Technological Advancements in Artificial Intelligence and Blockchain for Smart and Secure IoT Applications(TAABSI)

Enhancing Hepatitis C Disease Detection: A Study Using SMOTE, Optuna, and SHAP

S. M. Mehzabeen[1], Dr. R. Gayathri[2], Nirmalkumar. T. K[3], Pratosh Karthikeyan[3], Pranav. A[3], Pranav Manikandan Sundaresan[3].

[1]Assistant Professor at Department of Electronics and Communication Engineering
Sri Venkateswara College of Engineering.
[2] Professor at Department of Electronics and Communication Engineering
Sri Venkateswara College of Engineering.
[3]Undergraduate Student at Department of Electronics and Communication Engineering
Sri Venkateswara College of Engineering.

***Abstract*:** Hepatitis C virus (HCV) detection is a critical aspect of early intervention and effective management of the disease. This research paper presents a comprehensive study focused on enhancing the detection accuracy of HCV through the integration of advanced techniques - SMOTE, Optuna, and SHAP - alongside extensive exploratory data analysis (EDA). The study addresses class imbalance using Synthetic Minority Over-sampling Technique (SMOTE), optimizes model performance with Optuna for hyperparameter tuning, and provides model interpretability using SHAP (SHapley Additive exPlanations). EDA is leveraged to gain valuable insights into the dataset's characteristics, ensuring robust data preprocessing and feature engineering. The results demonstrate improved HCV detection performance, highlighting the efficacy of the proposed methodology in medical diagnostics and aiding healthcare professionals in making informed clinical decisions.

Keywords: *Hepatitis C virus, Synthetic Minority Over-sampling Technique, exploratory data analysis, SHapley Additive exPlanations, machine learning, classification algorithms, OPTUNA.*

INTRODUCTION

Hepatitis C is a RNA virus that predominantly affects the liver and is brought on by the hepatitis C virus (HCV). It is one of the main viral hepatitis strains and is regarded as a global public health issue. The virus is primarily spread through coming into contact with the blood of an infected person, most frequently when sharing needles when using drugs, getting contaminated medical care, or from infected mothers to their newborns while giving

birth. It has been estimated that over 150 million people worldwide have been affected chronically by this disease. The majority of HCV infections develop into chronic conditions, in which the virus stays in the body for a longer period of time and frequently causes permanent liver damage. Hepatitis C chronic infection can advance covertly for years without showing any signs. For detecting Hepatitis C, the first screening test would be ‘HCV antibody test’ which is a blood test which tests the levels of antibodies present in the blood relating to the Hepatitis C virus.

In recent years, the use of artificial intelligence and machine learning in the medical field has shown tremendous potential, notably in the areas of disease diagnosis and prognostic modeling. These cutting-edge methods have the potential to have a big impact on healthcare by offering precise, quick, and affordable diagnostic options. In order to improve the identification of hepatitis C disease, this study aims to harness the potential of cutting-edge approaches such as Synthetic Minority Over-sampling Technique (SMOTE), Optuna, and SHAP (SHapley Additive exPlanations).

SMOTE is a mathematical technique which is used for increasing the number of cases of a particular attribute in the dataset. This imbalance can affect the machine learning model, leading to a lower accuracy and hindering the performance of the model. With using SMOTE we can ensure that there is a balanced dataset that is being used and hence improving the performance of the model.

Building powerful machine learning models requires careful consideration of the hyperparameter tuning process. The choice of the best hyperparameters has a significant impact on the model's performance, yet manually investigating every combination is time- and resource-intensive. Enter Optuna, an automated framework for hyperparameter optimisation that uses cutting-edge algorithms to quickly find the optimal hyperparameter configuration. We can optimize the hepatitis C detection model's accuracy and resilience by using Optuna in our research.

It's important to interpret machine learning models, especially in the medical industry where trust-building and clinical decision-making depend on openness. By assigning feature priority, the advanced model interpretability technique SHAP offers illuminating justifications for certain predictions. Healthcare practitioners can better grasp the elements influencing the detection of hepatitis C disease by analyzing the contribution of each feature to the model's output. This allows them to obtain useful insights into the diagnostic process.

This study aims to enhance hepatitis C disease detection by employing SMOTE, Optuna, and SHAP techniques in the machine learning process.Section 1 reviews relevant literature on hepatitis C virus detection using machine learning models. Section 2 presents exploratory data analysis of the dataset, while Section 3 details the

machine learning flow with the mentioned methods. In Section 4, we present and discuss the results, concluding the findings and potential implications for medical diagnostics.

Literature Survey

In 2023, Ahmed M. Elshewey *et al.* [14] had proposed the hyOPTGB Model for hepatitis C prediction. The hyOPTGB is an hyperparameter optimized Gradient Boosting Model in which 8 specific hyperparameters of Gradient Boosting are optimized. The dataset is preprocessed using Min- Max normalization followed by feature selection using Forward selection wrapped method. For the same dataset different machine learning models such as SVM, DT, DC, BC and RC were evaluated based on their accuracy, F-1 score, recall and precision and their performance is compared with hyOPTGB model, where the proposed hyOPTGB model outperforms with 95.3% accuracy.

In 2023, Ali Mohd Ali *et al.* [15] implemented various machine learning models for comprehensive evaluation of their effect in predicting hepatitis C. The proposed framework consists of Sequential Forward Selection(SFS) to distinguish most relevant features from others. The impact of the synthetic minority oversampling technique (SMOTE) on the accuracy is investigated and conclusions were made that the SMOTE didn't significantly affect the accuracy of the models. An average of 83% accuracy was obtained when machine learning models such as LR, KNN, DT, NN and RF were used on the dataset. Shapley Additive Explanations (SHAP) method is used to interpret the predictions of the machine learning models.

In 2022, Michael Onyema Edeh *et al.* [17] built an AI based ensemble model after examining various machine learning models for predicting hepatitis. The proposed method had the capacity to forecast progressive fibrosis using clinical data and blood biomarkers.Individual models have been found to be capable of providing accuracy of up to 94.67%.The ensemble model, which includes a Bayesian network, MLP, and QUEST decision trees, was then created. The Ensemble node combines three model nuggets to generate more accurate predictions than any of the separate models (MLP, Bayesian Network, and QUEST). Limitations in the MLP, Bayesian Network, and QUEST models were removed by combining predictions from several models, resulting in a higher overall accuracy. MLP, Bayesian Network, and QUEST models combined in this manner frequently perform at least as well as, if not better than, the best MLP, Bayesian Network, and QUEST models. The accuracy obtained was 94.10%, 94.47%, 94.63%, 95.59% for MLP, Bayesian, QUEST and Proposed Ensemble Model respectively.

In 2020,Satish CR Nandipati *et al.* [16] used an Egyptian patient's dataset to conduct a study and compare the performance between multi and binary class labels of the same dataset. In data preprocessing data normalization is done to replace the missing values of the dataset with mode values based on age attribute. Python and R are the two

machine learning tools used to rescale the variables. A 1385-instance dataset with 29 attributes was used to test the classification model. Data analysis was done using R-based CARET and Python-based Scikit learn, and seven machine learning techniques and feature selection algorithms were applied. Adaboost and Bagging served as ensemble classifiers among the five models that were used. We used libraries and parameters to address feature selection methodologies. The effectiveness of the classifier was assessed using 10-fold cross-validation, accuracy, precision, recall, and overall average scores for datasets with multiple and binary labels. This investigation compared the effectiveness of tools and classifiers on the HCV dataset's multi- and binary-class labels. A random forest analysis of the multiclass dataset revealed that KNN (26.44%) and the multiclass dataset (28.36%) had the highest accuracy. The accuracy of NN's binary class labels was the greatest (53.12%), although KNN's performance in recall and precision was better. SVM, RF, KNN, and NB (51.31%) were the next most accurate models after the R multiclass dataset. KNN (53.66%) and boosting (54.23%) both demonstrated great accuracy. Precision and recall displayed varied results, despite the accuracy of multiclass and binary class labels performing similarly in both cases.

In 2021, Tsvetkov *et al.* [18] To detect the stage of liver fibrosis in patients, a machine learning model was proposed. The researchers examined 1240 chronic viral Hepatitis C patient records and created machine learning models utilizing data from 689 patients categorized by stage of liver fibrosis. In comparison to the "gold standard" of diagnosis (liver biopsy), the established approach for diagnosing the 3-4 stages of liver fibrosis in patients with chronic viral hepatitis C was 80.56% (95% CI: 69.53-88.94%), sensitivity — 66.67%, and specificity — 94.44%.

Exploratory Data Analysis

The dataset utilized in this study consists of laboratory findings from blood donors and Hepatitis C patients, alongside demographic data, notably age. The data was sourced from the esteemed University of California, Irvine Machine Learning Repository, accessible through the following link: 'https://archive.ics.uci.edu/ml/datasets/HCV.$+$data'.

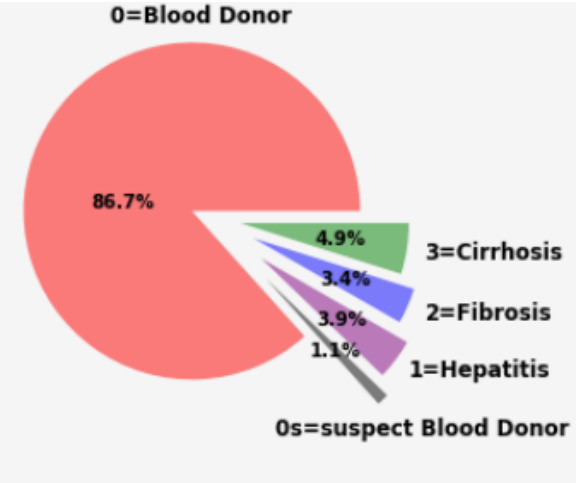

Fig.1. Initial look at the data set

From Fig.1 it is found that the distribution of liver fibrosis stages in patients with chronic viral hepatitis C. The majority of patients (66.7%) have stage 3 fibrosis, followed by stage 4 (22.2%). A small percentage of patients have stage 1 (11.1%) or stage 2 (0%) fibrosis.

There are a total of 31 missing values in this dataset. And also from our analysis it is found that ALP and CHOL contribute the most to the missing values in the dataset.

2.1 Univariate Data Analysis

A. Age

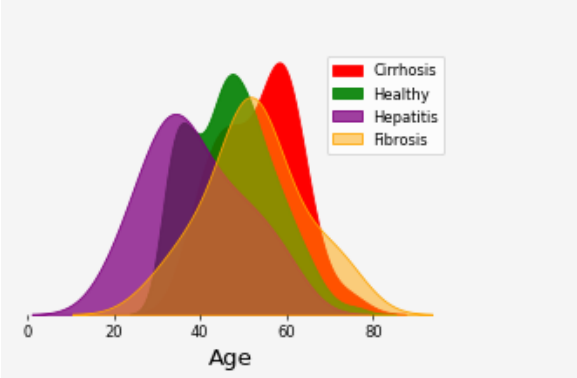

Fig.1.1. Hepatitis Rates Among Young People

Fig. 1.1 illustrates the distribution of hepatitis rates among the young population by age and liver status. *The overall age distribution demonstrates that the highest prevalence of hepatitis is in the population aged 20-30 years old.* Hepatitis is inflammation of the liver. Inflammation is swelling that occurs when
tissues of the body are injured or infected. The age distribution by liver status
indicates that the bulk of people with hepatitis are healthy, but there is a larger population with fibrosis and cirrhosis. This is because hepatitis can damage the liver, and if the damage is severe enough, it can lead to fibrosis and cirrhosis. Fibrosis is a criterion where scar tissue builds up in the liver, and cirrhosis is a more sophisticated phase of liver damage where the liver becomes scarred and incapable to function properly.

B. Albumin Level

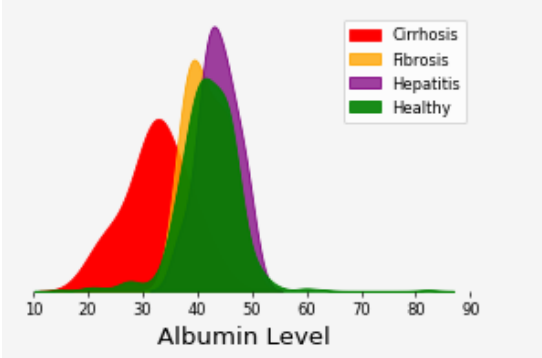

Fig.1.2. Albumin Level Indicators

Fig. 1.2 illustrates the distribution of albumin levels in populations with different liver status. The albumin level is a quantifier of the number of albumin in the blood. Albumin is a protein produced by the liver.. The chart indicates that people with a healthy liver have higher rates of albumin than the population with liver damage. This is because the liver is responsible for generating albumin. When the liver is damaged, it can no longer generate as much albumin, which can lead to low albumin levels. The graph also indicates that the intensity of liver damage is correlated to the level of albumin deficiency. *People with fibrosis have lower levels of albumin than people with healthy liver. People with cirrhosis have the lowest albumin levels of all.*

The above figure indicates that the average albumin level for population with healthy livers is 40 g/dL. The graph also illustrates that the average albumin level for people with fibrosis is 30 g/dL. The chart indicates that the average albumin level for the population with cirrhosis is 20 g/dL.

C. Alkaline Phosphatase Level

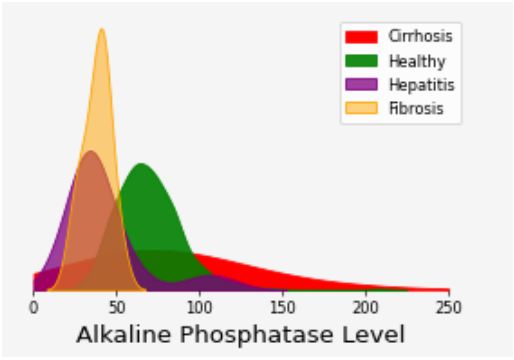

Fig.1.3 Alkaline Phosphatase Level Indicator

Fig. 1.3 illustrates the distribution of alkaline phosphatase (ALP) levels in populations with different liver status. ALP is an enzyme fabricated by the liver. It helps break down fats and proteins. When the liver is damaged, it can no longer generate as much ALP, which can lead to low ALP levels. The graph also indicates that the severity of liver damage is correlated to the level of ALP deficiency. *People with fibrosis have lower ALP levels than people with healthy livers. People with liver cirrhosis have the lowest ALP values of all*

D. Alanine Transaminase Level

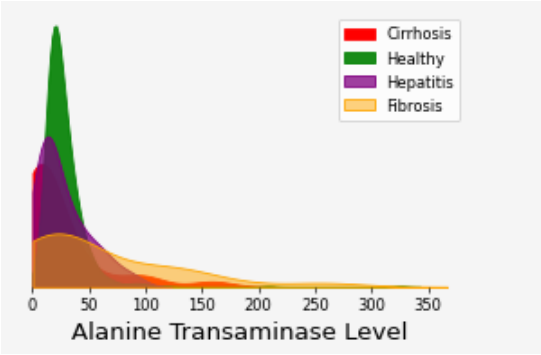

Fig.1.4. Alanine Transaminase Level Indicator

Fig.1.4. illustrates the distribution of alkaline transaminase (ALT) levels by liver status. ALT is an enzyme generated in the liver. It is an enzyme that helps the liver convert food into energy. When the liver is damaged, the level of ALT in the blood can rise. The above figure illustrates three different liver conditions: healthy, hepatitis, and cirrhosis. The healthy range for ALT levels is between 0 and 50 IU/L. ALT levels above 50 IU/L can signify a liver problem. *The overall ALT distribution is higher in people with healthy livers, while the distribution by liver status is lower in people with hepatitis, fibrosis, and cirrhosis.*

E. Aspartate Aminotransferase Level

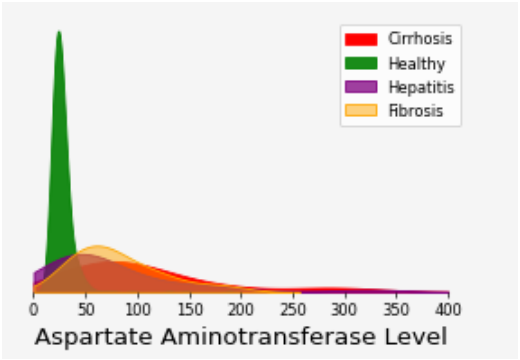

Fig.1.5. Aspartate Aminotransferase Level Indicator

Fig1.5. illustrates the distribution of aspartate aminotransferase (AST) levels by liver status. AST is an enzyme produced in the liver and released into the bloodstream when the liver is damaged. The graph illustrates three different liver conditions: healthy, hepatitis and cirrhosis. The healthy range for AST values is between 0 and 40 IU/L. AST levels above 40 IU/L can signify a liver problem. In people with hepatitis, most AST values are between 40 and 300 IU/L. *In people with liver cirrhosis, most AST values are above 300 IU/L.*

F. Bilirubin Level

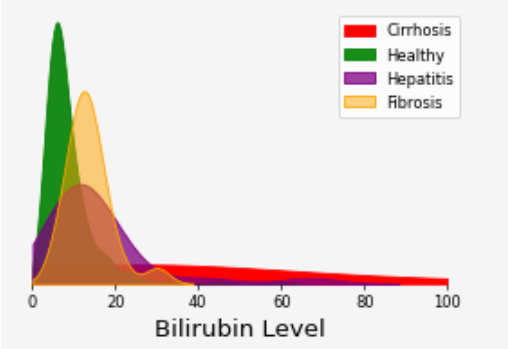

Fig.1.6. Bilirubin Level Indicator

Fig.1.6. illustrates the distribution of bilirubin levels by liver status. A consequence of the destruction of red blood cells is bilirubin. The level of bilirubin in the blood rises when the liver is not functioning properly because it cannot remove bilirubin from the blood. The chart indicates that people with healthy liver have a bilirubin range of 0-1.5 mg/dL. As the liver disease progresses, the bilirubin level increases. People with hepatitis have a bilirubin range of 1.6-3 mg/dL, people with fibrosis have a bilirubin range of 3.1-5 mg/dL, and people with cirrhosis have a bilirubin range of >5 mg/dL.

G. Cholinesterase Level

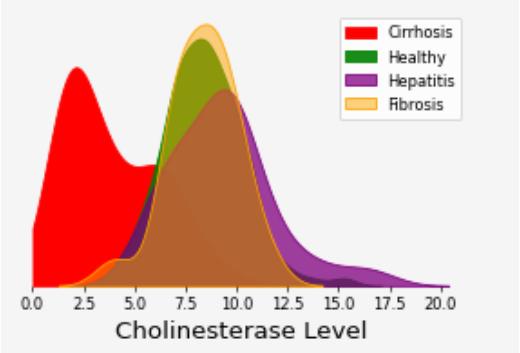

Fig.1.7. Cholinesterase Level Indicator

Fig1.7. shows the distribution of cholinesterase (CHE) levels in people with different liver statuses. CHE is an enzyme that is fabricated by the liver. It helps to break down fats and proteins. The bars are colored to signify the different liver conditions: healthy (green), fibrosis (yellow), and cirrhosis (red). The chart indicates that the average CHE level for people with healthy livers is 175 IU/L. The graph also indicates that the average CHE level for people with fibrosis is 125 IU/L. The graph indicates that the average CHE level for people with cirrhosis is 75 IU/L.

The graph also indicates that the distribution of CHE levels is bimodal, meaning that there are two distinct peaks in the distribution. This is possible due to the fact that there are two different causes for CHE elevation: liver damage and drug-induced liver injury. *The peak at the lower CHE levels is owing to liver damage, while the peak at the higher CHE levels is owing to drug-induced liver injury.*

H. Cholesterol Level

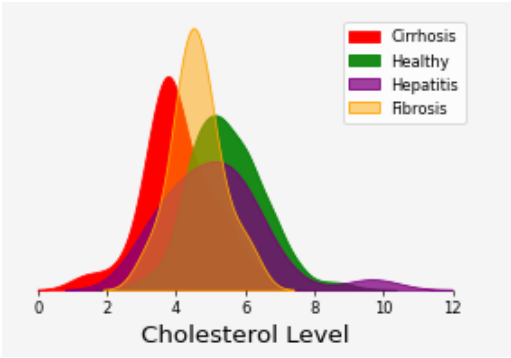

Fig.1.8. Cholesterol Level Indicator

Fig.1.8. shows the distribution of cholesterol levels in people with different liver status. Cholesterol is a type of fat that is found in all cells of the body. It is important for the production of hormones, vitamin D, and other substances. Bars are colored to indicate different liver conditions: healthy (green), fibrosis (yellow), cirrhosis (red), and hepatitis (blue). The graph shows that the average cholesterol level for people with a healthy liver is 200 mg/dl. The average cholesterol level in people with fibrosis is 250 mg/dl. *The average cholesterol level in people with liver cirrhosis is 300 mg/dl and the average cholesterol level in people with hepatitis is 225 mg/dl.*

I. Creatinine Level

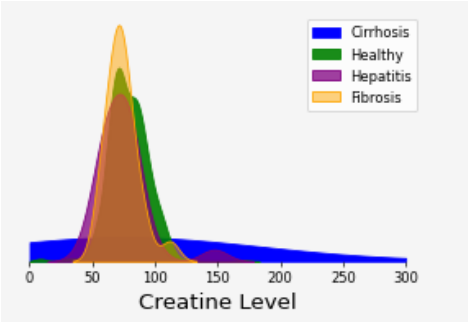

Fig.1.9. Creatine Level Indicator

Fig.1.9. shows the distribution of creatine levels in people with different liver statuses. The graph adopts a bar chart to show the distribution of creatine levels by liver status. The bars are colored to signify the different liver statuses: healthy (green), fibrosis (yellow), cirrhosis (red), and hepatitis (blue). The graph indicates that the average creatine level for people with healthy livers is 100 mg/dL. The graph also demonstrates that the creatine levels are distributed in a bimodal manner, with two separate peaks. This is probably because nutrition and genetic factors, which are two separate sources of creatine elevation, are involved. *Dietary variables contribute to the peak at the lower creatinine levels, whereas genetic factors contribute to the peak at the higher creatinine levels.*

J. Gamma-Glutamyl Transpeptidase Level

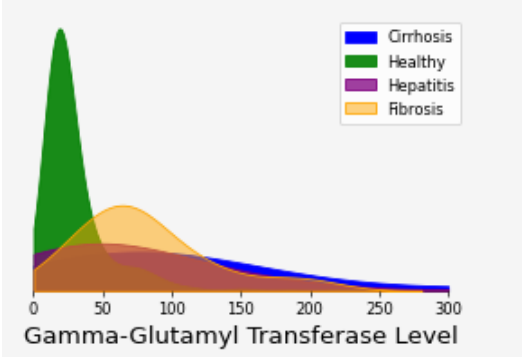

Fig.1.10. Gamma-Glutamyl Transpeptidase Level Indicator

Fig.1.10. illustrates distribution of gamma-glutamyl transpeptidase (GGT) levels in populations with different liver statuses. GGT is an enzyme that is generated in the liver and other organs. It is released into the bloodstream when the liver is damaged. The chart indicates that the overall distribution of GGT levels is bell-shaped, with the bulk of people having GGT levels between 100 and 200 units per liter. *However, people with liver problems tend to have higher GGT levels*. For example, people with hepatitis have an average GGT level of 250 units per liter, while people with cirrhosis have an average GGT level of 300 units per liter.

K. Protein Levels

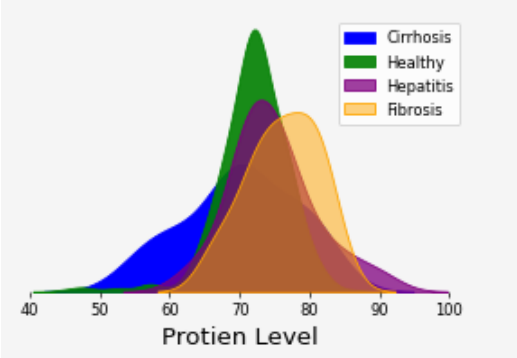

Fig. 1.11. Protein Level Indicator

Fig 1.11 shows a diagram of protein distribution by liver status. *The data shows that there is no clear correlation between protein levels and liver status.* The overall protein distribution is relatively uniform, with a slight increase in protein levels in people with hepatitis and cirrhosis. However, there is a wide range of protein levels in people with all liver statuses, and there are many people with healthy livers who have low protein levels.

2.2 Bivariate Analysis

While there was a comparison between each attribute in the dataset directly with the disease and the relation it presented, in this bivariate analysis we are going to compare 2 attributes against each other.

Bivariate analysis can be a useful way to identify relationships between variables that might not be immediately obvious

In this analysis we'll look at how "ATL vs. AST" and "ALP vs. CHE" together has impact on the dataset.

A. ATL vs. AST

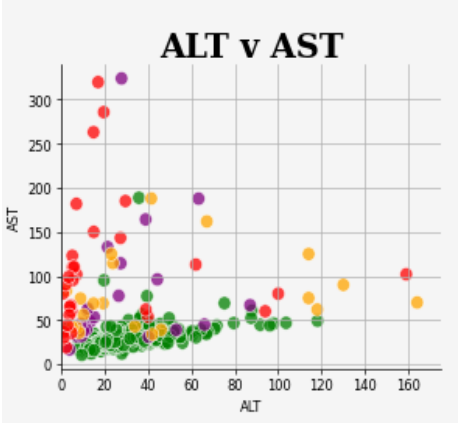

Fig. 2.1 ALT vs. AST Comparison

Fig. 2.1. shows that people with healthy livers have relatively low levels of ALT and AST. However, people with hepatitis C have much higher levels of ALT and AST. This is because the hepatitis C virus damages liver cells, which releases ALT and AST into the bloodstream. But, predominantly a very high AST is seen to be common among people who have been affected by hepatitis C virus.

B. ALP vs. CHE

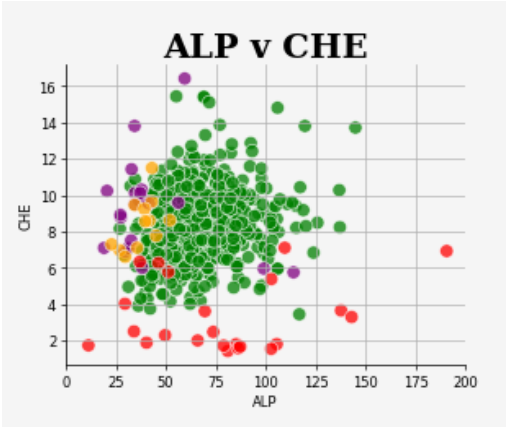

Fig. 2.2. ALP vs. AST comparison

Fig. 2.2 depicts the comparison between the alkaline phosphatase levels and the cholinesterase levels. It can be seen that low ALP levels and high CHE levels in a

patient's test result will lead to hepatitis C virus. Whereas on the other hand, healthy people seem to have a medium range of both substances.

METHODOLOGY

3.1 SMOTE

Synthetic Minority Over-sampling Technique is abbreviated as SMOTE. It is a machine learning data augmentation strategy for imbalanced datasets. When dealing with imbalanced datasets in which one class is severely underrepresented compared to others, SMOTE aids in class distribution balance by generating synthetic samples for the minority class.

Due to a lack of sufficient samples, the machine learning model may not be able to fully learn from the minority class if SMOTE is not used on an imbalanced dataset. As the model may be skewed towards the dominant class, this could result in biased and erroneous predictions. It selects a sample from the minority class, finds its k nearest neighbors (usually using Euclidean distance), and then creates new synthetic samples by randomly selecting a neighbor and adding a fraction of the difference between the two samples to the original sample. This process helps expand the minority class, making it more balanced with the majority class.

*New Sample = Sample_i + random_fraction * (Sample_i - Sample_j)*

where:

1. Sample_i is the original sample from the minority class.
2. Sample_j is one of its k nearest neighbors.
3. Random_fraction ranges from 0 to 1 and it is random.

For our dataset we use SMOTE to balance the various classes namely Healthy, Hepatitis, Fibrosis, Cirrhosis

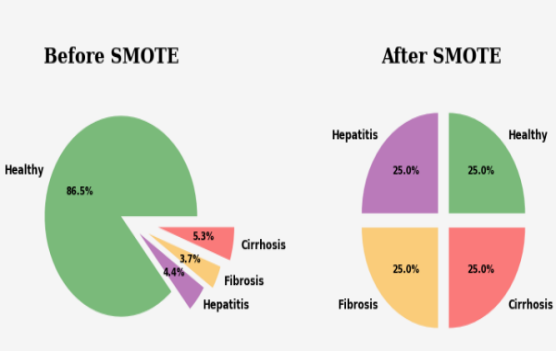

Fig.3.1.Data distribution with and without SMOTE

The Fig.3.1. pie charts depict the distribution of patients with liver diseases before and after SMOTE. T*he original dataset was imbalanced, with 56.5% of the patients being healthy, 25% having hepatitis, 5.3% having cirrhosis, and 5.3% having fibrosis*. SMOTE was used to balance the classes in the dataset by creating synthetic data points that were similar to the existing data points. *This resulted in a more evenly distributed dataset, with 25% of the patients in each class.*

Table 1 | Without SMOTE

Model	Accuracy	F-1 score	Precision
LR	0.903	0.628	0.615
SVM	0.946	0.609	0.736
KNN	0.849	0.605	0.611
Naive-Bayes	0.946	0.695	0.726

RF	0.946	0.653	0.702
MLP	0.956	0.613	0.619

Table 2 | With SMOTE

Model	Accuracy	F-1 score	Precision
LR	0.881	0.614	0.607
SVM	0.946	0.609	0.736
KNN	0.924	0.614	0.669
Naive-Bayes	0.946	0.695	0.726
RF	0.946	0.630	0.684
MLP	0.956	0.713	0.747

Table 1 shows the accuracy and precision of various machine learning models in the absence of SMOTE. The MLP model is the most accurate, with a score of 0.956, followed by RF, LR, SVM, KNN, and Naive Bayes. At 0.619, the MLP model has the highest precision. While RF and LR are both good performers, they fall short of the MLP model.

Table 2 shows the accuracy, F1-score, and precision of various machine learning models employing SMOTE (Synthetic Minority Over-sampling Technique). The MLP model has the highest accuracy (0.956), followed by RF, LR, SVM, K-Nearest Neighbors, Naive Bayes, Random Forest, and Multi-layer Perceptron. SMOTE improves the accuracy, f-score, and precision of these models.

From our study it is evident that all of the models' accuracy is slightly higher with SMOTE than without SMOTE. With and without SMOTE, the MLP model has the maximum accuracy. With SMOTE, RF and LR accuracy is slightly higher than without SMOTE. SVM, KNN, and Naive Bayes accuracy are not significantly different with and without SMOTE.

But overall there is no significant difference in accuracy between with and without SMOTE. However, all models with SMOTE have a minor gain in accuracy. This shows that SMOTE can be used to improve the accuracy of machine learning models on imbalanced datasets.

3.2 OPTUNA AND MODEL HYPERPARAMETERS

A. OPTUNA

Optuna is a sophisticated hyperparameter optimization framework that employs state-of-the-art algorithms to effectively explore the hyperparameter space of machine learning and deep learning models. Hyperparameters are configuration parameters that have a significant impact on the performance of a model. By automating the process of tuning hyperparameters, Optuna can help to find the optimal set of hyperparameters that maximize the model's performance metric.

Optuna requires the definition of a search space for hyperparameters. Practitioners specify hyperparameters and their respective distributions, such as continuous parameters with uniform or log-uniform distributions, categorical parameters with predefined choices, and discrete parameters with integer values. This search space serves as the basis for Optuna's exploration of different hyperparameter configurations.

Optuna uses a Bayesian optimization algorithm to select the hyperparameters for each trial. The Bayesian optimization algorithm uses a probabilistic model to represent the uncertainty about the hyperparameter space. This model is updated as new trials are run, and the algorithm uses this information to select the next set of hyperparameters to try.

The optimization process continues until a stopping criterion is met, such as a maximum number of trials or a maximum amount of time. At the end of the optimization process, Optuna returns the optimal set of hyperparameters.

For our model we utilized Optuna to optimize the hyperparameter. It aids in finding the best hyperparameter which results in optimal performance of the machine learning model. For the prediction of hepatitis C disease we have created an OPTUNA objective function that uses a voting classifier to ensemble six different machine learning models: logistic regression, KNN, SVM, random forest, naive Bayes, and MLP.
The objective function first defines the hyperparameters for each of the six models.

B. MODEL AND ITS HYPERPARAMETERS

1. **Logistic Regression:** It is a statistical model that predicts the probability of a binary outcome, such as whether a customer will click on an ad or not. The model is a linear regression model with a sigmoid function in the output layer. The sigmoid function is a nonlinear function that maps real numbers to the interval [0, 1]. This allows the logistic regression model to predict probabilities.

The mathematical equation for the logistic regression model is as follows:

$$p(y = 1 \mid x) = 1 / (1 + \exp(-wx))$$

where:

- p(y = 1 | x) is the probability that the outcome is 1 given the input x
- w is the vector of weights
- x is the vector of features
- exp() is the exponential function

The logistic regression model is trained by minimizing the following loss function:

$$L = -\text{sum}(y * \log(p(y \mid x)) + (1 - y) * \log(1 - p(y \mid x)))$$

where:

- L is the loss function
- y is the ground truth label
- p(y | x) is the predicted probability

Using a gradient descent algorithm the loss function is minimized.

The below are the hyperparameters we used in logistic regression:

- ***lr_penalty:*** *Type of regularization penalty ('l1', 'l2', or 'elasticnet') applied to logistic regression.*
- ***lr_solver1 and lr_solver2****: Solvers used for optimization in logistic regression, depending on 'lr_penalty' ('liblinear', 'saga', 'newton-cg', 'lbfgs', or 'sag').*

- ***lr_l1_ratio:*** *Mixing parameter for elasticnet penalty (0 for L2, 1 for L1), used when 'lr_penalty' is 'elasticnet'.*

- ***lr_tol:*** *Tolerance for stopping criteria during optimization (values between 1e-5 and 1e-2).*

- ***lr_C:*** *Inverse of regularization strength (C) for logistic regression (values between 0.0 and 1.0).*

The Logistic Regression achieved an *accuracy of 88.17 % with F-1 score and precision being 0.61 and 0.60* respectively.

2. **K-Nearest Neighbors (KNN):** It is a non-parametric machine learning technique that can be utilized for classification as well as regression tasks. The algorithm works by locating the k training instances that are the most similar to a new data point and then predicting the class or value of the new data point based on the classes or values of the k nearest neighbors.A distance metric can be used to assess the similarity of two data points. The Euclidean distance, the Manhattan distance, and the Minkowski distance are all common distance measures.

The distance metric employed in KNN most frequently is the Euclidean distance.It is defined as follows:

d(x, y) = sqrt(sum((x_i - y_i)^2))

where:

- d(x, y) is the distance between the two data points x and y
- x_i and y_i are the i-th features of the two data points x and y

The Manhattan distance is another common distance metric used in KNN. It is defined as follows:

d(x, y) = sum(| x_i - y_i |)

The Minkowski distance is a generalization of the Euclidean distance and the Manhattan distance. It is defined as follows:

d(x, y) = (sum (|x_i - y_i| ^ p) ^ (1/p))

where:

- p is a parameter that controls the weight of the distance between two data points

After measuring the similarity of two data points, the k most similar data points to a new data point can be discovered. The k nearest neighbors can then be used to anticipate the new data point's class or value.
The most common class among the k nearest neighbors can be used to forecast the class of the new data point. A new data point's value can be predicted by averaging the values of its k nearest neighbors

The hyperparameters we utilized in K-nearest neighbors are as follows:

- ***knn_neighbors:*** *An integer hyperparameter ranging from 2 to 100 that represents the number of neighbors utilized in K-Nearest Neighbors.*

- ***knn_weights:*** *A categorical hyperparameter with the options 'uniform' and 'distance' that determines how neighboring points are weighted for predictions ('uniform' for equal weight,' 'distance' for closer neighbors having more effect).*

- ***knn_p:*** *Categorical hyperparameter having values 1 and 2, corresponding to the power parameter for the Minkowski distance metric used in KNN ('1' for Manhattan distance, '2' for Euclidean distance).*

The K-Nearest neighbors (KNN) achieved an *accuracy of 92.47 % with F-1 score and precision being 0.61 and 0.66* respectively.

3. **Support Vector Machines (SVMs):** These models are a type of supervised machine learning that may be applied to both classification and regression. SVMs function by determining the optimum hyperplane between two classes of data.A hyperplane is a line or plane that splits data into two areas, with all data points in one region belonging to one class and all data points in the other region belonging to the other.The SVM method determines the hyperplane with the greatest margin between the two classes. The margin is the distance between the hyperplane and the data points in each class that are closest to it.

The mathematical equation of SVM is defined as follows:

min_{w, b} \frac{1}{2} ||w||^2 + C \sum_{i=1}^m \xi_i

where:

- ||w||^2 is the squared norm of the weight vector.
- C is a hyperparameter that controls the trade-off between maximizing the margin and minimizing the number of misclassifications

The hyperparameters we utilized in support vector machines are as follows:

- ***svm_C:*** *A uniform hyperparameter with values ranging from 0.0 to 1.0 that represents the regularization parameter (C) for SVM. Smaller C values indicate more regularization.*

- ***svm_kernel:*** *Categorical hyperparameter having values 'poly' and 'rbf' that determine the type of kernel used by SVM ('poly' for polynomial kernel, 'rbf' for radial basis function kernel). If'svm_kernel' is 'poly,' an integer hyperparameter with values ranging from 1 to 10, denoting the degree of the polynomial kernel. The number 3 is chosen as the default for 'rbf'.*

- ***svm_tol:*** *A uniform hyperparameter with values ranging from 1e-5 to 1e-2 that represents the tolerance for halting criteria during optimization.*

The Support vector machines (SVMs) achieved an *accuracy of 94.62 % with F-1*

score and precision being 0.60 and 0.73 respectively.

4. **Random Forest:** Random forest is an ensemble learning method that makes predictions by combining numerous decision trees. Each decision tree is trained on a random portion of the training data, and the multiple trees' predictions are then pooled to generate a final prediction.

To begin, the random forest method generates a bootstrap sample of the training data. A bootstrap sample is a random sample from the training data that has been replaced. This means that a data point can appear numerous times in the bootstrap sample.
A decision tree is trained on the bootstrap sample after it has been produced. A greedy approach is used to create the decision tree, which iteratively separates the training data into smaller and smaller sections.
The splitting criterion used by the decision tree algorithm is the Gini impurity criterion.
The Gini impurity criteria quantifies the impurity of a node in a decision tree. The impurity of a node is a measure of how well the data points in the node are categorized. A node with low impurity has well-classified data points.

The decision tree method will continue to split the training data until it hits a stopping threshold, such as a minimum number of samples or a maximum tree depth.Once all of the decision trees have been trained, the forecasts of the individual trees are combined to generate a final prediction. The projections of the different trees are often merged using a voting mechanism.The ultimate prediction in a voting scheme is the class anticipated by the majority of the decision trees.

The mathematical equation for random forest is as follows:

$$p(y \mid x) = sum_{i=1} \hat{} n \; w_i * p_i (y \mid x)$$

where:

- p(y | x) is the predicted probability of class y for the input x
- w_i is the weight of the i-th decision tree
- p_i(y | x) is the predicted probability of class y for the input x from the i-th decision tree

The mathematical equation for the Gini impurity criterion is as follows:

$$Gini = sum_{y} . p(y) * (1 - p(y))$$

The following are the hyperparameters we utilized in Random Forest:

- ***rf_estimators:*** *The number of decision trees in the Random Forest (numbers between 1 and 500).*

- ***rf_criterion:*** *A criterion for measuring split quality (also known as 'entropy' or 'gini').*

- ***rf_max_depth:*** *The maximum depth of decision trees (values between 1 and 100).*

- ***rf_min_samples_split:*** *The number of samples necessary to split an internal node (values between 2 and 50).*

- ***rf_min_samples_leaf:*** *The number of samples required to be at a leaf node (values between 1 and 25).*

The Random Forest achieved an *accuracy of 94.62 % with F-1 score and precision being 0.63 and 0.68* respectively.

5. **Naive Bayes:** The Bayes theorem is used to produce predictions with Naive Bayes. The Bayes theorem is a mathematical formula that predicts the likelihood of one event occurring given the likelihood of another. The chance of a data point belonging to a class is calculated in Naive Bayes as the product of the probabilities of the data point's features. The classifier is named Naive Bayes because the probabilities of the features are considered to be independent of one another.

The following are the hyperparameters we utilized in Naive Bayes:

nb_smoothing: *Uniform hyperparameter with values between 1e-10 and 1e-6. It represents the smoothing parameter (variance smoothing) used to avoid zero probabilities and improve the robustness of the Gaussian Naive Bayes model.*
Then the Gaussian Naive Bayes model (nb) is created using the GaussianNB class with the selected nb_smoothing hyperparameter, which enables the model to handle continuous data and make predictions based on the assumption of normal distribution for each feature.

The Naive Bayes achieved an accuracy of 94.62 % with F-1 score and precision being 0.69 and 0.72 respectively.

6. **Multilayer Perceptron:** MLP is a form of artificial neural network (ANN) that is made up of multiple layers of perceptrons. Perceptrons are basic units that can compute linear functions. An MLP's various layers of perceptrons enable the network to learn more complex functions.MLPs are frequently employed in classification and regression tasks. They are also utilized for image classification, natural language processing, and speech recognition, among other things.

An MLP's input layer is the top layer. Data enters the network at the input layer. The following layer is the concealed layer. The network learns to represent data in the hidden layer. An MLP can have a variable number of hidden layers..The output layer is the layer where the predictions are made.

Each layer's perceptrons are linked together. The connections between the perceptrons are weighted. Weights are taught during the training procedure. The training phase involves modifying the weights in the network so that it can generate correct predictions.
A backpropagation algorithm is commonly used in the training of an MLP. The backpropagation algorithm is an iterative technique that modifies the weights in the

network to reduce the error between anticipated and actual values.

The following are the hyperparameters utilized in Multilayer perceptron:

- ***mlp_hidden_layers:*** *Categorical hyperparameter with choices [1, 2, 3], representing the number of hidden layers in the MLP model.*

- ***mlp_hidden_units:*** *Integer hyperparameter with values between 16 and 128, representing the number of neurons in each hidden layer.*
- ***mlp_activation:*** *Categorical hyperparameter with choices ['relu', 'tanh', 'logistic'], determining the activation function used in the hidden layers.*

- ***mlp_alpha:*** *Uniform hyperparameter with values between 1e-6 and 1e-3, representing the L2 regularization parameter for weight decay to prevent overfitting.*

The Multilayer perceptron (MLP) achieved an *accuracy of 95.69 % with F-1 score and precision being 0.71 and 0.74* respectively.

7. **Ensemble Model and Voting Classifier**
An ensemble model is a grouping of numerous machine learning models (classifiers or regressors) that collaborate to deliver a more accurate and reliable prediction. A Voting Classifier is a machine learning ensemble model that combines the predictions of numerous base classifiers (or models) to provide a final prediction. It works on the majority voting concept, where each base classifier's prediction is treated as a "vote" for a specific class, and the class with the most votes becomes the final predicted class.

In our proposed framework a VotingClassifier class is used to build an ensemble model called vc, which combines predictions from six base models: Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Random Forest, Naive Bayes, and Multi-Layer Perceptron. Each base model is given a weight between 0.0 and 1.0 (lr_w, knn_w, svm_w, rf_w, nb_w, and mlp_w) to determine its influence on the final prediction. The ensemble model is then fitted on the balanced training data (X_bal, y_bal), and the accuracy (acc) is calculated using predictions made on the validation set (X_val). The goal of this ensemble strategy is to improve prediction.

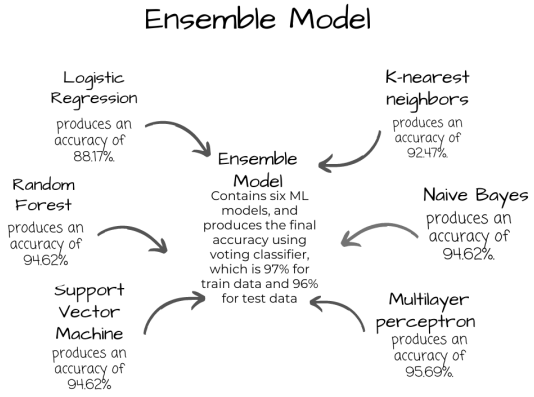

Fig. 4.. Ensemble Model

C. EVALUATING THE VALIDATION SET

Using Optuna, we obtain the findings of a hyperparameter optimization research. It shows the model accuracy on the validation set from the best trial is found to be 97%, as

well as the best hyperparameters discovered during the investigation. The best hyperparameters are displayed as a dictionary with their names and optimal values.

TABLE 3

Hyperparameter	**Values**
lr_penalty	'l2'
lr_solver2	'lbfgs'
lr_tol	0.003814159012031626
lr_C	0.42350470965469134
knn_neighbors	83
knn_weights	'distance'
knn_p	2
svm_C	0.9731171875077148
svm_kernel	'poly'
svm_degree	5
svm_tol	0.002292510698981856
Rf_estimators	70
rf_criterion	'gini'
rf_max_depth	38
rf_min_samples_split	29
rf_min_samples_leaf	14
nb_smoothing	2.596301857486734e-07
mlp_hidden_layers	2
mlp_hidden_units	113
mlp_activation	'tanh'
mlp_alpha	0.004011044419757664
et_n_estimators	38
et_criterion	'gini'
et_max_depth	29
et_min_samples_split	30
et_min_samples_leaf	21
lr_w	0.029872771851786117
knn_w	0.2574479288604744
svm_w	0.5286677776732325
rf_w	0.6069934816118663
nb_w	0.3189569725246839
mlp_w	0.8092536312976131
et_w	0.40729870664578616

Table 3 gives a detailed list of hyperparameter names and its corresponding optimal values.

Finally, we create an ensemble model using the best hyperparameters obtained from a hyperparameter optimization study. The best optimized hyperparameters are stored in the "study.best_params" attribute, which are used to initialize the ensemble model. The resulting model is stored in the variable model for further use and evaluation of the test data.

3.3 SHAP

Model interpretability is critical for understanding and trusting prediction models in the field of machine learning research. SHAP (SHapley Additive exPlanations) is a powerful and extensively used framework for evaluating the predictions of complex machine learning models. In this research, we explore the use of SHAP, concentrating on its permutation-based approach to computing SHAP values. The use of SHAP in our study entails developing a SHAP explanation to help comprehend the model's predictions.First, we created a SHAP explanation using the shap.Explainer functionality. The explanation is put up to interpret our model's predictions on a certain dataset. To provide explicit feature attribution, we label the features appropriately by supplying the relevant feature names. We compute SHAP values using a permutation-based technique.

The SHAP explanation gives information about the computation's progress and timing throughout the procedure.The output message *"Permutation explainer: 94it [17:53, 11.55s/it]" indicates that the explainer has processed 94 iterations*

(samples) out of the total samples in the test dataset. The first component "17:53" represents the elapsed time in minutes and seconds since the start of the computation, while "11.55s/it" is the average time taken each iteration (sample) in seconds.

By using SHAP we were able to visually depict the importance of the attributes in the dataset.

The accompanying bar graph Fig.5.1 depicts the importance of each attribute. The average SHAP value for each characteristic is shown in the bar graph. The SHAP value of a feature indicates how much it contributes to the model's prediction.

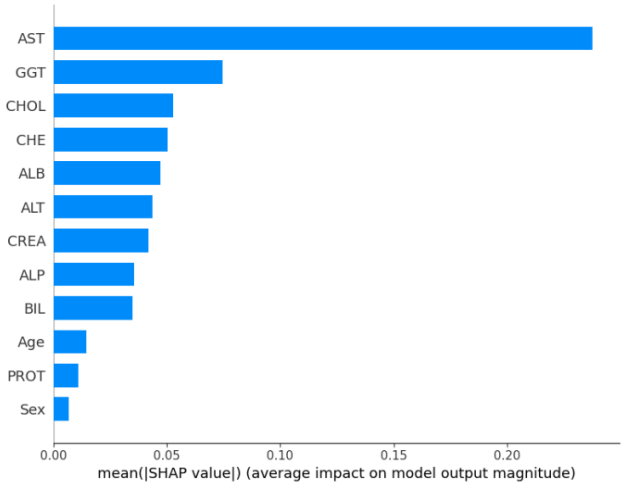

Fig. 5.1. Importance of each feature(Bar Graph)

The bar graph Fig.5.1 is separated into two sections: positive and negative SHAP values. Positive SHAP values imply that the feature positively adds to the model's prediction. Negative SHAP values imply that the feature has a negative impact on the model's prediction.

The magnitude of the SHAP values is also used to sort the bar graph. The properties with the highest SHAP values are the most crucial to the model's predictions.

The aspartate aminotransferase "AST" is the most essential feature for the model's predictions. This characteristic is an enzyme found primarily in the liver, but also in muscles and other organs. When damaged cells contain AST, the AST is released into the bloodstream.
The feature with the second highest SHAP score is Gamma-glutamyl Transferase (GGT) test. This feature is a test that determines the level of GGT in the blood. GGT is an enzyme that can be found all over the body, although it is most prevalent in the liver. If the liver is damaged, GGT could leak into the bloodstream.

This is also visualized using Force Plot

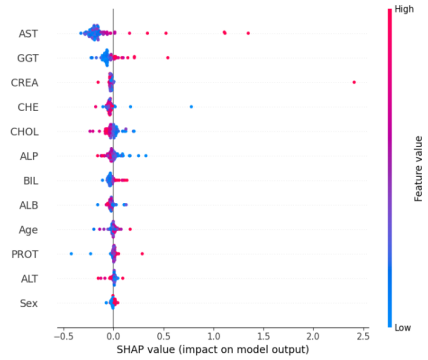

Fig.5.2. Importance of each feature(Force Plot)

The graphic Fig 5.2 depicts the SHAP force plot for a model that predicts whether the patient will be diagnosed with liver illness or not. The dataset contains the following features:age, AST, ALT, GGT, ALB, BIL, and PROT.

The arrows in the graphic represent each feature's contribution to the model's prediction. The size of the arrows represents the magnitude of the contribution. The contribution is indicated by the direction of the arrows. For example, *the arrow for the feature "AST" points to the right, indicating that a high AST score is connected with an increased likelihood of being diagnosed with liver illness.*

It also gives us the information that AST, ALT, and GGT are the most relevant features for the model's predictions. These characteristics are all connected to liver function and are all associated with an increased risk of being diagnosed with liver disease.

The other properties in the image are also significant for the model's predictions, *but not as much as the AST, ALT, and GGT.*

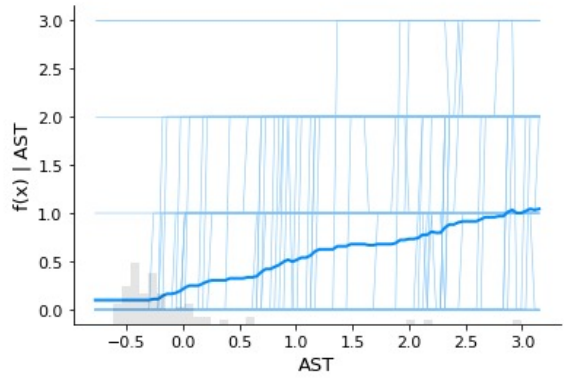

Fig.5.3. Effects of the AST feature

The graph Fig.5.3. *indicates that AST levels rise before any other indications of liver disease arise. This suggests that AST levels could be used to evaluate people for liver illness before they show any symptoms. If AST levels are confirmed to be elevated, additional tests may be performed to confirm the diagnosis of liver disease.*

The graph also indicates that AST levels can change over time in persons with liver disease. This means that a single AST test may not be enough to diagnose liver disease. However, if AST values are consistently excessive, this is a clear indication of liver disease.

Results and Discussions

The proposed framework presents a comprehensive approach to enhance the performance of a classification model. It begins by employing SMOTE to address data imbalance, creating a balanced dataset. *According to the study, SMOTE boosts model accuracy marginally, with the MLP model having the highest accuracy. SMOTE improves RF and LR accuracy slightly, but SVM, KNN, and Naive Bayes accuracy remains insignificant.* Subsequently, hyperparameter optimization using Optuna is applied to fine-tune the ensemble model, consisting of SVM, KNN, RF, Naive Bayes, LR, and MLP classifiers.*When models were observed individually, the accuracy of MLP tops the other models with an accuracy of 95.69%. And the optimized ensemble model achieved an impressive 97% accuracy on the validation data and maintained its robustness, yielding 96% accuracy on the test data.* Moreover, the implementation of SHAP features provided valuable insights into the model's predictions and increased its interpretability. *The framework provides importance of each feature using Bar Graph and Force Plot;* it also demonstrates the effectiveness of combining data

preprocessing techniques, hyperparameter tuning, and interpretability tools to build powerful and reliable classification models.

Table 4 presents the Model name with their corresponding individual accuracy score.

TABLE 4 | Comparison of Accuracy

Name of Model	Accuracy (%)
Logistic Regression	88.17
Random Forest	94.62
K- nearest neighbors	92.47
Naive Bayes	94.63
Support Vector Machines	94.62
Multilayer perceptron	95.69

Table 5 summarizes the techniques used, reason and outcome of that.

TABLE 5 | Techniques and their uses

Techniques	Uses and results
SMOTE	Balanced the dataset, So that the output produced is unbiased.
OPTUNA	Tuned and Optimized the hyperparameters of the machine learning model thereby producing high accuracy values
SHAP	Increased the interpretability of the model's prediction.

CONCLUSION

Finally, our research offered a complete framework that successfully improved the performance of a classification model. We got outstanding results by using SMOTE to handle data imbalance and Optuna for hyperparameter optimization in an ensemble model consisting of SVM, KNN, RF, Naive Bayes, LR, and MLP classifiers. The MLP classifier achieved the greatest individual accuracy of **95.69%**. The optimized ensemble model, on the other hand, beat all individual models, achieving an outstanding **97%** accuracy on the validation data and displaying robustness with **96%** accuracy on the test data. Furthermore, the inclusion of SHAP features improved the model's interpretability, providing significant insights into its predictions.

Visualizations such as the Bar Graph and Force Plot, which highlighted feature relevance, added to the framework's efficacy. To potentially attain even greater accuracies, we advocate studying advanced data preprocessing approaches, including more diverse classifiers, and investigating other hyperparameter optimization strategies in future research. Furthermore, extending our model's application to larger datasets or various domains can provide useful insights into its generalizability and scalability. Finally, our research provides vital insights into the development of robust and trustworthy classification models and opens up new opportunities for developments in the field of machine learning.

Funding: This research received no external funding.

Declarations Conflict of interest: The author declares no competing interests.

Reference Papers:

[1] Smith, D.B., Bukh, J., Kuiken, C., Muerhoff, A.S., Rice, C.M., Stapleton, J.T. and Simmonds, P., 2014. Expanded classification of hepatitis C virus into 7 genotypes and 67 subtypes: updated criteria and genotype assignment web resource. Hepatology, 59(1), pp.318- 327.

[2] Borgia, S.M., Hedskog, C., Parhy, B., Hyland, R.H., Stamm, L.M., Brainard, D.M., Subramanian, M.G., McHutchison, J.G., Mo, H., Svarovskaia, E. and Shafran, S.D., 2018. Identification of a novel hepatitis C virus genotype from Punjab, India: expanding classification of hepatitis C virus into 8 genotypes. The Journal of infectious diseases, 218(11), pp.1722-1729.

[3] Louie, K.S., St Laurent, S., Forssen, U.M., Mundy, L.M. and Pimenta, J.M., 2012. The high comorbidity burden of the hepatitis C virus infected population in the United States. BMC infectious diseases, 12, pp.1-11.

[4] Kashif, A.A., Bakhtawar, B., Akhtar, A., Akhtar, S., Aziz, N. and Javeid, M.S., 2021. Treatment response prediction in hepatitis C patients using machine learning techniques. International Journal of Technology, Innovation and Management (IJTIM), 1(2), pp.79-89.

[5] Yang, L. and Shami, A., 2020. On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing, 415, pp.295-316.

[6] Cai, J., Luo, J., Wang, S. and Yang, S., 2018. Feature selection in machine learning: A new perspective. Neurocomputing, 300, pp.70-79.

[7] Tran, N., Schneider, J.G., Weber, I. and Qin, A.K., 2020. Hyper-parameter optimization in classification: To-do or not-to-do. Pattern Recognition, 103, p.107245.

[8] Nugroho, A. and Suhartanto, H., 2020, September. Hyper-parameter tuning based on random search for densenet optimization. In 2020 7th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE) (pp. 96-99). IEEE.

[9] Cai, Z., Long, Y. and Shao, L., 2019. Classification complexity assessment for hyperparameter optimization. Pattern Recognition Letters, 125, pp.396-403.

[10] Akiba, T., Sano, S., Yanase, T., Ohta, T. and Koyama, M., 2019, July. Optuna: A next generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 2623-2631).

[11] Yağanoğlu, M., 2022. Hepatitis C virus data analysis and prediction using machine learning. Data & Knowledge Engineering, 142, p.102087.

[12] Tonmoy, S.T.I. and Zaman, S.M., 2022, December. OOG-Optuna Optimized GAN Sampling Technique for Tabular Imbalanced Malware Data. In 2022 IEEE International Conference on Big Data (Big Data) (pp. 6534-6539). IEEE.

[13] Shams, M.Y., El-kenawy, E.M., Ibrahim, A., and Elshewey, A.M., 2023. A Hybrid Dipper Throated Optimization Algorithm and Particle Swarm Optimization (DTPSO) Model for Hepatocellular Carcinoma (HCC) Prediction. Biomedical Signal Processing and Control, 85(2023), p. 104908.

[14] Ahmed M. Elshewey.,hyOPTGB: An Efficient OPTUNA Hyperparameter Optimization Framework for Hepatitis C Virus (HCV) Disease Prediction in Egypt.

[15] Ali Mohd Ali 1 , Mohammad R. Hassan 1 , Faisal Aburub 2 , Mohammad Alauthman 3 , Amjad Aldweesh 4 , Ahmad Al-Qerem 5 , Issam Jebreen 6 and Ahmad Nabot 6.,Explainable Machine Learning Approach for Hepatitis C Diagnosis Using SFS Feature Selection.

[16] Satish CR Nandipati1 , Chew XinYing1 , Khaw Khai Wah2.,Hepatitis C Virus (HCV) Prediction by Machine Learning Techniques

[17] Michael Onyema Edeh1 , Surjeet Dalal 2 , Imed Ben Dhaou3 , Charles Chuka Agubosim4 , Chukwudum Collins Umoke5 , Nneka Ernestina Richard-Nnabu6 and Neeraj Dahiya7.,Artificial Intelligence-Based Ensemble Learning Model for Prediction of Hepatitis C Disease. Intelligence-Based Ensemble Learning Model for Prediction of Hepatitis C Disease.

[18] Tsvetkov, V., Tokin, I. and Lioznov, D., 2021. Machine learning model for diagnosing the stage of liver fibrosis in patients with chronic viral hepatitis C.

[19] Dritsas, E.; Trigka, M. Supervised Machine Learning Models for Liver Disease Risk Prediction. Computers 2023, 12, 19.

[20] Alauthman, M.; Aldweesh, A.; Al-qerem, A.; Aburub, F.; Al-Smadi, Y.; Abaker, A.M.; Alzubi, O.R.; Alzubi, B. Tabular Data Generation to Improve Classification of Liver Disease Diagnosis. Appl. Sci. 2023, 13, 2678.