LIVER DISEASE PREDICTION USING ENSEMBLE STACKING APPROACH INTEGRATING ADVANCED MACHINE LEARNING MODELS

Abstract*—In recent years, the early detection of liver disease has become crucial due to its escalating prevalence and potential for fatal outcomes. Traditional diagnostic methods often lack the convenience needed for early-stage identification. This study aims to address this gap by utilizing machine learning algorithms for liver disease prediction. We employ an Ensemble Stacking Classifier, integrating XGBoost, Random Forest, and CART as base models, with Logistic Regression serving as the meta-model. This methodology was chosen based on its superior test accuracy over traditional classifiers. We further enhance the model's performance by imputing missing values using the MICE technique, balancing the dataset through minority up-sampling using the mean strategy and tuning the hyper-parameters through GridSearchCV . Our results indicate that this ensemble approach achieved an outstanding accuracy of 91.01% and an AUC-ROC score of 97.98%. These findings demonstrate the efficacy of machine learning methods, specifically Ensemble Stacking, in the early detection of liver disease, thereby providing a robust tool for clinical application.

***Index Terms*—Liver Disease, MICE, Ensemble Stacking, GridSearchCV, Machine Learning**

I. Introduction

The rise of liver diseases, such as cirrhosis, hepatitis, and liver cancer, continues to be a global health concern, resulting in millions of new cases and hundreds of thousands of fatalities each year. While advancements in medical treatments have led to a decrease in overall mortality rates, liver-related conditions still account for a significant percentage of global deaths. Traditional diagnostic methods involve an array of tests, including urinalysis, complete blood count, and comprehensive metabolic panel (CMP). However, these diagnostic approaches can be time-consuming, invasive, and expensive, particularly for diseases that necessitate multiple types of tests or repeated sampling [1].

Liver diseases often develop inconspicuously, with symptoms easily overlooked until advanced stages. Therefore, early diagnosis is not just beneficial but essential for effective management and improving the quality of life for patients. Traditional diagnostic methods often involve invasive procedures like biopsies or expensive imaging techniques that may not be readily accessible to everyone. Furthermore, these approaches may have their limitations, including the risk of complications, time consumption, and in some cases, delayed or incorrect diagnosis. Given the gravity and prevalence of liver diseases, there's a dire need for quicker, more accessible, and accurate diagnostic tools [2].

Machine learning and deep learning technologies offer a compelling alternative, one that is increasingly proving to be more effective for diagnostic healthcare. These algorithms possess the capability to analyze extensive datasets, thereby detecting patterns and connections that may be less apparent or completely disregarded during manual study. The automation and predictive capability inherent in these technologies not only speed up the diagnosis but also hold promise for higher accuracy and, consequently, better treatment planning [3].

However, the use of machine learning in the medical domain is not devoid of obstacles. For instance, the high dimensionality of medical data sets often introduces noise, leading to overfitting and reduced model performance. Additionally, the handling of missing data, a pervasive issue in medical research, affects the quality of diagnosis and introduces bias. More importantly, existing models often struggle with imbalanced classes, which can further compromise the diagnostic accuracy [4].

Our study, therefore, aims to explore and integrate machine learning algorithms, enhanced by projection-based statistical methods, for the diagnosis and prediction of liver diseases. In doing so, we intend to address the existing challenges like high-dimensionality, missing data, and class imbalances that conventional models often struggle with. The intent is to propose an integrated approach to overcome existing limitations, aiming for a diagnostic paradigm that is quicker, more accurate, and widely accessible.

II. Literature Review

Thirunavukkarasu K et al. (2018) [5], used several classification algorithms, including Logistic Regression, Support Vector Machine, and K-Nearest Neighbor, in their research endeavor aimed at predicting liver disease. The performance of these algorithms was evaluated by comparing their classification accuracy, which was assessed via the use of a confusion matrix. Based on the empirical findings, it was noted that Logistic Regression and K-Nearest Neighbour exhibited the highest level of accuracy, reaching a rate of 73.97%. On the other hand, Logistic Regression and Support Vector Machine exhibited the maximum sensitivity value of 0.952. Consequently, it may be inferred that Logistic Regression is a viable approach for predicting liver disease.

In their study, Ketan Gupta et al. (2022) [6], employed several machine learning algorithms, including Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors, Gradient Boosting, Extreme Gradient Boosting, and LightGB, to predict the occurrence of liver disease. The Random Forest and LightGB models had the highest accuracy, with a rate of 63%. Additionally, the Random Forest model demonstrated superior precision. Based on the conducted study and subsequent calculations, it has been determined that these algorithms have achieved marginally improved accuracy following the process of feature selection. The study's findings indicate that suboptimal data pre-processing techniques resulted in average accuracy ratings.

In the approach proposed by Ruhul Amin et al. (2023) [1], multiple classification algorithms including Random Forest, Multilayer Perceptron Network, KNNeighbor, Logistic Regression, Support Vector Machine, and Ensemble Stacking, were incorporated. The researchers put forth a statistical projection-based approach for integrating features extracted from various dimensionality reduction techniques, including PCA, FA, and LDA. Between the classifiers evaluated, it was observed that the Random Forest classifier exhibited the highest accuracy (88.10%) and AUC (88.20%) scores.

Elias Dritsas and Maria Trigka (2023) [7], present a comparative study on the effectiveness of various machine learning and ensemble methods for liver disease prediction. Utilizing metrics such as accuracy, precision, recall, F-measure, and AUC, the authors found that the Voting classifier exhibited superior performance. Specifically, it achieved an accuracy of 80.1%, a precision of 80.4%, and an AUC of 88.4% using 10-fold cross-validation post-SMOTE balancing. The paper also leverages the Random Forest model to evaluate feature importance through Gini impurity, enriching its methodological rigor.

In the paper authored by Srilatha Tokala et al. (2023) [8], the focus is on leveraging machine learning algorithms to develop classification models for predicting chronic liver disorders. Utilizing patient datasets, the study aims to alleviate the clinical burden by automating the diagnostic process. Various classification techniques were evaluated, and the Random Forest Classifier emerged as the most accurate with an 87% success rate. The outcomes are quantified using confusion matrices, illustrating the model's predictive reliability. This work not only advances automated healthcare diagnostics but also offers a viable solution for effective liver disorder classification when a training dataset is available.

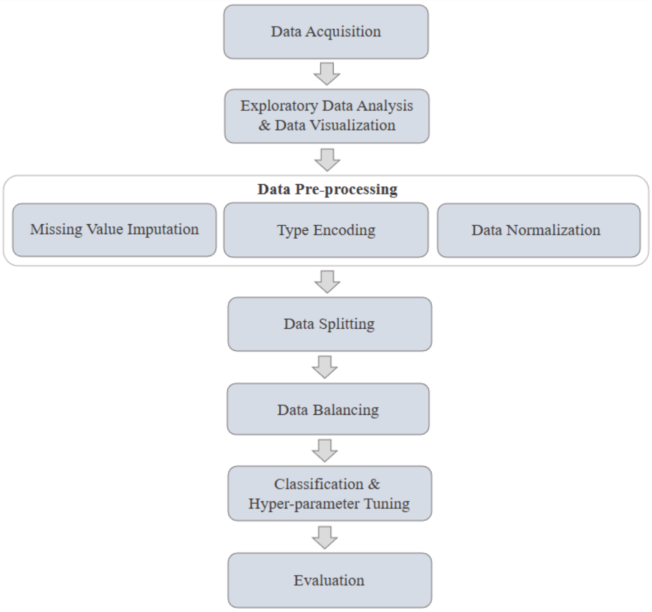


Fig. 1. Overview of the workflow

III. Methodology

This study utilizes a range of classification algorithms to analyze intricate medical data to provide accurate predictions regarding liver diseases. The objective was to optimize the accuracy of a specific classification algorithm by identifying the most effective combination of techniques for handling missing values, balancing data, normalizing data, reducing dimensionality and hyper-parameter tuning. Subsequently, through the process of evaluating diverse classifiers using many scoring criteria, the study identifies the most effective classifier for our suggested model. Fig. 1 exhibits a flowchart demonstrating the process of conducting the entire study.

A. Data Loading

The dataset utilized in this study [9] comprises an extensive compilation of medical records centered on liver diseases, spanning a total of 583 patient entries. The dataset utilized in this study originates from the northeastern region of Andhra Pradesh, India and was collected from UCI machine learning repository entitled ”Indian Liver Patient Dataset”. It consists of 416 records from patients who have been diagnosed with liver disease, and an additional 167 records from individuals who do not have liver disease. The dataset comprises 441 male patients and 142 female patients, indicating a gender distribution. The variables considered in this study cover the demographic characteristics of the patient, such as age and gender. Additionally, the levels of Total Bilirubin and Direct Bilirubin, Alkaline Phosphatase, Alamine Aminotransferase, Aspartate Aminotransferase, Total Proteins, Albumin, and the Albumin-Globulin Ratio are considered. The column labeled ”Dataset” is utilized to classify individuals into two distinct groups: those who have been diagnosed with liver disease and

TABLE I
ATTRIBUTES OF THE DATASET

Attribute Name	Data Type
Age of the patient	int64
Gender of the patient	object
Total Bilirubin	float64
Direct Bilirubin	float64
Alkaline Phosphatase	int64
Alamine Aminotransferase	int64
Aspartate Aminotransferase	int64
Total Proteins	float64
Albumin	float64
Albumin and Globulin Ratio	float64
Dataset	float64

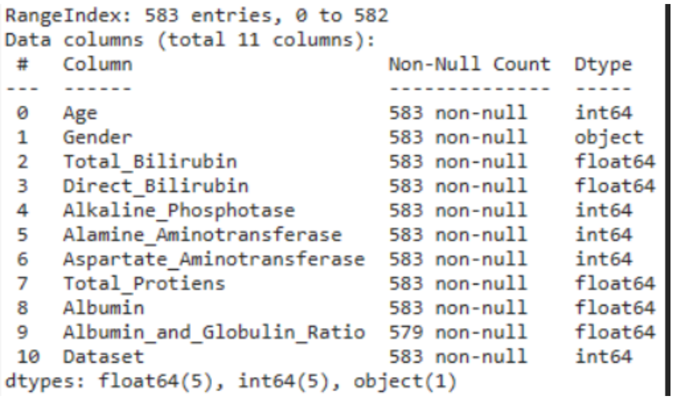
those who have not. Information on the attributes of the dataset is presented in table I.

B. Exploratory Data Analysis and Data Visualization

Exploratory data analysis (EDA) aims to provide an overview of the data set and guide future research and idea generation. Exploratory Data Analysis (EDA) plays a pivotal role in the identification and prognostication of liver diseases through the utilization of machine learning and deep learning algorithms. Initially, it offers initial understanding of the organization and features of the data, hence informing the selection of further research algorithms. Additionally, it aids in the identification of patterns and correlations among variables for the purpose of diagnostic modeling. The comprehension of the correlation between liver function metrics and the various phases of liver illness can have implications for the process of feature engineering and the selection of appropriate models. Furthermore, exploratory data analysis (EDA) plays a crucial role in identifying and resolving prevalent difficulties within medical datasets, such as the presence of missing or inconsistent data. These issues have the potential to undermine the trustworthiness of studies conducted using such datasets.

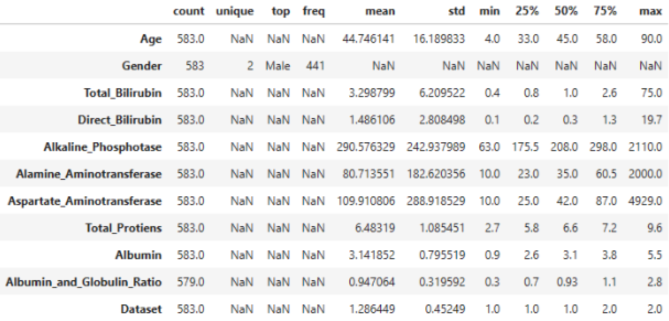
In the context of liver disease prediction, an intricate web of variables such as age, gender, and biochemical markers interact in ways that can be difficult to untangle through numerical data alone. The subtleties in this high-dimensional data space can be easily overlooked without an effective visualization strategy. Data visualization allows us to break down complex patterns into digestible insights. By presenting data in a graphical format, clinicians, researchers, and even patients can make sense of the complex interrelationships between different indicators of liver health.

The INDIAN LIVER PATIENT DATASET is structured as a Pandas Data Frame comprising 583 entries spread across 11 distinct attributes. Most of these attributes are numerical in nature, encoded either as integers (int64) or floating-point numbers (float64). Only the 'Gender' attribute is labeled as an object, signifying its categorical nature.



```
RangeIndex: 583 entries, 0 to 582
Data columns (total 11 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   Age                         583 non-null    int64  
 1   Gender                      583 non-null    object 
 2   Total_Bilirubin             583 non-null    float64
 3   Direct_Bilirubin            583 non-null    float64
 4   Alkaline_Phosphotase        583 non-null    int64  
 5   Alamine_Aminotransferase    583 non-null    int64  
 6   Aspartate_Aminotransferase  583 non-null    int64  
 7   Total_Protiens              583 non-null    float64
 8   Albumin                     583 non-null    float64
 9   Albumin_and_Globulin_Ratio  579 non-null    float64
 10  Dataset                     583 non-null    int64  
dtypes: float64(5), int64(5), object(1)
```

Fig. 2. Information on the 583x11 shaped dataset



	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Age	583.0	NaN	NaN	NaN	44.746141	16.189833	4.0	33.0	45.0	58.0	90.0
Gender	583	2	Male	441	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Total_Bilirubin	583.0	NaN	NaN	NaN	3.298799	6.209522	0.4	0.8	1.0	2.6	75.0
Direct_Bilirubin	583.0	NaN	NaN	NaN	1.486106	2.808498	0.1	0.2	0.3	1.3	19.7
Alkaline_Phosphotase	583.0	NaN	NaN	NaN	290.576329	242.937989	63.0	175.5	208.0	298.0	2110.0
Alamine_Aminotransferase	583.0	NaN	NaN	NaN	80.713551	182.620356	10.0	23.0	35.0	60.5	2000.0
Aspartate_Aminotransferase	583.0	NaN	NaN	NaN	109.910806	288.918529	10.0	25.0	42.0	87.0	4929.0
Total_Protiens	583.0	NaN	NaN	NaN	6.48319	1.085451	2.7	5.8	6.6	7.2	9.6
Albumin	583.0	NaN	NaN	NaN	3.141852	0.795519	0.9	2.6	3.1	3.8	5.5
Albumin_and_Globulin_Ratio	579.0	NaN	NaN	NaN	0.947064	0.319592	0.3	0.7	0.93	1.1	2.8
Dataset	583.0	NaN	NaN	NaN	1.286449	0.45249	1.0	1.0	1.0	2.0	2.0

Fig. 3. Data description

The 'Gender' attribute embodies two unique descriptors—'Male' and 'Female'. For machine learning algorithms to process this attribute, it needs to be numerically transformed via type encoding. Several columns reveal a considerable gap between the 75th percentile and the maximum values. This suggests the existence of outliers or extreme values that require attention. The outcome variable 'Dataset' is binary but is represented by the numbers 1 and 2. It should be recorded to fall within a [0, 1] range through binary encoding for compatibility with classification algorithms. The 'Albumin and Globulin Ratio' feature has a few missing entries that require imputation.

The dataset leans towards male representation with 441 male entries compared to 142 female entries, thus highlighting a gender imbalance. The outcome attribute 'Dataset' reveals an imbalance between the classes, which necessitates careful consideration during model training to mitigate bias. These initial exploration suggests that prior to model training, the dataset will benefit from data balancing procedures to establish balance between classes.

Observing the correlation table, one of the most prominent findings generated is the strong correlations observed among specific variables. For instance, 'Direct Bilirubin' and 'Total Bilirubin' displayed a high correlation, suggesting a possibility of redundant information. Similarly, 'Aspartate Aminotransferase' and 'Alamine Aminotransferase', as well as 'Total Proteins' and 'Albumin', were closely related. Another noteworthy correlation was between 'Albumin and Globulin Ratio' and 'Albumin'. These strong correlations indicate the necessity of feature engineering and dimensionality reduction techniques to remove multicollinearity, ensuring a more accurate and effective predictive model for liver disease. Addressing these correlations will refine the model's predictive accuracy while simplifying its complexity, a crucial step in developing a robust tool for liver disease diagnosis.

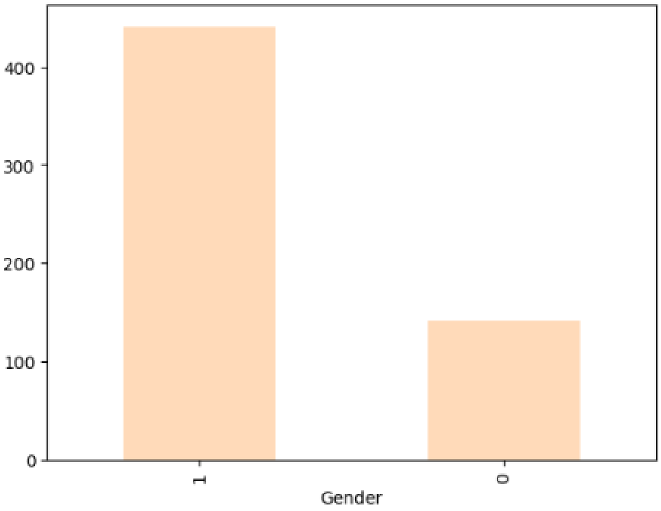


Fig. 4. Gender proportionality bar-plot

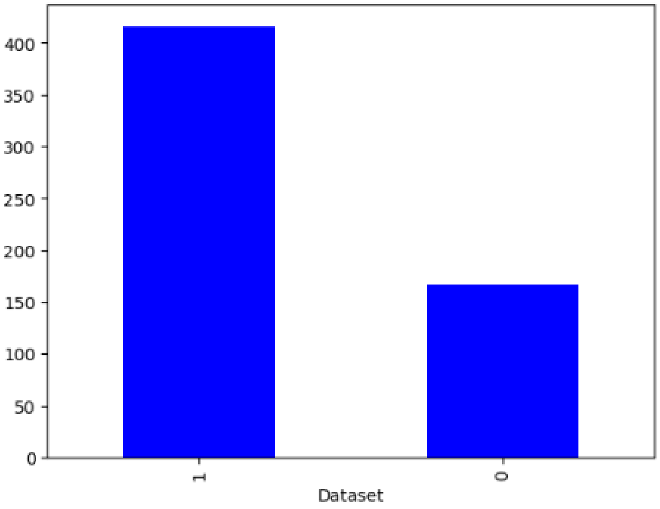
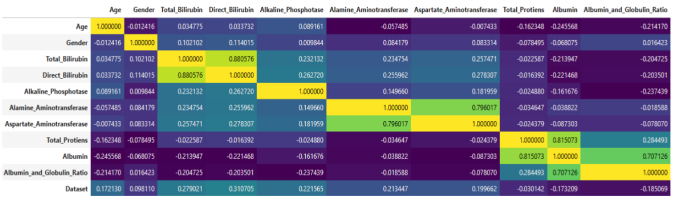


Fig. 5. 'Dataset' attribute's class proportionality bar-plot



	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio
Age	1.000000	-0.012416	0.034775	0.033732	0.089161	-0.057485	-0.007433	-0.162348	-0.245568	-0.214170
Gender	-0.012416	1.000000	0.102102	0.114015	0.009844	0.084179	0.083314	-0.078495	-0.068075	0.016423
Total_Bilirubin	0.034775	0.102102	1.000000	0.880576	0.232132	0.234754	0.257471	-0.022587	-0.213947	-0.204725
Direct_Bilirubin	0.033732	0.114015	0.880576	1.000000	0.262720	0.255962	0.278307	-0.016392	-0.221468	-0.203501
Alkaline_Phosphotase	0.089161	0.009844	0.232132	0.262720	1.000000	0.149660	0.181959	-0.024880	-0.161676	-0.237439
Alamine_Aminotransferase	-0.057485	0.084179	0.234754	0.255962	0.149660	1.000000	0.796017	-0.034647	-0.038822	-0.018588
Aspartate_Aminotransferase	-0.007433	0.083314	0.257471	0.278307	0.181959	0.796017	1.000000	-0.024379	-0.087303	-0.078070
Total_Protiens	-0.162348	-0.078495	-0.022587	-0.016392	-0.024880	-0.034647	-0.024379	1.000000	0.815073	0.284493
Albumin	-0.245568	-0.068075	-0.213947	-0.221468	-0.161676	-0.038822	-0.087303	0.815073	1.000000	0.707126
Albumin_and_Globulin_Ratio	-0.214170	0.016423	-0.204725	-0.203501	-0.237439	-0.018588	-0.078070	0.284493	0.707126	1.000000
Dataset	0.172130	0.098110	0.279021	0.310705	0.221565	0.213447	0.199662	-0.030142	-0.173209	-0.185069

Fig. 6. Full correlation table

C. *Data Pre-processing*

Predictive modeling relies heavily on the pre-processing of data, particularly when dealing with intricate datasets like the Indian Liver Patient Dataset. In our study, we mainly focus on four data pre-processing techniques: handling missing values, type encoding, data normalization and data balancing methods.

Missing Value Imputation—The initial stage of our pre-processing pipeline is addressing the issue of missing values. The most straightforward method for addressing missing values is by univariate missing value imputation. In this approach, each missing value within a specific feature is substituted with the mean of the observed values for that feature [10]. For tackling the missing values in our dataset, a more nuanced approach called Multivariate Imputation by Chained Equations (MICE) is applied. This method is especially useful when dealing with multiple variables that have missing entries and aims to create multiple imputations rather than single-point estimates.

The mathematical rigor behind MICE generally involves solving a series of chained equations. Suppose we have k variables X_1 ,X_2 ,$\ldots$,X_k with some missing data. We proceed by specifying a model for each X_i conditioned on X_{-i} , where X_{-i} represents all variables other than X_i. For each X_i ,we fit the model:

$$X_i \mid \mathbf{X}_{-i} = f\left(\mathbf{X}_{-i}; \theta_i\right)$$

Here, f can be any function that maps X_{-i} to X_i , and ${}_i$ are the parameters for this function, which are typically estimated using maximum likelihood. The "chained equations" in MICE come from iterating these models for each X_i with missing data, filling in estimated values at each step and then refining these estimates through subsequent iterations [11].

Type Encoding—In the scope of data analysis and machine learning, the transformation of categorical attributes into a numerical format is often necessary for the successful implementation of various algorithms. Within our dataset, the column labeled 'Gender' poses such a challenge, being designated as an object data type. To convert this into a machine-friendly format, we opt for the straightforward method known as binary encoding [12].

Binary encoding serves as an efficient means to convert categories into numerical identifiers. Given that our 'Gender' column contains only two distinct categories—'Male' and 'Other'—this method is ideally suited for our purposes. Here, we assign 'Male' the numerical value '1', while every other gender category is allocated the value '0'. In mathematical notation, this transformation is illustrated as [13]:

$$\text{Encoded Gender } = \begin{cases} 1 & \text{if Gender } = \text{ 'Male'} \\ 0 & \text{if Gender } = \text{ 'Other'} \end{cases}$$

Similarly, for our 'Dataset' column, which acts as an indicator of whether an individual is afflicted with liver disease, a binary encoding is also applied. In this setting, the numeral '1' corresponds to the existence of liver disease, while '0' signals its absence. This conversion is mathematically articulated as [14]:

$$\text{Encoded Dataset } = \begin{cases} 1 & \text{indicates Liver Disease} \\ 0 & \text{indicates No Liver Disease} \end{cases}$$

Data Normalization—The process of data normalization is an essential preprocessing step, particularly when confronted with characteristics that exhibit variations in terms of ranges or units. The performance of machine learning algorithms might be negatively affected by the presence of variables with varying scales, as these methods are highly sensitive to the size of the feature vectors. In the present study environment, wherein the objective is to categorize liver illnesses using diverse health parameters, the utilization of feature scaling approaches is especially relevant. Out of different normalization techniques like Min-Max Normalization, Maximum Absolute Scaling, Standardization and Robust Scaling, in the scope of this study, we used the 'StandardScaler' class from the scikit-learn library to implement standardization which rescales the feature values to have a mean of 0 and a standard deviation of 1. The conversion is mathematically articulated as [13], [14], [15]:

$$X_{\text{new }}=\frac{X-\bar{X}}{\sigma}$$

D. Data Splitting

In the realm of predictive analytics, especially in the context of medical diagnoses like liver disease, the validation of a model's ability to generalize is paramount. To achieve this, a well-structured partitioning of the original dataset into training and testing sets is essential. This procedure is commonly known as the "Train-Test Split."

The dataset D is bifurcated into two mutually exclusive subsets: the training set D_{train} and the testing set D_{test} . The training set is utilized for model training, while the testing set is reserved for model evaluation. The partitioning can be formally represented by the following equations [15]:

$$\mathcal{D}=\mathcal{D}_{\text{train }} \cup \mathcal{D}_{\text{test }}$$
$$\mathcal{D}_{\text{train }} \cap \mathcal{D}_{\text{test }}=\emptyset$$

In the present study, Python's scikit-learn library provides a train_test_split function to facilitate this partitioning. In this function, X represents the feature matrix, while y contains the class labels (liver disease presence). The function partitions 80% of the data into the training set and the remaining 20% into the testing set, as controlled by the parameter test_size=0.2. Additionally, the random_state=42 ensures reproducibility in the splitting process [16].

E. Data Balancing

Addressing unbalanced classes in machine learning and data science is challenging, especially when trying to develop a prediction model with high accuracy and predictability. Minority Up-sampling and Majority Down-sampling are typical solutions.

Minority Up-sampling— is crucial in cases where there is insufficient data. To achieve this, we will artificially increase the size of the minority group by duplicating their records. Given the size limitations of our dataset, where we are working with a lower volume of records for individuals without liver disease, up-sampling is mandatory. To precisely align the minority class (non-liver disease) with the majority class (liver disease), we increased the minority class records to 416. In this operation, we utilized the 'resample' function from Python's scikit-learn library. After the up-sampling, the overall dataset size increased to 832 records. In our scenario, using up-sampling to reduce class imbalance and increase the size of the training sample for machine learning models is quite beneficial [15], [17], [18].

Synthetic Minority Over-sampling Technique (SMOTE)— is a method that aims to mitigate the issue of class imbalance by creating artificial instances inside the feature space. The SMOTE algorithm acts inside the feature space spectrum, therefore enhancing the representation of the minority class by generating more intricate data points. These data points are formed by combining instances that are closely situated to one another. This approach contrasts with basic over-sampling techniques, since the replication of existing minority occurrences in simple over-sampling might result in overfitting.

Let $X = x_1, x_2, \ldots, x_N$ be the minority class samples, where x_i represents each feature vector. For each x_i , SMOTE selects k nearest neighbors, and the set of nearest neighbors is denoted by NN(x_i). To create a synthetic sample x_{new} , SMOTE performs the following computation:

$$x_{\text{new }}=x_i+\lambda \times\left(x_{\text{nn}}-x_i\right)$$

Here x_{nn} is a randomly chosen neighbor from NN(x_i) and is a random number between 0 and 1. The minority class and synthetic samples have been merged to create a balanced dataset. For our dataset, especially with the few records for patients without liver illness, SMOTE improves the model's learning power without bias [15], [17], [19].

Random Over-Sampling Examples (ROSE)— algorithm is another technique employed to tackle the issue of class imbalance in datasets. Unlike SMOTE, which generates synthetic instances, ROSE generates new samples by applying a smoothed bootstrap approach. The method is particularly effective when dealing with smaller datasets, making it an excellent choice for our study on liver diseases.

Let $C = c1, c2, \ldots, cM$ be the minority class samples, where c_i represents each feature vector. Let N be the desired number of synthetic samples. A synthetic sample cnew is generated using a smoothed bootstrap sample from C. The smoothed bootstrap sample c_{new} can be mathematically represented as:

$$c_{\text{new }}=c_i+h \cdot Z$$

where c_i is a randomly selected sample from C, is a smoothing parameter, and Z is a random vector from a d-dimensional standard normal distribution. The newly generated samples produced by ROSE are subsequently merged with the initial dataset to establish a more equitable distribution of classes. In the given setting of our liver illness dataset, which is of limited size, the utilization of ROSE is crucial for augmenting the training set of the model [15], [20], [21].

F. Classification

In machine learning, classification serves as a form of supervised learning where the objective is to assign labels to instances based on the relationships discovered in the training data. It is particularly vital in medical research for predictive analytics, such as the diagnosis of liver diseases, which is the focus of this thesis. A variety of classifiers have been employed in this study, each with its unique mathematical foundations, strengths, and weaknesses. This study leverages a broad array of machine learning algorithms to ensure comprehensive analysis and robust conclusions.

eXtreme Gradient Boosting (XGBoost)— is an advanced implementation of gradient boosted trees designed for speed and performance. The method is an ensemble learning technique that aims to optimize a differentiable loss function. Within the context of this thesis, XGBoost is chosen for its prowess in managing structured data and its ability to handle the high dimensionality and potential class imbalance prevalent in liver disease datasets. The objective function for XGBoost can be formalized as:

$$\mathcal{O}(\Phi)=\sum_{i=1}^{N} l\left(y_i, \hat{y}_i\right)+\sum_{k=1}^{K} \Omega\left(f_k\right)$$

where denotes the ensemble of K trees, l is a differentiable convex loss function that measures the difference between the actual output y_i and the predicted output $\hat{Y}_i$, and $\Omega\left(f_k\right)$ is the regularization term. The regularization term $\Omega\left(f_k\right)$ can be defined as:

$$\Omega\left(f_k\right)=\gamma T+\frac{1}{2} \lambda\|w\|^2$$

where T is the number of leaves in the tree, w is the vector of scores on the leaves, is a parameter that controls complexity, and is the L_2 regularization term on the leaf scores [22], [23].

Random Forest (RF—) approach belongs to the ensemble learning category, which leverages the principle of bagging to generate a collection of decision trees. The Random Forest method aims to enhance forecast accuracy and mitigate overfitting by combining predictions from numerous decision trees. It is an apt choice for the problem of liver disease prediction owing to its robustness against overfitting and its ability to handle high-dimensional data with potentially imbalanced classes.

Given a dataset D with n samples and m features, Random Forest aims to construct B decision trees $T_1, T_2, ..., T_B$, each trained on a bootstrapped subset of D. The final classification $C(x)$ for a new input x is obtained by majority voting:

$$C(x)=\operatorname{mode}\left\{T_1(x), T_2(x), \ldots, T_B(x)\right\}$$

In the case of regression, the final output is the average of the outputs from all decision trees [24], [25].

Classification and Regression Trees (CART)— constitute a non-parametric, hierarchical approach to decision-making, which is essential in predictive analytics and machine learning. Unlike ensemble methods, CART builds a single tree, but its ability to partition the feature space into homogenous regions makes it highly effective. Its suitability for the liver disease prediction task lies in its intuitive interpretation and the ability to handle both categorical and continuous features.

For a given dataset D consisting of n samples, each with m features, the goal is to partition the feature space into J distinct and non-overlapping regions $R_1, R_2, ..., R_J$. The tree predicts the output $C(x)$ based on the mean or mode of the training samples falling into region R_j that contains the new input x [26], [27]:

$$C(x)=\operatorname{mode}\left(y_i \mid x_i \in R_j\right) \quad \text{for classification}$$

Or for regression,

$$C(x)=\operatorname{mean}\left(y_i \mid x_i \in R_j\right)$$

Logistic Regression— is a probabilistic, parametric classifier that belongs to the class of linear models known as GLMs. Unlike linear regression, which is designed to predict continuous outcomes, Logistic Regression is tailored for binary classification problems. In this study, Logistic Regression is employed both as a standalone classifier and as a meta-classifier in ensemble stacking techniques.

The algorithm utilizes the logistic function, also known as the sigmoid function, to transform its output into a probability value. The logistic function $\sigma(z)$ is defined as follows: Logistic Regression is a probabilistic, parametric classifier that belongs to the class of linear models known as GLMs. Unlike linear regression, which is designed to predict continuous outcomes, Logistic Regression is tailored for binary classification problems. In this study, Logistic Regression is employed both as a standalone classifier and as a meta-classifier in ensemble stacking techniques.

The algorithm utilizes the logistic function, also known as the sigmoid function, to transform its output into a probability value. The logistic function $\sigma(z)$ is defined as follows:

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

Here z is the weighted sum of the input features and the bias term, mathematically represented by:

$$z=\theta_0+\theta_1 x_1+\theta_2 x_2+\cdots+\theta_n x_n$$

Given m instances in the dataset and n features, X is the feature matrix and is the parameter vector. The objective is to find the optimal that maximizes the log-likelihood function $\ell(\theta)$:

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{m} \left[y_i \log\left(\hat{p}_i\right) + \left(1 - y_i\right) \log\left(1 - \hat{p}_i\right) \right]$$

where $\hat{p}_i$ is the predicted probability of the i^{th} instance belonging to the positive class [28], [29].

Ensemble stacking— is an advanced machine learning paradigm that combines multiple classifiers' outputs to produce a final predictive model. This technique aims to leverage the individual strengths of an ensemble of base classifiers to improve prediction accuracy. In the context of liver disease prediction, ensemble stacking offers an integrated approach, thereby refining and enhancing predictive accuracy and robustness.

Let M be the set of N base classifiers $M_1, M_2, \ldots, M_N$. Given a feature vector x, each base classifier M_i produces a prediction y_i , where y_i =M_i (x). In stacking, these predicted outputs are used as input features for a second-level classifier F, which provides the final prediction y'.

$$y' = F\left(M_1(x), M_2(x), \ldots, M_N(x)\right)$$

For a binary classification problem, y' will be in the set 0,1, representing the two possible classes. In this study, ensemble stacking is implemented using the StackingClassifier from the scikit-learn library. The ensemble comprises base classifiers like XGBoost, Random Forest, CART, among others, while logistic regression is employed as the second-level classifier [15], [30], [31].

Except these classifiers Artificial Neural Networks (ANN), K-Nearest Neighbors (KNN), Support Vector Machines (SVM) and Gaussian Naive Bayes(GNB) were utilized for iterative experimentation in order to finding the best suited classifier for our model.

G. Proposed Model

The workflow includes various stages from data acquisition to classification and hyper-parameter tuning, all aimed at achieving the most accurate predictive model. Fig. 7 is a comprehensive outline of the adopted methodology.

After initial analysis and visualization of the dataset [9], the four missing values in the ”Albumin_and_Globulin_Ratio” feature are filled using Multivariate Imputation by Chained Equations (MICE). Moving on to the binary encoding, categorical features in the ’Gender’ column are encoded into numerical types. Additionally, a binary encoding is applied to the ’Dataset’ column, which indicates whether or not a person has liver disease.

After standardizing the features using StandardScaler, three different data balancing techniques (Mean, SMOTE and ROSE) are experimented for minority class up-sampling. On the final model, up-sampling using mean was chosen as it provided us with the highest accuracy.

After initial attempts at dimensionality reduction and feature selection proved unfruitful while being combined with different classifiers, an ensemble stacking model is selected for classification. This stacking model comprises XGBoost, Random Forest, and CART as base classifiers, while Logistic Regression serves as the meta-model. GridSearch is employed to fine-tune the hyperparameters of the ensemble stacking model. The best parameters found are as follows: CART max depth: 10, Logistic Regression C: 1.0, Random Forest max depth: 20, Random Forest n_estimators: 50, XGBoost learning rate: 0.01, XGBoost n_estimators: 100.

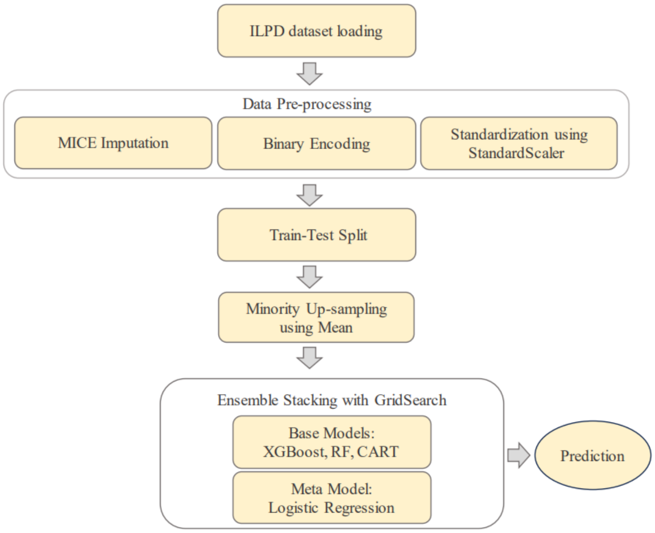

ILPD dataset loading
Data Pre-processing
MICE Imputation
Binary Encoding
Standardization using StandardScaler
Train-Test Split
Minority Up-sampling using Mean
Ensemble Stacking with GridSearch
Base Models: XGBoost, RF, CART
Meta Model: Logistic Regression
Prediction

Fig. 7. Workflow of the proposed methodology

IV. Result and Discussion

This section serves as the crux of this study, presenting an in-depth analysis of the outcomes derived from various experimental setups. This section will also elucidate on the computational environment used for the training and testing phases of the models. All computational tasks were executed on Kaggle, providing transparency on the hardware and software configurations that were instrumental in achieving the reported outcomes. Various performance metrics, including but not limited to, the accuracy of the classifier, AUC-ROC score, ROC curve, and the confusion matrix will be meticulously evaluated and discussed.

A. Training and Testing Environment

Kaggle is a widely acclaimed platform that serves as a rich ecosystem for data scientists, providing access to a plethora of datasets and specialized computational resources. Within the scope of this study, Kaggle was employed as a one-stop solution to facilitate every step of our research—from dataset acquisition to model evaluation.

Kaggle provides access to robust hardware infrastructure, specifically designed to cater to machine learning tasks. Most prominently, the availability of Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs) contributes significantly to the reduction of computational time. In this research, a single GPU was primarily utilized for model training and evaluation. The selected GPU boasts multiple cores designed explicitly for accelerated computations, thereby significantly reducing the time required for training complex machine learning algorithms. TPUs, although available, were not utilized in this particular study due to the proficient performance of the GPU in handling the dataset and computation needs. By leveraging Kaggle's versatile and powerful platform, our work was not only expedited but was also accomplished with a degree of thoroughness and efficiency that significantly contributed to the rigor and credibility of the findings presented in this study.

B. Performance Metrics

In the field of machine learning, especially in classification tasks, various metrics like the Accuracy Score, AUC-ROC Score, ROC Curve, and Confusion Matrix offer nuanced insights into the model's performance. This section mathematically formalizes these metrics [32].

Accuracy Score— is quantified as the proportion of true results (both True Positives and True Negatives) among the total number of cases examined. Mathematically, it is given by the equation:

$$\text{Accuracy } = \frac{\text{True Positive }(\mathrm{TP}) + \text{ True Negative }(\mathrm{TN})}{\text{Total Samples}}$$

In cases where the classes are imbalanced, a high accuracy may not indicate a well-performing model. For example, in a dataset where 95% of the samples belong to Class A and 5% to Class B, a naive model that predicts every sample as Class A will still have an accuracy of 95%.

AUC-ROC Score— The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at varying classification thresholds. Mathematically, these rates are defined as follows:

$$\text{TPR } = \frac{\text{True Positive }(\mathrm{TP})}{\text{True Positive }(\mathrm{TP}) + \text{ False Negative (FN)}}$$

$$\text{FPR } = \frac{\text{False Positive }(\mathrm{FP})}{\text{True Negative }(\mathrm{TN}) + \text{ False Positive (FP)}}$$

Here, TPR and FPR stands for True Positive Rate and False Positive Rate respectively. The AUC-ROC Score is the area under this curve and provides a single scalar value that ranges from 0 to 1, encapsulating the model's ability to discriminate between the positive and negative classes [33].

Receiver Operating Characteristic (ROC) Curve— is an important graphical tool that visualizes the performance of a binary classification model over varying thresholds. While the AUC-ROC score offers a singular scalar value representation of the classifier's performance, the ROC curve provides a more nuanced view. The curve is plotted with True Positive Rate (TPR) on the y-axis and False Positive Rate (FPR) on the x-axis. It allows for a comprehensive understanding of the classifier's behavior across different levels of sensitivity and specificity.

Each point on the ROC curve represents a distinct threshold value, and the curve essentially quantifies the trade-off between the TPR and FPR at each threshold. A model whose curve hugs the upper left corner of the graph is considered optimal, as it signifies a high TPR and a low FPR. On the other hand, a curve that aligns closely with the diagonal line represents a classifier that performs no better than random chance.

The convexity of the ROC curve is another significant aspect. A convex curve usually signifies that the classifier is consistently performing better than random guessing across various thresholds. If the curve dips below the diagonal, it indicates that the model is performing worse than random chance for certain thresholds, which is usually a red flag for further investigation [34], [35], [36].

Confusion Matrix— exposes the model's flaws and is used in the evaluation of classification algorithms. The matrix shows four key metrics: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives. These metrics show how well the model performed on positive and negative classes.

Simple interpretation is a Confusion Matrix strength. It allows multidimensional model behavior evaluation beyond correctness. For instance, it evaluates Sensitivity (Recall or True Positive Rate), Specificity (True Negative Rate), Precision, and F1 Score.

Unbalanced categorization issues are another important Confusion Matrix use. Accuracy metrics might be misleading in strongly skewed class distributions. Through Precision, Recall, and the F1 Score, the Confusion Matrix provides a more detailed appraisal of the minority class.

The Confusion Matrix also underpins advanced evaluation metrics like the ROC Curve and AUC-ROC Score. It can also be used to assess how well the model distinguishes across classes in a multi-dimensional feature space for multi-class classification issues.

When errors have varying consequences in cost-sensitive categorization tasks, the Confusion Matrix is typically utilized. In medical diagnostics, a False Negative might have far worse effects than a False Positive, and the Confusion Matrix can help tune the classifier to prevent such high-cost errors [33], [37], [38].

C. Evaluation Report

Before diving into the results, it is crucial to briefly revisit the methodology. An iterative work procedure was employed, encompassing 12 distinct iterations. In each iteration, the following constant elements were maintained: Missing value handling using Multiple Imputation by Chained Equations (MICE), Binary encoding applied to Gender and Dataset columns and Data standardization using the StandardScaler function from the scikit-learn library. Additionally, each iteration experimented with various data balancing methods and dimensionality reduction techniques.

At the 1st iteration, minority up-sampling was chosen as the data balancing method. No dimensionality reduction techniques was utilized and classifier with the best accuracy came out to be Ensemble Stacking algorithm using Random Forest, XGBoost and CART as base model and Logistic Regression as meta model. The accuracy was 0.9101, AUC-ROC score was 0.9798.

2nd iteration utilizes SMOTE as the data balancing method and the classifier with the best accuracy was XGBoost with 0.8622 accuracy and 0.9095 AUC-ROC score.

The 3rd one uses ROSE Data balancing method and achieves 0.8982 accuracy and 0.9459 AUC-ROC score with XGBoost classifier.

4th iteration combines PCA with minority upsample using 'resample' function from scikit-learn library and attains 0.8682 accuracy and 0.9391 AUC-ROC score with the Random Forest classifier.

The 5th iteration employs SMOTE for data balancing and utilizes PCA for dimensionality reduction, with Random Forest as the best classifier achieving 0.8143 accuracy and 0.8697 AUC-ROC score.

In the 6th iteration, ROSE is used for data balancing along with PCA for dimensionality reduction, the most accurate classifier being XGBoost with an accuracy of 0.8742 and an AUC-ROC of 0.9436.

The 7th iteration utilizes minority up-sampling by 'resample' function from scikit-learn's 'sklearn.utils' package and applies Factor Analysis (FA) for dimensionality reduction, with the Random Forest classifier optimized using GridSearch yielding 0.8323 accuracy and 0.8676 AUC-ROC.

The 8th iteration integrates SMOTE for data balancing and uses Factor Analysis (FA) for dimensionality reduction, deploying CART as the classifier with feature selection and GridSearch for hyper-parameter tuning, reaching 0.7125 accuracy and 0.7330 AUC-ROC.

The 9th iteration incorporates ROSE as the data balancing method and Factor Analysis (FA) for dimensionality reduction, achieving 0.7964 accuracy and 0.8247 AUC-ROC with the Random Forest classifier.

The 10th iteration combines minority up-sampling using the 'resample' function from scikit-learn's 'sklearn.utils' package with Linear Discriminant Analysis (LDA) for dimensionality reduction, and the optimized Random Forest classifier realizes 0.8383 accuracy and 0.8698 AUC-ROC.

In the 11th iteration, SMOTE is used for data balancing and LDA for dimensionality reduction, with the ANN classifier producing 0.7724 accuracy and 0.7882 AUC-ROC.

Lastly, the 12th iteration employs ROSE for data balancing and LDA for dimensionality reduction, with the CART classifier achieving 0.8862 accuracy and 0.8792 AUC-ROC.

Across these iterations, XGBoost and Random Forest showed consistent performance. However, none surpassed the Ensemble Stacking Classifier's performance in the first iteration.

The Ensemble Stacking Classifier emerged as the most accurate model with an accuracy of 0.9101 and an AUC-ROC score of 0.9798. Notably, this model did not employ any dimensionality reduction, which suggests that feature reduction may not always yield better classification outcomes.

The ROC curve and the confusion matrix for the suggested model utilizing the Ensemble Stacking Classifier are presented in fig. 8 and 9 respectively.

D. Comparison

In the journey to develop an effective model for liver disease prediction based on the ILPD dataset, multiple classifiers were explored, each coupled with varying data preprocessing techniques. This section aims to shed light on the comparative performance of these classifiers, focusing on key metrics such as accuracy and AUC-ROC scores. The experimentation process was iterative, where classifiers were fine-tuned with different hyperparameters, feature selections, and data preprocessing techniques.

TABLE II
COMPARISON BETWEEN DEPLOYED CLASSIFIERS

Classifier	Accuracy	AUC-ROC
XGBoost	0.8982	0.9459
Random Forest	0.8742	0.9486
Gaussian Naïve Bayes	0.7789	0.8315
KNN	0.8035	0.8117
Logistic Regression	0.7868	0.7912
SVM	0.8319	0.8562
ANN	0.7851	0.7982
Ensemble Stacking	**0.9101**	**0.9798**

As can be seen from the table II, the Ensemble Stacking Classifier outperforms all the individual classifiers on both the accuracy and AUC-ROC metrics. This comparative evaluation underlines the efficacy of ensemble methods, particularly the Ensemble Stacking Classifier, in tackling the challenges of liver disease prediction.

TABLE III
COMPARISON WITH RELATED STUDY ON ILPD DATASET

Authors	Suggested Classifier	Accuracy(%)
Thirunavukkarasu K et al. [5]	Logistic Regression	73.97
Ketan Gupta et al. [6]	Random Forest	63
Ruhul Amin et al. [1]	Random Forest	88.10
Elias Dritsas et al. [7]	Voting Classifier	80.10
Srilatha Tokala et al. [8]	Random Forest	87
This Study	**Ensemble Stacking**	**91.01**

As evident from table III, the proposed model surpasses the existing methods in terms of accuracy. Specifically, the closest study in terms of accuracy is by Ruhul Amin et al., which achieved an 88.10% accuracy using Random Forest. Even in this case, the proposed model shows a notable increase in accuracy, affirming the effectiveness of using an ensemble stacking approach.Stacking of multiple algorithms has been shown to reduce overfitting and improve the model's ability to generalize to new data. Additionally, techniques like MICE for missing value imputation and StandardScaler for feature scaling contributed to creating a more robust model. The methodical iterative process allowed for the fine-tuning of hyperparameters and the selection of the most effective data balancing and dimensionality reduction techniques.

These results validate the proposed Ensemble Stacking Classifier model as a powerful tool for liver disease prediction, offering significant improvements over existing methods. In summary, the findings of this study not only demonstrate the high predictive accuracy of the proposed model but also suggest the substantial potential of ensemble methods in medical diagnosis applications.

V. Conclusion

This research primarily aimed to construct a precise and effective predictive model for diagnosing liver disease using the ILPD dataset, vital for early detection in healthcare settings. By extensively evaluating various classifiers and data preprocessing techniques, it aspired to refine existing predictive tools. The pinnacle of this study is the formulation of an Ensemble Stacking Classifier, achieving a remarkable 91.01% accuracy and a 0.9546 AUC-ROC score, surpassing existing models on this dataset. The meticulous approach to classifier selection, hyperparameter tuning, and data pre-processing has yielded a model that surpasses set objectives, introducing the Ensemble Stacking Classifier as a valuable tool for intricate healthcare analytics tasks. This comprehensive methodology acts as a scaffold for future research, offering insights into classifier performance under different conditions and serving as a supplementary tool for medical practitioners for early diagnosis and individualized treatment strategies.

Even though this study has made considerable progress in liver disease prediction, there are unexplored areas that could lead to enhancement in model predictions. There is a critical need to address data skewness in machine learning models, possibly through the use of logarithmic transformation like the log1p function, to uphold the validity of model assumptions and refine the interpretation of feature importance. Additionally, while Standard Scaling was utilized for feature scaling, exploring alternative scaling techniques such as Robust Scaling is essential to assess capability to handle outliers and it's overall impact on model performance and data normalization, potentially leading to more precise predictions. A deeper exploration into feature selection and the implementation of varying feature weights during the prediction phase can also optimize the model by concentrating on the most crucial attributes. Addressing these aspects can aid in the development of more accurate and robust diagnostic tools for liver diseases, enhancing healthcare analytics and patient outcomes.

In closing, high reliability and accuracy of the suggested model make it a credible ally in clinical decisions, facilitating earlier interventions and assuring patients of a robust, data-driven diagnostic process. Having met its primary goal with notable accuracy, this study paves the way for further research and applications in medical diagnostics, poised to positively impact patient outcomes and healthcare systems.

References

[1] R. Amin, R. Yasmin, S. Ruhi, M. H. Rahman, and M. S. Reza, "Prediction of chronic liver disease patients using integrated projection based statistical feature extraction with machine learning algorithms," *Informatics in Medicine Unlocked*, vol. 36, p. 101155, 2023.

[2] A. Q. Md, S. Kulkarni, C. J. Joshua, T. Vaichole, S. Mohan, and C. Iwendi, "Enhanced preprocessing approach using ensemble machine learning algorithms for detecting liver disease," *Biomedicines*, vol. 11, no. 2, p. 581, 2023.

[3] D. Bhupathi and N. Tan, "Liver disease detection using machine learning techniques," 2022.

[4] F. Mostafa, E. Hasan, M. Williamson, and H. Khan, "Statistical machine learning approaches to liver disease prediction," *Livers*, vol. 1, no. 4, pp. 294–312, 2021.

[5] A. S. Singh, M. Irfan, A. Chowdhury, *et al.*, "Prediction of liver disease using classification algorithms," in *2018 4th international conference on computing communication and automation (ICCCA)*, pp. 1–3, IEEE, 2018.

[6] K. Gupta, N. Jiwani, N. Afreen, and D. Divyarani, "Liver disease prediction using machine learning classification techniques," in *2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 221–226, IEEE, 2022.

[7] E. Dritsas and M. Trigka, "Supervised machine learning models for liver disease risk prediction," *Computers*, vol. 12, no. 1, p. 19, 2023.

[8] S. Tokala, K. Hajarathaiah, S. R. P. Gunda, S. Botla, L. Nalluri, P. Nagamanohar, S. Anamalamudi, and M. K. Enduri, "Liver disease prediction and classification using machine learning techniques," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 2, 2023.

[9] B. Ramana and N. Venkateswarlu, "ILPD (Indian Liver Patient Dataset)." UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C5D02C.

[10] C. K. Enders, *Applied missing data analysis*. Guilford Publications, 2022.

[11] S. Van Buuren, *Flexible imputation of missing data*. CRC press, 2018.

[12] J. Huang and C. X. Ling, "Using auc and accuracy in evaluating learning algorithms," *IEEE Transactions on knowledge and Data Engineering*, vol. 17, no. 3, pp. 299–310, 2005.

[13] C. Sammut and G. I. Webb, *Encyclopedia of machine learning*. Springer Science & Business Media, 2011.

[14] S. B. Kotsiantis, I. Zaharakis, P. Pintelas, *et al.*, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[16] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, pp. 1137–1145, Montreal, Canada, 1995.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[18] H. He and Y. Ma, "Imbalanced learning: foundations, algorithms, and applications," 2013.

[19] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[20] N. Lunardon, G. Menardi, and N. Torelli, "Rose: a package for binary imbalanced learning.," *R journal*, vol. 6, no. 1, 2014.

[21] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Rusboost: A hybrid approach to alleviating class imbalance," *IEEE transactions on systems, man, and cybernetics-part A: systems and humans*, vol. 40, no. 1, pp. 185–197, 2009.

[22] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[23] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[24] Y. Mansour and M. Schain, "Learning with maximum-entropy distributions," *Machine Learning*, vol. 45, pp. 123–145, 2001.

[25] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, pp. 3–42, 2006.

[26] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, pp. 81–106, 1986.

[27] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[28] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*, vol. 398. John Wiley & Sons, 2013.

[29] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer, 2009.

[30] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[31] M. A. Arbib, *The handbook of brain theory and neural networks*. MIT press, 2003.

[32] J. A. Motta, L. Capus, and N. Tourigny, "Evaluation of efficiency of linear techniques to optimize attribute space in machine learning: Relevant results for extractive methods of summarizing," *Computer and Information Science*, vol. 5, no. 6, p. 58, 2012.

[33] F. Bellocchio, P. Carioni, C. Lonati, M. Garbelli, F. Martínez-Martínez, S. Stuard, and L. Neri, "Enhanced sentinel surveillance system for covid-19 outbreak prediction in a large european dialysis clinics network," *International Journal of Environmental Research and Public Health*, vol. 18, no. 18, p. 9739, 2021.

[34] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[35] I. Mogotsi, "Christopher d. manning, prabhakar raghavan, and hinrich schütze: Introduction to information retrieval: Cambridge university press, cambridge, england, 2008, 482 pp, isbn: 978-0-521-86571-5," 2010.

[36] J. A. Swets, "Measuring the accuracy of diagnostic systems," *Science*, vol. 240, no. 4857, pp. 1285–1293, 1988.

[37] K. J. Cios, W. Pedrycz, R. W. Swiniarski, K. J. Cios, W. Pedrycz, and R. W. Swiniarski, *Data mining and knowledge discovery*. Springer, 1998.

[38] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, pp. 1137–1145, Montreal, Canada, 1995.