Journal of Electronics, Electromedical Engineering, and Medical Informatics
Multidisciplinary: Rapid Review: Open Access Journal **Vol. 7, No. 2, April 2025, pp: 508-519; eISSN: 2656-8632**

RESEARCH ARTICLE **OPEN ACCESS**

Manuscript received February 8, 2025; Accepted April 20, 2025; date of publication April 15, 2025

Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i2.730

How to cite: Lista Tri Nalasari, Syaiful Anam, and Nur Shofianah, "Liver Cirrhosis Classification using Extreme Gradient Boosting Classifier and Harris Hawk Optimization as Hyperparameter Tuning", Journal of Electronics, Electromedical Engineering, and Medical Informatics, vol. 7, no. 2, pp. 508-519, April 2025.

Liver Cirrhosis Classification using Extreme Gradient Boosting Classifier and Harris Hawk Optimization as Hyperparameter Tuning

Lista Tri Nalasari, Syaiful Anam, and Nur Shofianah

Mathematics Department, Brawijaya University, Malang, East Java, Indonesia

Corresponding author: Syaiful Anam (e-mail: syaiful@ub.ac.id).

This work was supported by Brawijaya University for providing valuable resources and support.

ABSTRACT This study proposes an early diagnosis model based on Machine Learning for liver cirrhosis classification using the Hepatitis C dataset, which is the leading cause of cirrhosis, from UCI ML. The classification is performed using the XGBoost algorithm because it provides high accuracy and time efficiency based on previous studies. However, these advantages depend on the combination of its hyperparameters set. XGBoost has a large number of hyperparameters, which can be time-consuming for researchers to manually configure. Therefore, this study proposes combining XGBoost with the Harris Hawks Optimization (HHO) algorithm for hyperparameter tuning. HHO is implemented with a hawk population of 40 and maximum iterations set at 25. The proposed XGBoost-HHO model provides an average performance of 99.34% for accuracy, MAR, MAP and 99.33% for Macro F1-*score*. These performances are achieved with the shortest processing time across 25 experiments compared to other combination models. The performance of the XGBoost-HHO model shows more significant increase in performance and reduction in overfitting compared to the standard XGBoost, SVM, RF models, as well as several other combined models including RF-HHO, SVM-HHO, XGBoost-PSO, and XGBoost-BA. Additionally, based on the feature importance analysis of the XGBoost-HHO algorithm, *Alanine Aminotransferase* (ALT), Protein, and *Gamma-glutamyltransferase* (GGT) contribute the most to the classification process, with gain values of 11.21, 9.51, and 7.98, respectively. Overall, the findings of this study show that the XGBoost-HHO algorithm combination provides competitive performance and can serve as an excellent alternative for liver cirrhosis classification in terms of both accuracy and time efficiency.

INDEX TERMS XGBoost, Harris Hawk Optimizer, Hyperparameter Tuning, Liver Cirrhosis.

I. INTRODUCTION

A vital organ in the human body, the liver, is essential for controlling metabolism and preserving general health. Around two million people die each year from liver disease worldwide, and liver cirrhosis responsible for half of these fatalities, with the rest from hepatitis and liver cancer [1]. As the last stage of all chronic liver disorders, liver cirrhosis is typified by necrosis, or extreme destruction to the liver's tissue [2]. Liver cirrhosis has several causes, including excessive alcohol consumption, hepatitis B (HBV) and hepatitis C (HCV) viral infections, non-alcoholic fatty liver disease, and autoimmune conditions. Globally, HCV infection is the leading cause of liver cirrhosis progression [3].

Globally, liver cirrhosis is the eleventh most common cause of death [4]. Additionally, in the US, it ranks as the ninth most common cause of mortality, with a prevalence of 17% per 100,000 death cases [5]. The World Health Organization (WHO) reported that approximately 51.1% of men and 27.1% of women out of 100,000 global death cases in 2016 were caused by liver cirrhosis. Furthermore, the prevalence of deaths due to liver cirrhosis in South Asia and Southeast Asia is 44.9% [6]. However, the significant global impact of liver cirrhosis-related deaths is not yet matched by adequate preventive efforts. About 73% of patients who died from this disease were hospitalized for the first time only after reaching

the decompensated stage [7]. This is due to the lack of clearly visible symptoms in the early stages of liver damage, making timely identification difficult. Therefore, the development of early diagnostic methods is crucial to ensure appropriate treatment can be determined sooner, thereby increasing patients' life expectancy [4].

The development of early diagnosis based on artificial intelligence technology has significantly helped improve the efficiency of medical professionals in determining patient care [8]. This aligns with the numerous studies that have tried to develop disease diagnosis models using clinical data supported by Machine Learning (ML). [8] developed a liver cirrhosis classification model using ML approaches with Support Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF) algorithms. This study showed that SVM and RF algorithms produced the best accuracy. However, in 2019, [9] developed a liver damage classification model and found that Extreme Gradient Boosting (XGBoost) outperformed DT, RF, k-Nearest Neighbor (kNN), SVM, and Artificial Neural Network (ANN). Additionally, a study by [10] on breast cancer diagnosis models showed that the performance of the XGBoost algorithm surpassed RF. Similarly, [11] compared several ML algorithms, including XGBoost, Logistic Regression (LR), Light Gradient Boosting Machine (LGBM), DT, and SVM, for hepatitis C disease prediction. The study found that XGBoost delivered the best results. This is consistent with research by [12], which developed a Chronic Kidney Disease (CKD) diagnosis model using XGBoost, LR, SVM, and Classification and Regression Tree (CART). That study concluded that XGBoost outperformed the other algorithms, achieving an accuracy, sensitivity, and specificity of 1.00 [12].

The advantages of XGBoost in several studies are supported by the proper configuration of its parameters [13] [14]. However, XGBoost has a relatively large number of parameters, which makes manual tuning time-consuming. Therefore, this study will implement hyperparameter tuning to automatically find parameter combinations, making the process more effective.

Hyperparameter tuning can be applied using various algorithms. Metaheuristic algorithms are effective for complex optimization, efficiently exploring large solution spaces and finding near-optimal solutions in fewer iterations. However, genetic algorithms take longer due to sequential crossover and mutation processes, while SI offers greater computational efficiency with large-scale parallelization support [15]. Therefore, this study employs one of the SI algorithms, namely the Harris Hawks Optimizer (HHO). The algorithm was chosen because it provides superior and more consistent performance in multi-dimensional problems compared to Genetic Algorithms, Particle Swarm Optimization (PSO), and Differential Evolution (DE) [16]. HHO has also demonstrated excellent performance in hyperparameter optimization for the RF algorithm in predicting pile setup parameters [17], as well as for XGBoost in predicting drill penetration rates [18]. Thus, this study proposes a combination of XGBoost optimized using the HHO algorithm for hyperparameter tuning to develop a more accurate and efficient classification model for liver cirrhosis patients as an early diagnosis method. The contributions of this study are as follows:

1. This study proposes an early diagnosis model for liver cirrhosis based on clinical data classification using XGBoost optimized with the HHO algorithm for hyperparameter tuning to enhance classification performance and time efficiency.
2. The XGBoost-HHO model is assessed and contrasted with alternative machine learning algorithms and hyperparameter optimization techniques to confirm its superior performance.

II. METHODS

This study examines the classification of liver cirrhosis disease using Hepatitis C data was sourced from the UC Irvine Machine Learning repository that can be accessed at https://archive.ics.uci.edu/ml/datasets/HCV+data. This data was chosen because HCV is the main factor in liver cirrhosis. The process begins with a preprocessing stage prepares the data for training and testing with the XGBoost classifier. This study highlights hyperparameter optimization performed using HHO and evaluates the model's classification performance using specific metrics. The research workflow is presented in FIGURE 1.

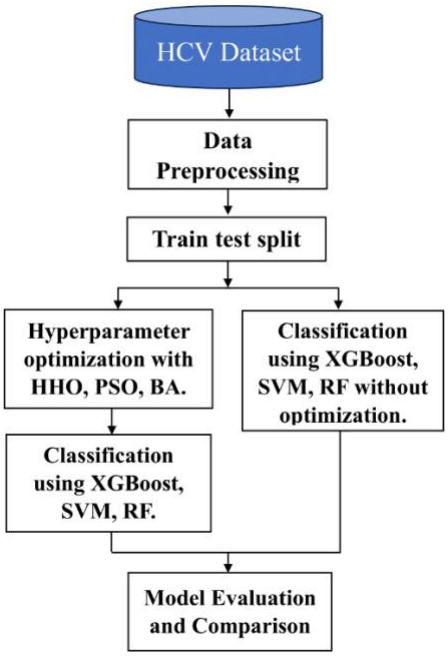

FIGURE 1. Research workflow

This study was conducted using Python within a Jupyter Notebook environment on a Windows 10 Pro 64-bit system, equipped with 8GB of RAM, a 256GB SSD, an AMD Ryzen 3 2200U processor (2.1 GHz), and integrated Radeon Vega Mobile Graphics. The dataset analysis was conducted using Sklearn, Matplotlib, Pandas, and Numpy libraries. Additionally, model validation was performed using evaluation metrics from the Sklearn package, including classification_report.

A. DATA COLLECTION

This study uses Hepatitis C data comprising 615 observations with 14 variables. These 14 variables include 13 features and 1 target class. The target class in this classification is the 'Category' variable, which consists of the classes 0 represents healthy, 1 indicates hepatitis, 2 correspond fibrosis, and 3 denotes cirrhosis. Before being processed in the classification model, the dataset will undergo a descriptive statistical analysis to identify the necessary data preprocessing steps. All variables included in this study's dataset are presented in TABLE 1.

TABLE 1
Dataset variable

Variable	Description	Data Type
ID	Patient ID	Integer
Age	Patient age	Integer
Sex	Patient gender	Categorical
ALB	Albumin	Continuous
ALP	*Alkalin Phosphotase*	Continuous
AST	*Aspartate Aminotransferase*	Continuous
BIL	Bilirubin	Continuous
CHE	*Cholinesterase*	Continuous
GGT	*Gamma-glutamyltransferase*	Continuous
PROT	Protein	Continuous
CHOL	Cholesterol	Continuous
CREA	Kreatinin	Continuous
ALT	*Alanine Aminotransferase*	Continuous
Category	0=Healthy, 1=Hepatitis, 2=Fibrosis, 3=Cirrhosis	Categorical

Liver Function Tests (LFTs) assess liver health through various parameters. AST and ALT indicate hepatocyte injury, with an AST/ALT ratio >1 suggesting cirrhosis progression. Increased bilirubin signals impaired excretion, while decreased albumin and PROT reflect reduced liver synthesis. Elevated ALP and GGT suggest cholestasis, often linked to cirrhosis. Low CHOL levels indicate advanced cirrhosis, whereas high levels suggest cholestasis or NAFLD. Reduced CHE serum and increased creatinine point to declining liver function and potential hepatorenal syndrome. Evaluating these markers alongside age helps classify disease severity into hepatitis, fibrosis, or cirrhosis [19].

B. DATA PREPROCESSING

The data preprocessing in this study begins with data identification and label encoding for features with object data types into integers using the Label Encoding method to facilitate analysis in ML algorithm. Outlier detection follows, employing the Interquartile Range (IQR) method to detect anomalous data or input errors. The IQR, which is chosen for its flexibility and robustness against data distribution variations. The IQR method works by measuring how far data points deviate from the average [20]. Additionally, missing values are handled using median imputation, which replaces missing values with the median of the respective feature. This approach is preferred due to its stability, resistance to outliers, and ease of implementation [21].

Feature scaling is the next stage to preserve the original data distribution and convert the value range into a uniform scale [22]. The two common methods are normalization and standardization. Both reducing bias from varying feature ranges [23]. This study applies normalization to ensure that all numerical columns in the dataset are scaled uniformly without altering their distribution. This approach enhances the model's training performance and maintains consistency [1]. This study uses the min-max normalization calculation as written in Eq. (1) [24],

$$x_{i_{norm}} = \frac{x_i - x_{i_{min}}}{x_{i_{max}} - x_{i_{min}}} \quad (1)$$

where $x_{i_{norm}}$ is the new value of the data sample x_i, $x_{i_{min}}$ and $x_{i_{max}}$ are the smallest and largest values in the feature column, respectively [24].

The normalization stage is crucial because this study requires handling data imbalance, because each class has an unequal sample size. This handling involves calculations sensitive to the distribution of data values. Imbalanced data can cause the model to perform well only on the majority class [25]. Therefore, resampling is necessary to balance the data. The resampling method used in this study is SMOTE-NC (Synthetic Minority Over-sampling Technique for Nominal and Continuous). This method can handle data with both nominal and continuous features, which aligns with this study's dataset that contains both types of data [26]. Although SMOTE-NC may generate less representative synthetic samples if the initial data distribution is suboptimal, this can be mitigated through distribution analysis and visualization. While alternatives like Borderline-SMOTE exist, SMOTE-NC remains the preferred choice due to the presence of categorical features in this dataset.

According to [27], SMOTE-NC is applied by identifying samples from the minority class. The median and standard deviation of each feature are used as benchmarks to measure the difference between these samples and their nearest neighbors. Synthetic samples for continuous features are constructed as described in Eq. (2) [28],

$$x_{m_{syn}} = x_{mi} + \lambda\left(x_{mj} - x_{mi}\right) \quad (2)$$

where x_{mi} represents the m-th continuous feature value of sample $\boldsymbol{x}_i$, x_{mj} represents the same feature of sample $\boldsymbol{x}_j$, and λ is a random number within the range $[0,1]$. Nominal features are assigned the most frequent value among the nearest neighbors.

C. EXTREME GRADIENT BOOSTING (XGBOOST)

XGBoost is an enhanced version of the Gradient Boosting (GB) algorithm in terms of optimization and regularization. This algorithm combines several CART decision trees to build a more robust model [29]. The XGBoost algorithm adds f_t as many as T to predict the output as Eq. (3) [30],

$$\hat{y}_i = \sum_{t=1}^{T} f_t(\boldsymbol{x}_i), f_t \in F \quad (3)$$

where $F = \{f(\boldsymbol{x}) = w_{q(x)}\}$ $(q: \mathbb{R} \to J, \boldsymbol{w} \in \mathbb{R}^J)$ represents the space of decision trees. q represents the structure of the

decision tree that maps the dataset to the corresponding leaf index, J is the number of leaves, and $w_{q(x)}$ is the weight of each leaf. Each f_t corresponds to an independent tree structure, q, with leaf weights, $\boldsymbol{w}$. The objective function in XGBoost is generally expressed as shown in Eq. (4) [31],

$$Obj^{(t)} = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \Omega(f_t) \tag{4}$$

where n represents the number of data samples, L represents the loss function that evaluates the model's performance on the training dataset, $\hat{y}_i$ is the predicted class of the i-th sample, y_i is the actual class of the i-th sample, and $\Omega(f_t)$ is the regularization term. $\Omega(f_t)$ regulates the complexity of the training trees by applying a penalty. f_t represents the functions of the constructed trees. The formula for the regularization term is written as shown in Eq. (5) [30],

$$\Omega(f_t) = \gamma J + \frac{1}{2}\lambda \sum_{j=1}^{J} w_j^2 \tag{5}$$

where γ and λ are hyperparameter, each leaf node is penalized by γ, and w_j^2 defines the L2 regularization of leaf weights, controlled by the value of λ.

XGBoost applies Taylor series expansion to the loss function to approximate the objective function [13]. This approach allows the algorithm to more accurately approximate changes in the loss function [31]. The optimization of the objective function after applying the second-order Taylor series expansion is shown in Eq. (6) [31],

$$Obj^{(t)} \approx \sum_{i=1}^{n} (L(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(\boldsymbol{x}_i) + \frac{1}{2} h_i f_t^2(\boldsymbol{x}_i)) + \Omega(f_t) \tag{6}$$

where g_i is the gradient or the first derivative of the loss function, and h_i is the second-order derivative of the gradient in the loss function, known as the Hessian.

Based on the expansion result in Eq. (6), the constant term $L(y_i, \hat{y}_i^{(t-1)})$ does not depend on the new model $f_t(\boldsymbol{x}_i)$ to be added, nor does it affect the gradient to be calculated during the optimization process. As a result, a simplified objective function for the ttt-th iteration is obtained, as shown in Eq. (7) [31].

$$Obj^{(t)} \approx \sum_{i=1}^{n} (g_i f_t(\boldsymbol{x}_i) + \frac{1}{2} h_i f_t^2(\boldsymbol{x}_i)) + \Omega(f_t). \tag{7}$$

If the set of j leaves is defined as $I_j = \{i | q(\boldsymbol{x}_i) = j\}$, then Eq. (7) can be written as Eq. (8) [32].

$$Obj^{(t)} = \sum_{j=1}^{K} \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma J. \tag{8}$$

The optimal value of a function can be found using the first derivative of the function. This can be applied to Eq. (8), resulting in the optimal weight for leaf j based on Eq. (9) [32].

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}. \tag{9}$$

The substitution of w_j^* yields the optimal value for the objective function in Eq. (10) [32]. This equation is used as a metric to evaluate the structure of the tree $q(\boldsymbol{x}_i)$.

$$Obj^* = -\frac{1}{2} \sum_{j=1}^{J} \frac{\left(\sum_{i \in I_j} g_i \right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma J. \tag{10}$$

If I is the set of samples at a leaf node before the split such that $I = I_{Left} \cup I_{Right}$, in addition I_{Left} and I_{Right} are the sets of samples split to the left and right sides as a result of the partition, then both parts will be measured for their contribution to reduce the residual value. This calculation is used to measure the quality of the split, shown in Eq. (11) [32].

$$gain = \frac{1}{2} \left(\frac{\left(\sum_{i \in I_{Left}} g_i \right)^2}{\sum_{i \in I_{Left}} h_i + \lambda} + \frac{\left(\sum_{i \in I_{Right}} g_i \right)^2}{\sum_{i \in I_{Right}} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right) - \gamma \tag{11}$$

Several hyperparameters of XGBoost that will be configured in this study are displayed in TABLE 2. The range of values and the default values for each hyperparameter are based on previous studies, including [31], [11], [33].

TABLE 2
Hyperparameter of XGBoost

Hyperparameter	Description	Default	Range
Learning rate	Reduces the weight of each step.	0.3	[0,1]
n_estimators	Number of trees.	100	[1,1000]
Max_depth	Maximum tree depth.	6	[1,10]
Gamma	Minimum error value reduction.	0	[0,5]
Reg_lambda	L2 regularization.	1	[0,100]
Subsample	Distribution of random data samples.	1	[0,1]
Colsample_ Bytree	Column subsample ratio.	1	[0,1]

D. HYPERPARAMETER TUNING

Hyperparameters play a role in directly controlling the behavior of the algorithm during the training process in ML. Determining specific hyperparameter values is done before the model training process. The process of finding a combination of hyperparameters that provides the best performance on the data for an ML algorithm is called hyperparameter tuning. Hyperparameter optimization is expressed in Eq. (12) [31],

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in \boldsymbol{X}} f(\boldsymbol{x}) \tag{12}$$

where $f(\boldsymbol{x})$ represents the objective score to be minimized and is evaluated on the validation set, $\boldsymbol{x}$ is the set of hyperparameters that yield the lowest score, and $\boldsymbol{x}$ can take any value within the domain $\boldsymbol{X}$.

E. HARRIS HAWK OPTIMIZATION

HHO is a nature-inspired algorithm from the cooperative hunting strategies of Harris hawks, introduced by Heidari et al. in 2019 [34]. Hawks are among the most intelligent birds in nature. The hawks' hunting process begins by perching in various locations around potential prey spots, usually rabbits, and then hunting the rabbits when the conditions are favorable. However, rabbits also exhibit survival behavior by escaping [35]. HHO has two phases in its hunting process: the exploration phase and the exploitation phase. These phases are controlled by the rabbit's energy to escape, E_{rabbit}. When $|E_{rabbit}| \geq 1$, it indicates that the rabbit still has sufficient energy to escape, so the HHO algorithm operates in the exploration phase. When the rabbit's energy to escape is $|E_{rabbit}| < 1$, the algorithm is in the exploitation phase. The calculation of E_{rabbit} is shown in Eq. (13) [36],

$$E_{rabbit} = 2E_0 \times \left(1 - \frac{t}{t_{max}}\right) \quad (13)$$

where t and t_{max} refer to the current iteration and the total number of iterations, respectively, E_0 is the initial energy of the rabbit, with its value randomly selected from the range $(-1,1)$.

1) EXPLORATION PHASE

This phase describes the strategy of Harris hawks perching randomly in various locations while scanning for prey based on two distinct approaches. This phase consists of two strategies determined by the value of q, which is defined as the probability of the hawk catching the rabbit. The probability q follows a uniform distribution within the range $(0,1)$. If $q \geq 0.5$, the hawk has a high probability of catching the rabbit in its current position. Thus, it will perch based on the position of other hawks of the hunting group and the rabbit. If $q < 0.5$, the hawk has a low probability, so it will choose to perch in another tree randomly to search for a more promising rabbit. If $q \geq 0.5$, the hawk's position is updated using Eq. (14) and Eq. (15) when $q < 0.5$ [36].

$$\boldsymbol{X}(t+1) = \boldsymbol{X}_r(t) - c_1|\boldsymbol{X}_r(t) - 2c_2\boldsymbol{X}(t)| \quad (14)$$

$$\boldsymbol{X}(t+1) = \left(\boldsymbol{X}_{rabbit}(t) - \boldsymbol{X}_{mean}(t)\right) - c_3\left(\boldsymbol{L_B} + c_4(\boldsymbol{U_B} - \boldsymbol{L_B})\right) \quad (15)$$

Here, $\boldsymbol{X}(t+1)$ represents the hawk's position in the next iteration, $\boldsymbol{X}(t)$, $\boldsymbol{X}_{mean}(t)$, $\boldsymbol{X}_r(t)$, and $\boldsymbol{X}_{rabbit}(t)$ are the current position of the hawk, the mean position of the population, the position of a randomly selected hawk, and the rabbit's position. The variable t indicates the current iteration, c_1 , c_2, c_3, and c_4 are random values within the range $(0, 1)$. $\boldsymbol{L_B}$ and $\boldsymbol{U_B}$ define the lower and upper boundaries of the search space. The population's mean position is calculated as shown in Eq. (16) [36],

$$\boldsymbol{X}_{mean}(t) = \frac{1}{P}\sum_{i=1}^{P} \boldsymbol{X}_i(t) \quad (16)$$

where P is the population size, and $\boldsymbol{X}_i(t)$ represents the position of every individual within the population during the present iteration.

2) EXPLOITATION PHASE

The hawks employ four different strategies during exploitation phase. The selection of strategy is based on the rabbit's escape energy and also the probability of its successfull escape. It is assumed that r represents the rabbit's chance of escaping, where r is a random variable that follows a uniform distribution between 0 and 1. If $r < 0.5$, the rabbit successfully escapes, and if $r \geq 0.5$, the rabbit fails to escape.

2.1) SOFT BESIEGE

This phase begins when the rabbit's energy satisfies $|E_{rabbit}| \geq 0.5$ and $r \geq 0.5$. This condition indicates that the rabbit possesses sufficient energy and a strong likelihood of escape. The hawk's position will be updated as shown in Eq. (17) [36],

$$\boldsymbol{X}(t+1) = \boldsymbol{X}_{rabbit}(t) - \boldsymbol{X}(t) - E_{rabbit}|J\boldsymbol{X}_{rabbit}(t) - \boldsymbol{X}(t)| \quad (17)$$

where hawk's position is denoted by $\boldsymbol{X}$, E_{rabbit} is the rabbit's energy, t current iteration, and J signifies the jump strength determined using Eq. (18) [36],

$$J = 2(1 - c_5) \quad (18)$$

where c_5 is a randomly generated value within the range $(0,1)$, which is updated at each iteration.

2.2) HARD BESIEGE

This phase occurs when $|E_{rabbit}| < 0.5$ and $r \geq 0.5$. This condition indicates that the rabbit is highly exhausted and also has low escape energy. However, it may still succeed in avoiding capture. The positions of the hawks are adjusted according to Eq. (19) [36].

$$\boldsymbol{X}(t+1) = \boldsymbol{X}_{rabbit}(t) - E_{rabbit}|\boldsymbol{X}_{rabbit}(t) - \boldsymbol{X}(t)| \quad (19)$$

2.3) SOFT BESIEGE WITH PROGRESSIVE RAPID DIVES

This phase is happened when $|E_{rabbit}| \geq 0.5$ and $r < 0.5$. This condition implies that the rabbit has enough energy to evade the attack but has a low probability of escaping successfully. The hawk continues to build a soft siege under these conditions. The hawks' positions are updated using Eq. (20) [36].

$$\boldsymbol{Y} = \boldsymbol{X}_{rabbit}(t) - E_{rabbit}|J\boldsymbol{X}_{rabbit}(t) - \boldsymbol{X}(t)| \quad (20)$$

Then, the hawks evaluate the potential outcomes of their actions by comparing them with previous movements If the outcomes are not beneficial, they will begin making abrupt, unpredictable, and swift movements. The hawks' positions are updated using Eq. (21) [36],

$$\boldsymbol{Z} = \boldsymbol{Y} + \boldsymbol{S} \times \boldsymbol{Levy}(dim) \quad (21)$$

where dim represents the number of dimensions in the optimization problem, $\boldsymbol{S}$ is a randomly generated vector of size $1 \times dim$, and the operation $\boldsymbol{S} \times \boldsymbol{Levy}(dim)$ is defined as the element-wise multiplication of $\boldsymbol{S}$ and $\boldsymbol{Levy}(dim)$. $Levy$ denotes the Levy flight function, which is computed using Eq. (22) [36],

$$Levy(\boldsymbol{x}) = 0.01 \times \frac{\boldsymbol{u} \times \sigma}{|\boldsymbol{v}|^{\frac{1}{\beta}}} \quad (22)$$

where $\boldsymbol{u}$ and $\boldsymbol{v}$ is a random value within the range $(0,1)$, β is a constant set to 1.5, and σ is calculated based on Eq. (23) [36].

$$\sigma = \left(\frac{\Gamma(1+\beta) \times sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2} \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}\right)} \right) \quad (23)$$

In this situation, the hawk's position is updated based on Eq. (24) [36],

$$\boldsymbol{X}(t+1) = \boldsymbol{A}, jika\ F(\boldsymbol{A}) < F(\boldsymbol{X}(t)) \quad (24)$$

where $\boldsymbol{A}$ includes Y and Z in Eq. (20) and Eq. (21) [36].

2.4) HARD BESIEGE WITH PROGRESSIVE RAPID DIVES

This stage is happened when $|E_{rabbit}| < 0.5$ and $r < 0.5$. This condition signifies that the rabbit has extremely low energy to evade the attack, prompting the hawks to execute a hard siege simultaneously. At this stage, the situation resembles a soft siege with progressive motion, but this time, the hawks strive to minimize the gap between their average position and the escaping rabbit as part of a tight siege effort. A new solution is calculated based on Eq. (25) [36]. If this solution does not yield good results, Eq. (21) will also be computed,

$$\boldsymbol{Y} = \boldsymbol{X}_{rabbit}(t) - E_{rabbit}|J\boldsymbol{X}_{rabbit}(t) - \boldsymbol{X}_{mean}(t)| \quad (25)$$

where $\boldsymbol{X}_{mean}(t)$ denotes the average position of the hawks in the current population. The hawks' positions are updated based on Eq. (24) [36].

F. EVALUATION METRICS

This study uses a confusion matrix to evaluate the model's effectiveness. A confusion matrix is a $K \times K$ table (with K being the number of classes) that illustrates how well the classification model performs. The entries in the confusion matrix are devided into four categories: True Positive (TP), when the model accurately identifies the positive class, False Positive (FP), when the model mistakenly classifies a negative instance as positive, True Negative (TN), when the model correctly identifies the negative class, and False Negative (FN) when the model fails to identify a positive instance [37]. In multi-class classification, an index k is introduced in TP_k, TN_k, FP_k, and FN_k to indicate the evaluation for a specific class. Since multi-class classification requires a balanced assessment, a macro-averaging approach is used, ensuring that each class contributes equally to the final evaluation, regardless of class distribution disparities [38][39]. The confusion matrix includes various evaluation metrics, such as accuracy, precision, recall, and F1-score can be derived as follows. The overall effevtiveness of the model is measured by accuracy, which calculates the percentage of correctly classified instances out of the total test data. Accuracy can be calculated as Eq. (26) [38].

$$Accuracy_k = \frac{TP_k + TN_k}{TP_k + FP_k + TN_k + FN_k}. \quad (26)$$

Macro precision is an evaluation metric that calculates the average precision for each class. Precision is a performance evaluation measure that determines how many positive classes are correctly classified. Macro precision can be calculated as Eq. (27) [38] .

$$Precision_k = \frac{TP_k}{TP_k + FP_k},$$

$$Macro\ Avg\ Precision\ (MAP) = \frac{\sum_{k=1}^{K} Precision_k}{K}. \quad (27)$$

Macro recall is an evaluation metric that calculates the average recall for each class. Recall, or sensitivity, measures the model's effectiveness in correctly identifying's all actual positive instances within the dataset. Macro recall can be calculated as Eq. (28) [38].

$$Recall_k = \frac{TP_k}{TP_k + FN_k},$$

$$Macro\ Avg\ Recall\ (MAR) = \frac{\sum_{k=1}^{K} recall_k}{K}. \quad (28)$$

The Macro F1-score is used to obtain the optimal combination of both precision and recall metrics. Macro F1-score can be calculated as Eq. (29) [38].

$$Macro\ F1 - Score = \frac{2 \times MAP \times MAR}{MAP + MAR}. \quad (29)$$

I. FEATURES IMPORTANCE

This study will also present the most important features of the dataset that significantly influence the classification process of liver disease indications in an individual. The determination of important features will also be conducted using the XGBoost algorithm. This algorithm can identify the best features based on the gain value of each feature produced during the training process. Features with high gain values indicate they are highly informative and make a significant contribution to improving the model's performance [40]. The calculation of gain is outlined in Eq. (11).

III. RESULT

This section contains the results of the applied data preprocessing, the evaluation of the proposed model, and its comparison with other models. Additionally, this section will also highlight the features that have the most influence in classifying the stages of liver disease in an individual.

A. DATA PREPROCESSING

Data preprocessing began with the removal of the 'ID' feature, as the patient number does not impact the liver condition in the classification process. Next, label encoding was applied to the categorical feature, "Sex" and "Category", this was done so that the data in the categorical feature could be used in the classifier model. Additionally, The outlier detection results show that eight features in the dataset contain outliers, including ALB and ALP (3 each), ALT (19), AST (45), BIL (26), CREA (5), GGT (29), and PROT (4). Further analysis revealed that these outliers still hold relevant information about medical data variations. Therefore, they were retained in this study to preserve valuable insights into high and low value characteristics in medical data. missing values handling was applied to the 'ALB,' 'ALP,' 'AST,' 'CHOL,' and 'PROT' features. In addition, this study applies median imputation due to its robustness against outliers. The affected features are 'ALB', 'ALP', 'AST', 'CHOL', and 'PROT', with missing values distributed as follows: 1 in 'ALB', 'ALT', and 'PROT'; 18 in 'ALP'; and 10 in 'CHOL'.

Data imbalance was addressed using the SMOTE-NC method after data normalization. The resampled data is presented in TABLE 3. After resampling, the dataset was divided into two subsets: 80% for training and 20% for testing using Sklearn's train_test_split function, chosen after testing ratios like 90:10 and 85:15 on XGBoost, SVM, and RF baseline model, with 80:20 yielding the best performance. The ratio selection based on baseline models ensures fair and consistent comparisons. Each model, whether baseline or combined, was run 25 times to assess stability based on its mean and standard deviation.

TABLE 3
Data resampling

Class	Amount of data		Percentage	
	Original	Resampling	Original	Resampling
0	540	540	87.80%	25%
1	30	540	4.88%	25%
2	24	540	3.90%	25%
3	21	540	3.42%	25%

B. XGBOOST-HHO PARAMETER SETTING

The XGBoost algorithm was optimized with HHO for hyperparameter tuning. The process began by defining the accuracy of the XGBoost model as the objective function for HHO. In the tuning process, the maximum iterations were set to 25, and the hawk population was initialized to 40. The hawk population was selected based on the average best accuracy from 25 model repetitions. The outcomes are presented in TABLE 4.

As observed in TABLE 4, an increase in population size leads to noticeable changes, the computation time for the XGBoost-HHO algorithm also increases. The model was able to classify the data with the best accuracy when the population was initialized at 40 and 100. However, with a hawk population of 100, the computation time was longer. Therefore, this study applied a hawk population initialization of 40. The relatively low standard deviation at this point indicates a small range of data variation and a stable distribution around the average.

TABLE 2
XGBoost-HHO Accuracy with Various Hawk Population

Hawks	Avg. Accuracy (%)	Std. Accuracy	Avg. Time Computation
10	0.9480	0.0063	**27s**
20	0.9745	0.0084	45s
30	0.9907	0.0066	66s
40	**0.9934**	0.0021	70s
50	0.9930	**0.0009**	109s
60	0.9927	0.0021	118s
70	0.9928	0.0009	180s
80	0.9930	0.0011	194s
90	0.9931	0.0013	211s
100	**0.9934**	0.0021	240s

C. RESULT OF XGBOOST-HHO

Each best fitness from the HHO algorithm provides a different hyperparameter combination for XGBoost. Several hyperparameter combinations were able to deliver optimal performance, one of which is listed in TABLE 5. The performance of the XGBoost-HHO model with these hyperparameters during the testing phase is detailed in TABLE 6.

TABLE 5
Hyperparameter chosen by XGBoost-HHO

Hyperparameter	*Value*
Learning rate	0.8
n_estimators	680
Max_depth	3
Gamma	0.1
Reg_lambda	0.5
Subsample	0.9
Colsample_bytree	0.75

TABLE 6
XGBoost-HHO evaluation performance

Phase	Accuracy	MAP	MAR	Macro F1-score
Training	1.0000	1.0000	1.0000	1.0000
Testing	0.9977	0.9976	0.9956	0.9977

D. COMPARATIVE RESULT WITH OTHER MODELS

A comparative analysis of the proposed XGBoost-HHO model's performance with various other models is conducted using the average accuracy, MAP, MAR, and Macro F1-score over 25 runs, as presented in TABLE 7. Additionally, the standard deviation of all evaluation metrics and the average computation time for various models are presented in TABLE 8 to illustrate the spread of performance values from their averages.

This study compares two other ML algorithms, RF and SVM. These models were chosen due to their strong performance in previous studies on similar datasets and their frequent use as optimal models in various disease diagnosis cases [8], [41]. Both models were also trained on the dataset that underwent the same preprocessing phase and optimized with HHO for hyperparameter tuning. One of the optimal hyperparameter configurations for the RF model was obtained in the 12th iteration, which includes n_estimators = 250,

max_depth = 14, min_samples_split = 2, min_samples_leaf = 1, and max_features = 0.1. An optimal hyperparameter configuration for the SVM model was obtained in the 14th iteration, consisting of $C = 77$, $\gamma = 0.58$, $r = 0.35$, and degree = 4.

In addition, several other algorithms were also applied for hyperparameter optimization on XGBoost, including Particle Swarm Optimization (PSO) and Bat Algorithm (BA). PSO and BA were applied for XGBoost hyperparameter optimization to benchmark HHO's performance in accuracy and stability. While PSO relies on velocity updates and BA mimics echolocation, HHO uses adaptive escaping and rapid dives, making convergence analysis essential [42], [43], [44]. All algorithms shared the same population size and maximal iterations. Each model was run 25 times, and the average accuracy, MAP, MAR, and Macro F1-score were calculated to validate effectiveness and consistency.

TABLE 7
Evaluation Model Comparison using accuracy, MAP, MAR, Macro F1-score

Classifier	Accuracy	MAP	MAR	Macro F1-Score
SVM	0.8402	0.8445	0.8397	0.8388
RF	0.9902	0.9901	0.9901	0.9899
XGBoost	0.9897	0.9899	0.9908	0.9902
SVM-HHO	0.9836	0.9838	0.9839	0.9833
RF-HHO	0.9910	0.9911	0.9913	0.9911
XGBoost-PSO	0.9915	0.9916	0.9919	0.9916
XGBoost-BA	0.9512	0.9524	0.9512	0.9508
XGBoost-HHO	**0.9934**	**0.9934**	**0.9934**	**0.9933**

Based on TABLE 7, it can be seen that XGBoost-HHO improves the performance of standard XGBoost that applied with default hyperparameters from TABLE 1. XGBoost-HHO achieved an increase of 0.0037 in accuracy, 0.0035 in MAP, 0.0026 in MAR, and 0.0031 in F1-score compared to standard XGBoost. Hyperparameter optimization using the HHO algorithm applied to XGBoost, RF, and SVM successfully improved the average performance in terms of accuracy, MAP, MAR, and Macro F1-score for each standard model. The proposed XGBoost-HHO model also outperformed the performance of other combined models in all four evaluation metrics. The higher accuracy indicates that more instances were correctly classified. An increase in MAP reflects the model's ability to prioritize relevant classes, reducing false positives. A higher MAR shows improved sensitivity in detecting all instances of each class, minimizing missed cases. The improved Macro F1-Score confirms a balanced performance between precision and recall across all classes. The training phase performance results of each standard model and comparison combinations yielded a 100% score. These results demonstrate that hyperparameter optimization using the HHO algorithm not only enhanced predictive accuracy but also improved the fairness and consistency of classification outcomes, with the XGBoost-HHO model achieving the best overall performance and lower overfitting risk.

TABLE 8
Std. Comparison of Testing Performance

Classifier	Std. of Testing			
	Accuracy	MAP	MAR	Macro F1-Score
SVM	0.0266	0.0251	0.0268	0.0268
RF	0.0066	0.0068	0.0067	0.0069
XGBoost	0.0047	0.0047	0.0043	0.0040
SVM-HHO	0.0173	0.0171	0.0172	0.0172
RF-HHO	0.0028	0.0028	0.0028	0.0028
XGBoost-PSO	**0.0012**	**0.0012**	**0.0012**	**0.0012**
XGBoost-BA	0.0379	0.0362	0.0385	0.0388
XGBoost-HHO	0.0021	0.0024	0.0023	0.0025

Based on the comparison of the standard deviation of the average testing performance in TABLE 8, XGBoost-PSO yields the most consistent performance with the smallest standard deviation across all evaluation metrics. On the other hand, the largest variation is observed in the SVM-HHO combination model. The proposed XGBoost-HHO model shows the second-highest performance consistency after XGBoost-PSO. This indicates that HHO optimization with XGBoost also contributes to maintaining performance stability across all evaluators.

TABLE 9
Time Computation

Classifier	Avg. Time Computation (s)	Std. Time Computation
SVM	0.1099	0.0356
RF	0.5025	0.2603
XGBoost	0.4216	0.1083
SVM-HHO	111.4031	19.8744
RF-HHO	233.2904	202.312
XGBoost-PSO	76.1101	106.7995
XGBoost-BA	149.4891	143.7964
XGBoost-HHO	**70.2252**	14.7185

Based on the computation time for each model shown in TABLE 9, SVM has the fastest average processing time at 0.1099, followed by XGBoost at 0.4216 seconds, and RF at 0.5025 seconds. Combined models with optimization algorithms require much longer computation times compared to the base models. It can be seen that the RF-HHO model has the highest average computation time, around 233 seconds. Meanwhile, XGBoost-HHO has the lowest average computation time among the other combined models, around 70 seconds. This demonstrates its efficiency as an optimization model.

Based on the provided accuracy in TABLE 7, the XGBoost-HHO model demonstrates the best performance compared to other models. To ensure that this difference is not merely due to chance or random variation, a Wilcoxon test was conducted as a statistical significance analysis presented in

TABLE 10. The test results show that all models compared to XGBoost-HHO have a p-value of less than 0.05, indicating that the performance difference is statistically significant. Additionally, the comparison between XGBoost-HHO and the SVM, XGB, and XGB-BA models yields a W-statistic of 0, which indicates that in every comparison, XGBoost-HHO consistently outperforms the other models without a single case where another model achieves higher accuracy. Thus, this difference can be categorized as highly significant.

TABLE 10
Wilcoxon Signed Rank Test Results

Compared Models	W-statistic	Z-value	p-value	Significance
SVM	0	-4.6212	0.0000	$p < 0.01$
RF	22	-3.0853	0.0010	$p < 0.05$
XGBoost	0	-4.6212	0.0000	$p < 0.01$
SVM-HHO	1	-4.4753	0.0000	$p < 0.01$
RF-HHO	36	-2.3951	0.0083	$p < 0.05$
XGBoost-PSO	20	-3.1915	0.0007	$p < 0.05$
XGBoost-BA	0	-4.6212	0.0000	$p < 0.01$

Overall, base models provide higher stability and time efficiency compared to combination models because optimization algorithms require more time to search for optimal hyperparameter configurations. This is reflected in the superior performance of each evaluation metric provided by the combined models. The proposed XGBoost-HHO model offers a good compromise between computation time and hyperparameter exploration ability, delivering the highest performance and shortest processing time compared to other combined models.

G. FEATURE IMPORTANCE

The important features that have a significant impact on the classification process based on their gain values from the XGBoost-HHO algorithm are presented in FIGURE 2. Based on FIGURE 2, the most influential feature in classifying the stages of liver disease in the given dataset is ALT, with the highest Feature score (F score). This suggests that ALT plays a crucial role in distinguishing between different liver disease stages. The next most important features are PROT, CGT, CREA, and CHOL, indicating their strong relevance in the classification process. These features contribute significantly to the model's decision-making.

Furthermore, features such as CHE, BIL, AST, ALP, and ALB also have notable influence but to a lesser extent than the top-ranked features. Meanwhile, Sex and Age show the lowest F-score values, with Age being the least impactful feature in the classification task. This implies that demographic factors like Age and Sex have minimal influence compared to biochemical indicators.

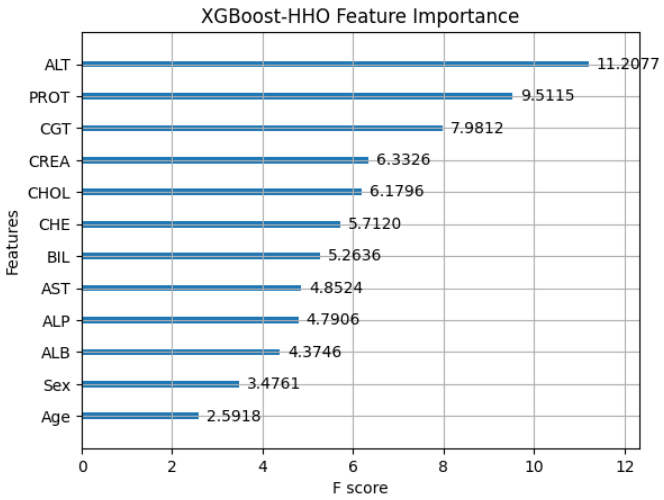

FIGURE 2. Gain of each Feature

IV. DISCUSSION

This study aims to develop an early diagnosis model for an individual's liver condition, with the hope of reducing the high mortality rate caused by chronic liver disease, specifically cirrhosis. The model is built using one of the Machine Learning algorithms, XGBoost, with a focus on optimizing the algorithm to improve classification accuracy and efficiency. The optimization is performed on the hyperparameter tuning process using the HHO algorithm.

The study uses the Hepatitis C dataset, which is the leading cause of cirrhosis, obtained from the Kaggle repository. The dataset consists of 615 observations, 14 features, and four target classes. Before the classification process, the dataset is preprocessed through several steps, including label encoding for categorical features, enabling the classification algorithm to interpret the observation values properly. Data normalization is then applied to maintain its distribution, followed by resampling with SMOTE-NC to handle imbalanced data.

Based on the results presented in TABLE 7, hyperparameter optimization can improve the classification performance of the standard models for every evaluation metric. This is evident as the combinations of SVM-HHO and RF-HHO improve the performance of standard SVM and RF models based on MAP, MAR, and Macro F1-score evaluations. Similarly, all combinations of XGBoost with various hyperparameter optimizations also enhance the performance of standard XGBoost. The most superior improvement in performance is achieved by the XGBoost-HHO model. The smallest average difference in training and testing accuracy is also provided by XGBoost-HHO, indicating that the proposed model has the lowest overfitting level among the others. This suggests that the XGBoost-HHO model can generalize well to new data, which in this study is represented by the test data. These findings align with studies [31], [45], which applied hyperparameter tuning to XGBoost. The optimization of hyperparameters in those studies also improved model performance and reduced the overfitting level of standard XGBoost.

TABLE 3
Some previous studies with HCV dataset comparison

Author and reference	Year	Method	Accuracy (%)
Li et. al. [41]	2022	Cascade RF-LR (optimized by ABC algorithm)	96.19%
Singh et. al. [11]	2022	XGBoost optimized with Monotonic Cosntraints and feature interaction constraints	98.60%
Alizargar et. al. [21]	2023	XGBoost	98.40%
Proposed model	**-**	**XGBoost-HHO**	**99.34%**

In addition, this study presents a comparison with previous research that applied similar datasets using various different models. Li et. al. employed a cascade two-stage method combining RF and Logistic Regression (LR) algorithms. In addition, ABC algorithm was used to optimize searches to obtain the essential threshold value to separate both models. The new component of this research is its examination of the unbalanced data that exists in clinical data [41]. Singh et. al. proposed an optimization method for XGBoost algorithm using monotonic cosntraints and feature interaction constraints. The proposed method achieving a prediction accuracy of 98.6%, outperforms LR, Light Gradient Boosting Machine (LGBM), DT, and SVM-RBF [11]. Besides, Alizargar et. al. (2023) employ various ML to predict hepatitis, including SVM, KNN, LR, DT, XGBoost and ANN. The results showed that XGBoost and SVM achieving the highest accuracy of 95% and Area Under the Curve (AUC) of 98.4%.

As shown in TABLE 11, the XGBoost-HHO model achieves the highest accuracy (99.34%), surpassing previous studies on this dataset. The 0.74%–3.15% improvement in accuracy suggests that HHO effectively optimizes hyperparameters to enhance model generalization and minimize classification errors. Overall, XGBoost-HHO proves to be one of the superior solutions for improving cirrhosis classification performance with the given dataset. From a clinical perspective, this improvement is crucial, as higher accuracy ensures more precise classification of liver cirrhosis stages, enabling timely interventions for severe cases and reducing complications. The proposed model demonstrates the potential for more reliable early diagnosis, leading to better treatment planning and patient outcomes. In a longitudinal context, periodic retraining is necessary to ensure that the model remains relevant to the latest patterns in the data for maintaining model accuracy and adaptability. Monitoring performance metrics can help optimize the retraining for long-term effectiveness.

However, the XGBoost-HHO model still has limitations, including potential dataset bias that may affect generalization. The dataset that used in this study may not fully represent the diversity of liver cirrhosis cases. Diverse datasets with variations in medical records and differences in diagnostic equipment are needed to ensure model reliability across different conditions. Moreover, broader validation strategies such as cross validation may improve the roubstness of the model. Besides that, although HHO improves accuracy, it increases computational complexity and risks premature convergence, which can contribute to overfitting. Implementing Opposition-Based Learning (OBL) or Brownian Mutation can enhance convergence and efficiency without compromising accuracy, helping to further mitigate overfitting and improve model generalization [36], [44].

Algorithm-based models can rapidly analyze large datasets, reduce subjectivity, and detect complex patterns that humans may overlook, thereby improving clinical assessment efficiency [8]. However, external validation and clinical testing are crucial for widespread adoption. This is necessary because the interpretability of the model must be improved for clinical use. This model has the potential to assist physicians in decision-making, enhance diagnostics, and support Clinical Decision Support System (CDSS). Additionally, Integrating it into Electronic Health Records (EHR) allows direct access to cirrhosis severity predictions.

V. CONCLUSION

This study aims to build an early diagnosis model based on Machine Learning using Hepatitis C data from the Kaggle repository, consisting of 615 observations and 14 variables. The data provides blood test results, which in this study are mapped into four classes representing stages of liver condition: healthy, hepatitis, fibrosis, and cirrhosis. Classification is performed using the XGBoost algorithm. The numerous hyperparameters in XGBoost would require significant time if determined manually. Therefore, this study proposes optimization using the HHO algorithm for hyperparameter tuning. The research results show that XGBoost-HHO achieved an average accuracy, MAP and MAR of 99.34% and achieved an average macro f1-score performance of 99.33% for 25 trials. This performance outperformed other models compared in this study and is competitive with previous research using similar datasets. The XGBoost-HHO model also achieved the lowest overfitting rate among the other models. The average processing time for XGBoost-HHO is 70.2252 seconds, which is competitive compared to other models. This processing time was achieved with the population size and maximum iterations initialized at 40 and 25, respectively. Feature importance analysis showed that the features ALT, PROT, and GGT contributed the most to the classification, with gain values of 11.21, 9.51, and 7.98, respectively. Based on the results, it can be concluded that the proposed XGBoost-HHO model is capable of improving the classification quality of the standard XGBoost model in this case, both in terms of accuracy and time efficiency. The performance of this model also outperformed other models compared in this study and several previous studies. In the future, the proposed XGBoost-HHO model can still be developed by enhancing the HHO algorithm's search quality for better performance. However, dataset bias and

computational complexity remain challenges. Broader validation and techniques like OBL can improve reliability. Enhancing interpretability supports clinical use, CDSS, and EHR integration for cirrhosis severity prediction.

REFERENCES

[1] R. Islam, A. Sultana, and M. N. Tuhin, "A comparative analysis of machine learning algorithms with tree-structured parzen estimator for liver disease prediction," *Healthc. Anal.*, vol. 6, no. August, p. 100358, 2024, doi: 10.1016/j.health.2024.100358.

[2] D. Badvath, A. safali Miriyala, S. chaitanya K. Gunupudi, and P. V. K. Kuricheti, "ONBLR: An effective optimized ensemble ML approach for classifying liver cirrhosis disease," *Biomed. Signal Process. Control*, vol. 89, no. January, p. 105882, 2024, doi: 10.1016/j.bspc.2023.105882.

[3] H. Enomoto *et al.*, "Transition in the etiology of liver cirrhosis in Japan: a nationwide survey," *J. Gastroenterol.*, vol. 55, no. 3, pp. 353–362, 2020, doi: 10.1007/s00535-019-01645-y.

[4] Z. Wang *et al.*, "Clinical prediction of HBV-associated cirrhosis using machine learning based on platelet and bile acids," *Clin. Chim. Acta*, vol. 551, no. July, p. 117589, 2023, doi: 10.1016/j.cca.2023.117589.

[5] CDC, "Underlying Cause of Death, 2018-2022," 2024.

[6] World Health Organization (WHO), "Liver Cirrhosis, age-standardized death rates (15+), per 100.000 population," 2024.

[7] X. Kuang *et al.*, "Transcriptomic and Metabolomic Analysis of Liver Cirrhosis," pp. 922–932, 2024, doi: 10.2174/1386207326666230717094936.

[8] I. Hanif and M. M. Khan, "Liver Cirrhosis Prediction using Machine Learning Approaches," *2022 IEEE 13th Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2022*, no. 2019, pp. 28–34, 2022, doi: 10.1109/UEMCON54665.2022.9965718.

[9] R. Bhardwaj, R. Mehta, and P. Ramani, "A Comparative Study of Classification Algorithms for Predicting Liver Disorders BT - Intelligent Computing Techniques for Smart Energy Systems," A. Kalam, K. R. Niazi, A. Soni, S. A. Siddiqui, and A. Mundra, Eds., Singapore: Springer Singapore, 2020, pp. 753–760.

[10] Y. Shinde, A. Kenchappagol, and S. Mishra, "Comparative Study of Machine Learning Algorithms for Breast Cancer Classification BT - Intelligent and Cloud Computing," D. Mishra, R. Buyya, P. Mohapatra, and S. Patnaik, Eds., Singapore: Springer Nature Singapore, 2022, pp. 545–554.

[11] K. R. Singh, R. Gupta, R. K. Kadian, and R. Singh, "An Optimized XGBoost approach for Predicting Progression of Hepatitis C using Hyperparameter Tuning and Feature Interaction Constraint," *2022 2nd Asian Conf. Innov. Technol. ASIANCON 2022*, pp. 1–8, 2022, doi: 10.1109/ASIANCON55314.2022.9909086.

[12] A. Ogunleye and Q. G. Wang, "XGBoost Model for Chronic Kidney Disease Diagnosis," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 17, no. 6, pp. 2131–2140, 2020, doi: 10.1109/TCBB.2019.2911071.

[13] P. Zhang, Y. Jia, and Y. Shang, "Research and application of XGBoost in imbalanced data," *Int. J. Distrib. Sens. Networks*, vol. 18, no. 6, 2022, doi: 10.1177/15501329221106935.

[14] B. Bischl *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 13, no. 2, pp. 1–43, 2023, doi: 10.1002/widm.1484.

[15] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020, doi: 10.1016/j.neucom.2020.07.061.

[16] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Futur. Gener. Comput. Syst.*, vol. 97, pp. 849–872, 2019, doi: 10.1016/j.future.2019.02.028.

[17] L. Duan, M. Wu, and Q. Wang, "Predicting the CPT-based pile set-up parameters using HHO-RF and WOA-RF hybrid models," *Arab. J. Geosci.*, vol. 15, no. 7, 2022, doi: 10.1007/s12517-022-09843-4.

[18] M. M. K. Kazemi, Z. Nabavi, and D. J. Armaghani, "A novel Hybrid XGBoost Methodology in Predicting Penetration Rate of Rotary Based on Rock-Mass and Material Properties," *Arab. J. Sci. Eng.*, vol. 49, no. 4, pp. 5225–5241, 2024, doi: 10.1007/s13369-023-08360-0.

[19] M. A. Kalas, L. Chavez, M. Leon, P. T. Taweesedt, and S. Surani, "Abnormal liver enzymes: A review for clinicians," *World J. Hepatol.*, vol. 13, no. 11, pp. 1688–1698, 2021, doi: 10.4254/wjh.v13.i11.1688.

[20] K. Sumwiza, C. Twizere, G. Rushingabigwi, P. Bakunzibake, and P. Bamurigire, "Enhanced cardiovascular disease prediction model using random forest algorithm," *Informatics Med. Unlocked*, vol. 41, no. July, p. 101316, 2023, doi: 10.1016/j.imu.2023.101316.

[21] A. Alizargar, Y. L. Chang, and T. H. Tan, "Performance Comparison of Machine Learning Approaches on Hepatitis C Prediction Employing Data Mining Techniques," *Bioengineering*, vol. 10, no. 4, 2023, doi: 10.3390/bioengineering10040481.

[22] S. Raschka, Y. (Hayden) Liu, and V. Mirjalili, *Machine Learning with PyTorch and Scikit Learn*. 2022.

[23] M. Ahmed Ouameur, M. Caza-Szoka, and D. Massicotte, "Machine learning enabled tools and methods for indoor localization using low power wireless network," *Internet of Things (Netherlands)*, vol. 12, p. 100300, 2020, doi: 10.1016/j.iot.2020.100300.

[24] H. Benhar, A. Idri, and J. L Fernández-Alemán, "Data preprocessing for heart disease classification: A systematic literature review.," *Comput. Methods Programs Biomed.*, vol. 195, 2020, doi: 10.1016/j.cmpb.2020.105635.

[25] E. Pusporani, S. Qomariyah, and I. Irhamah, "Klasifikasi Pasien Penderita Penyakit Liver dengan Pendekatan Machine Learning," *Inferensi*, vol. 2, no. 1, p. 25, 2019, doi: 10.12962/j27213862.v2i1.6810.

[26] Q. H. Doan, S. H. Mai, Q. T. Do, and D. K. Thai, "A cluster-based data splitting method for small sample and class imbalance problems in impact damage classification[Formula presented]," *Appl. Soft Comput.*, vol. 120, p. 108628, 2022, doi: 10.1016/j.asoc.2022.108628.

[27] E. C. Gök and M. O. Olgun, "SMOTE-NC and gradient boosting imputation based random forest classifier for predicting severity level of covid-19 patients with blood samples," *Neural Comput. Appl.*, vol. 33, no. 22, pp. 15693–15707, 2021, doi: 10.1007/s00521-021-06189-y.

[28] A. Al Ahad, B. Das, M. R. Khan, N. Saha, A. Zahid, and M. Ahmad, "Multiclass liver disease prediction with adaptive data preprocessing and ensemble modeling," *Results Eng.*, vol. 22, no. February, p. 102059, 2024, doi: 10.1016/j.rineng.2024.102059.

[29] R. Valarmathi and T. Sheela, "Heart disease prediction using hyper parameter optimization (HPO) tuning," *Biomed. Signal Process. Control*, vol. 70, no. March, p. 103033, 2021, doi: 10.1016/j.bspc.2021.103033.

[30] S. Li and X. Zhang, "Research on orthopedic auxiliary classification and prediction model based on XGBoost algorithm," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 1971–1979, 2020, doi: 10.1007/s00521-019-04378-4.

[31] K. Budholiya, S. K. Shrivastava, and V. Sharma, "An optimized XGBoost based diagnostic system for effective prediction of heart disease," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 7, pp. 4514–4523, 2022, doi: 10.1016/j.jksuci.2020.10.013.

[32] S. Singh, S. K. Patro, and S. K. Parhi, "Evolutionary optimization of machine learning algorithm hyperparameters for strength prediction of high-performance concrete," *Asian J. Civ. Eng.*, vol. 24, no. 8, pp. 3121–3143, 2023, doi: 10.1007/s42107-023-00698-y.

[33] D. A. Anggoro and S. S. Mukti, "Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 6, pp. 198–207, 2021, doi: 10.22266/ijies2021.1231.19.

[34] K. D. Chaudhuri and B. Alkan, "A hybrid extreme learning machine model with harris hawks optimisation algorithm: an optimised model for product demand forecasting applications," *Appl. Intell.*, vol. 52, no. 10, pp. 11489–11505, 2022, doi: 10.1007/s10489-022-03251-7.

[35] S. Gupta, K. Deep, A. A. Heidari, H. Moayedi, and M. Wang, "Opposition-based learning Harris hawks optimization with advanced transition rules: principles and analysis," *Expert Syst. Appl.*, vol. 158, p. 113510, 2020, doi: 10.1016/j.eswa.2020.113510.

[36] H. Kang, R. Liu, Y. Yao, and F. Yu, "Improved Harris hawks optimization for non-convex function optimization and design optimization problems," *Math. Comput. Simul.*, vol. 204, pp. 619–

639, 2023, doi: 10.1016/j.matcom.2022.09.010.
[37] K. Ali, Z. A. Shaikh, A. A. Khan, and A. A. Laghari, "Multiclass skin cancer classification using EfficientNets – a first step towards preventing skin cancer," *Neurosci. Informatics*, vol. 2, no. 4, p. 100034, 2022, doi: 10.1016/j.neuri.2021.100034.
[38] C. A. D. Lestari, S. Anam, and U. Sa'adah, *Tomato Leaf Disease Classification with Optimized Hyperparameter: A DenseNet-PSO Approach*, no. Icamsac 2023. Atlantis Press International BV, 2024. doi: 10.2991/978-94-6463-413-6_23.
[39] M. Grandini, E. Bagli, and G. Visani, "Metrics for Multi-Class Classification: an Overview," pp. 1–17, 2020, [Online]. Available: http://arxiv.org/abs/2008.05756
[40] Z. Jiang, J. Che, M. He, and F. Yuan, "A CGRU multi-step wind speed forecasting model based on multi-label specific XGBoost feature selection and secondary decomposition," *Renew. Energy*, vol. 203, no. November 2022, pp. 802–827, 2023, doi: 10.1016/j.renene.2022.12.124.
[41] T. H. S. Li, H. J. Chiu, and P. H. Kuo, "Hepatitis C Virus Detection Model by Using Random Forest, Logistic-Regression and ABC Algorithm," *IEEE Access*, vol. 10, no. August, pp. 91045–91058, 2022, doi: 10.1109/ACCESS.2022.3202295.
[42] L. Abualigah, "Particle Swarm Optimization: Advances, Applications, and Experimental Insights," *Comput. Mater. Contin.*, vol. 82, no. 2, pp. 1539–1592, 2025, doi: 10.32604/cmc.2025.060765.
[43] M. Z. Rehman, M. Aamir, N. M. Nawi, A. Khan, S. A. Lashari, and S. Khan, "An Optimized Neural Network with Bat Algorithm for DNA Sequence Classification," *Comput. Mater. Contin.*, vol. 73, no. 1, pp. 493–511, 2022, doi: 10.32604/cmc.2022.021787.
[44] M. Shehab *et al.*, "Harris Hawks Optimization Algorithm: Variants and Applications," *Arch. Comput. Methods Eng.*, vol. 29, no. 7, pp. 5579–5603, 2022, doi: 10.1007/s11831-022-09780-1.
[45] S. Dalal, E. M. Onyema, and A. Malik, "Hybrid XGBoost model with hyperparameter tuning for prediction of liver disease with better accuracy," *World J. Gastroenterol.*, vol. 28, no. 46, pp. 6551–6563, 2022, doi: 10.3748/wjg.v28.i46.6551.

AUTHOR BIOGRAPHY

Lista Tri Nalasari was born in Kediri, East Java, Indonesia in 2000. She received her bachelor degree in Mathematics Education from State University of Malang, Indonesia, in 2023. She is continuing his studies at the Mathematics Department, Faculty of Mathematics and Natural Sciences, Brawijaya University, since 2023. Currently, she dedicates his efforts to conducting in the field of data science that aligns with his academic interests, with a focus on exploring how data can be utilized to solve real-world problems. Involvement in research has provided valuable experience in working with data, especially in contexts that support societal benefits. She can be contacted at email: listatrinalasari@gmail.com.

Syaiful Anam received a Doctor of Natural Science and Mathematics degree from Yamaguchi University, Japan in 2015. He also received his Bachelor's Degree in Mathematics from Brawijaya University, Indonesia in 2001 and his Master Degree from Sepuluh Nopember Institute of Technology, Indonesia in 2006. He is currently an assistant professor at Mathematics Departmen, Brawijaya University, Malang, Indonesia. His research includes span across various fields including data science, computational intelligence, machine learning, digital image processing, and computer vision. He has actively contributed to scientific research, with published over 35 papers in international journals and conferences. He can be contacted at email: syaiful@ub.ac.id.

Nur Shofianah was born in Gresik, East Java, Indonesia in 1984. She received the Ph.D. degree in Mathematics from Graduate School of Natural Science and Technology, Kanazawa University, Japan, in 2014. She received the bachelor and master degree in 2007 and 2009 from Sepuluh Nopember Institute of Technology, Indonesia.

Since 2010, she is a lecturer in the Faculty of Mathematics and Natural Sciences, Brawijaya University, Indonesia. Her research interests are in numerical methods for PDE, dynamical analysis and optimal control of mathematical models. She is the author of about 20 articles. Nur Shofianah Ph.D. is member of IndoMS (Indonesian Mathematical Society) and IBMS (Indonesian Bio-Mathematical Society).

.