ABSTRACT

Hepatitis B virus (HBV) is a significant global health threat, responsible for severe liver diseases such as liver failure, cirrhosis, and hepatocellular carcinoma. The burden is especially high in low-income regions, where early diagnosis and treatment are critical for mitigating its impact. This study investigates the effectiveness of various machine learning (ML) techniques for diagnosing HBV and predicting patient outcomes. The Chi-squared test was used for feature selection to find the most important factors, which were later applied to train and evaluate various machine learning models. To address the class imbalance in the dataset, Synthetic Minority Over-sampling Technique (SMOTE) was used to balance the data. Among individual models, Support Vector Machines (SVM) and Logistic Regression (LR) each achieved the accuracy of 92.5%. By implementing a Voting Classifier that combined SVM and LR, the overall accuracy was improved to 95%. SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) were used to improve the models' interpretability. The results showed that higher levels of some risk factors, especially in older patients, greatly raise the risk of death. These insights provide healthcare professionals and policymakers with valuable information to develop better HBV diagnostic and patient care strategies.

1. Introduction

When we think about infectious diseases that have devastated populations across the globe, names like tuberculosis, malaria, and HIV/AIDS often come to mind. Yet, there is another disease that affects millions of lives, though it doesn’t always command the same level of attention: Hepatitis. In 2016, out of an estimated 257 million people worldwide with chronic HBV infection, only a small percentage were diagnosed and received treatment [1]. This condition, the most common liver disease worldwide, has become a significant global health concern.

Hepatitis is the swelling of the liver, which causes damage to this important organ. The liver is the largest organ inside the body and it performs many critical functions such as cleaning toxins from the blood, helping with digestion through bile production and storing energy in the form of glycogen. When hepatitis inflames the liver, these essential processes can become disrupted, leading to severe health complications throughout the body. Depending on its duration, hepatitis may be categorized as either acute or chronic. Acute hepatitis often resolves on its own but

if the inflammation persists for more than six months, it becomes chronic. The progression of the disease varies greatly between individuals, influenced by factors such as age, gender, genetics, and lifestyle. Fig. 1 shows the symptoms of the Hepatitis B Virus.

Hepatitis is most commonly caused by viral infections, classified into hepatitis A, B, C, D, E, and G. Hepatitis A, B, and C are the most widespread with the ability to cause severe illness. Of these, Hepatitis B is caused by the hepatitis B virus (HBV). HBV is a serious infection that can become chronic, putting people at a high risk of liver damage, cirrhosis, and liver cancer. The African region (6.1%) and the Western Pacific region (6.2%) have the highest prevalence of infection [2]. In contrast, North America has seen a significant decline in HBV infections, thanks to the widespread use of vaccines. However, HBV remains a silent threat. It is primarily transmitted from mother to child during childbirth, but it can also spread through unprotected sexual contact, the reuse of contaminated needles, tattooing, body piercings and even the sharing of razors or other personal items exposed to infected blood. Bodily fluids such as blood, saliva, menstrual and vaginal fluids and semen are all potential carriers of the virus.

For those infected with HBV, the symptoms can be severe. Acute cases typically show a range of symptoms such as jaundice (yellowing of the skin and eyes), dark urine, clay-colored stool, abdominal pain, fatigue, and loss of appetite. Although some individuals recover naturally, others progress to chronic infection, where the risk of developing life-threatening liver complications increases significantly. As the impact of Hepatitis B continues to unfold across the globe, it becomes clear that timely diagnosis and effective treatment are critical. Understanding the nature of this disease, its transmission and its symptoms provides a foundation upon which healthcare strategies can be built, allowing us to combat its devastating effects more effectively.

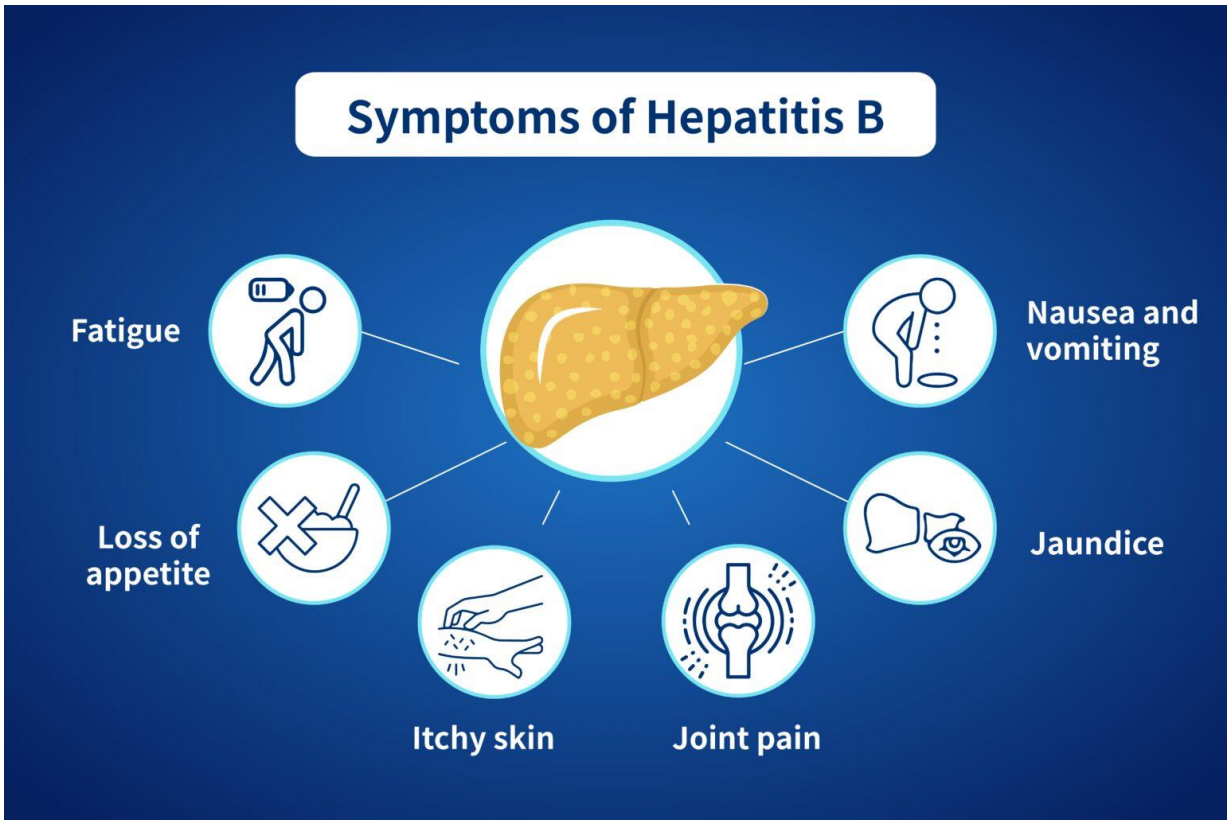


Fig. 1. Symptoms of Hepatitis B Virus.

Chronic HBV develops slowly so signs and symptoms may be hard to notice. According to World Health Organization (WHO) estimation, about 254 million people worldwide are infected chronically with HBV [3]. Most people do not have any symptoms when they are newly infected, but some may get acute illness with symptoms that last for weeks. A small group of people with acute HBV can develop liver failure, which can be deadly. In some cases, HBV can lead to chronic liver failure, cirrhosis (scarring of the liver), or liver cancer. This chronic HBV has worse outcomes than acute HBV. Table 1 illustrates the phases of chronic HBV.

The HBV consists of a protective outer shell or capsid enclosing its DNA. The virus initiates infection by binding to specific receptors on the surface of liver cells, known as hepatocytes. This receptor-ligand interaction is analogous to a key fitting into a lock, allowing the virus to enter the cell. Once inside, the capsid disassembles and the viral DNA is transported to the cell's nucleus, the central command of cellular operations. Using the hepatocyte's machinery, HBV replicates its genetic material and synthesizes the necessary proteins to form new viral particles. After replication, newly formed viruses exit the infected hepatocyte and continue the cycle by invading adjacent cells. Throughout this process, the body's immune system

recognizes the infected hepatocytes as compromised and mounts an immune response, often leading to inflammation and further damage to the liver.

Table 1

Stages of chronic HBV infection. [4]

Phase	ALT	HBV DNA	HBeAg	Liver Histology
Immune Tolerant	Normal	Elevated Typically >1 million IU/mL	Positive	Minimal inflammation and fibrosis
HBeAg-Positive Immune Active	Elevated	Elevated ≥20,000 IU/mL	Positive	Moderate-to-severe inflammation and fibrosis
Inactive Chronic Hepatitis B	Normal	Low or undetectable <2,000 IU/mL	Negative	Minimal necroinflammation but variable fibrosis
HBeAg-Negative Immune Reactivation	Elevated	Elevated ≥2,000 IU/mL	Negative	Moderate-to-severe inflammation or fibrosis

Several biochemical and virological criteria are utilized in conventional medical procedures to identify individuals with HBV infection based on the degree of liver damage and viral activity [5]. Research has also demonstrated that using clinical data [6-8], other conventional statistical techniques can forecast both acute and chronic HBV. Since variability affects extremely dimensional medical data, such forecasts made using the traditional statistical approach may be skewed. Moreover, applying sophisticated data analysis, such machine learning (ML) algorithms, may have several benefits in complicated viral illness scenarios where standard laboratory testing might not provide enough information about the clinical history and patient development. With the use of these techniques, new biomarkers for predictive and treatment strategies may be discovered by gaining understanding of the pathogen and host variables. Structures that reveal undiscovered correlations among characteristics can be used by ML systems to make predictions about upcoming occurrences. But in the field of medical health, ML algorithms have drawn

a lot of interest and have been applied to a variety of medical issues [9-13]. In order to gather information from medical data and produce precise predictions for HBV diagnosis, ML algorithms have been applied successfully as a classification technique [14-17].

Artificial intelligence (AI) has made it possible for us to identify and treat complicated illnesses. Medical professionals and health officials may struggle to understand the model findings because many ML studies in the literature have not considered model explainability. In the meantime, stakeholders in the health sector and real-world applications may benefit from the interpretation of ML models to comprehend the characteristics that lead to HBV infection. Thus, the main achievements of this work consist of

- **Feature Selection:** Employed the Chi-squared (Chi2) test for feature selection, identifying the most significant factors contributing to the diagnosis of HBV.
- **Data Balancing and Preprocessing:** Addressed data imbalance using SMOTE and applied standardization to scale the dataset for improved model performance.
- **Machine Learning Models:** Implemented and evaluated ten ML classifiers to assess their effectiveness in HBV diagnosis.
- **Ensemble Techniques:** Utilized ensemble learning methods Voting Classifier to achieve the highest predictive accuracy across models.
- **Model Interpretability:** Applied explainability techniques SHAP and LIME, to elucidate model predictions by analyzing feature contributions, enhancing transparency and trustworthiness of the results.

2. Related Work

Numerous studies have been conducted on the diagnosis of HBV using a variety of classification techniques. Table 2 summarizes the most advanced methods for HBV diagnosis, highlighting their performance metrics and corresponding ML algorithms.

Tian et al. [18] used a number of ML algorithm to improve the prognosis for individuals with chronic HBV. They created models using four algorithms: decision tree (DT), logistic regression (LR), random forest (RF), and extreme gradient boosting (XGBoost). The best model was determined by the area under the receiver operating characteristic curve (AUC). The AUC values were 0.619 for DT, 0.680

for LR, 0.829 for RF, and 0.891 for XGBoost, with XGBoost providing the best predictive results. The study concluded that these ML models were effective and demonstrated improved predictive performance with the available clinical data.

G. Obaido et al. [19] were proposed to make an early HBV diagnosis. To develop the models, they used SHapley Additive exPlanations (SHAP), a game-based approach for explaining and visualizing the predictions of ML models used for HBV diagnosis. The algorithms included decision tree (75% accuracy), logistic regression (82%), support vector machines (SVM) (75%), random forest (86%), adaptive boosting (AdaBoost) (90%), and XGBoost (92%). Additionally, the SHAP values indicated that bilirubin is the most important feature linked to a higher mortality rate.

V. K. Yarasuri et al. [20] created a method to predict the risk of HBV using data from the UCI ML repository, looking at different clinical cases. The dataset includes 155 examples with 20 features, one of which is a class that helps determine the life expectancy of hepatitis patients. They used various machine learning techniques and neural networks for hepatitis diagnosis. The study compared the accuracy of different ML and artificial neural network (ANN) methods to find the best tool for diagnosing hepatitis. They measured performance based on accuracy rates and mean squared error. The models used included Support Vector Machine (SVM), K-Nearest Neighbor (KNN) ans ANN for predicting the disease. The results showed that the ANN had the highest accuracy at 96% and the lowest mean squared error among all the models.

V. M. Putri et al. [21] used a ML approach to predict Hepatitis B vaccination status with Naive Bayes and random forest classifiers. Since the dataset was imbalanced, the models were biased toward the majority class. To fix this, they applied SMOTE to balance the data. The study, based on the 2017 National Socio-Economic Survey for Aceh province (with 2264 cases and 14 variables), showed that SMOTE improved the accuracy of Naive Bayes by 26.09% and random forest by 30.08%. Naive Bayes with SMOTE was the best model, achieving 73.22% accuracy. Key factors affecting vaccination status included the mother's education, mother's and father's occupations, father's education, and the number of health facilities.

Table 2
Current methods for diagnosis HBV and their performance metrics.

Existing work	Dataset	Methods	Performance Metrics
Tian et al. [18]	2,235 record of real patients	XGBoost, RF, DT, LR	Max AUC 0.891 (XGBoost)
G. Obaido et al. [19]	UCI ML Repo: 155 Hepatitis B patients	DT, LR, SVM, RF, AdaBoost, XGBoost	Max Accuracy of 92% (AdaBoost)
V. K. Yarasuri et al. [20]	UCI ML Repo: 155 Hepatitis B patients	SVM, KNN, ANN	Max Accuracy of 96% (ANN)
V. M. Putri et al. [21]	2, 264 record of real patients	RF, NB	Max Accuracy of 82.19% (RF without SMOTE)
A. Alamsyah et al. [22]	UCI ML Repo: 155 Hepatitis B patients	SVM + PCA	Max Accuracy of 93.55%
Fahad R. Albogamy et al. [23]	UCI ML Repo: 155 Hepatitis B patients	BiLSTM	Max Accuracy of 95.08%
Shipeng Chen et al. [24]	307 record of real patients	RF	Max Accuracy of 90%
Junfeng Peng et al. [25]	UCI ML Repo: 155 Hepatitis B patients	LR + DT + KNN, XGBoost + SVM + RF	Max Accuracy of 91.9%

A. Alamsyah et al. [22] used data mining techniques to improve the accuracy of hepatitis diagnosis. A SVM was combined with principal component analysis (PCA) to enhance classification performance. PCA was applied to extract and reduce features, effectively decreasing data dimensionality while retaining critical information. The reduced data was then classified using SVM and the performance was evaluated through a confusion matrix. The study achieved a hepatitis prediction accuracy of 93.55%, outperforming the SVM classification without PCA.

Fahad R. Albogamy et al. [23] developed a deep learning decision support system (DSS) to classify survivability and mortality in patients with severe hepatitis. They used a bidirectional long short-term memory (BiLSTM) model with balanced data to improve accuracy. The results showed that the BiLSTM model was effective, reaching 95.08% accuracy, 93% recall, 94% precision and 93% F1-score. This approach outperformed existing methods in hepatitis detection, showing significant improvements in classification accuracy.

Shipeng Chen et al. [24] designed a ML approach to predict Hepatocellular Carcinoma (HCC) in patients with chronic hepatitis B (CHB) by using sequence features from rt nucleic acid and rt/s amino acid sequences. The random forest (RF) model, built with 10 combined features, achieved the highest predictive performance, with an AUC of 0.96 and an accuracy of 0.90 in both the training and validation sets. The RF model's HCC risk scores (AUC 0.966) outperformed α-fetoprotein in distinguishing between HCC and CHB patients. This study highlights the importance of HBV rt sequences and the integration of deep sequencing with ML for HCC risk assessment.

Junfeng Peng et al. [25] proposed an explainable artificial intelligence (XAI) framework to improve the transparency of black-box models in hepatitis diagnosis. The study used both transparent models (LR, DT, KNN) and black-box models (XGBoost, SVM, RF) on a UCI hepatitis dataset. SHAP, LIME, and PDP were applied to enhance model interpretability. The RF model achieved the highest accuracy (91.9%), and the XAI framework provided insights into model decisions, improving treatment strategies and patient prognosis.

3. Materials and Methods

This section provides a comprehensive overview of the techniques and resources utilized in this study. The process began with importing the HBV dataset, followed by data preprocessing, where missing numerical values were handled using the median, and duplicate rows were checked. Feature selection was conducted using the Chi-squared (Chi2) test, and SMOTE was applied to address class imbalance. The dataset was scaled before training and testing different ML models. These models included Decision Tree, Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Random Forest, AdaBoost, Gaussian Naive Bayes, LightGBM, CatBoost, and Voting Classifier. Performance analysis was carried out using metrics such as the confusion matrix and ROC curve. Additionally, model explainability was explored using SHAP and LIME techniques. Fig. 2 shows the workflow of the proposed method for diagnosing HBV.

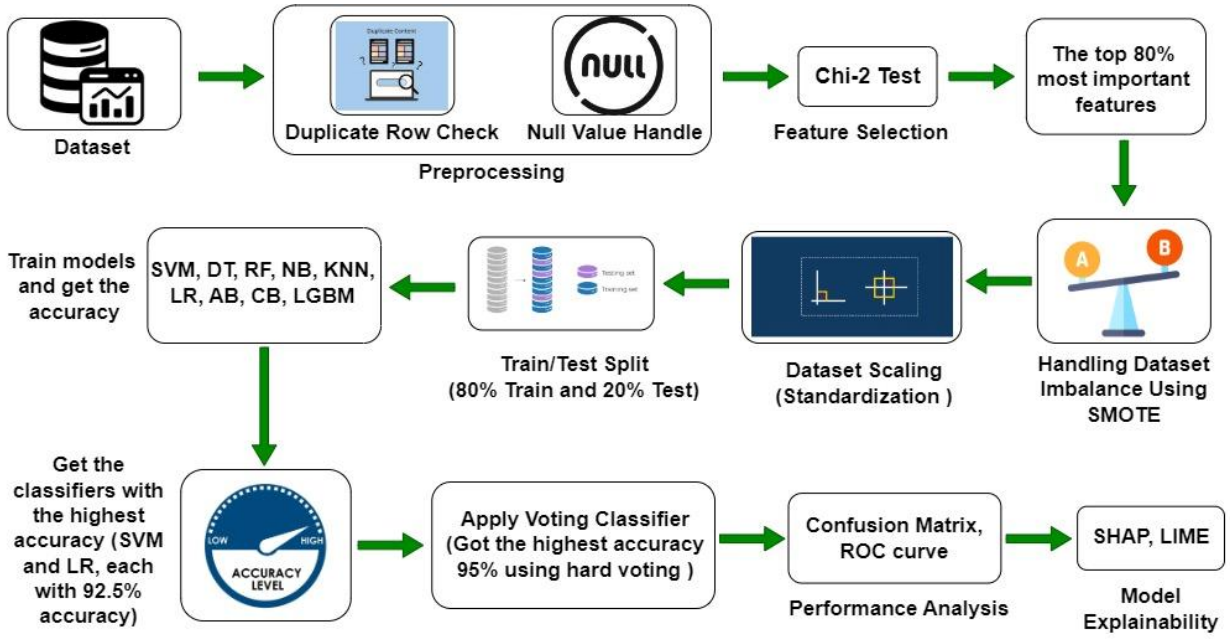


Fig. 2. Workflow of the proposed methodology.

3.1. Dataset preparation

The dataset used in this study was sourced from the University of California Irvine (UCI) Machine Learning Repository [26], selected due to its established role in providing curated datasets for the ML community. Table 3 provides a summary of the dataset's features. The choice to use hepatitis datasets from this repository was made deliberately, given the limited availability of relevant data in this area of

research. The dataset includes demographic and clinical information for 155 hepatitis B patients, encompassing 20 distinct features.

Table 3

A broad synopsis of the UCI ML repository's HBV dataset.

Dataset types	Multivariate
Attribute types	Categorical, Integer, Real
Total features (Col in general)	20
Total instances	155
Numeric features	6
Categorical features	14
Target column	Class
Classification Categories	Die/Live
Number of Live Instances	123
Number of Die Instances	32
Data Source	UCI ML Repo

To prevent data leakage, the dataset was first checked for duplicate samples, but none were found. Missing data were addressed using the fillna method, which handled 167 missing values across 20 features in a dataset of 155 records. A summary of the attributes in the dataset is presented in Table 4. The missing values were imputed with the median using the fillna technique, replacing the NULL values with specified values. After feature selection, the top 80% of the most important features were selected for further analysis. The dataset had an issue of imbalance, where one class had significantly fewer instances compared to the other, which could negatively impact the performance of ML models. To tackle this issue, SMOTE [27] was used to create synthetic samples for the minority class, which helped balance the dataset.

Additionally, the dataset was scale using a standardization scaling method, which transformed the features to have a mean of zero and a standard deviation of one. This ensured that all features were on the same scale, preventing any feature from dominating the model due to larger values, and improving the overall model performance. The features were further standardized using the Z-score scaling method [28]. In Eq. (1), X represents the original values, while μ and σ denote the mean and standard deviation of the feature f_i, respectively. The standardized values are denoted as Z, with the following equation:

$$Z = \frac{X-\mu}{\sigma} \quad (1)$$

where $X, \mu, \sigma \in f_i$. This process ensures that all features have an equal impact on the analysis, enhancing the performance of ML algorithms. After scaling the dataset, 80% was set aside for training, and the remaining 20% was kept for testing.

Table 4

Overall summary of the HBV dataset available in the UCI ML Repository.

SL	Features	Description	Data Type
1	Age	Age of the patient	Numerical
2	Sex	Patient's gender: Male or Female	Nominal
3	Steroid	Use of steroids, which may increase HBV replication	Nominal
4	Antivirals	Antiviral drugs to inhibit HBV replication	Nominal
5	Fatigue	Fatigue due to impaired functional capacity from HBV	Nominal
6	Malaise	General discomfort and unease	Nominal
7	Anorexia	Loss of appetite	Nominal
8	Liver Big	Presence of liver enlargement	Nominal
9	Liver Firm	Firmness of the liver	Nominal
10	Spleen Palpable	Palpable enlargement of the spleen	Nominal
11	Spiders	Unusual grouping of blood vessels closes to the skin's surface.	Nominal
12	Ascites	Build-up of fluid in the abdomen	Nominal
13	Varices	Swollen veins, commonly found in the esophagus or stomach	Nominal
14	Bilirubin	Yellowish pigment formed by red blood cell breakdown	Numerical
15	Alk Phosphate	Enzyme used to assess liver fibrosis levels	Numerical
16	Sgot	Liver enzyme indicating liver function	Numerical
17	Albumin	Protein produced by the liver	Numerical
18	Protime	Blood clotting protein produced by the liver	Numerical
19	Histology	Microscopic examination of liver tissue	Nominal
20	Class	Patient outcome: "Died" or "Lived"	Nominal

Fig. 3 presents a visualization of the distribution of six features along with their corresponding percentages of missing values. The distribution reveals that some

features contain distant outliers. Additionally, certain features exhibit a high proportion of missing values, making imputation with measures of central tendency inappropriate, as it would distort their distributions.

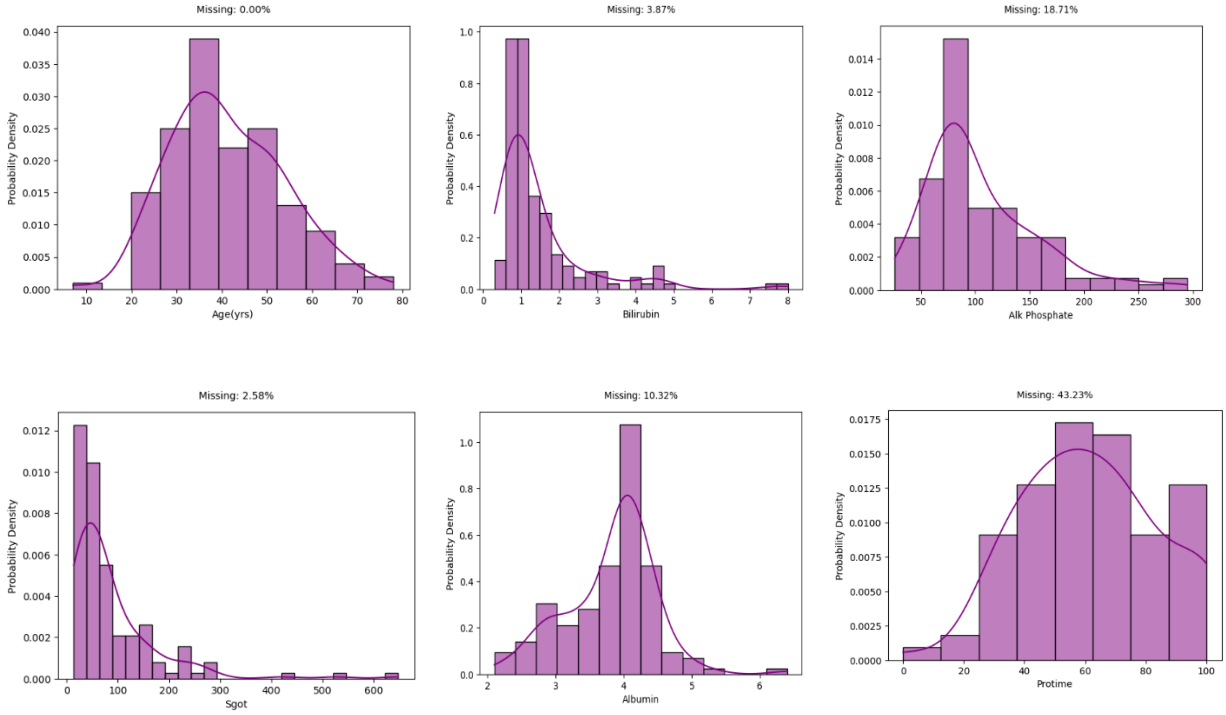


Fig. 3 Displays the percentage of missing values across the numerical features in the dataset's distribution.

In the first figure of Fig. 3 table, the number of patients with missing age data is displayed, with a missing percentage of 0%. The subsequent figure shows the percentage of missing values for Bilirubin, which is 3.87%. To the right, the missing data percentage for Alk Phosphate is shown to be 18.71%. The next figure presents the missing value percentage for SGOT, which is 2.58%. Additionally, the missing count for Albumin is 10.32%, while Protime has the highest missing value percentage at 43.23%.

Fig. 4 illustrates the distribution of the nominal features, along with the numeric of each attribute present in the dataset. The visualization reveals a class imbalance, which was addressed using SMOTE.

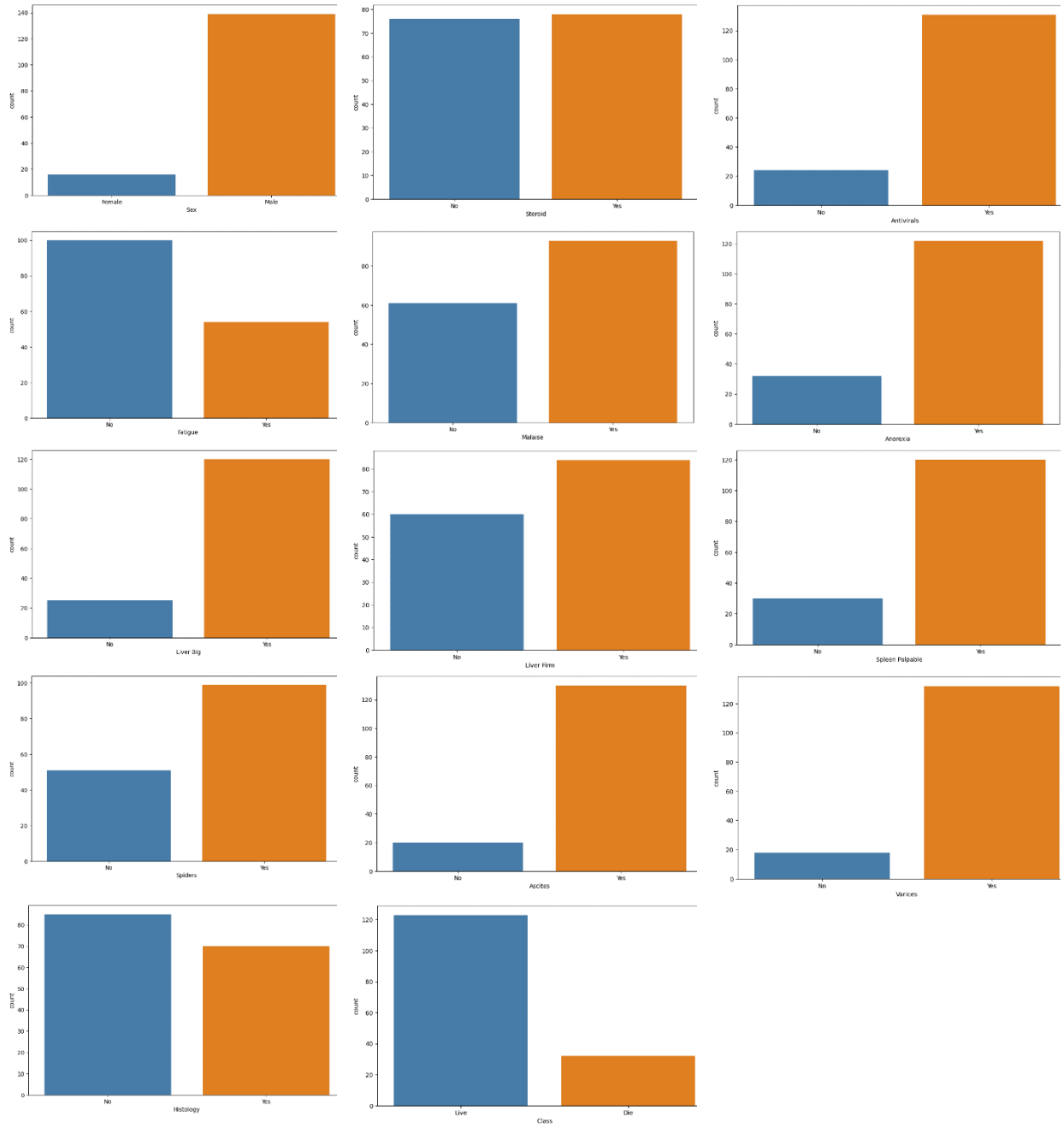


Fig. 4 provides a visual representation of the distribution of nominal features in the dataset.

Table of figure, Fig. 4 presents the distribution of the nominal features in the dataset. The first figure shows the distribution of Gender, with 139 males and 16 females. For Steroid, 76 patients have 'No', 78 have 'Yes', and 1 value is missing. The Antivirals count indicates 131 'Yes' and 24 'No'. Fatigue shows 54 'Yes', 100 'No', and 1 missing value. Malaise has 93 'Yes', 61 'No', with 1 missing entry. For Anorexia, 122 are 'Yes', 32 'No', and 1 is missing. Liver Big shows 120 'Yes', 25

'No', and 10 missing values. Liver Firm has 84 'Yes', 60 'No', and 11 missing values. Spleen Palpable counts 120 'Yes', 30 'No', and 5 missing values. For Spiders, 99 are 'Yes', 51 are 'No', and 5 are missing. Ascites has 130 'Yes', 20 'No', with 5 missing values. Varices shows 132 'Yes', 18 'No', and 5 missing values. Histology counts 70 'Yes', 85 'No', with no missing data. Finally, the Class attribute has 123 'Live' and 32 'Die' cases, with no missing values.

3.2 Features selection

As highlighted by Qin et al. (2020) [29], identifying the most critical risk factors in healthcare informatics can reduce the time required for ML algorithm training, improve data consistency, remove redundant features and increase prediction performance. In recent years, researchers have adopted multiple approaches for selecting relevant features, including the Chi2 test, Principal Component Analysis (PCA), Recursive Feature Elimination (RFE), and Mutual Information (MI). Among these methods, the Chi2 Test has emerged as a key approach for feature selection.

In this study, the Chi2 test was applied to identify the most important features from the dataset. The Chi2 test examines the statistical independence of each feature from the target variable, helping to identify those with the strongest association. The working principle of the Chi2 test is demonstrated in Eq. (2). In the equation, χ^2 represents the Chi-squared statistic, O denotes the observed value, and E is the expected value based on the null hypothesis.

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{2}$$

Following the Chi2 test, the features were ranked by importance and the top 80% of the most important features were selected for further analysis. Eq. (3) defines $\in_{\omega}$, which represents the set of important features identified through the Chi2 test. This set contains features that have been deemed statistically significant based on their Chi2 values and their relevance to the target variable. x represents a feature from the dataset. After chi2 test, $\in_{\omega}$ becomes $\in_{\omega} = \{Protime, Bilirubin, Age, Alk\ Phospate, Spiders, Albumin, Histology, Ascites,$

$$Fatigue, Malaise, Varices, Spleen\ Palpable, Sgot, Sex, Anorexia, Antivirals\}$$

$$\in_{\omega} = \{x \mid x \in \chi^2\} \quad (3)$$

A Pearson correlation check was performed to identify highly correlated features, particularly those with a correlation coefficient exceeding 80%. However, no strongly correlated features were found. The Pearson correlation matrix, illustrated in Fig. 5, visualizes the relationships between features. The formula for calculating the Pearson correlation coefficient is given in Eq. (4):

$$r = \frac{\sum(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum(a_i - \bar{a})^2 \sum(b_i - \bar{b})^2}} \quad (4)$$

In this equation, r denotes the correlation coefficient, a_i and b_i represent individual data points for two variables, while $\bar{a}$ and $\bar{b}$ are their respective means. This formula assesses both the strength and direction of the linear relationship between the two variables.

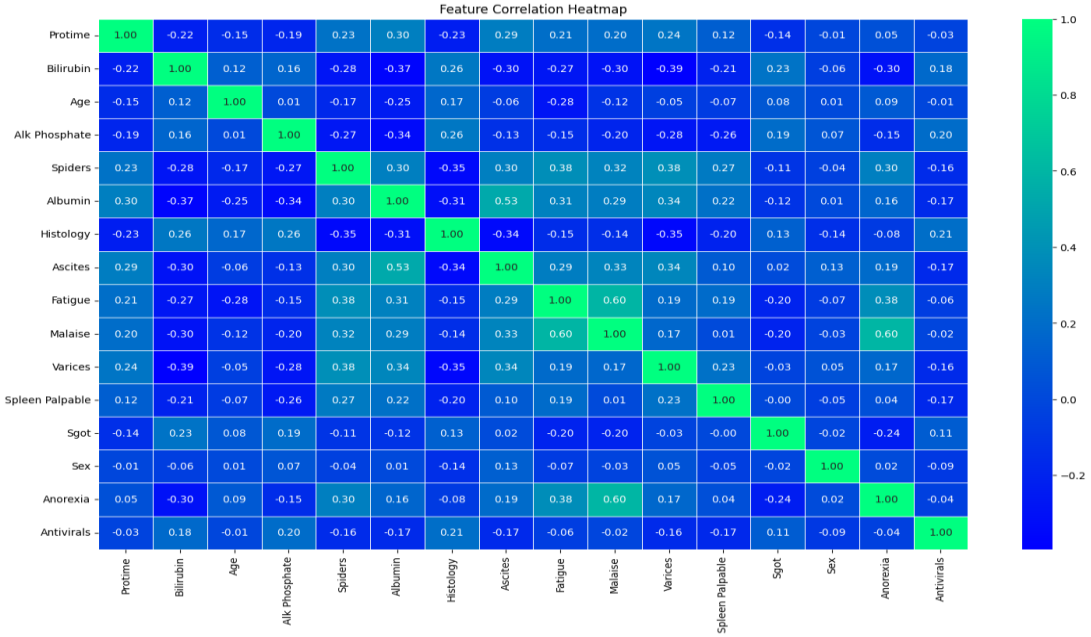


Fig. 5 Pearson correlation matrix of selected features.

Fig. 5 represents the correlation matrix of the selected features identified after the feature selection process. The correlation values can vary from -1 to +1. A correlation value of 0 means there is no linear relationship between the two features. As the value gets closer to +1, it indicates a strong positive correlation, which means the features increase together. On the other hand, values near -1 suggest a strong negative correlation, where an increase in one feature corresponds with a decrease in the other, indicating an inverse relationship.

3.3. Machine learning models

With the use of data structures, ML models enable predictions systems to produce precise forecasts in a variety of fields, improving efficiency and decision-making. Regression and classification in clinical and health diagnosis involve the development of models based on historical analysis in both supervised and unsupervised learning methodologies. 10 classifiers—Decision Tree, Random Forest, Logistic Regression, Gaussian naïve bayes, Support Vector Machines, K-Nearest Neighbor, Light-GBM classifier, Ada Boost, Cat-Boost and Voting are used to evaluate the performance; these are all covered in this section. The best accurate model among these classifiers is then evaluated, indicating its superior performance.

3.3.1 Decision tree classifier

The decision-making process of the human brain served as the model for a class of ML algorithms called decision trees. In order to forecast the value of a target variable using a decision model or tree-like graph, this nonparametric supervised learning technique for regression and classification learns fundamental decision rules from the properties of the data. Under the category of supervised machine learning, decision trees are frequently used to solve categorization problems [30]. As a result of its node at the beginning that branches into several options, decision trees usually have a tree-like shape. Tree performance is independent of nonlinear relationships between parameters, and it can handle both numerical and category data. Through a top-down recursive strategy, its learning approach computes the information contribution of each feature type, based on the theory of loss function minimization. Several DT techniques include ID3, C4.5, and classification and regression trees (CART) [31-33]. Applications for classification and hypothesis-supporting decisions have been seen by segmenting the data category based on this information [34-37].

In DT, leaf nodes represent classes, branching represents outcomes, and internal nodes represent input factors. This structure is made up of numerous layers

of nodes, with the root node at the top and the leaves at the bottom [30]. While the entropy is calculated using two features by Eq. (6), the entropy of each class is determined by Eq. (5). Using Eq. (7), we calculate the feature-level gain G. The total proportionate entropy is obtained by applying Eq. (8) to the entropy of each branch that results from a progressive separation of the data features. In this instance, the feature level's height gain ratio is determined by G/I.

$$B(L) = \sum_{i=1}^{c} -P_i log_2 P_i \quad (5)$$

$$B(L, M) = \sum_{c \in H} P(c).E(c) \quad (6)$$

$$G(L, M) = B(L) - B(L, M) \quad (7)$$

$$I(L, M) = \sum_{j=1}^{v} \frac{M_i}{M} log_2 \frac{M_i}{M} \quad (8)$$

3.3.2 Random-forest classifier

An ensemble classifier called a Random Forest (RF) classifier creates multiple decision trees by using a randomly selected subset of training samples and variables. Many interconnected decision trees are produced by this algorithm. Decision trees act as the cornerstones of this methodology. It is applicable for both regression and classification tasks. For classification problems, the algorithm predicts the class that most decision trees select. In contrast, for regression tasks, it provides the average prediction from the various decision trees. The random forest algorithm addresses the overfitting issue associated with decision trees [38]. Additionally, the method builds individual trees based on several bootstrap samples; the accuracy increases with the number of trees in the forest [39]. Moreover, the RF significantly outperforms traditional decision tree algorithms in feature selection, as it does it automatically [40]. The final ensemble prediction for a given test sample x with T trees and C classes can be computed as follows:

$$H_T(x) = argmax_j \sum_{k=1}^{K} I(h_k(x) = j), for\ j = 1, \dots, C \quad (9)$$

A feature's importance is assessed by dividing its proportional reduction in node impurity by the probability of reaching that node. One way to calculate the node probability is to divide the entire quantity of possible samples by the total quantity of samples that arrive at the node. A higher score indicates greater significance for the trait.

$$P_{ij} = L_j C_j - L_{left(j)} C_{left(j)} - L_{right(j)} C_{right(j)} \quad (10)$$

$$Pi\ sub\ (j) = the\ significance\ of\ node\ j$$

$L\ sub\ (j) = samples\ arriving\ at\ node\ j$

$C\ sub\ (j) = the\ node\ j's\ impurity\ value$

$left\ (j) = split\ on\ node\ j\ by\ the\ child\ node\ from\ the\ left$

$right\ (j) = split\ on\ node\ j\ by\ the\ child\ node\ from\ the\ right$

3.3.3 Logistic Regression

The 1958 David Cox-created logistic regression (LR) model has become a well-known statistical classification technique that is frequently used in biological, medical, and therapeutic research [41]. It is particularly useful for modeling binary-dependent variables. To establish the relationship between binary-dependent and independent variables, LR estimates probabilities using a logistic function or cumulative logistic distribution, which helps determine if an individual is healthy or ill. Since it mainly predicts a binary target variable based on one or more predictor variables, it is often referred to as binomial LR [42, 43].

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_m x_m \quad (11)$$

where x_i stands for the predictor variables, β_i for the coefficients of the regression, and π for the probability of a specific event, such as the existence or absence of HBV [44].

3.3.4 Gaussian Naïve Bayes

For mathematical categorization, the concept of Bayes is applied via a Gaussian Naïve Bayes classifier. The procedure is anxious to acquire information. It is efficient at classifying the latest sample because it doesn't wait on test data to learn. It can classify data in a way that is comparable to decision trees and neural networks [45]. The conditional likelihoods of the collected data fitting within the period are computed by the NBs classifier using the total amount of HBV and non-HBV data in every individual measurement interval. As a supervised ML technique, this type of classification is employed for healthcare analysis of statistical information because of its remarkable performance on large datasets and capacity to yield meaningful results. It assigns an individual i containing an array of characteristics x_i to class c utilizing a Naive Bayes classifier and a Bayesian probabilistic equation in order to optimize the subsequent probability $P(c|x_{i1,} \ldots, x_{iM})$ [46]. Utilizing Eq. (12) and $P(L|N)$ as a possible input proposition over a specific collection of

information that includes $P(N|L), P(L)$ and $P(N)$ possibilities, the resulting classifier is constructed.

$$P(L|N) = \frac{P(N|L)\times P(L)}{P(N)} \quad (12)$$

For every input occurrence $N = N_1, N_2, \ldots N_n$ for the response element L, $P(N|L)$ concentrates on the conditional chance distribution. Relative odds of the response attribute are written as $P(L)$, and marginal possibility of the input parameter occurrence is written as $P(N)$. The NBs classifier's labeling estimation is given a function by Eq. (13).

$$L = argmax_L P(L) \prod_{k=1}^{n} P(N_k|L) \quad (13)$$

3.3.5 Support Vector Machine

For the purposes of data characterization and regression, various ML approaches are now in use. Techniques for ML essentially allow the machine to carry out the assigned work. SVM is one such instance. SVM is regarded as an efficient classifier since, in addition to classifying the data, it maximizes the separation among data clusters in order to optimize the decision boundaries. Support vectors are a collection of values closest to both sides of the hyperplane. Kernel functions, also known as transformation functions, are used to change data that cannot be separated sequentially into a new, higher-dimensional space where they may be divided sequentially. Kernel functions come in several forms, including Gaussian radial basis function (RBF), polynomial, quadratic, and linear [47, 48]. To determine the SVM's answer for a dual optimization issue where $y \in \{1, -1\}$ and $\propto \epsilon R^N$, let's assume that we have an attribute vector x with d dimension and an initial set of N separated by linearity samples.

$$\underset{a}{maximize} \sum_{i=1}^{n} \propto_i - \frac{1}{2}\sum_{i=1}^{n} \propto_i \propto_j y_i y_j (X_i^T, X_j) \quad (14)$$

$$subject\ to\ \propto_i \geq 0\ and\ \sum_{i=1}^{n} \propto_i y_i = 0 \quad (15)$$

Utilizing the experimental risk-management approach, the SVM algorithm's primary objective is to calculate the best accessible surface for separating both

positive and negative instances from the data. An algorithmic decision boundary computation is attempted [49]. Using a hyperplane to divide the set of simulated variables into two separate groups is the primary SVM problem. Eq. (15) illustrates the SVM calculating procedure.

$$D = \{(x^1, y^1), \dots, (x^1, y^1)\}, x \in \mathbb{R}, y \in \{-1,1\} \quad (16)$$

If there was no error and the disparity among the variables that were closest to the hyperplane was maximal, then the hyperplane was deemed to have divided the total number of vectors effectively. Eq. (17) limits both parameters c and q for a standard hyperplane, which can be examined because Eq. (16) includes a certain repetition.

$$(c, x) + q = 0 \quad (17)$$

$$min_i |\,(\,c, x^i\,) \,+\, q \,=\, 1 \quad (18)$$

Eq. (19) computes the distance $d\ (c, q; x)$ among a point x and the hyperplane (c, q).

$$d(c, q;\ x) \ = \ \frac{|(c,x^i\,) + q|}{||c||} \quad (19)$$

The optimal hyperplane is produced by optimizing the margin, in accordance with the constraints of Eq. (18). Here's how the margin is determined:

$$\begin{aligned} p(c,q) &= \min_{x^i y^i=-1} d\left(c, q; x^i\right) + \min_{x^i y^i=1} d(c, q; x^i) \\ &= \min_{x^i y^i=-1} \frac{|\,(c,x^i)+q|}{||c||} + \min_{x^i y^i=1} \frac{|\,(c,x^i)+q|}{||c||} \\ &= \frac{1}{||c||}\left(\min_{x^i y^i=-1} \frac{|\,(c,x^i)+q|}{||c||} + \min_{x^i y^i=1} \frac{|\,(c,x^i)+q|}{||c||}\right) \\ &= \frac{|2|}{||c||} \end{aligned} \quad (20)$$

Consequently, use Eq. (21) to reduce yields the most suitable hyperplane to optimize data separation.

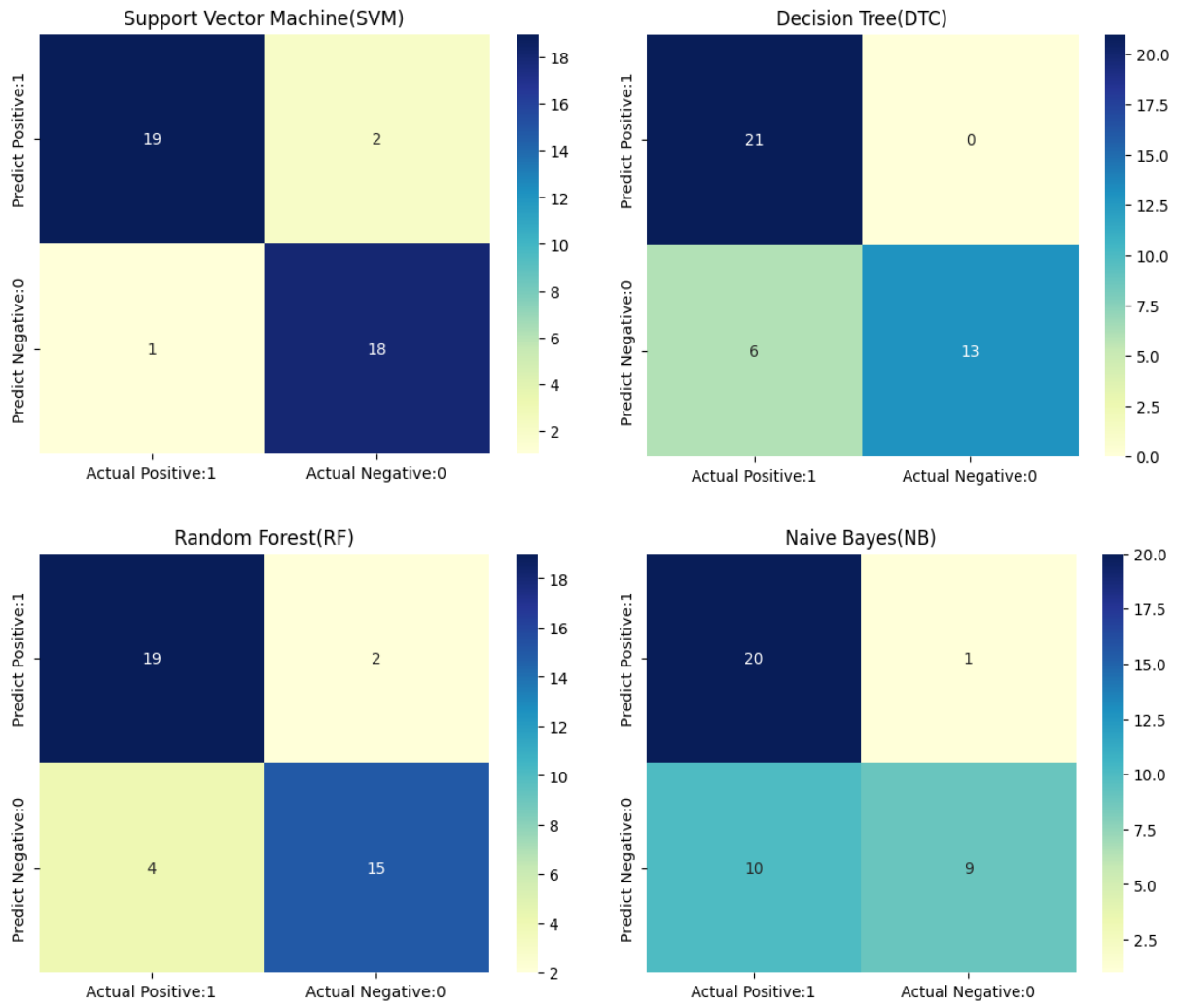
$$\varphi(c) \ = \ \frac{1}{2}||c||^2 \quad (21)$$

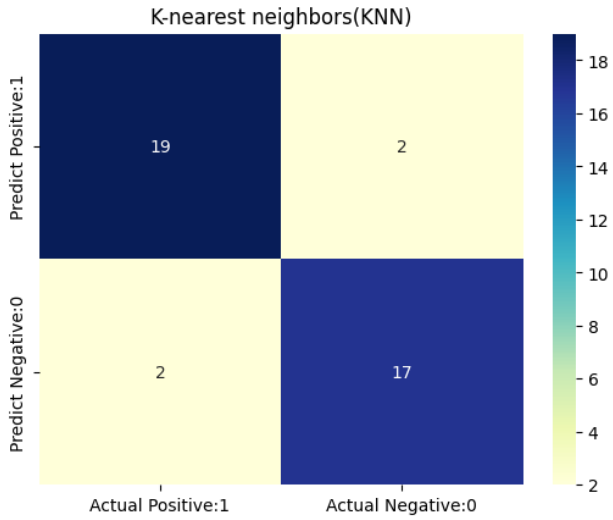
To illustrate the similarity between minimizing Eq. (21) and applying the structural risk minimization (SRM) concept, consider the subsequent limitation.

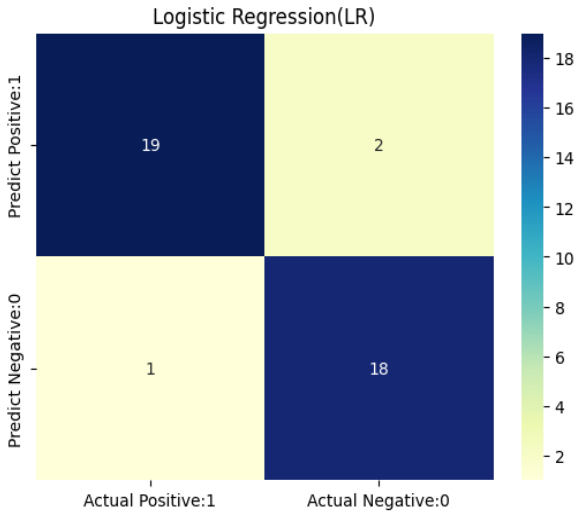
$$||c|| < A \tag{22}$$

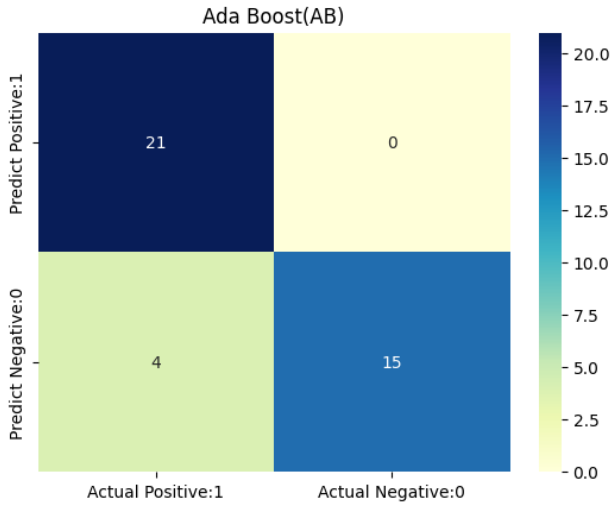
Following that, construct an additional formula by utilizing Eqs. (18) and (19).

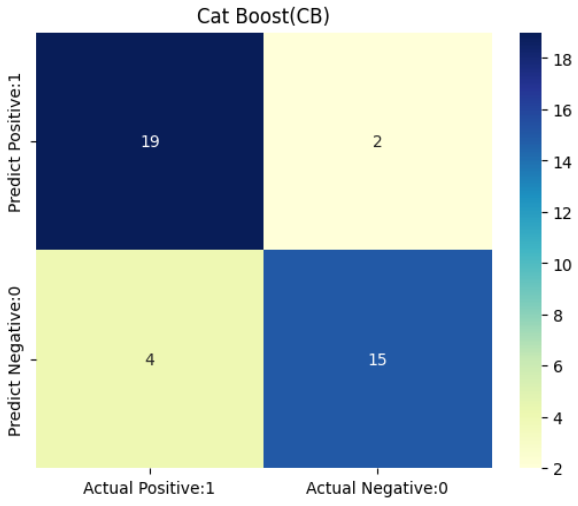
$$d(c, q;\ x) \geq \frac{1}{A} \tag{23}$$

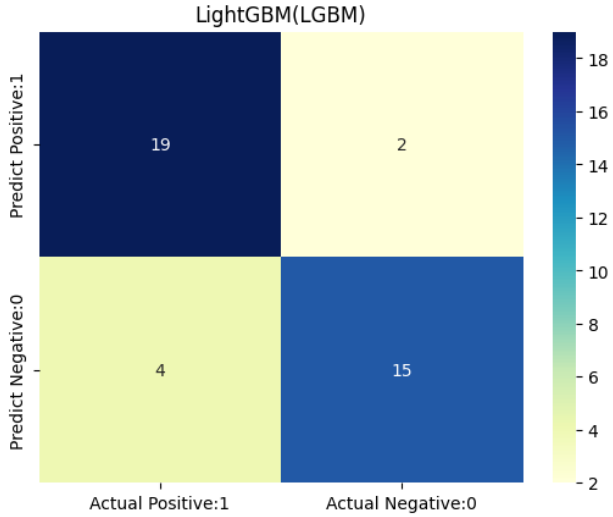


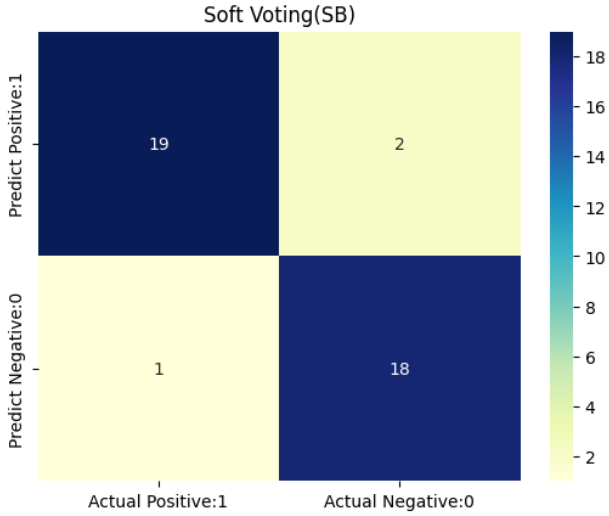

K-nearest neighbors(KNN)
Predict Positive:1
Predict Negative:0
19
2
2
17
Actual Positive:1
Actual Negative:0
18
16
14
12
10
8
6
4
2


Logistic Regression(LR)
Predict Positive:1
Predict Negative:0
19
2
1
18
Actual Positive:1
Actual Negative:0
18
16
14
12
10
8
6
4
2


Ada Boost(AB)
Predict Positive:1
Predict Negative:0
21
0
4
15
Actual Positive:1
Actual Negative:0
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0


Cat Boost(CB)
Predict Positive:1
Predict Negative:0
19
2
4
15
Actual Positive:1
Actual Negative:0
18
16
14
12
10
8
6
4
2


LightGBM(LGBM)
Predict Positive:1
Predict Negative:0
19
2
4
15
Actual Positive:1
Actual Negative:0
18
16
14
12
10
8
6
4
2


Soft Voting(SB)
Predict Positive:1
Predict Negative:0
19
2
1
18
Actual Positive:1
Actual Negative:0
18
16
14
12
10
8
6
4
2

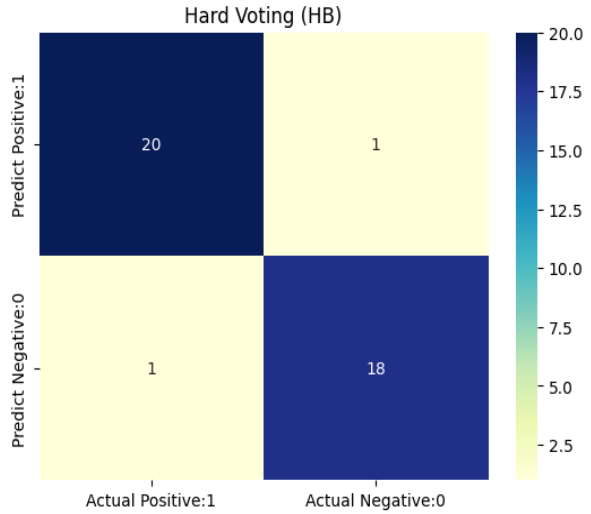


Fig. 6 Confusion Matrix of different ML models.

3.3.6 K-nearest neighbor classifier (KNN)

The KNN technique employs unsupervised ML to address classification and regression problems. It was developed in 1951 by Evelyn Fix and Joseph Hodges and later improved by Thomas Cover. The comparable metric is used by this method to categorize the data. "K" indicates how many of the closest neighbors should be taken into account. The closest points are determined by taking into account various distance metrics, such as Euclidean, Manhattan, Chebyshev, Hamming, and others. It doesn't need any training of the data. The KNN algorithm requires standardized data that is in order to be used [50]. KNN maintains a record of every instance and classifies each one separately using a correlation score. In K-nearest neighbors, text can be expressed using a coordinate vector $S = S(D_1, E_1; D_2, E_2; \ldots\ldots\ldots D_n, E_{n2};)$ Any text can be made comparable by using the training text to identify and evaluate its connection. The works that have the greatest degree of correspondence are selected. Lastly, the classes are ascertained using K neighbors.

The characteristic vectors about every piece of training text and the newly received text are compared using the subsequent technique.

$$sim\,(Pi, Pj) = \frac{\sum_{k=1}^{N} DikDjk}{\sqrt{\sum_{k=1}^{N} Dik^2}\sqrt{\sum_{k=1}^{N} Djk^2}} \quad (24)$$

The feature vectors of the simulated text are denoted by P_j and the input text by P_i in this case. N determines how big the feature vector is. Specifically, D_{ik} and D_{jk} represent the k^{th} constituents of vectors P_i and P_j. These are the real formulas utilized by KNN:

$$Q(Pi, Cm) = \sum_{j=1}^{k} sim(Pi, Pj)\ \delta(Pi, Cm) \quad (25)$$

$$\delta(Pi, Cm) = \begin{cases} 1\ if\ Pi \in Cm \\ 0\ if\ Pi \in Cm \end{cases} \quad (26)$$

3.3.7 Light-GBM classifier

A well-known gradient enhancement approach that makes use of tree-based learning techniques is called LGBM (Light Gradient Boosting Machine). As part of an automated distributed gradient-boosting ML mechanism, it forms a part of the free and open-source Light-GBM library. Large datasets are especially well-suited for its efficient and distributed training approach. Continuous characteristics are bucked through specific segments using a histogram-based method by LGBM, which expedites training and requires less memory. LGBM grows tree leaf-wise as opposed to level-wise, as is the case with typical gradient boosting algorithms. Performance is enhanced and dimensionality is decreased when LGBM handles categorical information directly instead of requiring one-hot encoding. Improved accuracy may result from this. The LGBM composite demonstrated greater temporal, computational, and operational performance. $X = (xi, yi)$ represents a dataset with attributes x and a label y. Eq. (27), with P_0 serving as the original fit optimization target and Γ serving as the loss function.

$$\hat{P} = arg_P\ minE_{x,y}[\Gamma(y, P(x)] \quad (27)$$

Eq. (28) provides the gradient or pseudo-relativistic C_m for the m^{th} iteration, this is fitted by the decision tree $hm(x)$.

$$C_m = -\frac{\partial \Gamma(y_i,\ \hat{P})}{\partial P} \quad (28)$$

The iterative criterion in Eq. (29), where λ_m is a multiplier optimized by the linear optimization strategy and acts as a step size, is used by GBDT to optimize the loss function. Eq. (29) can be used to obtain λ_m.

$$P_m(x) = P_{m-1}(x) + \lambda_m h_m(x) \quad (29)$$

$$\lambda_m = \arg_\lambda \min \sum_{i=1}^{N} \Gamma(y_i, P_{m-1}(x_i) + \lambda h_m(x_i)) \quad (30)$$

3.3.8 AdaBoost classifier

Another technique for improving collaborative learning is the AdaBoost algorithm. This approach, which is well-liked for enhancing quality, builds powerful classifiers by linearly combining poor classifiers. Using Classification and Regression Trees (CART) as the base classifier, the boosting decision tree algorithm is a boosting technique. In this case, it can be considered a specific instance of the AdaBoost method. The boosting decision tree is also a highly effective learning technique because the linear arrangement of decision trees may suit the training data adequately [51]. Starting with an initial learner typically a decision tree—the AdaBoost method evenly weights each training sample. Each round of estimation updates the weights of each of the instances; samples that were properly predicted have lower weights than poorly predicted samples, and vice versa [52]. As a result, samples that were incorrectly predicted are given greater weights in later training cycles, ensuring that the algorithm gives them more consideration. By summing the weighted predictions from the different base models, the ultimate prediction is obtained [53]. The amounts of weight are allocated for each example in B at each iteration, assuming that $B = \{(k_1, l), \dots, (k_i, l_i), \dots, (k_n, l_n)\}$ symbolizes the training set and that l_i indicates the class label of sample k_i and $l_i = -1, +1$ [54]. After training $h_t(k)$ weak classifiers with the basic learner L for $t = \{1, \dots, T\}$ iterations, the sample weight B_1 and weight update B_{t+1} are calculated utilizing:

$$B_1(i) = \frac{1}{n}, i = 1,2, \dots, n \quad (31)$$

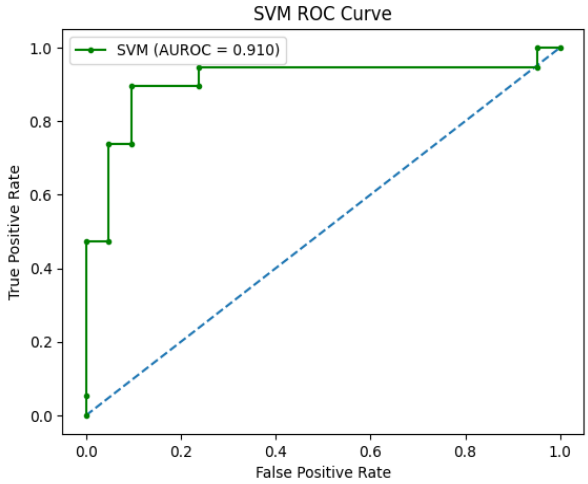
$$B_{t+1}(i) = \frac{B_t(i)}{Z_t} \exp\left(-\propto_t l_i h_t(k_i)\right), i = 1,2, \dots, n \quad (32)$$

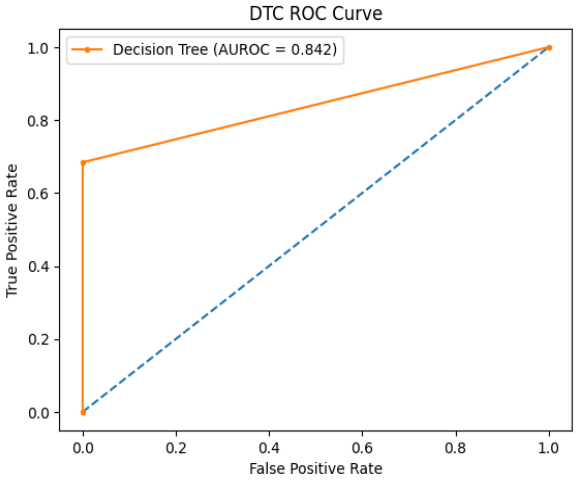
Where $\propto_t$ is the classifier $h_t(k)$ weight and Z_t is a leveling variable. When determining the ultimate estimated forecasting, the significance of $h_t(k)$ is gauged by the classifier weight $\propto_t$. In the meantime, in the $t + 1$ training round [55, 56], greater weights are allocated to the instances that were incorrectly predicted in $h_t(k)$. The weak classifiers' objective is to reduce the error rate ϵ_t:

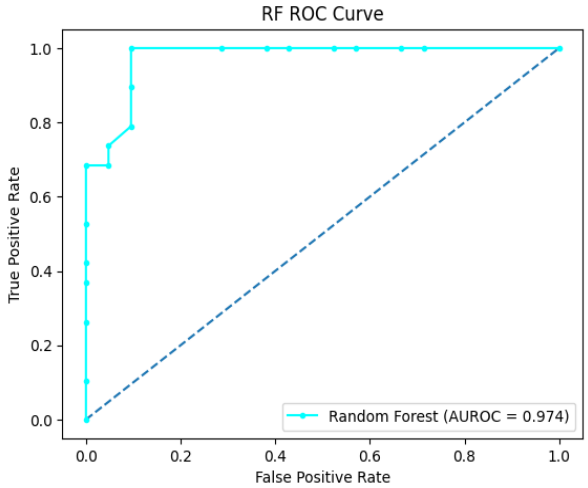
$$\epsilon_t = P[h_t(k_i) \neq l_i] = \sum_{i=1}^{n} B_i(i) I[h_t(k_i) \neq l_i] \quad (33)$$

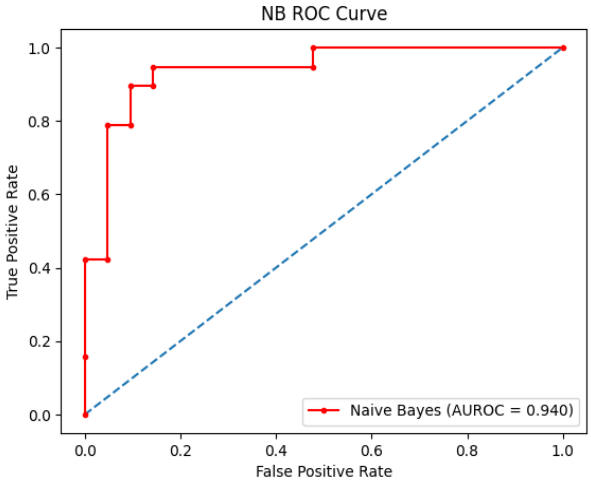
Ultimately, following the specified quantity of learning cycles, the collective classifier is produced by means of:

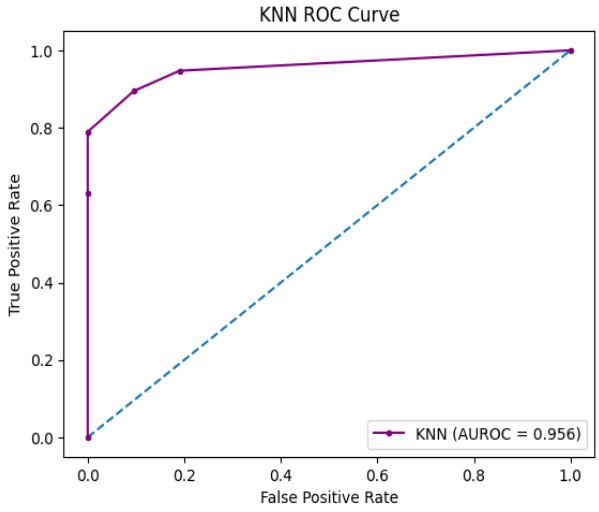
$$H(k) = sign(\sum_{t=1}^{T} \propto_t h_t(k)) \quad (34)$$

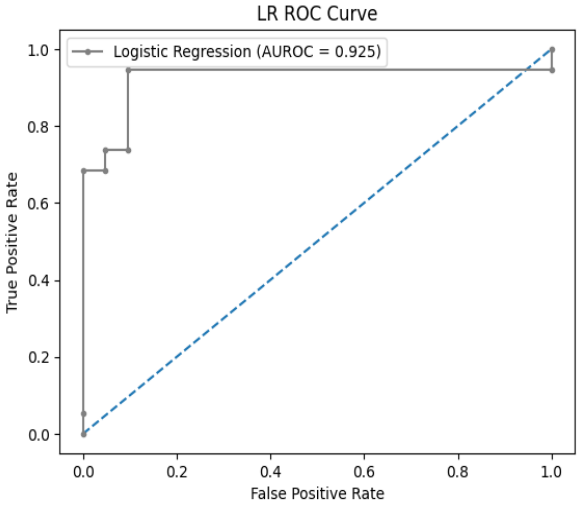


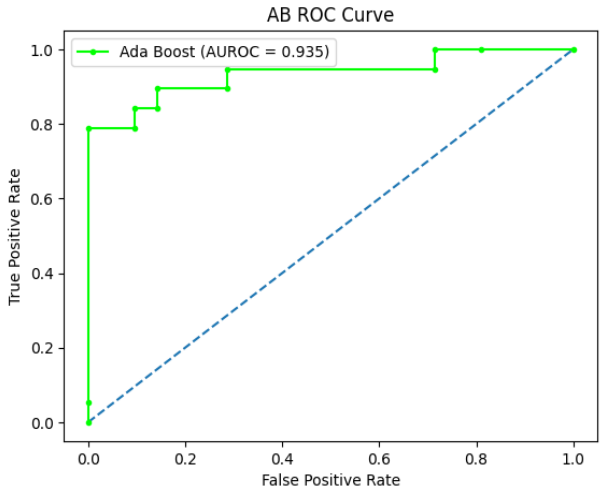


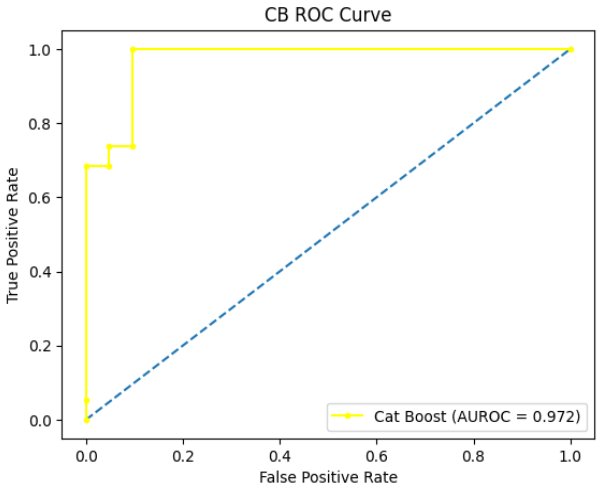


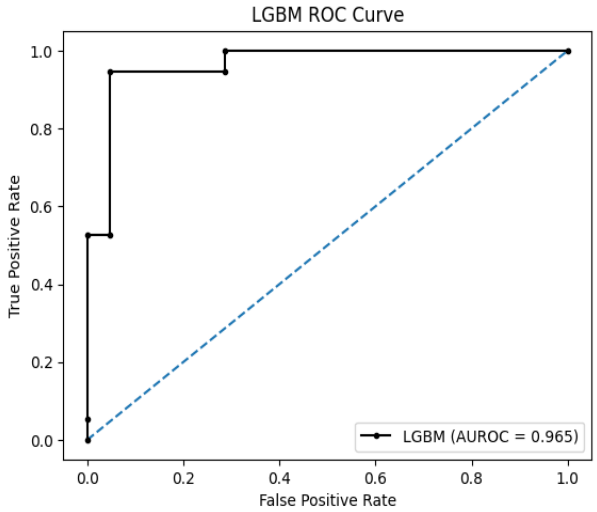


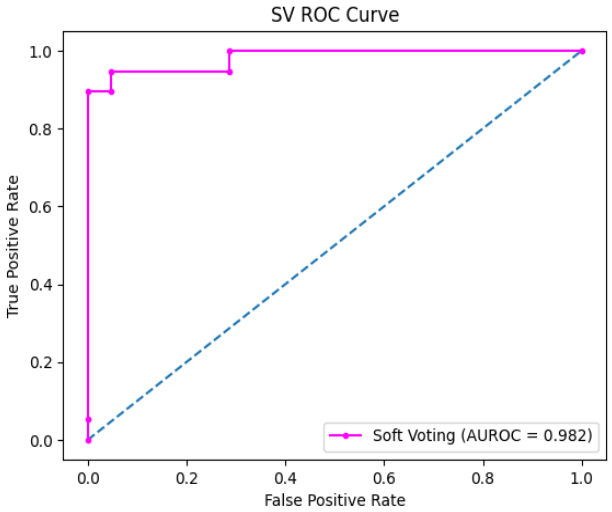

KNN ROC Curve
True Positive Rate
False Positive Rate
KNN (AUROC = 0.956)


LR ROC Curve
True Positive Rate
False Positive Rate
Logistic Regression (AUROC = 0.925)


AB ROC Curve
True Positive Rate
False Positive Rate
Ada Boost (AUROC = 0.935)


CB ROC Curve
True Positive Rate
False Positive Rate
Cat Boost (AUROC = 0.972)


LGBM ROC Curve
True Positive Rate
False Positive Rate
LGBM (AUROC = 0.965)


SV ROC Curve
True Positive Rate
False Positive Rate
Soft Voting (AUROC = 0.982)

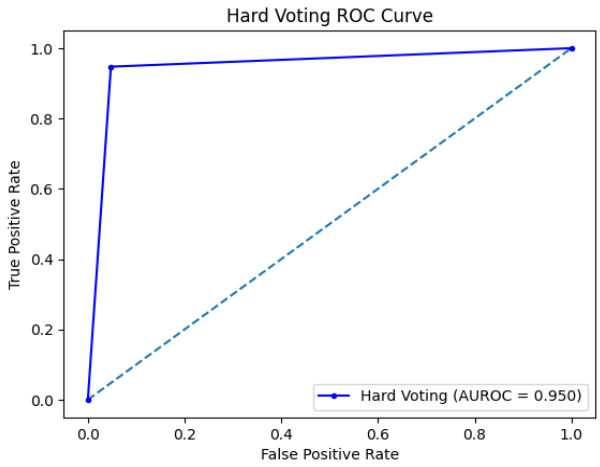


Fig. 7 ROC Curve of different classifiers.

3.3.9 Cat-Boost classifier

Yandex has developed a gradient boosting library called Cat-Boost (Categorical Boosting). Categorical characteristics are effectively processed by Cat-Boost. It doesn't require a lot of preprocessing because it employs a unique encoding technique to transform category elements through numerical values. By dividing data into training and validation sets in a novel way throughout the enhancement phase, this method helps minimize overfitting. The idea of gradient boosting on decision trees (GBDT) is the foundation of Cat-Boost. An expression for a GBDT's fundamental equations is as follows:

$$F(x) = \sum_{m=1}^{M} \gamma_m h_m(x) \quad (35)$$

Where, $F(x)$ is the prediction for input x.

M is the number of tress (iterations).

$h_m(x)$ is the $m - th$ decision tree.

γ_m is the weight associated with the $m - th$ tree.

Assuming that the data are samples $D = (Xj, yj)\ j = 1, \dots, m$, we can see that $Xj = (x1\ j,\ x2\ j, .. xn\ j)$ is a vector with response feature and n attributes. Binary (yes or no) or numeric (0 or 1) properties can be found in $y_j \in \mathbb{R}$. An uncertain probability $p(.,.)$ is followed in the random arrangement of the samples (Xj, yj). In order to

reduce the anticipated loss given in Eq. (36), the learning job aims to construct a function $M: \mathbb{R}^n \to \mathbb{R}$.

$$\mathcal{L}(M) := \Psi L(y, M(X)) \quad (36)$$

(X, y), represents the evaluation data that was chosen from the initial data D, and $L(.,.)$ displays the simulated function of loss.

3.3.10 Voting classifier

A meta classifier called the Ensemble Voting Classifier is used to combine similar or skillfully exceptional ML classifiers for recognition as well as categorization. The Ensemble Voting Classifier employs both 'hard' and 'soft' voting methods [57]. Hard and Soft both voting is used in this study.

i. *Hard Voting:* In majority voting, hard ensemble voting is the simplest scenario. In this case, the class label L is chosen by each classifier Q_j casting a majority vote:
$$L = mode\,\{Q_1(x), Q_2(x), \dots, Q_m(x)\} \quad [j = 1,2,3, \dots, m] \quad (37)$$

ii. *Soft Voting:* The predicted probability P_{ij} of each occurrence $'i'$ classifier determine the class names in soft ensemble voting. When the predictors are highly coordinated, this way is typically recommended.
$$L = \arg max_i \sum_{j=1}^{m} V_j P_{ij} \quad [j = 1,2,3, \dots, m; i = 1,2,3, \dots, n] \quad (38)$$

where V_j is the load that can be distributed with the j-th classifier.

3.4 Model Explainability Techniques

Model explainability is crucial in ML, especially in fields like healthcare and finance where understanding model decisions is essential. As models become more complex, interpretability techniques help clarify their decision-making processes. Two widely used model-agnostic methods are SHAP and LIME. SHAP uses game theory to assign values to features, showing how they influence predictions both overall and for specific cases. LIME simplifies the model by creating an easier-to-understand version, explaining individual predictions by changing the input data and observing how features affect the results.

These techniques are key to ensuring transparency and accountability in ML. By explaining how models arrive at their predictions, SHAP and LIME facilitate debugging, reduce bias, and enhance user trust. Their use is particularly valuable in high-stakes applications, where understanding and validating decisions can improve the reliability and fairness of predictive models. As AI continues to advance, explainability methods like SHAP and LIME will play a vital role in developing interpretable and trustworthy systems.

3.4.1 SHapley Additive exPlanations (SHAP)

One of the most widely used state-of-the-art ML explainability tools is SHAP. This method leverages concepts from cooperative game theory to characterize a model's performance [58, 59]. Different versions of SHAP are effective in explaining various types of ML models, including Tree-based SHAP, Deep SHAP, and Kernel SHAP. For example, Tree-based SHAP is especially suitable for models like decision trees, gradient-boosted trees such as XGBoost and CatBoost, as well as random forests.

SHAP employs a sequential attribute assignment technique, which can be viewed as a continuous summation of input variables to elucidate a model's performance. The function $g(x^{!})$ representing the explanation of the initially utilized model $f(x)$ is defined as follows, assuming an input variable vector $x = (x_1, x_2, x_3, \dots, x_n)$:

$$f(x) = g(x^{!}) = \theta_0 + \sum_{i=1}^{M} \theta_i x_i^{!} \quad (39)$$

In Eq. (39), θ_0 represents the baseline value, and M denotes the total number of input attributes. The SHAP value represents the overall mean of all possible configurations' marginal contributions. SHAP values ensure consistency in the explanations for each model's specific predictions [60]. Moreover, SHAP effectively differentiates between the positive and negative contributions of features, with higher SHAP values reflecting a more substantial positive influence on the model's capacity to predict outcomes, like the presence of HBV. SHAP is applicable to both regression and classification problems [61].

3.4.2 Local Interpretable Model-agnostic Explanations (LIME)

One of the most widely used machine learning explainability techniques is LIME. LIME provides local explanations by approximating the predictions of complex models using simpler, interpretable models around specific instances [62]. This approach allows for insights into how the model makes predictions while remaining agnostic to the underlying structure of the original model. LIME can be applied to various types of ML algorithms, including classifiers and regressors, making it a versatile tool for model interpretability.

LIME approximates the complex model $f(x)$ with a linear model $g(x^!)$ that is locally interpretable around the instance of interest x. The mathematical formulation can be expressed as follows:

$$g(x^!) = \theta_0 + \sum_{i=1}^{M} \theta_i x_i^! \quad (40)$$

In Eq. (40) $g(x^!)$ is the prediction from the local surrogate model, θ_0 is the intercept term, θ_i represents the weight or contribution of the i-th feature $x_i^!$ to the prediction, M is the total number of features used.

The surrogate model g is trained on a perturbed dataset generated by sampling around the original instance x. The coefficients θ_i represent the significance of each feature in the prediction, offering insights into the decision-making process of the complex model in the vicinity of x. This local interpretability is particularly beneficial for understanding model behavior in specific scenarios, making LIME a valuable tool for ensuring transparency in machine learning applications.

4. Experimental Results

This section presents a thorough overview of the experimental results. The dataset was randomized and separated into training and testing sets. Several data preparation methods were applied initially to the dataset to prevent data leakage and overfitting before it was utilized.

4.1 Environmental Setup

The experiments were conducted under several conditions to obtain the results and the experimental setup is detailed in Table 5. All methods were implemented locally using Python, with tools such as Python notebooks and libraries like Matplotlib, NumPy, Pandas, Seaborn, and Scikit-Learn. The project code was implemented in Google Colab.

Table 5

Specifications of the System for Developing the ML Model

CPU	1 × 11TH Gen Intel®Core™i5- 1135G7 @ 2.40GHz
RAM	8 GB
Cache	46 MB
GPU	Intel iRISX^{e}
GPU Memory	8 GB
Session Limit	10 h
Disk Space	256 GB

4.2 Performance Matrix

For assessing a ML model's effectiveness for identifying objects on a given dataset, a confusion matrix is an essential instrument. It summarizes how well the model predicts categorical labels for input instances, which is an important way of gauging its efficiency. The desired variable's actual and anticipated values are compared in this Table 6.

	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Table 6: Confusion Matrix

The model that follows assessment metrics can be computed using a matrix:

i. *Accuracy:* It is the percentage of forecasts for which our model was adjusted. A greater precision number indicates a better algorithm.

$$Accuracy = \frac{TN+TP}{TN+TP+FN+FP} \times 100\% \quad (41)$$

ii. *Precision:* It is the percentage of instances of positive that were anticipated accurately. A superior model is indicated by greater precision.

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (42)$$

iii. *Recall:* It is the percentage of real positive cases that were anticipated accurately. Sensitivity, or True Positive Rate (TPR), is the term for it. More accurate models are implied by greater values.

$$Recall = \frac{TP}{TN+TP+FN+FP} \times 100\% \quad (43)$$

iv. *F1 score:* It is an assessment model's average of precision (P) and recall (R) values. A model with a higher F1 score is considered superior.

$$F1\ score = 2 \times \frac{P \times R}{P+R} \times 100\% \quad (44)$$

Additionally, various assessment criteria were employed to evaluate the effectiveness of the available machine learning classifiers. Here, TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively. Test samples labeled as positive are considered true positives, while those classified as negative are termed false positives. A positive test sample is labeled as a false negative if it is a true negative, and a negative test sample is classified as a true negative if it is indeed negative. Furthermore, two key metrics used to demonstrate the success of machine learning algorithms are the receiver operating characteristic curve (ROC) and the area under the ROC curve (AUC). To better illustrate the performance of the machine learning models, we utilized both measures in this study. The AUC indicates the algorithm's ability to differentiate between healthy (positive) and sick (negative) individuals, as it represents the ROC curve, which graphs the true positive rate (TPR) against the false positive rate (FPR) at different threshold levels. A higher AUC value indicates better discrimination between the two classes.

4.3 Performance Analysis

Fig. 8 shows the accuracy achieved in distinguishing between Live and Die groups using the dataset [26] excluding voting. The 20% dataset allocated for testing and 80% for training. Table 7 presents the calculations for accuracy, precision, recall, and F1-score. Among the 9 ML classifiers, the SVM classifier and LR performed the best, achieving a maximum accuracy of 92.5%, with 90% precision, 95% recall, and a 92% F1-score for both. Subsequently, a voting classifier was implemented, combining the most accuracy classifiers SVM and LR. Using soft voting, the classifier achieved a maximum accuracy of 92.5%, with a precision of 90%, recall of 95%, and an F1-score of 92%.

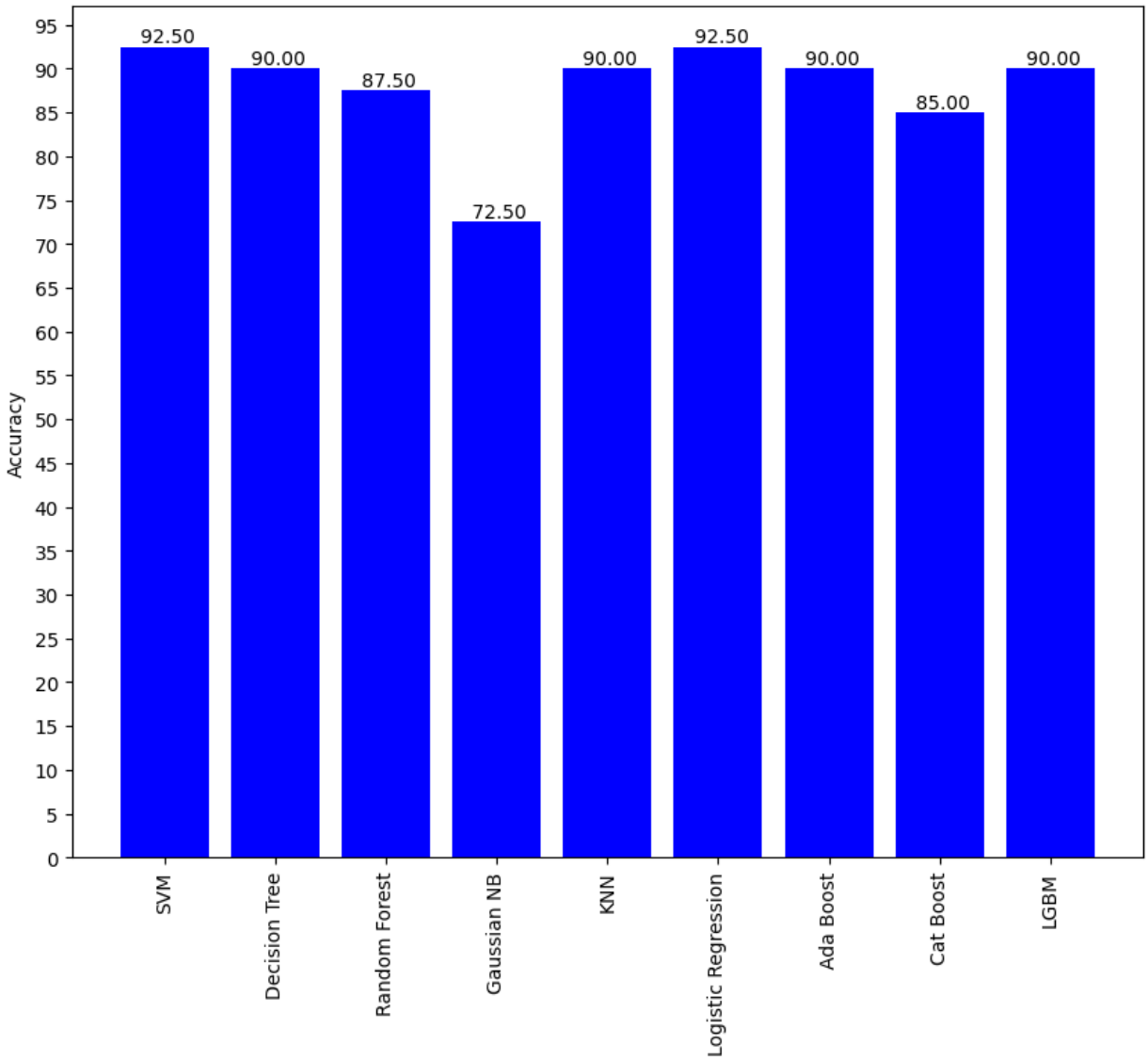


Fig. 8 Overall classification accuracy percentages (excluding voting).

However, the highest performance was observed with hard voting, where the classifier attained an accuracy of 95%, with 95% precision, 95% recall, and a 95% F1-score. Overall, the hard voting classifier showed the highest performance in terms of accuracy and other metrics, while the Gaussian Naïve Bayes classifier had the lowest accuracy at 72.50%. Fig. 9 illustrates the accuracy comparison across different classifiers, including both soft and hard voting classifiers.

Table 7

Measures of the ML model's performance throughout the whole dataset.

ML Algorithms	Accuracy	Precision	Recall	F1-score
Support Vector Machines	92.50%	90%	95%	92%
Decision Tree Classifier	90.00%	94%	84%	89%
Random Forest Classifier	87.50%	89%	84%	86%
Naïve Bayes Classifier	72.50%	90%	47%	62%
KNN Classifier	90.00%	89%	89%	89%
Logistic Regression	92.50%	90%	95%	92%
Ada Boost Classifier	90.00%	100%	79%	88%
Cat Boost Classifier	85.00%	88%	79%	83%
LGBM Classifier	90.00%	88%	79%	83%
Soft Voting Classifier	92.50%	90%	95%	92%
Hard Voting Classifier	95.00%	95%	95%	95%

Table 7 presents a comparative analysis of the performance of several ML algorithms using four key metrics: F1-score, recall, precision, and accuracy. Among the algorithms, the Hard Voting Classifier stood out, achieving the highest accuracy of 95%, along with balanced performance across precision, recall, and F1-score, each at 95%. This demonstrates its greater capacity to accurately identify both positive and negative cases. The SVM and LR models followed closely with 92.5% accuracy, 90% precision, 95% recall, and an F1-score of 92%, showing strong performance with minimal errors in both false positives and false negatives.

The DT Classifier achieved 90% accuracy with high precision (94%) but lower recall (84%), indicating it missed some true positives. Similarly, the Random Forest Classifier had 87.5% accuracy with a slight imbalance between precision (89%) and recall (84%). The Naïve Bayes Classifier performed the worst, with 72.5% accuracy and low recall (47%) despite high precision (90%). KNN showed a balanced

performance with 90% accuracy and 89% across all metrics. The AdaBoost Classifier achieved 90% accuracy and perfect precision (100%), though its recall was lower (79%). Both CatBoost and LGBM classifiers had 85% accuracy, with slightly higher precision (88%) and lower recall (79%). The Soft Voting Classifier matched SVM and Logistic Regression with 92.5% accuracy, 90% precision, and 95% recall, while the Hard Voting Classifier outperformed all with 95% accuracy and balanced metrics across the board.

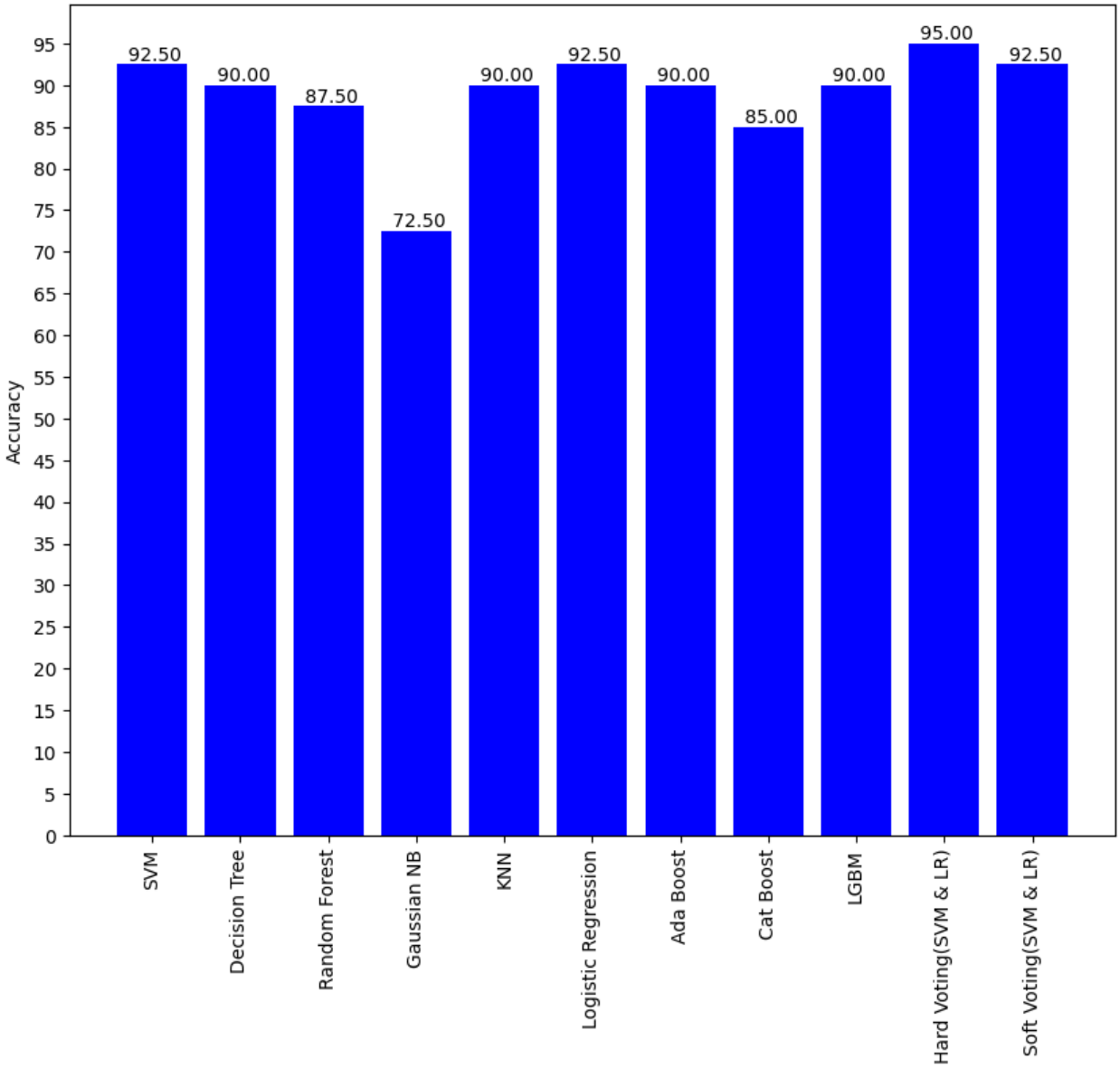


Fig. 9 Overall classification accuracy percentages (including voting).

Table 8

Comparison of the proposed model's performance with previous studies.

Reference	Accuracy	Applied feature selection	Model Explainability
Tian et al. [18]	89.1% (AUC)	No	No
G. Obaido et al. [19]	92.00%	No	Yes
V. K. Yarasuri et al. [20]	96.15%	No	No
V. M. Putri et al. [21]	82.19%	No	No
A. Alamsyah et al. [22]	93.55%	No	No
Fahad R. Albogamy et al. [23]	95.08%	No	No
Shipeng Chen et al. [24]	90.00%	No	No
Junfeng Peng et al. [25]	91.90%	No	No
Our Proposed Model	95.00%	Yes	Yes

Table 8 represents the proposed model's demonstrates clear superiority when compared to previous studies in terms of accuracy, applied feature selection, and model explainability. With an accuracy of 95.00%, it ranks among the top-performing models, closely matching Fahad R. Albogamy et al. (95.08%) and V. K. Yarasuri et al. (96.15%). Although our proposed model slightly trails Fahad R. Albogamy et al. and V. K. Yarasuri et al in term of accuracy, it is superior due to its use of feature selection and model explainability, which their models lack. These aspects make our model more efficient, reduce the risk of overfitting, and provide transparency, allowing healthcare professionals to better understand and trust the predictions. Our proposed model surpassing others like Shipeng Chen et al. (90.00%) and V. M. Putri et al. (82.19%). Many of these models, while achieving high accuracy, such as A. Alamsyah et al. (93.55%) and Junfeng Peng et al. (91.90%), do not include feature selection or model explainability. In contrast, the proposed model incorporates feature selection, optimizing its focus on the most relevant attributes and improving its overall efficiency and accuracy by reducing potential noise or irrelevant data.

Moreover, the proposed model provides model explainability, which is a key differentiator. Most other models, such as Tian et al. (89.1% AUC) and V. K. Yarasuri et al., did not offer any explainability, operating more like "black boxes." Only G. Obaido et al. (92.00%) included model explainability, but it lacked the feature selection component, which is a crucial factor in refining predictive models. By combining both feature selection and explainability, the proposed model achieves not only high predictive power but also transparency in how decisions are made—an essential attribute in healthcare contexts where understanding model outputs is critical for patient trust and decision-making. This makes the proposed model more robust, interpretable, and ultimately more practical for real-world applications compared to others.

4.4 Interpretability

Understanding how a model arrives at its predictions is crucial. SHAP was applied to the Hard Voting Classifier, as it demonstrated the highest performance with the accuracy of 95%. SHAP values were derived from the training data, offering insights into how the model mapped features to its predictions.

Fig. 10, shows how certain features influence the model's prediction, shifting it from the 'base' value (1.437) to the final predicted value (2.00) for a particular instance. The base value represents the average prediction of the model across the entire training set. The higher value of 2.00 suggests that the model predicts the patient is dead from HBV, influenced by the contribution of multiple features. Features in red, such as Varices (0.5517), Protime (0.1789), Spleen Palpable (0.6643), and Ascites (0.6783), push the prediction higher, indicating that they contribute positively to the likelihood of the outcome. Bilirubin (-0.9481) contributes negatively, but still raises the overall prediction because it's part of the features pushing the value higher. Albumin (0.9385) is another important feature that raises the prediction. On the other hand, features in blue, such as Spiders (-1.033) and Sgot (-0.7751), decrease the model’s prediction, suggesting these features lower the likelihood of the condition.

The color red indicates features that push the prediction higher, whereas blue shows features that lower the prediction. This breakdown shows how each feature contributes to the final prediction, explaining the model’s decision-making process for this specific instance.

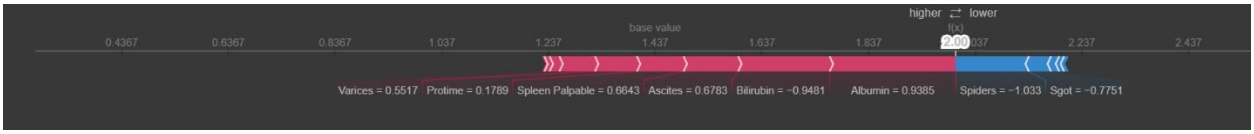


Fig. 10 SHAP predictions generated using Hard Voting.

In Fig 11, the SHAP summary plot provides a clear interpretation of how different clinical and demographic features influenced the hard-voting classifier, which combined SVM and LR, in diagnosing HBV. In this case, the SHAP plot reveals that certain features, such as "Albumin," "Bilirubin," "Sgot," and "Age," played a particularly significant role in determining the final prediction. These features had the highest SHAP values, indicating that they had a strong influence on the model's output. The plot also provides information about the interactions between features. By examining the clustering of dots on the x-axis, it is possible to see how different features interacted with each other to influence the prediction. For example, when two features have dots clustered together to the right of the plot, it suggests that they had a positive interaction, meaning that the combined effect of these features on the prediction was greater than their individual effects.

Furthermore, the SHAP plot allowed the assessment of the global importance of each feature. By observing the overall position of the dots for a particular feature on the y-axis, one could determine whether it was consistently important across different predictions. Features with dots that were consistently high or low across the plot were considered globally important.

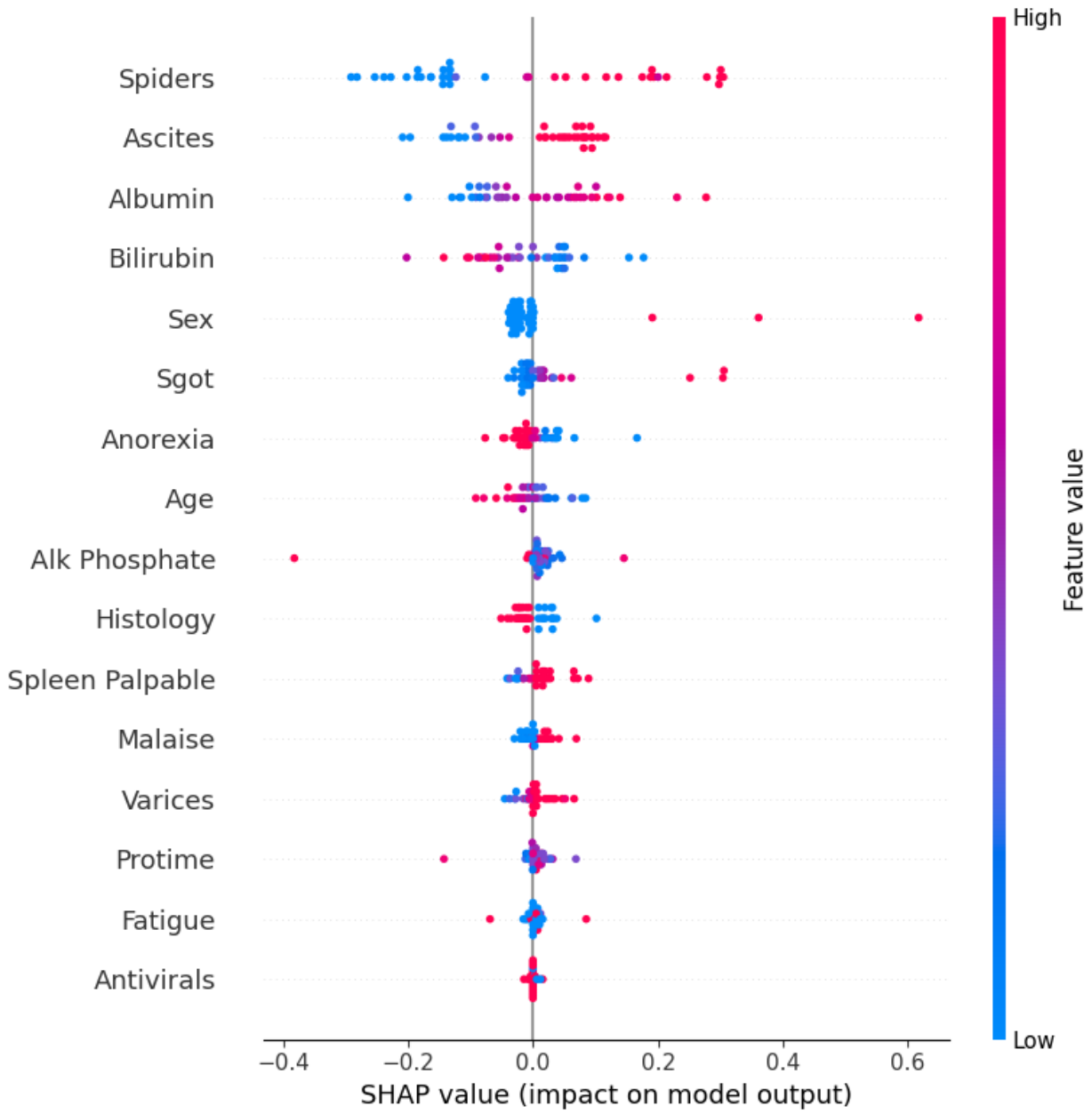


Fig. 11 Summary plot of the predictions.

The LIME explanation plot offers a detailed view of how different features influence the hard-voting classifier for a specific prediction of diagnosis for HBV. In the Fig. 12, Ascites with values between -0.84 and 0.68, has the strongest positive influence on the model's prediction. This means that higher levels of ascites significantly increase the likelihood of a positive HBV diagnosis, reflecting the clinical relevance of fluid accumulation in the abdomen. Albumin, when greater than

0.61, also contributes positively to the prediction, suggesting that elevated albumin levels play a role in supporting a positive diagnosis in this particular case.

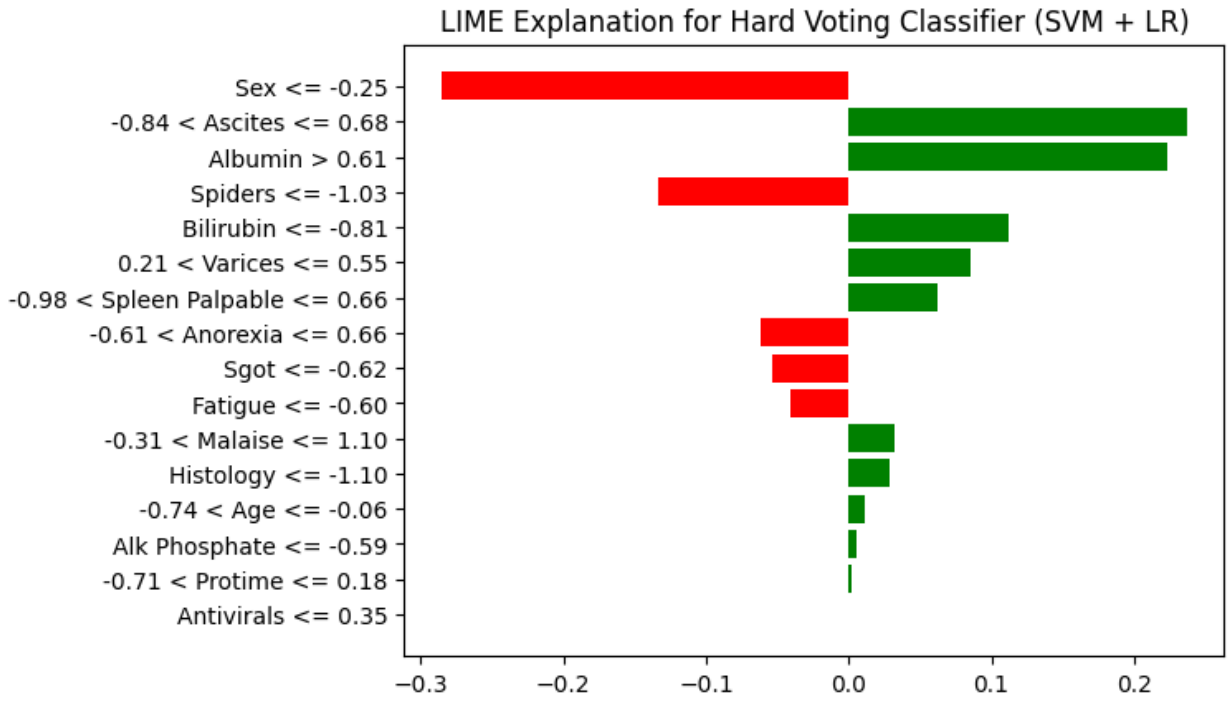


Fig. 12 Summary plot of the predictions.

In contrast, Sex with values less than or equal to -0.25, negatively impacts the prediction, indicating that for this instance, the patient's sex decreases the probability of HBV being diagnosed. Bilirubin, which is less than or equal to -0.81, surprisingly has a negative influence, where lower bilirubin levels reduce the likelihood of a positive diagnosis, even though bilirubin is generally a marker of liver dysfunction. Spiders (spider nevi), with values less than or equal to -1.03, also decrease the likelihood of an HBV diagnosis, suggesting that in this local interpretation, the presence of spider nevi steers the model toward a negative result.

On the other hand, the feature Spleen Palpable, with values less than or equal to -0.98, strongly contributes to a positive HBV diagnosis, likely due to its association with advanced liver disease. Anorexia, with a range of -0.61 to 0.66, slightly increases the likelihood of a positive diagnosis, while Malaise, ranging from -0.31 to 1.10, negatively influences the prediction, particularly at higher values. Other features such as Varices (0.21 to 0.55), Sgot (-0.62), and Fatigue (-0.60) have smaller impacts, either positively or negatively, influencing the prediction to a lesser extent.

This LIME explanation provides a transparent, instance-based understanding of how specific features and their values contribute to the classifier's decision, offering clear insights into the factors driving each individual prediction in the diagnosis of HBV.

5. Conclusion

This research presents a comprehensive analysis of ML techniques for diagnosing HBV using a diverse set of classifiers and advanced feature selection methods. The use of the Voting Classifier, which combines SVM and LR, resulted in a 95% accuracy, outperforming individual classifiers. The integration of feature selection using the Chi-squared test and the application of SMOTE to address class imbalance contributed significantly to the model's performance. The explainability of the model, enhanced by SHAP and LIME, provides crucial insights for healthcare professionals, ensuring that the predictions are interpretable and trustworthy.

The strength of this study lies in its well-rounded approach, addressing both predictive performance and model transparency. By combining SVM and LR in a Voting Classifier, this research achieved a superior accuracy compared to other studies, making it a reliable tool for HBV diagnosis. Additionally, the use of feature selection and explainability techniques allows healthcare professionals to better understand the influence of key features, such as elevated bilirubin levels, in predicting HBV-related outcomes.

One limitation of this work is the relatively small dataset (155 patients), which may impact the generalizability of the model to larger and more diverse populations. Despite the application of SMOTE for data balancing, the limited sample size could influence the robustness of the results in real-world settings. Future work could benefit from testing the model on larger datasets to validate its effectiveness across different demographics and regions.

This research stands out as the best due to its holistic approach in balancing high predictive accuracy with model interpretability, making it highly suitable for practical healthcare applications. Unlike other models that function as black boxes, the integration of SHAP and LIME in this study ensures that decision-making is transparent, which is essential in the healthcare domain where trust and understanding of predictions are critical. Furthermore, the careful selection of relevant features and the innovative combination of classifiers set this work apart, offering a robust, accurate, and explainable solution for HBV diagnosis.

References

1 Tan M.; Bhadoria A.; Estimating the proportion of people with chronic hepatitis B virus infection eligible for hepatitis B antiviral treatment worldwide: a systematic review and meta-analysis, 2021 [CrossRef]

2 Breakwell L.; Marke D.; Assessing the impact of the routine childhood hepatitis B immunization program and the need for hepatitis B vaccine birth dose in Sierra Leone, 2018, April 2022 [CrossRef]

3 World Health Organization (WHO) | Hepatitis B [CrossRef]

4 Core Concepts - Hepatitis B Coinfection - Co-Occurring Conditions - National HIV Curriculum (uw.edu) [CrossRef]

5 Tai, W.; He, L.; Zhang, X.; Pu, J.; Voronin, D.; Jiang, S.; Zhou, Y.; Du, L. Characterization of the receptor-binding domain (RBD) of 2019 novel coronavirus: Implication for development of RBD protein as a viral attachment inhibitor and vaccine. Cell. Mol. Immunol. 2020, 17, 613–620. [CrossRef]

6 Strother, H.W.; David, B.D. Estimation of the probability of an event as a function of several independent variables. Biometrika 1967, 54, 167–179. [CrossRef]

7 Uttreshwar, G.S.; Ghatol, A. Hepatitis B Diagnosis Using Logical Inference And Generalized Regression Neural Networks. In Proceedings of the 2009 IEEE International Advance Computing Conference, Patiala, India, 6–7 March 2009; pp. 1587–1595. [CrossRef]

8 Wang, H.; Liu, Y.; Huang, W. Random forest and Bayesian prediction for Hepatitis B virus reactivation. In Proceedings of the 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Guilin, China, 29–31 July 2017; pp. 2060–2064. [CrossRef]

9 Agbele, K.K.; Oriogun, P.K.; Seluwa, A.G.; Aruleba, K.D. Towards a model for enhancing ICT4 development and information security in healthcare system. In Proceedings of the 2015 IEEE International Symposium on Technology and Society (ISTAS), Dublin, Ireland, 11–12 November 2015; pp. 1–6. [CrossRef]

10 Ho, T.K. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 14–16 August 1995; Volume 1, pp. 278–282. [CrossRef]

11 Aruleba, K.; Obaido, G.; Ogbuokiri, B.; Fadaka, A.O.; Klein, A.; Adekiya, T.A.; Aruleba, R.T. Applications of Computational Methods in Biomedical Breast Cancer Imaging Diagnostics: A Review. J. Imaging 2020, 6, 105. [CrossRef]

12 Aruleba, R.T.; Adekiya, T.A.; Ayawei, N.; Obaido, G.; Aruleba, K.; Mienye, I.; Aruleba, I.; Ogbuokiri, B. COVID-19 Diagnosis: A Review of Rapid Antigen, RT-PCR and Artificial Intelligence Methods. Bioengineering 2022, 3, 153. [CrossRef]

13 Mienye, I.D.; Obaido, G.; Aruleba, K.; Dada, O.A. Enhanced Prediction of Chronic Kidney Disease Using Feature Selection and Boosted Classifiers. In International Conference on Intelligent Systems Design and Applications; Springer: Cham, Switzerland, 2022; pp. 527–537. [CrossRef]

14 Xiaolu, T.; Yutian, C.; Yutao, H.; Pi, G.; Mengjie, L.; Wangjian, Z.; Zhicheng, D.; Xiangyong, L.; Yuantao, H. Using Machine Learning Algorithms to Predict Hepatitis B Surface Antigen Seroclearance. Comput. Math. Methods Med. 1967, 2019, 2019. [CrossRef]

15 Akbar, K.W.; Peter, D.R.H. Machine learning in medicine: A primer for physicians. Am. J. Gastroenterol. 2010, 105, 1224–1226. [CrossRef]

16 Rohan, G.; Devesh, S.; Mehar, S.; Swati, T.; Rashmi, K.A.; Pravir, K. Artificial intelligence to deep learning: Machine intelligence approach for drug discovery. Mol. Divers 2021, 25, 1315–1360. [CrossRef]

17 Qing-Hai, Y.; Lun-Xiu, Q.; Marshonna, F.; Ping, H.; Jin, W.K.; Amy, C.P.; Richard, S.; Yan, L.; Ana, I.R.; Yidong, C.; et al. Predicting hepatitis B virus–positive metastatic hepatocellular carcinomas using gene expression profiling and supervised machine learning. Nat. Med. 2003, 9, 416–423. [CrossRef]

18 Tian, X.; Chong, Y.; Huang, Y.; Guo, P.; Li, M.; Zhang, W.; Du, Z.; Li, X.; Hao, Y. Using machine learning algorithms to predict hepatitis B surface antigen seroclearance. Comput. Math. Methods Med. 2019, 2019, 6915850. [CrossRef]

19 Obaido, G.; Ogbuokiri, B.; Swart, T.G.; Ayawei, N.; Kasongo, S.M.; Aruleba, K.; Mienye, I.D.; Aruleba, I.; Chukwu, W.; Osaye, F.; et al. An Interpretable Machine Learning Approach for Hepatitis B Diagnosis. *Appl. Sci.* **2022**, *12*, 11127. [CrossRef]

20 V. K. Yarasuri, G. K. Indukuri and A. K. Nair, "Prediction of Hepatitis Disease Using Machine Learning Technique," *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2019, pp. 265-269. [CrossRef]

21 Putri, V.; Masjkur, M.; Suhaeni, C. Performance of SMOTEinarandomforestandnaiveBayesclassifierforimbalancedHepatitis-B vaccination status. J. Physics: Conf. Ser. 2021, 1863, 012073. [CrossRef]

22 Alamsyah, A. Increased accuracy of prediction hepatitis disease using the application of principal component analysis on a support vector machine, 2021 J. Phys.: Conf. Ser. 1968 012016 [CrossRef]

23 Albogamy F. R, Asghar J. , Subhan F. Decision Support System for Predicting Survivability of Hepatitis Patients, 14 April 2022 [CrossRef]

24 Chen S., Using Quasispecies Patterns of Hepatitis B Virus to Predict Hepatocellular Carcinoma With Deep Sequencing and Machine Learning, 2021 Jun 4 [CrossRef]

25 Peng J., An Explainable Artificial Intelligence Framework for the Deterioration Risk Prediction of Hepatitis Patients, 13 April 2021 [CrossRef]

26 University of California Irvine (UCI) Machine Learning Repository | Hepatitis B dataset [CrossRef]

27 Chawla N. V, SMOTE: Synthetic Minority Over-sampling Technique, June 2002 [CrossRef]

28 Ali P. J. M., Data Normalization and Standardization: A Technical Report, January 2014 [CrossRef]

29 Qin, J., Chen, L., Liu, Y., Liu, C., Feng, C., & Chen, B. (2020). A machine learning methodology for diagnosing chronic kidney disease. IEEE access : practical innovations, open solutions, 8, 20991–21002. [CrossRef]

30 Zhao, L., Lee, S., & Jeong, S.-P. (2021). Decision tree application to classification problems with boosting algorithm. Electronics, 10, 1903. [CrossRef]

31 Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. Classification and Regression Trees; Routledge: London, UK, 2017. [CrossRef]

32 Quinlan, J.R. Induction of decision trees. *Mach Learn* **1**, 81–106 (1986). [CrossRef]

33 Quinlan, J.R. C4. 5: Programs for Machine Learning; Elsevier: Amsterdam, The Netherlands, 2014. [CrossRef]

34 Wieczorek, W.; Kozak, J.; Strąk, Ł.; Nowakowski, A. Minimum query set for decision tree construction. Entropy 2021, 23, 1682. [CrossRef]

35 Moshkov, M. On the depth of decision trees with hypotheses. Entropy 2022, 24, 116. [CrossRef]

36 Azad,M.; Chikalov, I.; Hussain, S.; Moshkov, M. Entropy-based greedy algorithm for decision trees using hypotheses. Entropy 2021, 23, 808. [CrossRef]

37 Ma, E.J.; Kummer, A. Principled Decision-Making Workflow with Hierarchical Bayesian Models of High-Throughput Dose Response Measurements. Entropy 2021, 23, 727. [CrossRef]

38 Mienye, I.D.; Sun, Y.; Wang, Z. An improved ensemble learning approach for the prediction of heart disease risk. Inform. Med. Unlocked 2020, 20, 100402. [CrossRef]

39 Schonlau, M.; Zou, R.Y. The random forest algorithm for statistical learning. Stata J. 2020, 20, 3–29. [CrossRef]

40 Lin, W.; Wu, Z.; Lin, L.; Wen, A.; Li, J. An ensemble random forest algorithm for insurance big data analysis. IEEE Access 2017, 5, 16568–16575. [CrossRef]

41 S. H. Walker and D. B. Duncan, “Estimation of the probability of an event as a function of several independent variables,” Biometrika, vol. 54, no. 1-2, pp. 167–179, 1967. [CrossRef]

42 Wu, Y.; Zhang, Q.; Hu, Y.; Sun-Woo, K.; Zhang, X.; Zhu, H.; Li, S. Novel binary logistic regression model based on feature transformation of XGBoost for type 2 Diabetes Mellitus prediction in healthcare systems. Future Gener. Comput. Syst. 2022, 129, 1–12. [CrossRef]

43 Ogbuokiri, B.; Ahmadi, A.; Nia, Z.M.; Mellado, B.; Wu, J.; Orbinski, J.; Ali, A.; Jude, K. Vaccine Hesitancy Hotspots in Africa: An Insight from Geotagged Twitter Posts. TechRxiv 2022. [CrossRef]

44 Mienye, I.D.; Obaido, G.; Aruleba, K.; Dada, O.A. Enhanced Prediction of Chronic Kidney Disease Using Feature Selection and Boosted Classifiers. In International Conference on Intelligent Systems Design and Applications; Springer: Cham, Switzerland, 2022; pp. 527–537. [CrossRef]

45 G.S. Tomar, S. Verma & Ashish Jha; "Web Page Classification using Modified naïve Baysian Approach", IEEE TENCON-2006, pp 1-4, 14-17 Nov 2006. [CrossRef]

46 Linero AR, Antonelli JL. The how and why of Bayesian nonparametric causal inference. Wiley Interdisciplinary Reviews: Comput Stat 2023;15(1):e1583. [CrossRef]

47 Bernhard Schlkopf and Alexander J. Smola (2001) Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning), MIT Press. [CrossRef]

48 Huang, M.W.; Chen, C.W.; Lin, W.C.; Ke, S.W.; Tsai, C.F. SVM and SVM ensembles in breast cancer prediction. PLoS ONE 2017, 12, e0161501. [CrossRef]

49 Ghosh, M.; Sanyal, G. An ensemble approach to stabilize the features for multi-domain sentiment analysis using supervised machine learning. J. Big Data 2018, 5, 1–25. [CrossRef]

50 M. Sameer and B. Gupta, "CNN based framework for detection of epileptic seizures," Multimed. Tools Appl., 2022, doi: 10.1007/s11042-022-12702-9. [CrossRef]

51 Caruana, R.; Niculescu-Mizil, A. An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 161-168. [CrossRef]

52 Sevinç, E. An empowered AdaBoost algorithm implementation: A COVID-19 dataset study. Comput. Ind. Eng. 2022, 165, 107912. [CrossRef]

53 Lee, S.J.; Tseng, C.H.; Yang, H.Y.; Jin, X.; Jiang, Q.; Pu, B.; Hu, W.H.; Liu, D.R.; Huang, Y.; Zhao, N. Random RotBoost: An Ensemble Classification Method Based on Rotation Forest and AdaBoost in Random Subsets and Its Application to Clinical Decision Support. Entropy 2022, 24, 617. [CrossRef]

54 Zheng, H.; Xiao, F.; Sun, S.; Qin, Y. Brillouin Frequency Shift Extraction Based on AdaBoost Algorithm. Sensors 2022, 22, 3354. [CrossRef]

55 Huang, X.; Li, Z.; Jin, Y.; Zhang, W. Fair-AdaBoost: Extending AdaBoost method to achieve fair classification. Expert Syst. Appl. 2022, 202, 117240. [CrossRef]

56 Ding, Y.; Zhu, H.; Chen, R.; Li, R. An Efficient AdaBoost Algorithm with the Multiple Thresholds Classification. Appl. Sci. 2022, 12, 5872. [CrossRef]

57 A Mahabub, MI Mahmud, MF Hossain, A robust system for message filtering using an ensemble machine learning supervised approach, ICIC Expr. Lett. Part B Appl. 10 (2019) 805–811. [CrossRef]

58 Mangalathu, S.; Hwang, S.H.; Jeon, J.S. Failure mode and effects analysis of RC members based on machine-learning-based SHapley Additive exPlanations (SHAP) approach. Eng. Struct. 2020, 219, 110927. [CrossRef]

59 García, M.V.; Aznarte, J.L. Shapley additive explanations for NO2 forecasting. Ecol. Inform. 2020, 56, 101039. [CrossRef]

60 Santos, R.N.; Yamouni, S.; Albiero, B.; Vicente, R.; A Silva, J.; FB Souza, T.; CM Freitas Souza, M.; Lei, Z. Gradient boosting and Shapley additive explanations for fraud detection in electricity distribution grids. Int. Trans. Electr. Energy Syst. 2021, 31, e13046. [CrossRef]

61 Meddage, P.; Ekanayake, I.; Perera, U.S.; Azamathulla, H.M.; Md Said, M.A.; Rathnayake, U. Interpretation of Machine-LearningBased (Black-box) Wind Pressure Predictions for Low-Rise Gable-Roofed Buildings Using Shapley Additive Explanations (SHAP). Buildings 2022, 12, 734. [CrossRef]

62 Rao S., Mehta S., A Study of LIME and SHAP Model Explainers for Autonomous Disease Predictions, 2022 [CrossRef]