Algerian Journal of Research and Technology
https://doi.org/10.58681/ajrt.25090106
A.J.R.T Volume 9 (N°1) (2025) 50– 66

An Empirical Study on the Effectiveness and Efficiency of Machine Learning Classifiers for Liver Disease Prediction

Mohamed Amine NEMMICH [1], Asmaa BOUDALI [2], Noureddine BOUKHARI [3], Fatima DEBBAT [4]

[1]*Department of Computer Science, Mathematics Laboratory, Djillaliliabes University of SidiBel Abbes, SidiBel Abbes, Algeria*
[2]*Department of Electronics, Laboratory of coding and security of information, Sciences and Technology University of Oran Mohamed Boudiaf, Oran, Algeria*
[3]*Department of Mathematics, Djillaliliabes University of SidiBel Abbes, SidiBel Abbes, Algeria*
[4]*Department of Computer Science, Mustapha Stambouli University of Mascara, Mascara, Algeria*

*Corresponding author. amine.nemmich@univ-sba.dz; *asmaa.boudali@univ-usto.dz*; *noureddine.boukhari@univ-sba.dz*; ebbat.fatima@univ-mascara.dz

Received. June 17, 2024. Accepted. December 06, 2024. Published. December 20, 2024.

DOI: https://doi.org/10.58681/ajrt.25090106

Abstract.Liver disease poses a significant global health burden, with high mortality rates exacerbated by challenges in early detection. Machine learning (ML) offers promising avenues for developing automated diagnostic tools to address this critical need. While various ML classifiers have been explored for liver disease prediction, a comprehensive, systematic comparison of a wide range of modern algorithms, incorporating robust pre-processing, handling of class imbalance, hyper parameter tuning with cross-validation, and analysis of computational efficiency, is essential to guide the selection of models for practical application. This study systematically evaluates thirteen diverse ML classification algorithms using the Liver Patient Dataset (LDPD). The methodology includes data pre-processing with imputation, encoding, and standardization within a pipeline to prevent data leakage, handling class imbalance using SMOTE, splitting data into training and testing sets, and employing RandomizedSearchCV with Stratified K-Fold cross-validation for hyper parameter optimization. Performance was assessed using key metrics including Accuracy, Precision, Recall, Specificity, F1-Score, and ROC AUC on an independent test set, alongside training time. Results demonstrate that ensemble and advanced tree-based methods achieve superior predictive performance. Hyper parametertuning further optimized performance, with Tuned Random Forest achieving the highest ROC AUC

(0.9995) and Specificity (0.9973), and Tuned LightGBM achieving the highest Recall (0.9996). The study highlights a crucial trade-off: while tuning yields peak performance, default configurations of efficient models like LightGBM and XGBoost offer exceptionally high performance (ROC AUC ≥ 0.9993) combined with significantly faster training times (≤ 0.41 seconds), providing a favorable balance for practical application. This research identifies highly effective and efficient ML models for liver disease prediction, contributing empirical evidence to support the development of automated diagnostic aids.

Keywords.Liver Disease Prediction, Machine Learning Classification,Class Imbalance, Hyperparameter Tuning, Ensemble Methods.

INTRODUCTION

Liver disease represents a significant global health challenge, contributing to substantial morbidity and mortality worldwide. As highlighted by recent data, the burden of liver disease is particularly acute in regions like India, where 264,193 deaths were reported in 2018, corresponding to an age-adjusted death rate of approximately 23.00 per 100,000 population [World Life Expectancy, 2022]. The liver, a vital organ responsible for detoxification and numerous metabolic functions, is susceptible to damage from various etiologies, including viral infections, metabolic disorders, excessive alcohol consumption, and genetic factors [SindhujaandPriyadarsini, 2016;Md et al., 2023]. While conditions like cirrhosis and liver failure represent advanced stages, early detection of liver damage is often challenging due to its insidious progression and non-specific initial symptoms [Md et al., 2023]. This delayed identification can severely limit therapeutic options and negatively impact patient outcomes, underscoring the critical need for timely and accurate diagnostic tools to facilitate early intervention and improve prognosis [Shaheamlung, KaurandKaur, 2020].

The growing availability of health data and advancements in computational capabilities have positioned machine learning (ML) as a powerful paradigm for enhancing medical diagnosis and prognosis [Md et al., 2023]. Classification techniques, in particular, have shown promise in developing automated tools for identifying various diseases based on patient data. In the context of liver disease, ML algorithms have been explored for tasks such as classifying liver fibrosis stages, predicting patient survival, and distinguishing between different liver conditions [Md et al., 2023]. However, the landscape of ML applications in liver disease prediction is continuously evolving. While numerous studies have investigated various algorithms, there remains a need for comprehensive, head-to-head comparisons of a wide array of modern and diverse ML classifiers on relevant datasets. Furthermore, the impact of critical steps like systematic data preprocessing, effective handling of class imbalance, and rigorous hyperparameter tuning on the performance of these models for liver disease prediction warrants further investigation to identify the most robust and reliable approaches for potential clinical application.

This study aims to address these gaps by conducting a systematic and comprehensive evaluation of multiple machine learning classification algorithms for liver disease prediction using a publicly available dataset. The primary objectives are: (1) to benchmark the performance of a diverse set of ML classifiers; (2) to identify the most effective and efficient models for this prediction task based on a thorough analysis of various performance metrics, including those crucial in medical diagnosis such as Recall and Specificity, alongside overall discrimination ability (ROC AUC) and computational efficiency (training time). The rationale behind this research is to provide a data-driven comparison to guide the selection of suitable ML models for developing automated liver disease screening or diagnostic support tools. This work contributes to the field by offering a detailed comparative analysis of numerous

algorithms, demonstrating the practical impact of different techniques, and highlighting the trade-offs between model performance and efficiency in the context of liver disease prediction. The implemented methodology involves standard data preprocessing techniques, addressing class imbalance using SMOTE, splitting the data into training and testing sets, training and evaluating a broad range of classifiers, and conducting a staged performance comparison analysis of both models.

The remainder of this paper is organized as follows: Section 2 presents a review of the existing literature on machine learning applications in liver disease classification and detection. Section 3 provides a detailed explanation of the dataset, the proposed architecture, the algorithms utilized, and the preprocessing steps. Section 4 describes the experimental setup and presents the evaluation results. Section 5 discusses the conclusion and outlines potential directions for future work.

LITERATURE REVIEW

This section reviews existing research on applying machine learning classification techniques for liver disease prediction and diagnosis, focusing on commonly used algorithms, datasets, and key findings to establish the context for this study.

Machine learning models such as Support Vector Machines (SVM), Logistic Regression, Naïve Bayes, Decision Trees (DT), Random Forest, K-Nearest Neighbors (KNN), and Artificial Neural Networks (ANN), along with various boosting algorithms, have been widely applied to classify liver diseases [Ramana et al., 2011]. Comparative studies on datasets like the Andhra Pradesh (AP), UCLA, UCI, and Indian Liver Patient Dataset (ILPD) show varied results regarding the best-performing algorithms. Some studies found KNN, backward propagation (a type of ANN), and SVM to be effective [Ramana et al.,2011], while others highlighted Decision Trees [Kumar andSahoo, 2013;Ayeldeen et al., 2015], C4.5 [Hashem et al., 2018, 1Durai et al., 2019], ANN [Sivakumar et al., 2019], or Bayesian networks [Jacob et al., 2018] as top performers in specific comparisons or on particular datasets. The influence of the dataset itself on model performance has also been noted [Ramana et al., 2011 Ramana et al., 2012].

Researchers have also explored specific techniques and algorithms. Studies have compared models like SVM and backpropagation [Ma et al., 2018], focused on predicting specific conditions like fibrosis [Ayeldeen et al., 2015; Sontakke et al., 2017] or fatty liver disease [Jacob et al., 2018], and investigated the utility of risk factors [Wu et al., 2019]. Techniques such as feature selection [Ramana et al., 2012 ;Gogi, 2018 ; GeethaandArunachalam, 2021], and data normalization [Gogi, 2018] have been incorporated to improve model performance. While some work has focused on single algorithms with preprocessing and tuning [GeethaandArunachalam, 2021], the diverse findings across studies using different methodologies and datasets underscore the complexity of the problem and the lack of a universally agreed-upon optimal approach.

Despite the extensive research, a key gap in the literature is the need for comprehensive, systematic comparisons of a wide range of modern machine learning classifiers evaluated under a consistent and rigorous methodology. Many studies focus on a limited set of algorithms or lack detailed consideration of crucial steps like robust preprocessing, handling class imbalance (although SMOTE is used in some implementations, its systematic evaluation across models is needed), and the impact on a broad scale. Furthermore, a thorough analysis that considers not only predictive performance metrics but also practical factors like computational efficiency (training time) is often missing but essential for real-world application.

This study aims to address these gaps by providing a comprehensive and systematic evaluation of a wide array of machine learning classifiers. By employing a consistent

methodology, including robust preprocessing pipelines and SMOTE for imbalance handling, and evaluating models across a standard set of performance metrics including training time, this research offers a valuable comparative analysis to identify effective and efficient models for liver disease prediction, contributing empirical evidence to the field. Furthermore, the analysis will explicitly consider the computational efficiency (training time) alongside predictive performance metrics, providing valuable insights for the practical application of these models in liver disease prediction.

RESEARCH METHODOLOGY

This study adopted a systematic machine learning workflow to develop and evaluate predictive models for the classification of liver disease. The comprehensive methodology encompasses data acquisition, rigorous preprocessing, strategies for handling class imbalance, model training, hyperparameter tuning, and a comprehensive performance evaluation process. The specific steps are elaborated in the following subsections.

Data Acquisition and Initial Inspection

The initial phase involved the acquisition of the dataset, identified as the Liver Patient Dataset (LDPD), which contains patient-specific information and related medical parameters relevant to liver disease diagnosis. The fundamental characteristics of the dataset, including its demographic scope, total number of records, and distribution across liver patient and non-liver patient categories, as well as gender distribution, are summarized in Table 1. The dataset comprises ten predictor variables and one target variable. The predictor variables encompass demographic information (age, gender) and various biochemical markers related to liver function (Total Bilirubin, Direct Bilirubin, Alkaline Phosphatase, Alamine Aminotransferase (SGPT), Aspartate Aminotransferase (SGOT), Total Proteins, Albumin, and Albumin-to-Globulin Ratio). The target variable indicates the diagnosis as either 'Liver Patient' or 'Non Liver Patient', expert-labeled to facilitate supervised learning. Detailed information regarding each attribute, including measurement units, value ranges, means, and standard deviations, is provided in Table 2.

Table1.LDPD Dataset Description.

Demography	Total records	Liver patients	Not liver patients	Male	Female
Worldwide liver patients	30691	21917	8774	21986	7803

Table2. Attributes' Information of Dataset.

Attribute	Measurement unit	Value range	Mean	Std
Age (AG)	Years	4–90	44.107	15.981
Gender (GN)	Categorical	0 or 1	0.775	0.483
Total bilirubin (TB)	mg/dl	0.4–75	3.370	6.256
Direct bilirubin (DB)	mg/dl	0.1–19.7	1.528	2.870
Alkaline phosphatase (AP)	U/L	63–2110	289.075	238.538
Alanine aminotransferase (ALA)	U/L	10–2000	81.489	182.159
Aspartate aminotransferase (ASA)	U/L	10–4929	111.470	280.851
Total proteins (TP)	g/dl	2.7–9.6	6.480	1.082

Albumin (AL)	g/dl	0.9–5.5	3.130	0.792
Albumin and globulin ratio (AGR)	g/dl	0.3–2.8	0.943	0.323
Liver disease or not (LD or NLD)	Categorical	0 or 1	0.286	0.452

Upon loading the data into a structured format, a preliminary inspection was conducted to ascertain the dataset's overall structure and identify variable types (numerical and categorical). Basic descriptive statistics were reviewed to understand the distribution and central tendencies of the attributes. To gain deeper insights into data distribution patterns and the relationships between variables, particularly concerning the target variable, visual exploratory data analysis (EDA) techniques were employed, including the generation of histograms for individual attributes and pair plots to visualize attribute distributions and their relationships with the liver disease outcome. A critical assessment was also performed to identify the presence and extent of missing values across different features, which is a necessary precursor to data cleaning. Ensuring data quality by addressing such redundancies and inconsistencies, including the identification and potential handling of duplicate instances, is essential for improving the efficiency and reliability of subsequent modeling. Initial steps also involved recognizing the need to convert the categorical 'Gender' feature into a numerical format suitable for machine learning algorithms, which was performed through data encoding in a subsequent preprocessing step.

Data Preprocessing

Data preprocessing constituted a crucial stage focused on transforming the raw data into a clean, consistent, and numerically compatible format for machine learning, while strictly adhering to principles that prevent data leakage. This stage involved several key procedures. Missing values, identified during the initial inspection (the counts of which are detailed in Table 3), were handled through Imputation. Specifically, a Median Imputation strategy was applied to numerical features, replacing missing entries with the median value calculated solely from the training data subset to avoid test set influence. For the categorical 'Gender' feature, Mode Imputation was utilized to fill missing values with the most frequent category observed in the training subset. Categorical features, such as 'Gender', were converted into a numerical representation through One-Hot Encoding, creating binary indicator variables to ensure no ordinal relationship was incorrectly imposed. Furthermore, numerical features, which often exhibit widely varying scales, were subjected to Standardization (Z-score scaling). This technique transforms features to have a mean of zero and a standard deviation of one, standardizing their range. The Z-score method was also employed to address the presence of significant outliers observed in certain attributes, effectively neutralizing their disproportionate impact. Feature Scaling is a fundamental step for algorithms sensitive to feature magnitudes, ensuring that no single feature dominates the learning process, regardless of its original unit or range.

All these preprocessing steps—imputation, encoding, and scaling—were encapsulated within a Pre-processing Pipeline using scikit-learn's Pipeline and ColumnTransformer classes. This theoretical framework guarantees that all fitting of preprocessing parameters occurs exclusively on the training data, and these learned parameters are then applied consistently to transform both the training and independent test sets, rigorously preventing data leakage.

Table3. No of Missing Values in the Dataset.

AG	GN	TB	DB	AP	ALA	ASA	TP	AL	AGR

AG	GN	TB	DB	AP	ALA	ASA	TP	AL	AGR
278	0	648	561	796	739	859	463	494	559

Handling Class Imbalance

The dataset utilized in this study exhibited a notable imbalance in the distribution of the target variable, with a higher prevalence of the positive class (Liver Patient). Addressing this inherent class imbalance was a critical step to mitigate potential model bias towards the majority class. This was achieved through Oversampling of the minority class. Specifically, the Synthetic Minority Over-sampling Technique (SMOTE) was applied to the pre-processed training data. SMOTE is a synthetic oversampling algorithm that generates artificial instances of the minority class by interpolating between existing minority samples and their k-nearest neighbours in the feature space. This process, applied only to the pre-processed training data, resulted in a training dataset with a more balanced class distribution, thereby enabling the subsequent models to learn the characteristics of the minority class more effectively. The independent test set was kept in its original class distribution to ensure performance evaluation reflected real-world scenarios.

Model Training and Evaluation

To establish a baseline performance and identify algorithms with high potential, a diverse suite of thirteen machine learning classification models was initially selected and trained using their default hyperparameters. These models were chosen to represent a broad spectrum of theoretical approaches to classification, encompassing Generalized Linear Modeling (Logistic Regression), Instance-Based Learning (K-Nearest Neighbors), Decision Tree Learning, various Ensemble Methods based on Bagging (Random Forest, Extra Trees) and Boosting (Gradient Boosting Machines, XGBoost, LightGBM, AdaBoost, CatBoost), a Kernel Method (Support Vector Machine with an RBF kernel), a Probabilistic Model (Gaussian Naïve Bayes), and an Artificial Neural Network (Multi-Layer Perceptron). Each selected model underwent Model Training by being fitted to the SMOTE-resampled and preprocessed training data. The diversity in algorithm selection was intentional, designed to enrich the comparative study by evaluating models with distinct underlying mechanisms and potential strengths in capturing different patterns within the data. Following training, each model's performance was evaluated on the independent preprocessed test dataset.

Hyperparameter Tuning

Following the initial evaluation of models with default parameters, hyperparameter tuning was performed on a subset of the most promising models to further optimize their performance. This process utilized RandomizedSearchCV, a robust technique for efficiently searching a predefined hyperparameter space. To ensure a reliable estimate of performance during tuning and mitigate the risk of overfitting to a single validation set, Stratified K-Fold cross-validation was employed with 5 splits (k=5). Stratification ensured that each fold maintained a representative distribution of the target classes. The optimization criterion for RandomizedSearchCV was the ROC AUC score, which is a suitable metric for evaluating classifier performance on imbalanced datasets by assessing the model's ability to discriminate between positive and negative classes across various thresholds. The tuning process involved fitting the models with various combinations of hyperparameters sampled from specified distributions and evaluating them using cross-validation on the resampled training data. The best set of hyperparameters for each model was selected based on the highest mean cross-validation ROC AUC score.

EXPERIMENTAL RESULTS AND PERFORMANCE EVALUATION

The experimental evaluation was conducted on a ThinkPad L390 laptop equipped with an Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz, 24.0 GB RAM, and a 256GB SSD, running the Windows 10 Pro 64-bit operating system. The implementation, coding, and visualization were performed using Python within a Jupyter Notebook environment.

Performance Evaluation Metrics

The performance of the developed prediction models was assessed using a rigorous experimental protocol. The dataset was initially divided into an 80% training set and a 20% testing set using stratified random sampling to ensure that the proportion of target classes was maintained in both subsets. The confusion matrix served as the fundamental basis for performance evaluation, providing a detailed breakdown of classification outcomes: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). The confusion matrix components for all evaluated default algorithms are presented in Table 4.

Model performance was quantified using a suite of widely accepted evaluation metrics derived from the confusion matrix. These included Accuracy, Precision, Recall (Sensitivity), F1-Score, Specificity, and the Area Under the Receiver Operating Characteristic curve (ROC AUC). Table 5 shows the calculation of each evaluationmetric. In the context of medical diagnosis, the following metrics are particularly important:

- Recall (Sensitivity): The proportion of actual positive cases (Liver Patients) that were correctly identified. High Recall is crucial for minimizing false negatives, which is paramount in medical diagnosis to avoid missing true cases.
- Specificity: The proportion of actual negative cases (Non Liver Patients) that were correctly identified. High Specificity is important for minimizing false positives, preventing healthy individuals from being incorrectly diagnosed.
- F1-Score: The harmonic mean of Precision and Recall, providing a balanced measure particularly useful for imbalanced datasets.
- ROC AUC: An aggregate measure of the model's ability to discriminate between positive and negative classes across all possible classification thresholds. A higher AUC indicates superior discriminatory power, representing the trade-off between True Positive Rate and False Positive Rate.
- Accuracy: The overall proportion of correctly classified instances. While a general indicator, it is not the primary comparison metric due to the potential for misleading results in the presence of class imbalance.
- Precision: The proportion of instances predicted as positive that were actually positive.

In addition to these predictive performance metrics, the training time for each model was recorded to consider computational efficiency. This allows for an analysis of the trade-offs between model performance and the resources required for training. The performance evaluation was conducted on the independent preprocessed test dataset using both the default models and the tuned versions of selected classifiers.

Table4. Confusion Matrix.

Model	TN	FP	FN	TP
Logistic Regression	967	145	1221	1541
K-Nearest Neighbors	979	133	357	2405
Decision Tree	1065	47	1120	1642
Random Forest	1105	7	8	2754
Gradient Boosting Machines	1050	62	490	2272
XGBoost	1101	11	6	2756

LightGBM	1103	9	10	2752
Support Vector Machine	1046	66	1181	1581
Gaussian Naïve Bayes	1069	43	1647	1115
AdaBoost Classifier	944	168	653	2109
Extra Trees Classifier	1102	10	10	2752
CatBoost Classifier	1094	18	25	2737
Deep Learning	1080	32	627	2135

Table5. Performance Evaluation Metrics.

Metric	Calculation
Accuracy	$(TP+TN) / (TP+TN+FP+FN)$
Precision (P)	TP/(TP+FP)
Recall (R)	TP/(TP+FN)
F1-score	$2\times(P\times R) / (P+R)$
Specificity	TN/(TN+FP)
ROC curve	TPR (y-axis) vs. FPR (x-axis)

Default Model Performance

An initial evaluation was conducted by training a diverse set of thirteen classification models using their default hyperparameters on the SMOTE-resampled training data and assessing their performance on the independent test set [Shrivastava, 2024)]. In addition to standard performance metrics, the training time for each model was recorded to consider computational efficiency. Table 6 presents the key performance metrics and training duration for all default models, sorted by their ROC AUC score. The visualizing performance comparison for the 13 models is displayed in Fig. 1.

For comparing the performance of the different machine learning models in this study, we primarily focus on ROC AUC and F1-Score as robust overall indicators of performance on imbalanced data. Additionally, Recall (Sensitivity) and Specificityarecarefully examined to understand the critical trade-off between minimizing false negatives and false positives, which is paramount in a medical diagnostic context. The results reveal distinct tiers of performance and highlight the trade-offs between predictive power and computational cost (training time) at the default settings.

The highest predictive performance, as measured by ROC AUC and other key metrics, is concentrated among the ensemble and tree-based models: Random Forest (0.9994 ROC AUC), Extra Trees Classifier (0.9994 ROC AUC), LightGBM (0.9994 ROC AUC), XGBoost (0.9993 ROC AUC), and CatBoost Classifier (0.9985 ROC AUC). These models consistently achieved Accuracy, Precision, Recall, F1-Score, and Specificity exceeding 0.98. While their predictive capabilities at default settings are very similar and exceptionally high, significant differences emerge in their training times. LightGBM stands out as particularly efficient, training in just 0.19 seconds, followed by XGBoost (0.31s), Extra Trees (0.41s), CatBoost (0.97s), and Random Forest (0.99s). For practical applications where rapid retraining or development cycles are important, the speed offered by LightGBM, XGBoost, and Extra Trees is a notable advantage.

Beyond this top group, Gradient Boosting Machines (0.9588 ROC AUC, 3.96s) and the Deep Learning (MLP) model (0.9511 ROC AUC, 103.13s) show a considerable drop in ROC AUC and generally higher training times compared to the leading boosted trees. The MLP's training

time is dependent on hyperparameters like epochs and batch size, but even 50 epochs resulted in a relatively longer duration compared to most other default models.

Simpler models like Decision Tree (0.06s), K-Nearest Neighbors (0.07s), Logistic Regression (0.13s), and Gaussian Naïve Bayes (0.01s) exhibit significantly lower training times, often completing in milliseconds or a fraction of a second. Gaussian Naïve Bayes is the fastest to train. However, this efficiency comes at the cost of predictive performance, with ROC AUC values ranging from 0.7361 to 0.9406. Among these faster models, KNN achieves the best balance of speed and performance, with a respectable ROC AUC of 0.9406. The Support Vector Machine, while theoretically powerful, shows the longest training time (119.77s) at default settings with the RBF kernel, coupled with relatively modest performance metrics compared to the faster top models.

In summary, the default evaluation reveals a clear trade-off between training time and predictive performance. While the top ensemble methods demonstrate exceptional classification accuracy and discriminatory power, models like LightGBM, XGBoost, and Extra Trees offer a compelling combination of high performance and computational efficiency. Simpler models are faster but generally less accurate. This initial analysis guides the selection of models for the hyperparameter tuning phase, prioritizing those with high potential based on their default performance metrics, while also keeping computational cost in mind for practical considerations.

The Receiver Operating Characteristic (ROC) curve provides a visual representation of a classifier's ability to distinguish between positive and negative classes across various probability thresholds. The Area Under the ROC Curve (AUC) quantifies this discriminatory power, with values closer to 1indicating better performance. Fig.2 displays the ROC curves for all models evaluated at default settings. Notably, a distinct group of models—Random Forest, Extra Trees Classifier, LightGBM, XGBoost, and CatBoost Classifier—exhibits curves tightly positioned near the top-left corner of the plot, corresponding to exceptionally high AUC values ranging from 0.9985 to 0.9994. This visually confirms their superior discriminatory ability, achieving high True Positive Rates while maintaining low False Positive Rates across different thresholds.

Table6.Performance Evaluation of ML Models.

Model	Accuracy	Precision	Recall (Sensitivity)	F1-Score	Specificity	ROC AUC	Training Time (s)
Random Forest	0.9961	0.9975	0.9971	0.9973	0.9937	0.9994	0.99
Extra Trees Classifier	0.9948	0.9964	0.9964	0.9964	0.9910	0.9994	0.41
LightGBM	0.9951	0.9967	0.9964	0.9966	0.9919	0.9994	0.19
XGBoost	0.9956	0.9960	0.9978	0.9969	0.9901	0.9993	0.31
CatBoost Classifier	0.9889	0.9935	0.9909	0.9922	0.9838	0.9985	0.97
Gradient Boosting Machines	0.8575	0.9734	0.8226	0.8917	0.9442	0.9588	3.96
Deep Learning (MLP)	0.8299	0.9852	0.7730	0.8663	0.9712	0.9511	103.13
K-Nearest Neighbors	0.8735	0.9476	0.8707	0.9075	0.8804	0.9406	0.07
AdaBoost Classifier	0.7881	0.9262	0.7636	0.8371	0.8489	0.9005	1.75
Decision Tree	0.6988	0.9722	0.5945	0.7378	0.9577	0.8439	0.06
Support Vector Machine	0.6781	0.9599	0.5724	0.7172	0.9406	0.8152	119.77

Logistic Regression	0.6474	0.9140	0.5579	0.6929	0.8696	0.7644	0.13
Gaussian Naïve Bayes	0.5638	0.9629	0.4037	0.5689	0.9613	0.7361	0.01

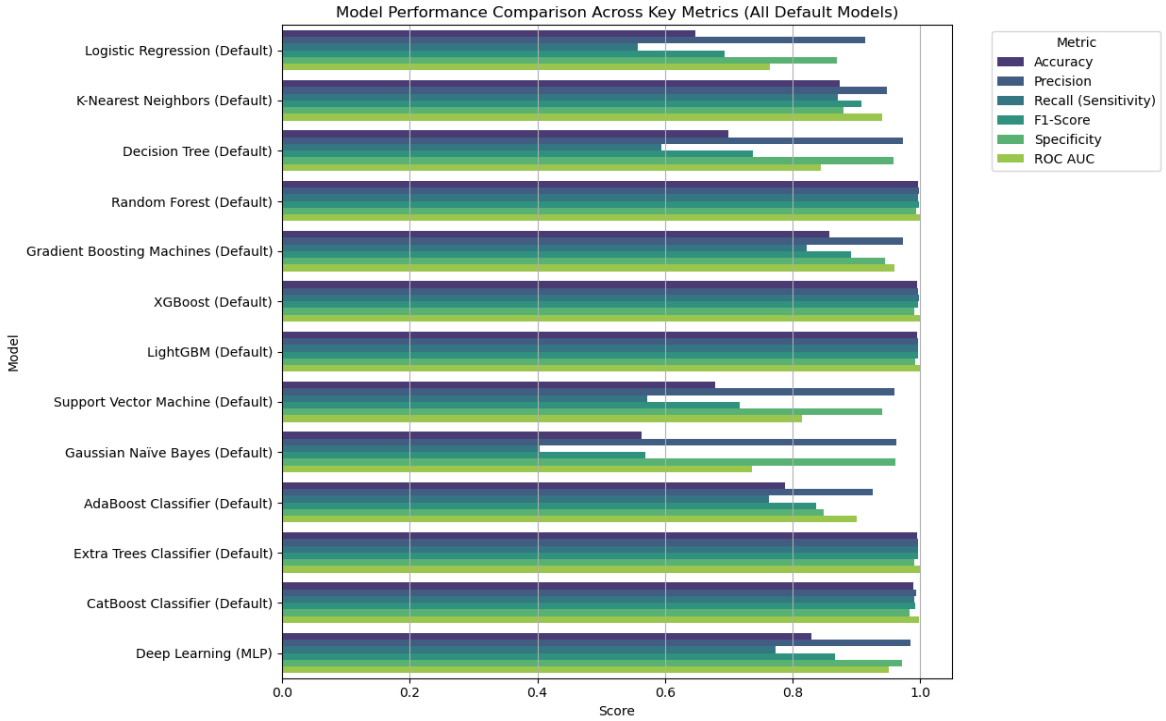

Fig. 1.Visualizing performance comparison (all models).

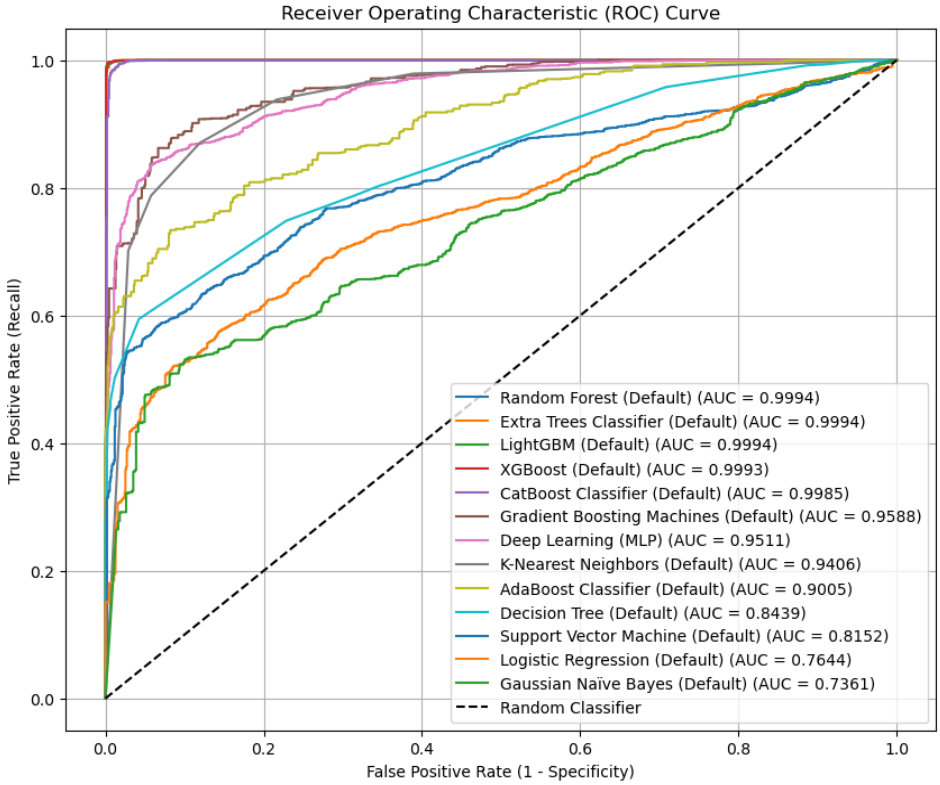

Fig. 2 ROC curves for all models evaluated.

Conversely,the remaining models show curves progressively closer to the diagonal random classifier line (AUC = 0.5), indicating lower discriminatory power. Models like Gradient Boosting Machines and the Deep Learning MLP perform moderately well (AUCs around

0.95), positioned below the top tier but still above random chance. Simpler models such as Logistic Regression and Gaussian Naïve Bayes yield curves closest to the diagonal, reflecting their limited capacity to separate the classes effectively compared to the more complex ensemble and neural network approaches. This ROC analysis visually reinforces that ensemble and advanced tree-based methods provide the strongest discrimination performance in this liver disease prediction task at default configurations.

Hyperparameter Tuning Results

Based on the promising performance of several models at their default settings, hyperparameter tuning was performed using RandomizedSearchCV with 5-fold Stratified K-Fold cross-validation, optimizing for ROC AUC. This process allowed for a more thorough exploration of the model's potential and provided a more statistically validated estimate of performance through cross-validation. Table 7 summarizes the best hyperparameters found and their corresponding best cross-validation ROC AUC scores for the selected models.

The high cross-validation ROC AUC scores achieved by the tuned models (all above 0.99) indicate that these models are consistently performing well across different subsets of the training data, providing statistical confidence in their predictive capability.

Final Performance Comparison (Top Default and Tuned Models)

For the final comparison, we selected the top 5 default models based on their initial ROC AUC from the default evaluation and included all models that underwent hyperparameter tuning. These models were then evaluated on the independent test set. Table 8 presents a comprehensive comparison of their performance metrics and training times, sorted by ROC AUC score in descending order.The visualizing performance comparison for the top default and tuned models is displayed in Fig. 3.

Discussion of Results

The final comparison, sorted by ROC AUC (Table 8), highlights that both the top performing default models and their tuned counterparts achieve exceptionally high performance metrics for liver disease prediction on this dataset. Specifically, the tuned versions of Random Forest, LightGBM, and XGBoost, along with the default versions of Random Forest, Extra Trees, and LightGBM, demonstrate the highest ROC AUC scores, all at 0.9994 or higher, indicating outstanding discriminatory power. Tuned Random Forest achieved the highest ROC AUC on the test set at 0.9995.

Comparing the default and tuned versions reveals the impact of hyperparameter optimization. While the default ensemble models already performed very well, tuning resulted in slight improvements in metrics like Recall, Precision, and F1-Score for some models, and importantly, led to the highest observed ROC AUC. For instance, Tuned LightGBM achieved a remarkable Recall of 0.9996, indicating its ability to correctly identify almost all positive cases. Tuned Random Forest not only achieved the highest ROC AUC but also the highest Specificity at 0.9973.

The use of Stratified K-Fold cross-validation during the hyperparameter tuning process provides statistical validation for the performance estimates of the tuned models. The high and consistent cross-validation scores (Table7) demonstrate that these models' performance is not overly sensitive to the specific data split used for training and validation, increasing confidence in their robustness.

A crucial consideration for practical application is the trade-off between performance and computational efficiency. While tuning generally improved performance and led to the best overall models by ROC AUC, it significantly increased the training time compared to using default parameters, as the reported training times for tuned models include the entire

RandomizedSearchCV process. Default LightGBM, XGBoost, and Extra Trees Classifier remain highly attractive options due to their combination of very high performance (ROC AUC of 0.9994 or 0.9993) and significantly faster training times (under 0.5 seconds) compared to their tuned versions (hundreds of seconds) or other models like Tuned Random Forest (over 1700 seconds). For scenarios where rapid model training or frequent retraining is required, the default configurations of these boosting algorithms present a favorable balance.
The ROC curve analysis for the models included in the final comparison (Fig. 4) visually reinforces these findings, with the curves for the top default and tuned models closely clustered near the top-left corner, demonstrating their superior ability to distinguish between liver patients and non-liver patients.

Table7.Hyperparameter Tuning Results (Optimized for ROC AUC).

Model	Best Cross-Validation ROC AUC	Best Parameters
XGBoost (Tuned)	0.99989	{'subsample': 0.7, 'reg_lambda': 0.01, 'reg_alpha': 0.01, 'n_estimators': 1000, 'min_child_weight': 1, 'max_depth': 10, 'learning_rate': 0.05, 'gamma': 0.1, 'colsample_bytree': 0.7}
LightGBM (Tuned)	0.99991	{'subsample': 0.8, 'reg_lambda': 0.001, 'reg_alpha': 0.1, 'num_leaves': 31, 'n_estimators': 200, 'min_child_samples': 20, 'max_depth': 10, 'learning_rate': 0.2, 'colsample_bytree': 0.9}
CatBoost Classifier (Tuned)	0.99984	{'subsample': 0.9, 'learning_rate': 0.2, 'l2_leaf_reg': 5, 'iterations': 200, 'depth': 10, 'border_count': 32}
Random Forest (Tuned)	0.99985	{'n_estimators': 300, 'min_samples_split': 5, 'min_samples_leaf': 2, 'max_features': 'sqrt', 'max_depth': 20, 'bootstrap': False}
Extra Trees Classifier (Tuned)	0.99986	{'n_estimators': 100, 'min_samples_split': 5, 'min_samples_leaf': 1, 'max_features': 'sqrt', 'max_depth': None, 'bootstrap': False}
K-Nearest Neighbors (Tuned)	0.99393	{'weights': 'distance', 'n_neighbors': 17, 'metric': 'manhattan'}

Table8.Final Model Performance Comparison on Test Set (Top 5 Default and Tuned Models).

Model	Accuracy	Precision	Recall (Sensitivity)	F1-Score	Specificity	ROC AUC	Training Time (s)*
Random Forest (Tuned)	0.9961	0.9989	0.9957	0.9973	0.9973	0.9995	1748.71
LightGBM (Tuned)	0.9979	0.9975	0.9996	0.9986	0.9937	0.9994	271.96
XGBoost (Tuned)	0.9977	0.9986	0.9982	0.9984	0.9964	0.9994	223.47
Random Forest (Default)	0.9961	0.9975	0.9971	0.9973	0.9937	0.9994	0.99
Extra Trees	0.9948	0.9964	0.9964	0.9964	0.9910	0.9994	0.41

Classifier (Default)							
LightGBM (Default)	0.9951	0.9967	0.9964	0.9966	0.9919	0.9994	0.19
XGBoost (Default)	0.9956	0.9960	0.9978	0.9969	0.9901	0.9993	0.31
CatBoost Classifier (Tuned)	0.9941	0.9949	0.9967	0.9958	0.9874	0.9993	509.98
Extra Trees Classifier (Tuned)	0.9951	0.9975	0.9957	0.9966	0.9937	0.9993	617.29
CatBoost Classifier (Default)	0.9889	0.9935	0.9909	0.9922	0.9838	0.9985	0.97
KNN (Tuned)	0.9378	0.9857	0.9261	0.9550	0.9667	0.9893	44.88

* Training Time for Tuned models includes the time taken for the RandomizedSearchCV process.

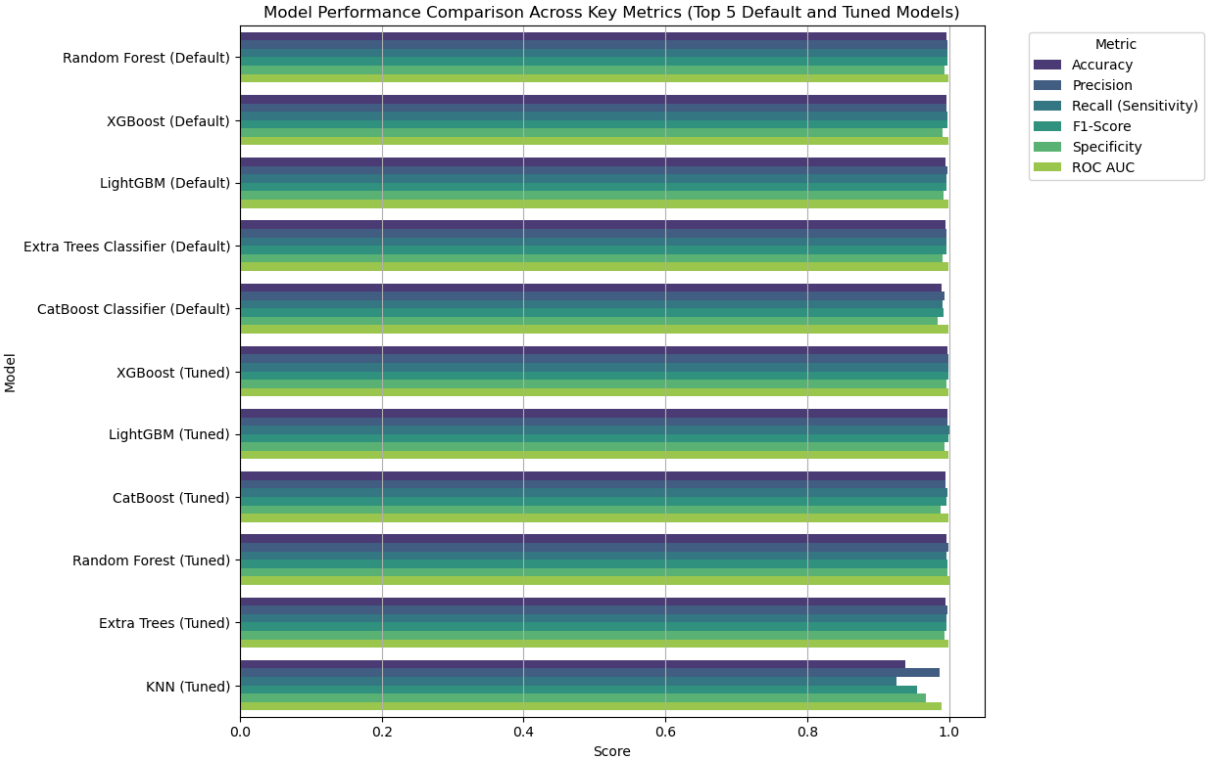

Fig. 3. Visualizing performance comparison (top 5 default and tuned models).

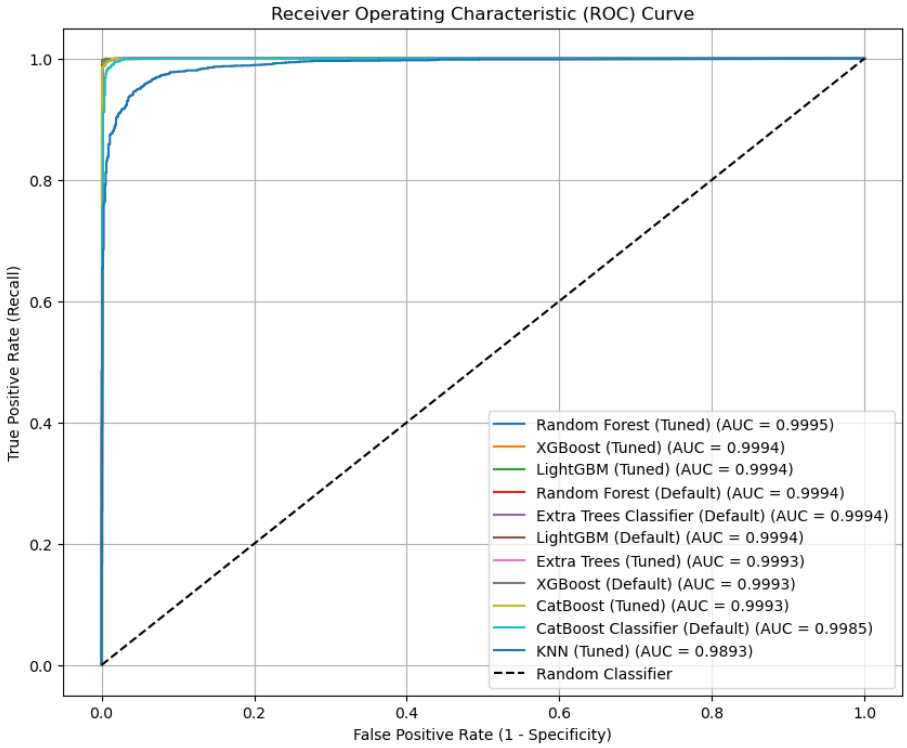

Fig. 4. ROC curve comparison (top 5 default and tuned models).

In addition to the aggregated performance metrics, analyzing the confusion matrices provides detailed insight into how each model performs in correctly classifying positive (Liver Patient) and negative (Non Liver Patient) instances. Table 4 presented these components for the default models.For the tuned models, the confusion matrix components on the independent test set are presented in Table 9.

Table9. Confusion Matrix Components for Tuned Models on Test Set.

Model	TN	FP	FN	TP
Random Forest (Tuned)	1109	3	12	2750
XGBoost (Tuned)	1108	4	5	2575
LightGBM (Tuned)	1105	7	1	2761
Extra Trees Classifier (Tuned)	1105	7	12	2750
CatBoost Classifier (Tuned)	1098	14	9	2753
K-Nearest Neighbors (Tuned)	1075	37	204	2558

Analysis of Table 9 shows that the top-performing tuned models, particularly LightGBM, XGBoost, Random Forest, and Extra Trees, exhibit very low numbers of False Positives (FP) and False Negatives (FN), aligning with their high Precision, Recall, and Specificity scores presented in Table 8. For instance, Tuned LightGBM has only 1 False Negative, highlighting its exceptional ability to identify positive cases. Tuned Random Forest shows only 3 False Positives, indicating a very high accuracy in classifying non-patients. Tuned KNN, while showing improvement over its default counterpart (Table 4), still has a considerably higher number of False Negatives compared to the tree-based ensemble models, which is reflected in its lower Recall. These detailed components further support the findings from the aggregated metrics and are crucial for understanding the specific types of errors each model makes, which is vital in a medical diagnostic context.

In summary, the experimental results strongly support the effectiveness of ensemble and advanced tree-based models, particularly Random Forest, LightGBM, XGBoost, Extra Trees, and CatBoost, for liver disease prediction. Hyperparameter tuning can yield marginal performance improvements, achieving the highest ROC AUC, but at the cost of significantly increased training time. The rigorous methodology, including preprocessing pipelines, SMOTE, and cross-validation during tuning, enhances the reliability and validity of these findings.

CONCLUSION AND FUTURE WORK

This study conducted a systematic and comprehensive evaluation of a diverse suite of machine learning classification algorithms for the prediction of liver disease using the Liver Patient Dataset (LDPD). The primary objective was to benchmark the performance of these models, identify those demonstrating superior predictive capabilities and computational efficiency, and explore the impact of hyperparameter tuning.

The experimental results demonstrate that machine learning classification is highly effective for this task, achieving exceptionally high performance metrics across several models, particularly within the ensemble and advanced tree-based categories. The initial evaluation with default hyperparameters established a strong baseline, with models like Random Forest, Extra Trees Classifier, LightGBM, XGBoost, and CatBoost Classifier achieving ROC AUC scores above 0.99.

Subsequently,hyperparameter tuning using RandomizedSearchCV with Stratified K-Fold cross-validation was applied to a subset of promising models. This process, while computationally more intensive, led to marginal but significant improvements in performance, achieving the highest observed metrics. The final comparison, incorporating both top default and tuned models, revealed that Tuned Random Forest achieved the highest ROC AUC (0.9995) and Specificity (0.9973) on the independent test set. Tuned LightGBM demonstrated the highest Recall (0.9996), alongside a very high ROC AUC (0.9994). Tuned XGBoost also exhibited outstanding performance across key metrics, with a ROC AUC of 0.9994. These results solidify the finding that ensemble methods, when appropriately tuned, can achieve near-perfect discrimination and high accuracy in identifying both positive and negative cases in this dataset.

A crucial insight from this study is the significant trade-off between model performance and computational efficiency (training time). While hyperparameter tuning yielded the highest performance, it drastically increased the training duration. Conversely, the default configurations of models like LightGBM, XGBoost, and Extra Trees Classifier provided exceptionally high performance (ROC AUC $\geqslant$ 0.9993) with significantly faster training times (under 0.5 seconds). This highlights that for practical applications where rapid model deployment or frequent retraining is necessary, prioritizing slightly lower, but still excellent, performance with significantly faster training from default configurations of efficient algorithms like LightGBM could be more suitable.

The rigorous methodology employed, including the use of preprocessing pipelines to prevent data leakage, SMOTE to address class imbalance, and Stratified K-Fold cross-validation during tuning for robust performance estimation, enhances the reliability and validity of these findings. Analysis of the confusion matrices provided detailed insights into the types of errors made, confirming the low rates of both false positives and false negatives among the top models.

Based on this comprehensive evaluation, the tuned versions of Random Forest, LightGBM, and XGBoost are identified as the top-performing models. Considering the performance-efficiency trade-off, the default configurations of LightGBM, XGBoost, and Extra Trees

Classifier are also highly promising candidates for practical implementation due to their strong performance combined with rapid training.

For future work, external validation on independent datasets is a crucial next step before these models can be considered for clinical application; additionally, exploring deeper model interpretability using techniques like SHAP and LIME can provide valuable insights into feature influence, and investigating the practical challenges of clinical integration is essential for real-world deployment.

REFERENCES

Ayeldeen, H., Shaker, O., Ayeldeen, G., & Anwar, K. M. (2015). Prediction of liver fibrosis stages by machine learning model: A decision tree approach. *Proceedings of the 2015 Third World Conference on Complex Systems (WCCS)*, 23–25.

Durai, V., Ramesh, S., & Kalthireddy, D. (2019). Liver disease prediction using machine learning. *International Journal of Advanced Research in Ideas and Innovations in Technology*, *5*(3), 1584–1588.

Geetha, C., & Arunachalam, A. R. (2021). Evaluation based approaches for liver disease prediction using machine learning algorithms. *Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI)*, 27–29.

Gogi, V. J. (2018). Prognosis of liver disease: Using machine learning algorithms. *Proceedings of the Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE)*, 27–28.

Hashem, S., Soliman, H., Elbahnasawy, H., Saleh, M., Elshamy, M., & El Kassas, M. (2018). Comparison of machine learning approaches for prediction of advanced liver fibrosis in chronic hepatitis C patients. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *15*(3), 861–868. https://doi.org/10.1109/TCBB.2017.2712365

Jacob, J., Mathew, J. C., Mathew, J., & Issac, E. (2018). Diagnosis of liver disease using machine learning techniques. *International Research Journal of Engineering and Technology*, *5*(5), 412–423.

Kumar, Y., & Sahoo, G. (2013). Prediction of different types of liver diseases using rule-based classification model. *Technology and Health Care*, *21*(5), 417–432.

Ma, H., Xu, C.-F., Shen, Z., Yu, C.-H., & Li, Y.-M. (2018). Application of machine learning techniques for clinical predictive modeling: A cross-sectional study on nonalcoholic fatty liver disease in China. *BioMed Research International*, *2018*, Article ID 4304376. https://doi.org/10.1155/2018/4304376

Maldonado, S., López, J., & Vairetti, C. (2019). An alternative SMOTE oversampling strategy for high-dimensional datasets. *Applied Soft Computing*, *76*, 380–389. https://doi.org/10.1016/j.asoc.2018.12.021

Md, Q., Kulkarni, S., Joshua, C. J., Vaichole, T., Mohan, S., & Iwendi, C. (2023). Enhanced preprocessing approach using ensemble machine learning algorithms for detecting liver disease. *Biomedicines*, *11*(2), 581. https://doi.org/10.3390/biomedicines11020581

Ramana, V., Babu, M. S. P., & Venkateswarlu, N. B. (2011). A critical study of selected classification algorithms for liver disease diagnosis. *International Journal of Database Management Systems*, *3*(4), 101–114.

Ramana, V., Babu, M. P., & Venkateswarlu, N. B. (2012). Liver classification using modified rotation forest. *International Journal of Engineering Research and Development*, *6*(4), 17–24.

Sameer, M., & Gupta, B. (2020). Detection of epileptical seizures based on alpha band statistical features. *Wireless Personal Communications*, *115*(2), 909–925. https://doi.org/10.1007/s11277-020-07542-5

Shaheamlung, G., Kaur, H., & Kaur, M. (2020). A survey on machine learning techniques for the diagnosis of liver disease. *Proceedings of the 2020 International Conference on Intelligent Engineering and Management (ICIEM)*, 17–19.

Shrivastava, A. (n.d.). *Liver disease patient dataset* [Data set]. Kaggle. https://www.kaggle.com/datasets/abhi8923shriv/liver-disease-patient-dataset/data

Sindhuja, D. R. J. P., & Priyadarsini, R. J. (2016). A survey on classification techniques in data mining for analyzing liver disease disorder. *International Journal of Computer Science and Mobile Computing*, *5*(5), 483–488.

Sivakumar, D., Varchagall, M., & Gusha, S. A. (2019). Chronic liver disease prediction analysis based on the impact of life quality attributes. *International Journal of Recent Technology and Engineering (IJRTE)*, *7*(6), 2111–2117.

Sontakke, S., Lohokare, J., & Dani, R. (2017). Diagnosis of liver diseases using machine learning. *Proceedings of the 2017 International Conference on Emerging Trends & Innovation in ICT (ICEI)*, 3–5.

World Life Expectancy. (2022, April 14). *Liver disease in India*. https://www.worldlifeexpectancy.com/india-liver-disease

Wu, C.-C., Yeh, M.-L., Liu, Y.-L., Hsu, W.-Y., Tsai, P.-C., Lin, Y.-H., ... & Huang, C.-F. (2019). Prediction of fatty liver disease using machine learning algorithms. *Computer Methods and Programs in Biomedicine*, *170*, 23–29. https://doi.org/10.1016/j.cmpb.2018.10.028