behind using Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN) for liver disease diagnosis and outlines the project's aim to

develop a non-invasive, data-driven detection model. The chapter also presents the problem statement, project scope, and specific objectives for classification accuracy. By analyzing various patient indicators, this project demonstrates how ANN and CNN models effectively classify individuals based on liver health [1][2].

The liver, located in the upper gastrointestinal tract, weighs between 1,200–1,800 grams and is vital for digestion, metabolism, detoxification, immune function, and nutrient storage [1]. Liver diseases, which can lead to serious health issues and even death, are categorized based on their causes—such as infections, injuries, genetic abnormalities, or exposure to toxins [3]—and their effects on liver function [4]. Key liver diseases include non-alcoholic fatty liver disease (NAFLD) [6], cirrhosis [5], and various forms of hepatitis [12]. NAFLD is characterized by lipid accumulation, often linked to obesity and diabetes [20]. Cirrhosis involves the replacement of healthy liver tissue with scar tissue, frequently caused by alcoholism or chronic hepatitis [17]. Hepatitis can be acute or chronic and is primarily caused by viral infections from a group of hepatitis viruses (A, B, C, D, and E) [13], each with distinct transmission methods [15]. Common liver diseases such as NAFLD, cirrhosis, and hepatitis have different health implications [3][4]. Advanced stages of fatty liver disease can significantly elevate mortality risk [19]. Preventative measures include reducing alcohol consumption [18], practicing safe sex [14], avoiding shared needles [16], getting vaccinated against hepatitis A and B [11], and maintaining a healthy lifestyle [10]. Traditional methods for diagnosing liver disease are often invasive and costly, emphasizing the need for non-invasive solutions like deep learning models [6][22]. Increasingly, liver disease is linked to obesity and diabetes [21], posing a silent but severe health threat. With advancements in artificial intelligence (AI) and machine learning (ML) [23], healthcare professionals are improving disease detection through better data collection and analysis [24]. Emerging research suggests that gut microbiota may play a significant role in liver disease progression and management [26]. Dysbiosis, or an imbalance in gut bacteria, has been linked to NAFLD and cirrhosis, highlighting potential therapeutic interventions through probiotics and dietary modifications [27]. Additionally, genetic predisposition is increasingly recognized as a contributing factor to liver diseases, with genome-wide association studies (GWAS) identifying specific genes linked to liver dysfunction [28].

This chapter provides an overview of liver disease, its prevalence, and the challenges of early detection. It introduces the motivation behind using Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN) for liver disease diagnosis and outlines the project's aim to develop a non-invasive, data-driven detection model. The chapter also presents the problem statement, project scope, and specific objectives for classification accuracy. This report highlights the potential of ANN and CNN for liver disease detection, focusing on classifying patients based on crucial features such as age [39], bilirubin levels [40], liver enzyme levels [42], and other liver indicators [45]. By integrating these technologies, clinicians can enhance diagnostic accuracy and

improve patient outcomes in liver health management [25].

DATASET DESCRIPTION

The dataset utilized in this study comprises 2,391 patient records, derived from three sources:

1. A publicly available dataset from Kaggle[7].
 2. Collected from local Laboratories from Maharashtra.
 3. Synthetic data generated using AI techniques to ensure diversity and balance in class distribution.
- Each record contains the following features:
 - **Demographics:** Age, Gender
 - **Liver Function Indicators:**
 - Bilirubin levels (Total and Direct)
 - Enzyme concentrations (e.g., Alanine Transaminase(ALT), Aspartate Transaminase (AST))
 - Protein levels (e.g., Albumin, Globulin ratio)

Table:1 Dataset Features

Feature	Type	Description
Gender [38]	Nominal	This feature illustrates the participant's gender.
Age (years) [39]	Numeric	The age range of the participants is 4– 90 years.
Total Bilirubin—TB (mg/dL) [40]	Numeric	This feature captures the participant's
Direct Bilirubin— DB (mg/dL) [40]	Numeric	This feature captures the participant's direct bilirubin.
Alkaline Phosphatase—ALP (IU/L) [41]	Numeric	This feature captures the participant's alkaline phosphatase.
Alanine Aminotransferase— SGPT (U/L) [42]	Numeric	This feature captures the participant's alanine aminotransferase.
Aspartate Aminotransferase— SGOT (U/L) [42]	Numeric	This feature captures the participant's aspartate aminotransferase.
Total Protein—TP (g/L) [43]	Numeric	This feature captures the participant's total protein.

Albumin—ALB (g/dL) [44]	Numeric	This feature captures the participant's albumin.
Albumin and Globulin Ratio—AGR [45]	Numeric	This feature captures the participant's albumin and globulin Ratio.
Liver Disease	Nominal	This feature stands for whether the participant has been diagnosed with liver disease or not.

METHODOLOGY

In this study, we proposed a deep learning-based approach for data classification. The methodology follows a structured pipeline that includes data preprocessing, feature selection, deep learning model training, classification, and result evaluation.

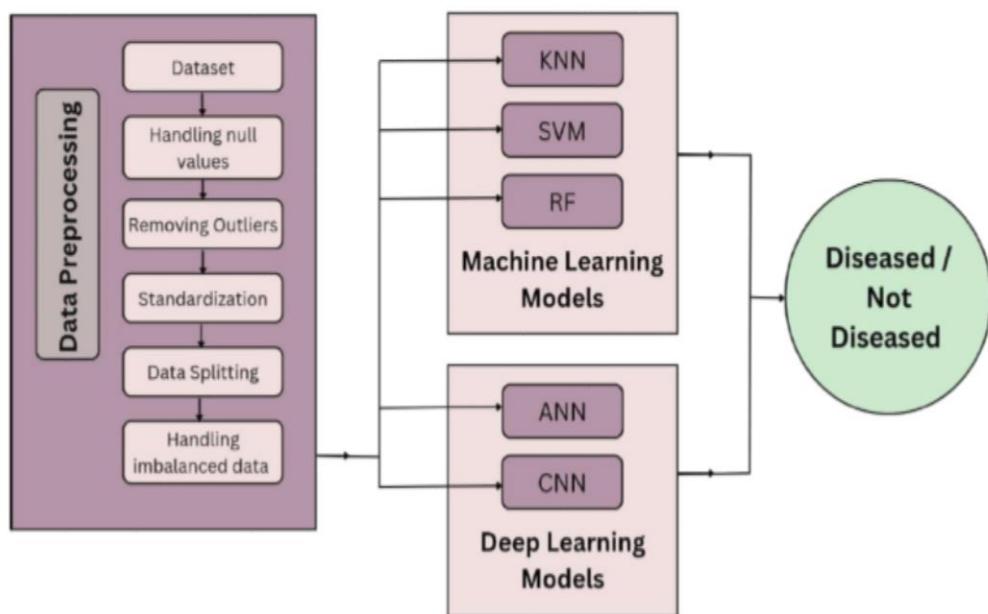


Fig.1 Methodology

- **Data Collection and Preprocessing:** The dataset used for this study consists of 2391 patient records. Each record includes demographic and clinical features relevant to liver function, such as age, gender, bilirubin levels, enzyme concentrations, and protein ratios. To ensure data quality, the preprocessing steps involved:
 - **Handling Missing Values:**

One of the first steps in data preprocessing is to deal with missing or incomplete data. Missing data can occur due to various reasons, such as errors in data collection or patients not undergoing specific tests. Handling missing values is crucial because most machine learning models, including ANNs, cannot handle missing data directly. In this dataset, we conducted an extensive check for missing values using pandas functions. Fortunately, the dataset had no missing values in any of the features. If missing data had been present, several imputation strategies could have been applied:

- **Mean/Median Imputation:** For continuous variables, replacing missing values with the mean or median of the available data.
- **Mode Imputation:** For categorical variables like gender, missing values could be replaced with the most frequent category.

- **Removing Outliers:**

To identify outliers, we used a Box Plot to visualize the distribution of the data and detect extreme values. We then applied the Interquartile Range (IQR) method to systematically remove outliers. The IQR method involves calculating the range between the first quartile (Q1) and third quartile (Q3), and using this range to identify and eliminate data points that lie outside the acceptable limits, typically beyond 1.5 times the IQR. After outliers were removed, imputation was performed on missing values to ensure that the dataset was complete and usable for further analysis. Depending on the nature of the data, different imputation methods, such as mean, median, or regression-based imputation, were applied to ensure consistency and preserve data integrity.

These preprocessing steps are essential to ensure that the data is clean, reliable, and ready for the subsequent analysis and modelling stages. By addressing outliers and missing data, we improve the quality of the data, reduce noise, and enhance the accuracy of any machine learning models or statistical analysis performed.

Feature Scaling:

Artificial Neural Networks (ANNs) are sensitive to the scale of input features. If one feature has a much larger range than others, it can dominate the learning process and lead to suboptimal performance. For example, the Age feature ranges from 0 to 90, whereas Albumin levels are in a

much smaller range, around 1-5. To ensure all features contribute equally, feature scaling was applied.

- **Standardization:** In this project, standardization (z-score normalization) was applied, which transforms each feature to have a mean of 0 and a standard deviation of 1. This is done using the formula:

- $$z = \frac{x - \mu}{\sigma}$$

Where,

- x is the original value,
- μ is the mean of the feature,
- σ is the standard deviation of the feature.

Standardization ensures that all the features have the same scale, regardless of their original ranges. This is particularly important for distance-based algorithms and neural networks, where feature magnitudes can affect the learning process. The **Standard Scaler** from the scikit-learn library was used to scale the dataset. The scaler was fit on the training data and then applied consistently to both the training and test data to ensure that the test data was processed in the same way.

- **Data Splitting:**

To evaluate the performance of both the **Artificial Neural Network (ANN)** and **Convolutional Neural Network (CNN)** models, the dataset was split into **training** and **test** sets. This is a standard practice in machine learning to ensure that the model can generalize well to unseen data.

- **Train-Test Split:** We used the **train_test_split** function from the **scikit-learn** library to split the dataset into 80% training data and 20% test data. This 80/20 ratio is commonly adopted in machine learning projects as it provides enough data for training while keeping a sufficient portion for testing the model's generalization ability.

The **training set** is used to teach the model, whether it's the ANN or CNN, by learning the patterns and relationships in the data. The **test set** is reserved to evaluate how well the trained model performs on **unseen data**, ensuring that it can generalize to new, real-world cases.

The use of **CNN** is particularly useful for tasks like image classification, object detection, or any task where spatial hierarchies and patterns are important. CNNs apply convolutional filters to detect local features, such as edges, textures, and shapes, making them highly effective in visual data analysis.

By splitting the data into training and test sets, we can prevent the models from overfitting,

where they might memorize the data instead of learning the underlying patterns. The test set provides a realistic evaluation of how the model would perform in real-world scenarios when deployed.

Additionally, using both ANN and CNN allows us to compare the performance of traditional neural networks (ANNs) and more specialized deep learning models (CNNs) in handling the dataset, helping us to choose the most effective model for the task at hand.

- **Handling Imbalanced Data (SMOTE)**

In many machine learning tasks, especially classification problems, the dataset may have an **imbalanced class distribution**, where one class has significantly more instances than the other. This imbalance can lead to biased model performance, as the model may become biased toward predicting the majority class and underperform in predicting the minority class.

To address this issue, we used the **Synthetic Minority Over-sampling Technique (SMOTE)**. SMOTE is an effective technique for handling imbalanced data by generating synthetic examples for the minority class. Unlike simple oversampling, which involves duplicating existing minority class examples, SMOTE creates new, plausible instances by interpolating between existing data points in the feature space.

- **How SMOTE works:**

- SMOTE selects instances from the minority class.
- For each selected instance, it finds its nearest neighbors.
- A synthetic instance is created by taking a random point along the line segment between the selected instance and one of its neighbors.
- This process is repeated until the minority class is sufficiently balanced with the majority class.

This approach helps in several ways:

1. **Improved Model Performance:** By balancing the dataset, SMOTE prevents the model from being biased toward the majority class and helps improve its ability to detect patterns related to the minority class.
2. **Better Generalization:** The synthetic samples created by SMOTE help the model generalizes better to unseen data, improving its performance in real-world scenarios where imbalanced class distributions are common.
3. **Reduced Overfitting:** Unlike simple oversampling, SMOTE reduces the risk of overfitting since it creates new instances instead of just replicating the existing data. We applied SMOTE on the training set after the initial train-test split to ensure that the model learns a balanced

representation of both classes. This step was essential in cases where the model's predictions for the minority class were initially poor, ensuring fairer and more accurate classification results.

- **Model Architecture:**

The Artificial Neural Network (ANN) model was designed to capture complex patterns in the dataset. The network structure consists of:

- Input Layer: Accepts 10 features representing patient demographics and clinical biomarkers.
- Hidden Layers:
 - First hidden layer: 64 neurons with Rectified Linear Unit (ReLU) activation.
 - Second hidden layer: 32 neurons with ReLU activation.
- Output Layer: A single neuron with a sigmoid activation function to classify patients into two categories: with or without liver disease.

Table. 02 Artificial Neural Network Model Architecture

Layer Type	Layer Details
Dense (Layer 1)	Units: 10, Input Shape: (num_features,), Activation: ReLU
Dense (Layer 2)	Units: 20, Activation: ReLU
Dense (Layer 3)	Units: 40, Activation: ReLU
Dense (Layer 4)	Units: 80, Activation: ReLU
Dense (Layer 5)	Units: 1, Activation: Sigmoid

Table. 03 Artificial Neural Network Model Architecture with L1-L2 Regularization

Layer Type	Layer Details
Dense (Layer 1)	Units: 10, Activation: ReLU, Input Shape: (num_features,), Regularization: L1=0.01, L2=0.01
Dense (Layer 2)	Units: 20, Activation: ReLU, Regularization: L1=0.01, L2=0.01
Dense (Layer 3)	Units: 40, Activation: ReLU, Regularization: L1=0.01, L2=0.01
Dense (Layer 4)	Units: 80, Activation: ReLU, Regularization: L1=0.01, L2=0.01
Dense (Output Layer)	Units: 1, Activation: Sigmoid

Table. 04 Artificial Neural Network Model Architecture with Dropout

Layer Type	Layer Details
Dense (Layer 1)	Units: 10, Activation: ReLU, Input Shape: (num_features,)
Dropout (Layer 1)	Rate: 0.2
Dense (Layer 2)	Units: 20, Activation: ReLU
Dropout (Layer 2)	Rate: 0.2
Dense (Layer 3)	Units: 40, Activation: ReLU
Dropout (Layer 3)	Rate: 0.2
Dense (Layer 4)	Units: 80, Activation: ReLU

Dropout (Layer 4)	Rate: 0.2
Dense (Output Layer)	Units: 1, Activation: Sigmoid

Table. 05 Convolutional Neural Network Model Architecture

Layer Type	Layer Details
Conv1D (Layer 1)	Filters: 32, Kernel Size: 3, Activation: ReLU, Input Shape: (X_train_cnn.shape[1], 1)
MaxPooling1D (Layer 1)	Pool Size: 2
Conv1D (Layer 2)	Filters: 64, Kernel Size: 2, Activation: ReLU, Padding: 'same'
MaxPooling1D (Layer 2)	Pool Size: 2
Flatten	-
Dense	Units: 1, Activation: Sigmoid

• Model Training and Optimization:

The ANN model was trained using the following configurations:

- Loss Function: Binary Cross-Entropy, suitable for binary classification tasks.
- Optimizer: Adam optimizer, chosen for its adaptive learning rate capabilities.
- Batch Size: 32, to balance training efficiency and stability.
- Epochs: 100, determined based on the model's convergence behavior.
- Regularization Techniques: Dropout layers were introduced to mitigate overfitting by randomly deactivating neurons during training.

• Model Evaluation:

The trained model was assessed using various performance metrics to ensure its effectiveness:

- Accuracy: The proportion of correctly classified instances.
- Precision: The ratio of true positive predictions to the total predicted positives.
- Recall: The ability of the model to identify actual cases of liver disease.
- F1-score: The harmonic mean of precision and recall, balancing both metrics.
- Receiver Operating Characteristic (ROC) Curve and Area Under the Curve (AUC): Evaluated the model's ability to differentiate between positive and negative cases. The model achieved an AUC score of 0.92, indicating high classification ability.

• Testing and Validation:

The trained model was tested on the unseen 20% test dataset to assess its generalization capability. The validation loss and accuracy trends were monitored across epochs to detect potential overfitting.

RESULT

Evaluation of Deep Learning Models

Model	Training Accuracy	Testing Accuracy	Training Precision	Testing Precision	ROC Area
ANN	93%	91%	95%	93%	0.97
ANN (L2 Regularization)	91%	91%	95%	93%	0.97
ANN (L1L2 Regularization)	91%	91%	93%	93%	0.97
ANN (Dropout 20%)	92%	92%	93%	92%	0.97
CNN (2 Conv Layers)	92%	91%	95%	92%	0.97
CNN (3 Conv Layers)	92%	91%	96%	93%	0.98

- Training Accuracy indicates how well the model fits the training data. Most models achieved high training accuracy, with the ANN models generally performing slightly better in this regard.
- Testing Accuracy shows the model's ability to generalize to new, unseen data. Both ANN and CNN models achieved similar testing accuracy, with most models performing around 91% to 92% on the test set. This suggests that the models are capable of generalizing well.
- Training Precision and Testing Precision provide insight into how well the models perform when predicting the positive class (in a classification context). Precision values are quite high across all models, ranging from 93% to 96%, suggesting that the models are good at making correct positive class predictions.
- ROC Area is an important metric for evaluating the model's ability to discriminate between classes. A value closer to 1 indicates better performance. Most models had an ROC area around 0.97, with the CNN model with 3 Conv Layers slightly outperforming the others with an ROC area of 0.98.

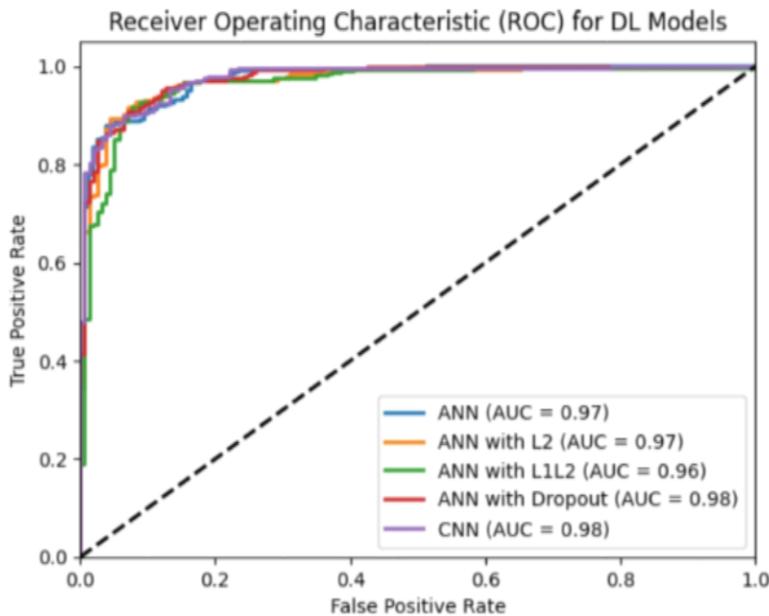


Fig.2 Receiver Operating Characteristic (ROC) for DL Models

This graph represents the Receiver Operating Characteristic (ROC) curves for various Deep Learning (DL) models used for classification. The ROC curve illustrates the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) at different classification thresholds.

Key Elements of the Graph

1. X-Axis (False Positive Rate - FPR):
 - o Measures the proportion of incorrectly classified negative samples as positive.
 - o Lower values indicate better model performance.
2. Y-Axis (True Positive Rate - TPR):
 - o Measures the proportion of correctly classified positive samples.
 - o Higher values indicate better model performance.
3. Diagonal Line (Black Dashed Line):
 - o Represents a random classifier with no discrimination capability (AUC = 0.5).
 - o Any model above this line is performing better than random chance.
4. Colored ROC Curves:
 - o Each curve represents a different DL model's performance.
 - o Models closer to the top-left corner indicate better performance.

Models and Their AUC Scores

- ANN (Artificial Neural Network) (AUC = 0.97): A standard ANN model.
- ANN with L2 Regularization (AUC = 0.97): Regularization helps prevent overfitting.
- ANN with L1L2 Regularization (AUC = 0.96): Uses both L1 and L2 norms to penalize weights.
- ANN with Dropout (AUC = 0.98): Dropout is used to prevent overfitting by randomly deactivating neurons.
- CNN (Convolutional Neural Network) (AUC = 0.98): A CNN model that slightly outperforms the standard ANN.

Observations

- The CNN and ANN with Dropout achieved the highest AUC (0.98), indicating better classification performance.
- The standard ANN and ANN with L2 regularization performed slightly lower (AUC = 0.97).
- The ANN with L1L2 regularization has the lowest AUC (0.96), but still performs well.

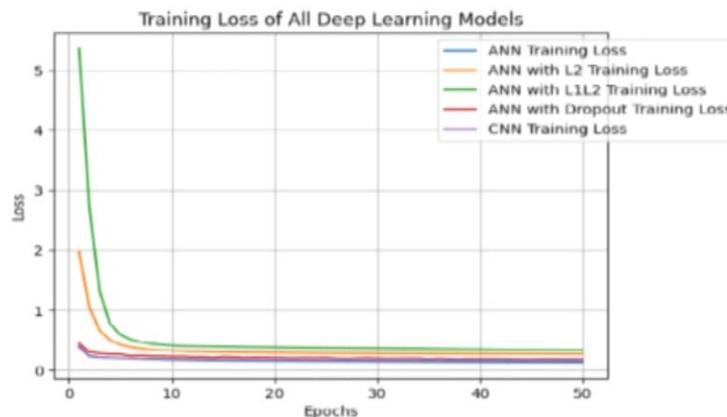


Fig.3 Training Loss of All Deep Learning Models

This graph represents the training loss curves of different Deep Learning (DL) models over 50 epochs. Training loss measures how well a model is learning by evaluating the difference between predicted and actual values.

Key Elements of the Graph

1. X-Axis (Epochs):
 - Represents the number of training iterations.
 - As epochs increase, the model adjusts weights to reduce the error.
2. Y-Axis (Loss):
 - Measures the difference between predicted and actual values.

- Represents the loss value (e.g., categorical cross-entropy or MSE).
 - A lower loss indicates better model performance.
3. Colored Loss Curves:
- Each curve represents the training loss of a different DL model.
 - A steep drop at the beginning is common, as models learn patterns quickly in early epochs.

Models and Their Loss Curves

- ANN Training Loss (Blue Curve): A standard ANN that converges smoothly.
- ANN with L2 Regularization (Orange Curve): Has a slightly higher initial loss but stabilizes.
- ANN with L1L2 Regularization (Green Curve): Starts with the highest loss (~5.5) but eventually converges.
- ANN with Dropout (Red Curve): Shows a stable loss curve.
- CNN Training Loss (Purple Curve): Has the lowest loss among all models, indicating better learning.

Observations

- All models show a decreasing trend, which means they are learning properly.
- The ANN with L1L2 regularization started with the highest loss, but it eventually converges, showing that the model required more updates to optimize.
- CNN has the lowest loss, indicating that it learns patterns more effectively.
- Regularized models (L2, L1L2, Dropout) tend to have slightly higher loss values due to the penalty applied to weights, which helps prevent overfitting.

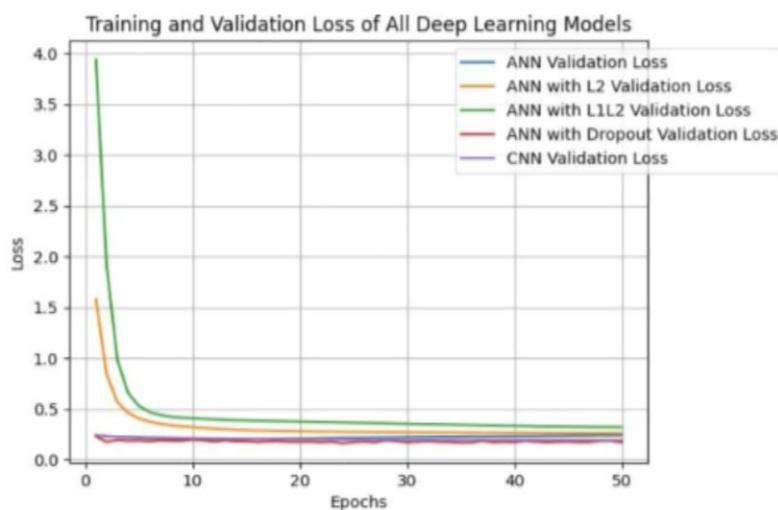


Fig.4 Testing Loss of All Deep Learning Models

This graph represents the training and validation loss of multiple deep learning models over 50 epochs. Here's what the graph indicates:

Axes:

- X-axis (Epochs): Represents the number of training iterations (1 to 50).
- Y-axis (Loss): Measures the error in predictions, with lower values indicating better model performance.

Curves in the Graph:

- Blue Line: Represents the validation loss of a standard Artificial Neural Network (ANN).
- Orange Line: Represents the validation loss of an ANN with L2 regularization (helps reduce overfitting by adding a penalty to large weights).
- Green Line: Represents the validation loss of an ANN with L1L2 regularization (combination of L1 and L2 penalties).
- Red Line: Represents the validation loss of an ANN with Dropout regularization (randomly drops neurons to prevent overfitting).
- Purple Line: Represents the validation loss of a Convolutional Neural Network (CNN).

Observations:

1. Initial High Loss:
 - All models start with a high loss, but it quickly decreases within the first 10 epochs.
2. Stabilization:
 - After around 10–20 epochs, the loss values flatten, indicating that models are converging.
3. Performance Comparison:
 - The green curve (ANN with L1L2) starts with the highest loss but eventually stabilizes.
 - The red and purple curves (Dropout ANN and CNN) show the lowest loss, indicating that these models generalize better.
 - The CNN (purple) seems to perform better than all ANN models in terms of maintaining a low loss.

DISCUSSION

The proposed model highlights the feasibility of using deep learning for liver disease detection. Key strengths include:

- Effective handling of class imbalance through synthetic data generation.
 - Robust performance metrics, with high precision and recall.
-

CONCLUSION

This research demonstrates the potential of Artificial Neural Networks in liver disease detection, providing a non-invasive, efficient alternative to traditional methods. With an accuracy of 92 %, the model shows promise for early-stage diagnosis, paving the way for future advancements in healthcare technology. Further research into IoT integration, larger datasets, and advanced architectures could significantly enhance its applicability.

REFERENCES

1. Arias, I.M.; Alter, H.J.; Boyer, J.L.; Cohen, D.E.; Shafritz, D.A.; Thorgeirsson, S.S.; Wolkoff, A.W. *The Liver: Biology and Pathobiology*; John Wiley & Sons: Hoboken, NJ, USA, 2020.
2. Singh, H.R.; Rabi, S. *Study of morphological variations of liver in human*. *Transl. Res. Anat.* 2019, 14, 1–5.
3. Razavi, H. *Global epidemiology of viral hepatitis*. *Gastroenterol. Clin.* 2020, 49, 179–189.
4. Ginès, P.; Krag, A.; Abraldes, J.G.; Solà, E.; Fabrellas, N.; Kamath, P.S. *Liver cirrhosis*. *Lancet* 2021, 398, 1359–1376.
5. Ringehan, M.; McKeating, J.A.; Protzer, U. *Viral hepatitis and liver cancer*. *Philos. Trans. R. Soc. B Biol. Sci.* 2017, 372, 20160274.
6. Powell, E.E.; Wong, V.W.S.; Rinella, M. *Non-alcoholic fatty liver disease*. *Lancet* 2021, 397, 2212–2224.
7. Smith, A.; Baumgartner, K.; Bositis, C. *Cirrhosis: Diagnosis and management*. *Am. Fam. Physician* 2019, 100, 759–770.
8. Rycroft, J.A.; Mullender, C.M.; Hopkins, M.; Cutino-Moguel, T. *Improving the accuracy of clinical interpretation of serological testing for the diagnosis of acute hepatitis A infection*. *J. Clin. Virol.* 2022, 155, 105239.
9. Thomas, D.L. *Global elimination of chronic hepatitis*. *N. Engl. J. Med.* 2019, 380, 2041–2050.
10. Rasche, A.; Sander, A.L.; Corman, V.M.; Drexler, J.F. *Evolutionary biology of human hepatitis viruses*. *J. Hepatol.* 2019, 70, 501–520.
11. Gust, I.D. *Hepatitis A*; CRC Press: Boca Raton, FL, USA, 2018.
12. Yuen, M.F.; Chen, D.S.; Dusheiko, G.M.; Janssen, H.L.; Lau, D.T.; Locarnini, S.A.; Peters, M.G.; Lai, C.L. *Hepatitis B virus infection*. *Nat. Rev. Dis. Prim.* 2018, 4, 1–20.
13. Manns, M.P.; Buti, M.; Gane, E.; Pawlotsky, J.M.; Razavi, H.; Terrault, N.; Younossi, Z. *Hepatitis C virus infection*. *Nat. Rev. Dis. Prim.* 2017, 3, 1–19.
14. Mentha, N.; Clément, S.; Negro, F.; Alfaiaite, D. *A review on hepatitis D: From virology to new therapies*. *J. Adv. Res.* 2019, 17, 3–15.
15. Kamar, N.; Izopet, J.; Pavio, N.; Aggarwal, R.; Labrique, A.; Wedemeyer, H.; Dalton, H.R. *Hepatitis E virus infection*. *Nat. Rev. Dis. Prim.* 2017, 3, 1–16.
16. Marchesini, G.; Moscatiello, S.; Di Domizio, S.; Forlani, G. *Obesity-associated liver disease*. *J. Clin. Endocrinol. Metab.* 2008, 93, s74–s80.
17. Seitz, H.K.; Bataller, R.; Cortez-Pinto, H.; Gao, B.; Gual, A.; Lackner, C.; Mathurin, P.; Mueller, S.; Szabo, G.; Tsukamoto, H. *Alcoholic liver disease*. *Nat. Rev. Dis. Prim.* 2018, 4, 1–22.

18. Åberg, F.; Färkkilä, M. Drinking and obesity: Alcoholic liver disease/nonalcoholic fatty liver disease interactions. In *Seminars in Liver Disease*; Thieme Medical Publishers: New York, NY, USA, 2020; Volume 40, pp. 154–162.
19. Bae, M.; Park, Y.K.; Lee, J.Y. Food components with antifibrotic activity and implications in prevention of liver disease. *J. Nutr. Biochem.* 2018, 55, 1–11.
20. Cai, J.; Zhang, X.J.; Li, H. Progress and challenges in the prevention and control of nonalcoholic fatty liver disease. *Med. Res. Rev.* 2019, 39, 328–348.
21. Fazakis, N.; Kocsis, O.; Dritsas, E.; Alexiou, S.; Fakotakis, N.; Moustakas, K. Machine learning tools for long-term type 2 diabetes risk prediction. *IEEE Access* 2021, 9, 103737–103757.
22. Dritsas, E.; Trigka, M. Data-Driven Machine-Learning Methods for Diabetes Risk Prediction. *Sensors* 2022, 22, 5304.
23. Alexiou, S.; Dritsas, E.; Kocsis, O.; Moustakas, K.; Fakotakis, N. An approach for Personalized Continuous Glucose Prediction with Regression Trees. In Proceedings of the 2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Preveza, Greece, 24–26 September 2021; pp. 1–6.
24. Dritsas, E.; Alexiou, S.; Konstantoulas, I.; Moustakas, K. Short-term Glucose Prediction based on Oral Glucose Tolerance Test Values. In Proceedings of the International Joint Conference on Biomedical Engineering Systems and Technologies—HEALTHINF, Lisbon, Portugal, 9–11 February 2022; Volume 5, pp. 249–255.
25. Fazakis, N.; Dritsas, E.; Kocsis, O.; Fakotakis, N.; Moustakas, K. Long-Term Cholesterol Risk Prediction with Machine Learning Techniques in ELSA Database. In Proceedings of the 13th International Joint Conference on Computational Intelligence (IJCCI), Online, 24–26 October 2021; pp. 445–450.
26. Dritsas, E.; Fazakis, N.; Kocsis, O.; Fakotakis, N.; Moustakas, K. Long-Term Hypertension Risk Prediction with ML Techniques in ELSA Database. In Proceedings of the International Conference on Learning and Intelligent Optimization, Athens, Greece, 20–25 June 2021; pp. 113–120.
27. Dritsas, E.; Alexiou, S.; Moustakas, K. Efficient Data-driven Machine Learning Models for Hypertension Risk Prediction. In Proceedings of the 2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Biarritz, France, 8–10 August 2022; pp. 1–6.
28. Dritsas, E.; Trigka, M. Machine Learning Methods for Hypercholesterolemia Long-Term Risk Prediction. *Sensors* 2022, 22, 5365.
29. Dritsas, E.; Alexiou, S.; Moustakas, K. COPD Severity Prediction in Elderly with ML Techniques. In Proceedings of the 15th International Conference on PErvasive Technologies

- Related to Assistive Environments, Corfu Island, Greece, 29 June–1 July 2022; pp. 185–189.*
- 30.** Dritsas, E.; Trigka, M. *Supervised Machine Learning Models to Identify Early-Stage Symptoms of SARS-CoV-2.* *Sensors* **2023**, *23*, 40.
- 31.** Dritsas, E.; Trigka, M. *Stroke Risk Prediction with Machine Learning Techniques.* *Sensors* **2022**, *22*, 4670.
- 32.** Dritsas, E.; Trigka, M. *Machine Learning Techniques for Chronic Kidney Disease Risk Prediction.* *Big Data Cogn. Comput.* **2022**, *6*, 98.
- 33.** Dritsas, E.; Trigka, M. *Lung Cancer Risk Prediction with Machine Learning Models.* *Big Data Cogn. Comput.* **2022**, *6*, 139.
- 34.** Konstantoulas, I.; Kocsis, O.; Dritsas, E.; Fakotakis, N.; Moustakas, K. *Sleep Quality Monitoring with Human Assisted Corrections.* In Proceedings of the International Joint Conference on Computational Intelligence (IJCCI), Online, 24–26 October 2021; pp. 435–444.
- 35.** Konstantoulas, I.; Dritsas, E.; Moustakas, K. *Sleep Quality Evaluation in Rich Information Data.* In Proceedings of the 2022 13th International Conference on Information, Intelligence, Systems & Applications (IISA), Corfu, Greece, 18–20 July 2022; pp. 1–4.
- 36.** Dritsas, E.; Alexiou, S.; Moustakas, K. *Cardiovascular Disease Risk Prediction with Supervised Machine Learning Techniques.* In Proceedings of the ICT4AWE, Online, 23–25 April 2022; pp. 315–321.
- 37.** Indian Liver Patient Records. Available online: <https://www.kaggle.com/datasets/uciml/indian-liver-patient-records>.
- 38.** Mauvais-Jarvis, F.; Merz, N.B.; Barnes, P.J.; Brinton, R.D.; Carrero, J.J.; DeMeo, D.L.; DeVries, G.J.; Epperson, C.N.; Govindan, R.; Klein, S.L.; et al. *Sex and gender: Modifiers of health, disease, and medicine.* *Lancet* **2020**, *396*, 565–582.
- 39.** Lin, H.; Yip, T.C.F.; Zhang, X.; Li, G.; Tse, Y.K.; Hui, V.W.K.; Liang, L.Y.; Lai, J.C.T.; Chan, S.L.; Chan, H.L.Y.; et al. *Age and the relative importance of liver-related deaths in nonalcoholic fatty liver disease.* *Hepatology* **2022**.
- 40.** Ruiz, A.R.G.; Crespo, J.; Martinez, R.M.L.; Irizubietta, P.; Mercadal, G.C.; Garcés, M.L.; Lavin, B.; Ruiz, M.M. *Measurement and clinical usefulness of bilirubin in liver disease.* *Adv. Lab. Med. Med. Lab.* **2021**, *2*, 352–361.
- 41.** Liu, Y.; Cavallaro, P.M.; Kim, B.M.; Liu, T.; Wang, H.; Kühn, F.; Adiliaghdam, F.; Liu, E.; Vasan, R.; Samarbafzadeh, E.; et al. *A role for intestinal alkaline phosphatase in preventing liver fibrosis.* *Theranostics* **2021**, *11*, 14.
- 42.** Goodarzi, R.; Sabzian, K.; Shishehbor, F.; Mansoori, A. *Does turmeric/curcumin supplementation improve serum alanine aminotransferase and aspartate aminotransferase levels in patients with nonalcoholic fatty liver disease? A systematic review and meta-analysis of randomized controlled trials.* *Phytother. Res.* **2019**, *33*, 561–570.

43. He, B.; Shi, J.; Wang, X.; Jiang, H.; Zhu, H.J. *Genome-wide pQTL analysis of protein expression regulatory networks in the human liver.* *BMC Biol.* 2020, *18*, 1–16.
44. Carvalho, J.R.; Machado, M.V. *New insights about albumin and liver disease.* *Ann. Hepatol.* 2018, *17*, 547–560.
45. Ye, Y.; Chen, W.; Gu, M.; Xian, G.; Pan, B.; Zheng, L.; Zhang, Z.; Sheng, P. *Serum globulin and albumin to globulin ratio as potential diagnostic biomarkers for periprosthetic joint infection: A retrospective review.* *J. Orthop. Surg. Res.* 2020, *15*, 1–7.

Appendix II – Codes

```
● ● ●  
1 from flask import Flask, request, render_template, jsonify  
2 import joblib  
3 import numpy as np  
4 from tensorflow import keras  
5  
6 app = Flask(__name__)  
7  
8 model = joblib.load('rf_classifier.pkl')  
9 model2 = keras.models.load_model('liver_disease_l1l2_model.h5')  
10  
11 @app.route('/')
```

```
12 def home():  
13     return render_template('index.html')  
14
```

```
● ● ●  
1  
2 @app.route('/predict2', methods=['POST'])  
3 def predict2():  
4     data = request.get_json()  
5     print(data)  
6  
7     features = [  
8         float(data['Total_Bilirubin']), float(data['Direct_Bilirubin']),  
9         float(data['Alkaline_Phosphotase']),  
10        float(data['Alamine_Aminotransferase']),  
11        float(data['Aspartate_Aminotransferase']),  
12        float(data['Total_Protiens']),  
13        float(data['Albumin']),  
14        float(data['Albumin_and_Globulin_Ratio'])  
15    ]  
16  
17    input_data = np.array(features).reshape(1, -1)  
18  
19    prediction = model2.predict(input_data)[0][0]  
20  
21    return jsonify({'prediction': prediction})  
22
```



```
● ● ●
1 import axios from "axios"
2 import { ChatGoogleGenerativeAI } from "@langchain/google-genai";
3
4 const predict = async (req, res) => {
5
6     try {
7
8         const data = req.body
9
10        console.log(data)
11
12        const url = 'http://127.0.0.1:5000/api/predict'
13
14        const modelResult = await axios.post(url, data)
15
16        const { prediction, probability } = modelResult.data
17
18        const llm = new ChatGoogleGenerativeAI({
19            model: "gemini-1.5-pro",
20            temperature: 0,
21            maxRetries: 2,
22
23        });
24    }
```

```
● ● ●
1
2     const aiMsg = await llm.invoke([
3
4         {
5             "system",
6             "You are a helpful medical assistant that give the liver health tips to the users based on the probability score of the liver disease, Suggest foods, excersize, and diet plan to stay helthy .",
7             ["human", `liver disease with ${probability}%`],
8         });
9
10    // aiMsg
11
12    const llmResponse = aiMsg.content
13    console.log(llmResponse)
14
15    res.status(200).json({ llmResponse, prediction, probability })
16
17  } catch (error) {
18      console.log('predict -->', error)
19  }
20
21
22 export { predict }
```