Journal of
Information Systems Engineering
and Business Intelligence

Vol.10, No.2, June 2024
Available online at: http://e-journal.unair.ac.id/index.php/JISEBI

Hepatitis Identification using Backward Elimination and Extreme Gradient Boosting Methods

Jasman Pardede [1)*], Desita Nurrohmah [2)]

[1)2)] *Department of Informatics, Faculty of Industrial Technology, National Institute of Technology, Bandung, Indonesia*
[1)]jasman@itenas.ac.id, [2)]desitanurrohmah89@gmail.com

Abstract

Background: Hepatitis is a contagious inflammatory disease of the liver and is a public health problem because it is easily transmitted. The main factors causing hepatitis are viral infections, disease complications, alcohol, autoimmune diseases, and drug effects. Some hepatitis variants such as B, C, and D can also cause liver cancer if left untreated.
Objective: This research aims to determine the effect of Backward Elimination feature selection on the performance of hepatitis disease identification compared to cases where Backward Elimination is not applied.
Methods: XGBoost classification, capable of handling machine learning problems, was utilized. Additionally, Backward Elimination was used as a featured selection to increase accuracy by reducing the number of less important features in the data classification process.
Results: The results for training XGBoost model with Backward Elimination, and applying Random Search for hyperparameter optimization, achieved an accuracy of 98.958% at 0.64 seconds. This performance was better than using Bayesian search, which produced the same accuracy of 98.958% but required a longer training time of 0.70 seconds.
Conclusion: The use of features obtained from Backward Elimination process as well as the use of feature average values for missing value treatment, produced an accuracy of 98.958%. Meanwhile, the precision in training XGBoost model with hyperparameter Bayesian search achieved accuracy, recall, and F1 score of 98.934%, 98.934%, and 98.934%, respectively. Consequently, the use of Backward Elimination in XGBoost model led to faster training, improved accuracy, and decreased overfitting.

Keywords: Hepatitis, Backward Elimination, XGBoost, Bayesian Search, Random Search

Article history: Received 2 August 2023, first decision 16 January 2024, accepted 13 March 2024, available online 28 June 2024

I. INTRODUCTION

Liver disease is often considered a silent killer because it tends not to show symptoms. Hepatitis is a contagious inflammatory liver disease that causes public health problem [1]. According to WHO, more than 3,000 people die every day from liver disease caused by hepatitis virus. Hepatitis variants such as B, C, and D can lead to liver cancer when left untreated [2]. In 2019, WHO reported that 78,000 deaths occurred worldwide due to complications of acute hepatitis A to E infection [3]. The main causes of hepatitis are viral infections, disease complications, alcohol, autoimmune diseases, and the effects of drugs [4].

The WHO's Regional Office in Southeast Asia reported that the prevalence of hepatitis B in Indonesia reached 7.1% or around 18 million cases, while hepatitis C reached 2.34% or around 6 million cases. The prevalence rate is the highest among other Southeast Asian countries after Myanmar and Thailand. Additionally, the 2018 Basic Health Research Report (riskesdas) stated that the highest prevalence of hepatitis cases in Indonesia occurred in children aged 5 to 14 years [5].

To avoid further transmission, there are various types of hepatitis which have different causes, symptoms, and treatments. This implies that the results of hepatitis diagnosis need to be known in order to administer proper treatment. Typically, adequate inspection is necessary to be carried out in a short time, while laboratory tests are usually required when diagnosing the disease. However, disease diagnosis based on laboratory results produces errors in the initial analysis or in determining the disease suffered by the patient. Based on the research conducted by [6], pre-analytical errors contribute to 46–68.2% of the total errors in laboratory tests, with incorrect patient identification accounting for 26.8% of these errors. Therefore, laboratory test results are necessary to be validated, specifically in hepatitis research.

* Corresponding author

ISSN 2443-2555 (online) 2598-6333 (print)
doi: http://dx.doi.org/10.20473/jisebi.10.2.302-313

It is crucial to be aware that the validation based on factors that influence the disease can be predicted using machine learning.

Machine learning has an impact on the rapid development of technology in various fields, including the medical field. In general, the technology uses computers to learn from data and make predictions. According to [7], predictions with high accuracy make it easier for explorers to evaluate an experiment quickly and accurately.

Following machine learning technology, Extreme Gradient Boosting (XGBoost) is an evolution of Gradient Tree Boosting algorithm based on ensemble algorithms, which can handle machine learning problems efficiently. XGBoost excels in solving a variety of classification, regression, and ranking problems. In addition, the algorithm has also succeeded in becoming one of the most popular methods in machine learning [8], particularly in identifying hepatitis disease, where precise predictions are essential for effective and accurate treatment.

In research conducted by [9], seven different machine learning algorithms were utilized, including XGBoost, used to predict liver disease. The exploration performed by [10] predicted hepatitis B Surface Antigen Seroclearance using several machine learning algorithms such as logistic regression, decision trees, random forests, and XGBoost. Furthermore, [11] predicted heart disease using machine learning such as multi-layer perceptron Random Forest, Decision Tree, and XGBoost. Similarly, George Obaido et al. [12] performed Diagnosis using various machine-learning methods such as Decision Trees, Logistic Regression, SVM, Random Forest, XGBoost, and AdaBoost.

To improve data processing efficiency and machine learning model performance, feature selection method can be adopted. This is because the method can reduce data dimensionality by selecting the most important and relevant features [13]. An example of feature selection that can increase accuracy and reduce the number of insignificant features in the data classification process is Backward Elimination. In machine learning-based data classification, accuracy is low due to the large number of attributes. When identifying diseases, the large number of attributes in medical data can present a complex challenge. In addition, feature extraction can make an important contribution by helping machine learning models identify the different characteristic features of each variant of hepatitis. This implies that the features improve the ability of the model to provide accurate predictions.

Backward Elimination was used as a feature selection method for the identification of volatile organic compounds (VOC) when applying SVM algorithm [14]. Consequently, the model achieved an accuracy of 75.6%, but when Backward Elimination was not used, the value was 73.2%. This implies that Backward Elimination method reduces the number of features and increases model accuracy. In order to improve the performance of data mining algorithms, such as KNN, Naïve Bayes, and C4.5, [15] classified diabetes using Backward Elimination. Based on the accuracy and AUC values, it is concluded that Backward Elimination can improve the total performance of the data mining algorithm.

According to the results of [14] and [15], the use of feature selection, specifically Backward Elimination has the potential to improve model performance and reduce the number of features. In addition, the popular adoption and effectiveness of XGBoost, an ensemble algorithm that combines multiple learning models have been investigated. Several research results from [9], [10], [11], and [12], showed that XGBoost has better performance in the classification process. Therefore, this research aims to validate hepatitis test results by using Backward Elimination feature selection, and to improve the classification of XGBoost in order to avoid errors in diagnosing hepatitis. The impact of Backward Elimination on hepatitis disease is determined by comparing the performance results of disease identification using Backward Elimination with those without using Backward Elimination.

II. METHODS

This section explains the datasets, methods, and system architecture used in this research.

A. Dataset

The dataset used in this research consisted of liver function test results which were a combination of primary and secondary data. Primary data were obtained from 375 laboratory results of patient medical records at Dustira Hospital Tk.II, Cimahi City. According to the agreement with Dustira Hospital, the research documents and files can only be accessed by officers involved in the research. Therefore, the primary dataset will be stored in a secure location and will remain unpublished. Meanwhile, the secondary data were sourced from the Kaggle website with the data used being the Indian Liver Patient Dataset https://www.kaggle.com/datasets/jeevannagaraj/indian-liver-patient-dataset containing 583 laboratory results collected in North East of Andhra Pradesh, India. Therefore, this research used 958 instances and 9 features (variables), which were categorized into 5 classes, namely hepatitis A, hepatitis B, hepatitis C, Unspecified Hepatitis, and Non-Hepatitis. Consequently, the variables used in this research were listed in Table 1.

Based on Table 1, some variables in the dataset have normal value limits, such as Total Bilirubin 0.1-1.2 mg/dL, Direct Bilirubin 0-0.3 mg/dL, SGOT 5-40 U/L, and SGPT 7-56 U/L. These variables help in establishing the disease

of hepatitis by observing the extent of liver function damage. Furthermore, the HBsAg, HAV, and HCV variables check the type of hepatitis suffered to ensure the normal value is negative.

TABLE 1
DATASET VARIABLES USED FOR IMPLEMENTATION AND TESTING

No	Variable	Description	Type of Data
1	Age	Age of Patient	Numeric
2	Gender	Gender of the Patient	Categorical
3	Bilirubin Total	The total amount of bilirubin present in the blood	Numeric
4	Bilirubin Direct	The total bilirubin that was directly excreted into the bile	Numeric
5	SGOT	serum glutamic oxaloacetic transaminase	Numeric
6	SGPT	serum glutamic pyruvic transaminase	Numeric
7	HbSAg	The surface antigen of hepatitis B virus (HBV)	Categorical
8	HAV	The virus that caused hepatitis A	Categorical
9	HCV	The virus that caused hepatitis C	Categorical

B. Backward Elimination

Backward Elimination is a method that could be used to remove insignificant attributes from the model [16]. The method is a wrapper-type feature selection technique performed by entering all predictor variables into a linear regression model, as shown in Equation (1). In addition, the method gradually eliminates the variables that do not meet the eligibility requirements, until a model was formed with only significant predictor variables [17]. The representation of Backward Elimination process could be seen in Fig. 1.

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + \ldots + b_n * x_n \quad (1)$$

where:
y : Dependent variable
x_n : Independent variable
b_n : Regression coefficients

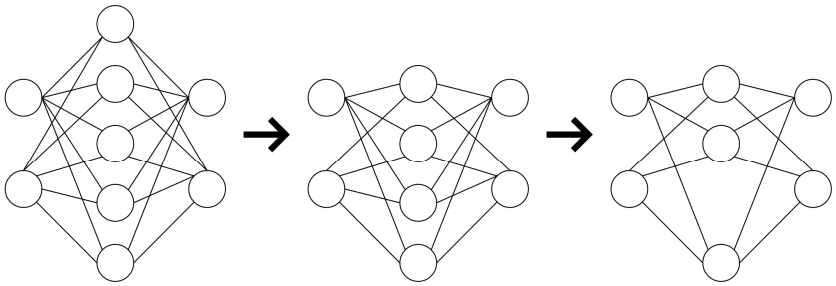

Fig. 1 Image of Backward Elimination

C. Extreme Gradient Boosting

XGBoost is a machine learning method, and it is a regression and classification algorithm with ensemble methods. The method is also a variant of Tree Gradient Boosting algorithm, developed with optimization that is 10 times faster than Gradient Boosting [18]. Furthermore, XGBoost could be formed from several decision trees, with each subsequent tree construction relying on the previous tree to form a stronger classification tree based on the sum of all tree weights. Fig. 2 showed an image of Extreme Gradient Boosting.

D. Bayesian Search Optimization

Bayesian Search Optimization is used as a hyperparameter search method to obtain the optimal XGBoost model. To achieve better performance, hyperparameter search utilized information from previous experiments to select hyperparameter combinations. Additionally, prior probability was used to determine the best point until the last iteration [20].

E. Random Search

Random search is a hyperparameter search method that efficiently and randomly selects a combination of hyperparameters from each iteration. Generally, the selection of optimal hyperparameters was performed by considering the highest cross-validation accuracy value from all the candidates generated. According to Li & Talwalker, random search is a simple method that has a strong basis compared to more complex algorithms [21].

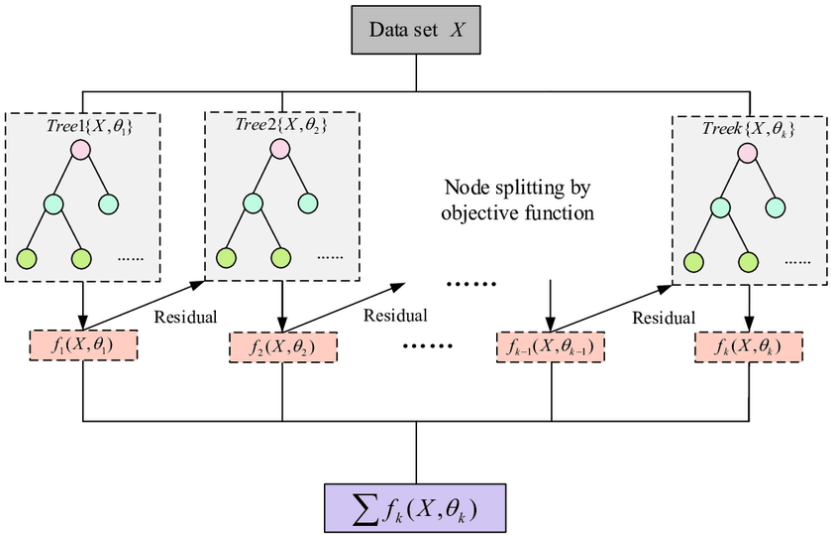

Fig. 2 Image of Extreme Gradient Boosting (based on the Flow chart of XGBoost [19])

F. System Architecture

The workflow of this research system was presented in the form of a business process model shown in Fig. 3.

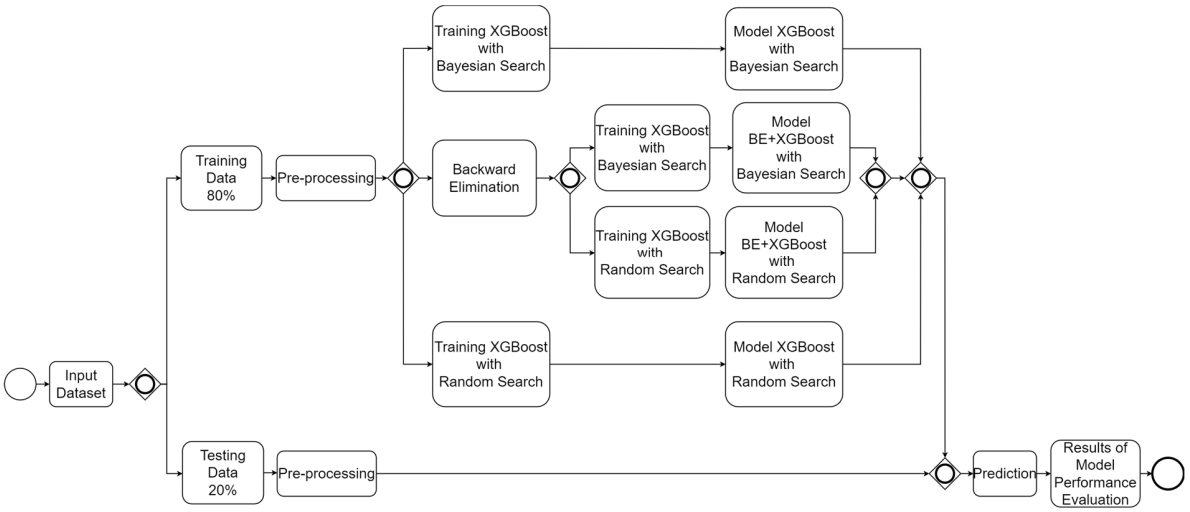

Fig. 3 System Architecture

According to Fig. 3, the system started by inputting the dataset and later divided the dataset into two parts, namely training and test data, with the training data comprising 80% and the test data being 20% of the total dataset. Furthermore, the necessary preprocessing process was carried out by performing Missing Value Treatment and Label Encoding method. To reduce features or data that were not very significant, Backward Elimination process was initially performed before developing XGBoost model using a combination of Bayesian and Random Search

hyperparameters. Subsequently, predictions were conducted to obtain model performance evaluation results in the form of a confusion matrix and k-fold cross-validation with 5-fold cross-validation

G. Evaluation Metrics

Classification model evaluation was performed to measure the performance of the classification model used [22]. Measurement of evaluation metrics in this result used the confusion matrix method and k-fold cross-validation with 5-fold cross-validation.

1) Confusion Matrix

Confusion matrix is a method commonly used to calculate accuracy in the classification model evaluation stage [23]. The method produced several values that were used as an evaluation of model performance, namely f1 score, accuracy, precision, and recall [24].

Accuracy: Accuracy refers to the percentage result of the number of correctly classified test data. The calculation of the accuracy value could be seen in Equation (2).

$$Accuracy = \frac{TP + TN}{Total\ Data} \tag{2}$$

Precision: Measures the certainty of the actual percentage of tuples labeled as positive were true in reality [25]. The calculation of the precision value is shown in Equation (3).

$$Precission = \frac{TP}{TP + FP} \tag{3}$$

Recall: Recall measures the completeness of the exact percentage of positive tuples that are positively labeled, and the calculation of recall value was shown in Equation (4).

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

F1 Score: The sum of the harmonic mean between precision and recall, and the calculation of the f1 score value was shown in Equation (5).

$$F1\ Score = 2 * \frac{Precission * Recall}{Precission + Recall} \tag{5}$$

where:
TP (True Positive): The number of positives that were correctly predicted as positive.
FP (False Positive): The number of negatives that were incorrectly predicted as positive.
TN (True Negative): The number of negatives that were correctly predicted as negative.
FN (False Negative): The number of positives that were incorrectly predicted as negative.

2) K-Fold Cross Validation

K-fold cross-validation was used to estimate prediction errors when evaluating model performance. The data was divided into k almost equal parts and the classification model was trained and tested k times. In addition, model classification accuracy was determined by averaging the accuracy at each repetition. In the duplication, a set of parts was used as training and testing data [26].

III. RESULTS

A. Pre-Processing

The initial preprocessing process was carried out by handling missing values in each row of data that had empty values. Various methods were used to handle missing values, which were described in Table 2. The next step comprises the process of converting categorical data into numerical values, which was performed by using the label encoding method. Subsequently, the data was split with a ratio of 80% for training data and 20% for test data.

TABLE 2
MISSING VALUE TREATMENT METHOD IN THE PRE-PROCESSING PROCESS

Missing Value Treatment	Description
Dropna	Deleted data rows that have empty values
Median	Filled empty values using the median value of the variable
Mean	Filled empty values using the mean value of the variable
Forward Fill	Filled the empty value with the closest value in front of the row

B. Training Model

Model training was performed by conducting a feature selection process using Backward Elimination. The process was performed by selecting features based on the p-value of each feature obtained through regressor_OLS.summary. The results of the p-value calculation could be seen in Table 3.

Based on Table 3, it could be seen that the variables x1 (age), x2 (gender), and x5 (SGOT) were not significant in the model, as evidenced by p-values exceeding the significance level ($\alpha = 0.05$). Meanwhile, the remaining variables (total bilirubin, direct bilirubin, SGPT, HBsAg, HAV, and HCV) were influential (significant) features in the model, indicated by their p-value, which was lower than the significant level. This difference was due to the use of Backward Elimination during the model development. Regarding the hyperparameter tuning process, Bayesian and random search was performed using a set of parameters listed in Table 4.

TABLE 3
THE SIGNIFICANCE OF THE RELATIONSHIPS IN THE MODEL

Variable	Coefficient	Standard Error	t	P-value
Const	3.2724	0.033	99.689	0.000
x1	-0.0012	0.003	-1.919	0.055
x2	0.0292	0.001	1.301	0.194
x3	-0.0109	0.022	-3.408	0.001
x4	0.0472	0.003	7.666	0.000
x5	-5.215e-05	0.006	-1.266	0.206
x6	0.0003	4.12e-05	6.419	0.000
x7	-2.3048	0.027	-85.551	0.000
x8	-3.3040	0.026	-124.684	0.000
x9	-1.3299	0.033	-39.806	0.000

*Significant $\alpha = 0{,}05$

TABLE 4
BAYESIAN SEARCH AND RANDOM SEARCH HYPERPARAMETER RANGE

Parameter	Description	Range		
n_estimators	Number of trees to be created	100	-	300
max_depth	Maximum depth of the tree	4	-	8
min_child_weight	Minimum number of weights of child nodes in the tree	0	-	7
learning_rate	Rate of learning patterns in the data	0.025	-	0.3
gamma	Minimum loss reduction value	0	-	2
colsample_bylevel	Column subsample ratio at each level	0.25	-	1
subsample	Number of samples used during the training process	0.5	-	1

Random Search hyperparameter tuning process used XGBoost RandomSearchCV model initialization, where the CV was cross-validated with K-fold cross-validation. Kfold used = 5. Based on the hyperparameter range in Table 3, the hyperparameter tuning process based on Bayesian Search and Random Search was performed for 100 iterations to get the best hyperparameter values, which could be seen in Table 5.

Model training experiments were performed with backward feature selection and without Backward Elimination, different missing value treatments, and each model with different hyperparameters, namely Bayesian search and random search. Furthermore, for each model training, the aim settings were “multi:softmax”, num_class = 5, and eval_metric = ['mlogloss', 'merror']. The following model training scenarios without Backward Elimination were shown in Table 6. When training the model with Backward Elimination, the following datasets were used, namely the total bilirubin, direct bilirubin, SGPT, HBsAg, HAV, and HCV features. The following model training scenario with Backward Elimination was shown in Table 7.

TABLE 5
BAYESIAN SEARCH AND RANDOM SEARCH HYPERPARAMETER TUNING RESULTS

Parameter	Bayesian Search	Random Search
n_estimators	240	200
max_depth	5	4
min_child_weight	1	0
learning_rate	0.2	0.025
gamma	0.3	0
colsample_bylevel	0.610	0.75
subsample	0.445	0.15

TABLE 6
SCENARIO OF XGBOOST MODEL TRAINING WITHOUT BACKWARD ELIMINATION

Number Of Models	Dataset	Missing Value Treatment	Hyperparameter Optimization
1	Normal	Dropna	Bayesian Search
2	Normal	Dropna	Random Search
3	Normal	Median	Bayesian Search
4	Normal	Median	Random Search
5	Normal	Mean	Bayesian Search
6	Normal	Mean	Random Search
7	Normal	Forward Fill	Bayesian Search
8	Normal	Forward Fill	Random Search

TABLE 7
SCENARIO OF XGBOOST MODEL TRAINING WITH BACKWARD ELIMINATION

Number Of Models	Dataset	Missing Value Treatment	Hyperparameter Optimization
1	Backward Elimination	Dropna	Bayesian Search
2	Backward Elimination	Dropna	Random Search
3	Backward Elimination	Median	Bayesian Search
4	Backward Elimination	Median	Random Search
5	Backward Elimination	Mean	Bayesian Search
6	Backward Elimination	Mean	Random Search
7	Backward Elimination	Forward Fill	Bayesian Search
8	Backward Elimination	Forward Fill	Random Search

C. Results of XGBoost Model Testing Without Backward Elimination

The results of testing 8 models using 20% of the test data were measured based on the degree of accuracy, precision, recall, F1 score, training time, and average precision of k-fold cross-validation results with a value of Kfold = 5. The following were the results of testing the model without Backward Elimination, which could be seen in Table 8.

According to Table 8, the results of training XGBoost model without Backward Elimination, using mean feature value to handle missing data, and applying "Random Search" for hyperparameter optimization with specific parameters (n_estimators = 200, max_depth = 4, min_child_weight = 0, learning_rate = 0.025, gamma = 0, colsample_bylevel = 0.75, and subsample = 0.15) produced an accuracy of 98.437% at 0.74 seconds. This performance was better than using Bayesian Search hyperparameter optimization with parameters n_estimators = 240, max_depth = 5, min_child_weight = 1, learning_rate = 0.2, gamma = 0.3, colsample_bylevel = 0.610, and subsample = 0.445, which achieved the same accuracy of 98.437%, but with a training time of 0.85s.

Even though model training without Backward Elimination achieved high accuracy based on the results of the confusion matrix or 5-fold cross-validation average, it could be seen from the graph in Fig. 4 that there were several models where overfitting occurred. In these cases, the training data had a high level of accuracy, but simultaneously, the accuracy of the validation data was low.

TABLE 8
RESULTS OF XGBOOST MODEL TESTING WITHOUT BACKWARD ELIMINATION

Number Of Models	Evaluation of Model Performance				TrnT	5-Fold CrossVal
	Acc.	Prec.	Rec.	F1 Scr		
1	95.890 %	95.252 %	95.894 %	95.521 %	0.68 s	0.983
2	96.575 %	96.061 %	96.421 %	96.227 %	0.59 s	0.982
3	95.833 %	95.495 %	96.466 %	95.878 %	0.92 s	0.980
4	97.916 %	97.636 %	98.233 %	97.906 %	0.72 s	0.973
5	98.437 %	98.263 %	98.583 %	98.416 %	0.85 s	0.982
6	98.437 %	98.263 %	98.583 %	98.416 %	0.74 s	0.983
7	96.354 %	96.033 %	96.817 %	96.365 %	0.77 s	0.971
8	97.916 %	97.500 %	98.596 %	97.939 %	0.65 s	0.973

*Acc: Accuracy, Prec: Precision, Rec: Recall, F1 Scr.: F1-Score, TrnT.: Training Time,
K-Fold Cross Val: Mean of 5-Fold Cross Validation

D. Test results of XGBoost model with Backward Elimination

The results of testing 8 models using 20% of the test data were measured based on the level of accuracy, precision, recall, F1 score, training time, and average precision of k-fold cross-validation results with a value of Kfold = 5. The following are the results of testing the model without Backward Elimination, which could be seen in Table 9.

Table 9 showed that XGBoost model trained with Backward Elimination, mean feature value treatment for missing value, and using "Random Search" with specific parameters (n_estimators = 200, max_depth = 4, min_child_weight = 0, learning_rate = 0.025, gamma = 0, colsample_bylevel = 0.75, and subsample = 0.15) achieved an accuracy of 98.958% at 0.64 seconds. This performance was better than using Bayesian Search (n_estimators = 240, max_depth = 5, min_child_weight = 1, learning_rate = 0.2, gamma = 0.3, colsample_bylevel = 0.610, and subsample = 0.445), which achieved the same accuracy but required a longer training time of 0.70 seconds.

TABLE 9
RESULTS OF XGBOOST MODEL TESTING WITH BACKWARD ELIMINATION

Number Of Models	Evaluation of Model Performance				TrnT	5-Fold CrossVal
	Acc.	Prec.	Rec.	F1 Scr		
1	97.945 %	97.921 %	97.473 %	97.682 %	0.64 s	0.983
2	97.945 %	97.921 %	97.473 %	97.682 %	0.57 s	0.982
3	96.354 %	95.979 %	97.180 %	96.420 %	0.87 s	0.980
4	97.395 %	97.048 %	97.882 %	97.404 %	0.69 s	0.973
5	98.958 %	98.934 %	98.934 %	98.934 %	0.70 s	0.982
6	98.958 %	98.934 %	98.934 %	98.934 %	0.64 s	0.983
7	98.437 %	98.064 %	98.947 %	98.442 %	0.65 s	0.971
8	98.958 %	98.666 %	99.298 %	98.953 %	0.64 s	0.973

*Acc: Accuracy, Prec: Precision, Rec: Recall, F1 Scr.: F1-Score, TrnT.: Training Time,
K-Fold Cross Val: Mean of 5-Fold Cross Validation

The model training using Backward Elimination feature selection according to the findings in Table 9 achieved high accuracy based on the results of the confusion matrix or five-fold cross-validation average. The graph in Fig. 5 showed that model training using Backward Elimination achieved high accuracy. Furthermore, feature selection increased the accuracy of the training data to the same level as the validation data, showing a reduction in the overfitting effect. The reduction could be understood by observing how the two accuracy curves, namely the training data and validation data, come closer together. The observation showed that the model could perform well on new, unseen data. However, some models still experienced overfitting, obviously due to a growing disparity between the training and validation data curves longer time.

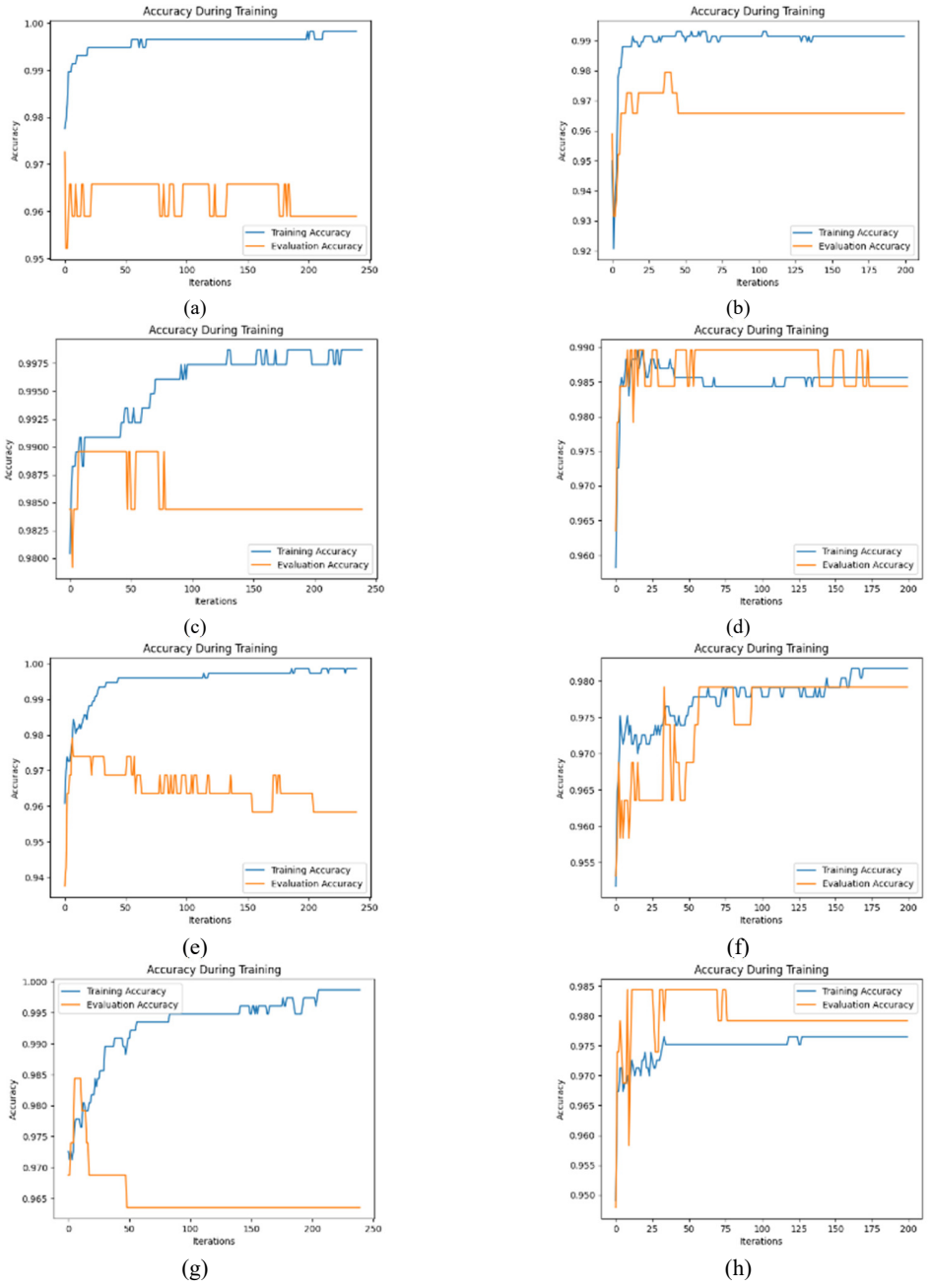

Fig. 4 Graph Of Accuracy For Xgboost Model Testing Without Backward Elimination (a) Drop+Bayesian, (b) Drop+Random, (c) Mean+Bayesian, (d) Mean+Random, (e) Median+Bayesian, (f) Median+Random, (g) Ffill+Bayesian, (h) Ffill+Random

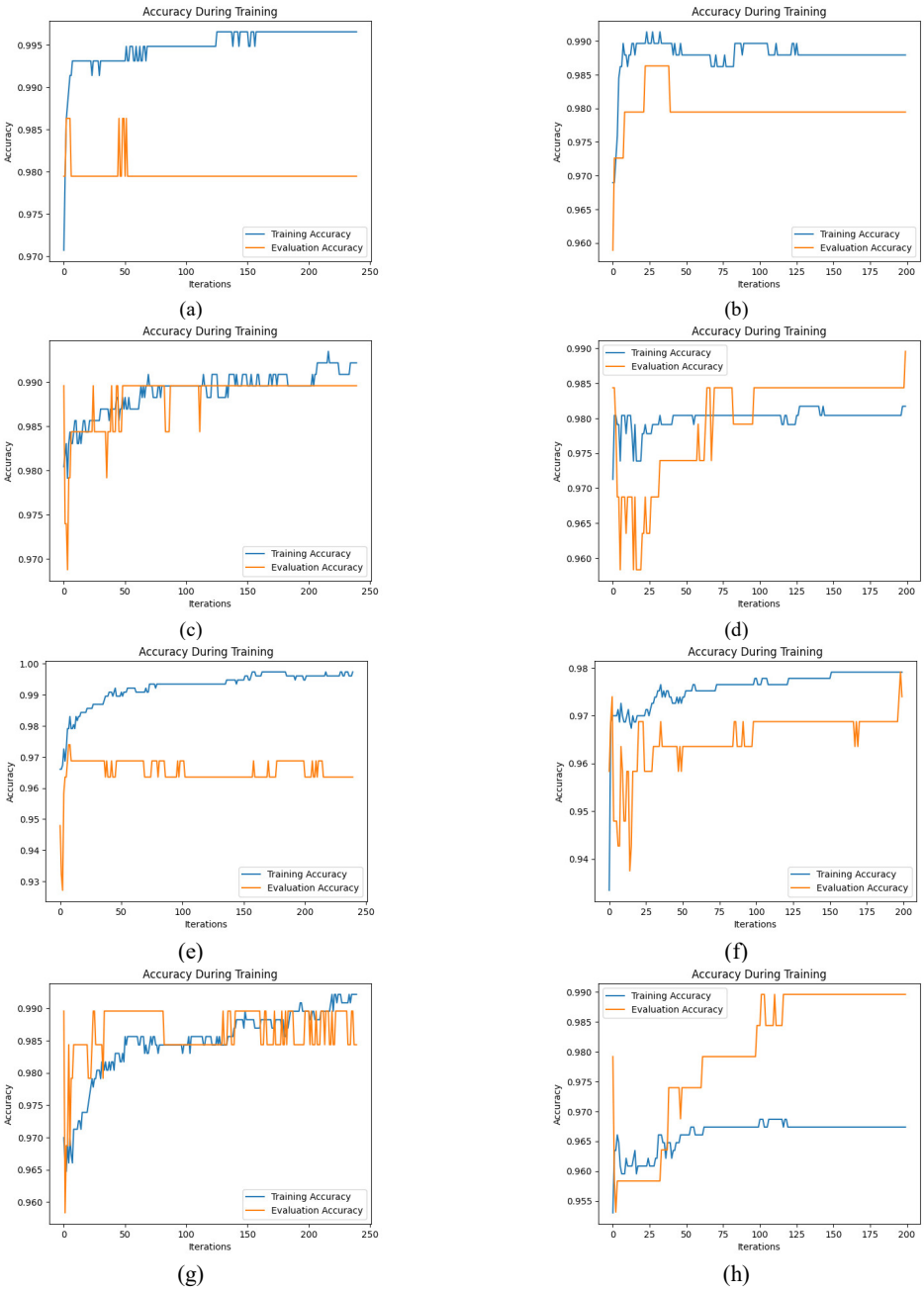

Fig. 5 Graph Of Accuracy For Xgboost Model Testing With Backward Elimination (BE) (a) Drop+BE+Bayesian, (b) Drop+ BE+Random, (c) Mean+ BE+Bayesian, (d) Mean+ BE+Random, (e) Median+ BE+Bayesian, (f) Median+ BE+Random, (g) Ffill+ BE+Bayesian, (h) Ffill+ BE+Random

IV. DISCUSSION

Several research including [9], had examined the implementation results of seven machine learning algorithms including SVM, Decision Tree, Random Forest, Naive Bayes, Logistic Regression, Adaptive Boosting, and Extreme Gradient Boosting for predicting liver disease. Among the algorithms, Extreme Gradient was used, which showed the highest accuracy reaching 81% [10]. Additionally, XGBoost showed superior accuracy of 95% in predicting hepatitis B surface antigen zero-clearance [11]. The use of this model for liver disease prediction obtained the highest accuracy

of 87.02%. According to George Obaido et al. [12], hepatitis B diagnosis using XGBoost achieved 90% accuracy. In this current research, model training using XGBoost achieved the highest accuracy of 98.437% at 0.74 seconds.

The test results using Backward Elimination when training XGBoost model showed an accuracy of 98.958% at 0.64 seconds. The improvement of 0.521% in accuracy was attributed to Backward Elimination which aided in reducing the number of features used.

The inclusion of Backward Elimination in the training process improved accuracy and reduced overfitting by minimizing the use of insignificant features. Furthermore, the choice of hyperparameters alongside Backward Elimination also affected the training time with "Random Search" showing faster training time compared to "Bayesian Search".

Other aspects that could be investigated in further research were feature selection methods. Although Backward Elimination was used in this exploration, different feature selection methods could be used to provide a more comprehensive comparison. Further exploration can also be carried out to understand the way in which different feature selection methods impact results and contribute to significant improvements. While considering the balance of each class, the number of datasets needs to be taken into account because an increase in the number could also affect performance.

V. Conclusions

In conclusion, the results of the research showed that feature selection using Backward Elimination method had a positive impact on the performance of XGBoost model. By using only six relevant features, the model achieved high accuracy, precision, recall, and F1 score. Apart from increasing the accuracy of hepatitis identification, the use of Backward Elimination also reduced overfitting. Additionally, the results of hyperparameter optimization of XGBoost model using Bayesian and Random Search methods showed that this method was effective.

Author Contributions: *Jasman Pardede*: Writing -Review, Supervision, Validation. *Desita Nurrohmah*: Conceptualisation, Methodology, Software, Writing -Original Draft.

All authors have read and agreed to the published version of the manuscript.

Funding: This research received no specific grant from any funding agency.

Acknowledgments: This research is supported by Department of Informatics, Faculty of Industrial Technology, National Institute of Technology.

Conflicts of Interest: The authors declare no conflict of interest.

Data Availability: The primary data that has been used is confidential. Meanwhile, the secondary data that support the findings of this study are openly available in Kaggle at https://www.kaggle.com/datasets/jeevannagaraj/indian-liver-patient-dataset.

Institutional Review Board Statement: This study received ethical approval from the LPPM-Institut Teknologi Nasional (Itenas) Institutional Review Board on June 23, 2023.

Informed Consent: Informed Consent was obtained, and a detailed explanation was presented in the Methods section.

Animal Subjects: There were no animal subjects.

ORCID:
Jasman Pardede: https://orcid.org/0000-0001-7773-0296
Desita Nurrohmah: https://orcid.org/0009-0002-5391-8256

References

[1] K. Y. Raharja, H. Oktavianto, and R. Umilasari, "Perbandingan Kinerja Algoritma Gaussian Naive Bayes Dan K-Nearest Neighbor (KNN) Untuk Mengklasifikasi Penyakit Hepatitis C Virus (HCV)," Undergraduate thesis, Department of Informatics Engineering, Universitas Muhammadiyah Jember, 2021. [Online]. Available: http://repository.unmuhjember.ac.id/id/eprint/8590

[2] Pan American Health Organization and World Health Organization, "5 Things to Know About Viral Hepatitis," PAHO.org. Accessed: August. 2, 2023. [Online.] Available: https://www.paho.org/en/topics/hepatitis/5-things-you-should-know-about-viral-hepatitis

[3] P. Khetrapal Singh, "Bringing hepatitis care closer to you," WHO.int, 2022. [Online]. Available: https://www.who.int/southeastasia/news/opinion-editorials/detail/bringing-hepatitis-care-closer-to-you.

[4] M. K. dr. Wening Sari, *Care Your self: Hepatitis*. Niaga Swadaya. [Online]. Available: https://books.google.co.id/books?id=jQdJz1maiXwC

[5] Kementerian Kesehatan, "Laporan Riskesdas 2018 Nasional," Kementrian Kesehatan RI, Indonesia, 2018. [Online]. Available: https://repository.badankebijakan.kemkes.go.id/id/eprint/3514/1/Laporan%20Riskesdas%202018%20Nasional.pdf

[6] A. K. Saurav, Patra MD; Mukherjee, Brijesh; Das, "Pre-analytical errors in the clinical laboratory and how to minimize them quality control view project," *Int. J. Bioassays*, vol. 2, no. May 2014, pp. 551–553, 2013, [Online]. Available: https://www.researchgate.net/publication/236020318

[7] Y. Rombe, "penggunaan metode XGboost untuk klasifikasi status obesitas di Indonesia," Thesis, Fakultas Matematika dan Ilmu Pengetahuan Alam, Hasanuddin University, 2022. [Online]. Available: http://repository.unhas.ac.id:443/id/eprint/13027

[8] A. N. Rachmi, "Implementasi metode Random Forest dan Xgboost pada klasifikasi customer churn," Undergraduate thesis, Faculty of Mathematics and Natural Sciences, Universitas Islam Indonesia, 2020. [Online]. Available: https://dspace.uii.ac.id/123456789/30082

[9] Pranitha Gadde, G. Deepthi, C. Shivani, K. Nagavinith, and K. H. Kumar, "Heart disease prediction using machine learning algorithms," *Int. J. Manag. Technol. Eng.*, vol. 11, no. 6, pp. 29–35, 2021, doi: 16.10089.IJMTE.2021.V10I6.21.50804.

[10] X. Tian *et al.*, "Using machine learning algorithms to predict hepatitis B surface antigen Seroclearance," *Comput. Math. Methods Med.*, vol. 2019, pp. 1–7, Jun. 2019, doi: 10.1155/2019/6915850.

[11] C. M. Bhatt, P. Patel, T. Ghetia, and P. L. Mazzeo, "Effective heart disease prediction using machine learning techniques," *Algorithms*, vol. 16, no. 2, p. 88, Feb. 2023, doi: 10.3390/a16020088.

[12] G. Obaido *et al.*, "An interpretable machine learning approach for hepatitis B diagnosis," *Appl. Sci.*, vol. 12, no. 21, p. 11127, Nov. 2022, doi: 10.3390/app122111127.

[13] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, Oct. 2007, doi: 10.1093/bioinformatics/btm344.

[14] M. Tharmakulasingam, C. Topal, A. Fernando, and R. La Ragione, "Backward feature elimination for accurate pathogen recognition using portable electronic nose," in *2020 IEEE International Conference on Consumer Electronics (ICCE)*, IEEE, Jan. 2020, pp. 1–5. doi: 10.1109/ICCE46568.2020.9043043.

[15] M. A. Wiratama and W. M. Pradnya, "Optimization of data mining algorithm using backward elimination for diabetes classification," *J. Nas. Pendidik. Tek. Inform.*, vol. 11, no. 1, p. 1, Apr. 2022, doi: 10.23887/janapati.v11i1.45282.

[16] D. H. Vu, K. M. Muttaqi, and A. P. Agalgaonkar, "A variance inflation factor and backward elimination based robust regression model for forecasting monthly electricity demand using climatic variables," *Appl. Energy*, vol. 140, pp. 385–394, Feb. 2015, doi: 10.1016/j.apenergy.2014.12.011.

[17] Kurniawan and B. Yuniarto, *Analisis Regresi: Dasar dan penerapannya dengan R*. Indonesia: Kencana Prenada Media Group, 2016.

[18] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[19] R. Guo, Z. Zhao, T. Wang, G. Liu, J. Zhao, and D. Gao, "Degradation State Recognition of Piston Pump Based on ICEEMDAN and XGBoost," *Appl. Sci.*, vol. 10, no. 18, p. 6593, Sep. 2020, doi: 10.3390/app10186593.

[20] R. Ubaidillah, M. Muliadi, D. T. Nugrahadi, M. R. Faisal, and R. Herteno, "Implementasi XGBoost Pada Keseimbangan Liver Patient Dataset dengan SMOTE dan Hyperparameter Tuning Bayesian Search," *J. MEDIA Inform. BUDIDARMA*, vol. 6, no. 3, p. 1723, Jul. 2022, doi: 10.30865/mib.v6i3.4146.

[21] L. Hertel, P. Baldi, and D. L. Gillen, "Quantity vs. Quality: On Hyperparameter Optimization for Deep Reinforcement Learning," Jul. 2020, [Online]. Available: http://arxiv.org/abs/2007.14604

[22] P. Purwono, A. Wirasto, and K. Nisa, "Comparison of Machine Learning Algorithms for Classification of Drug Groups," *SISFOTENIKA*, vol. 11, no. 2, p. 196, Jul. 2021, doi: 10.30700/jst.v11i2.1134.

[23] M. F. Rahman, D. Alamsah, M. I. Darmawidjadja, and I. Nurma, "Classification for Diabetes Diagnosis Using Bayesian Regularization Neural Network (RBNN) Method," *J. Inform.*, vol. 11, no. 1, p. 36, Jan. 2017, doi: 10.26555/jifo.v11i1.a5452.

[24] Suyanto, *Machine Learning Tingkat Dasar Dan Lanjut*. Informatika, 2018. [Online]. Available: https://books.google.co.id/books?id=QWbuzwEACAAJ

[25] I. Saputra and D. Rosiyadi, "Perbandingan kinerja algoritma K-Nearest Neighbor, Naïve Bayes Classifier dan Support Vector Machine dalam klasifikasi tingkah laku bully pada aplikasi Whatsapp," *Fakt. Exacta*, vol. 12, no. 2, p. 101, Jul. 2019, doi: 10.30998/faktorexacta.v12i2.4181.

[26] Nurhayati, I. Soekarno, I. K. Hadihardaja, and M. Cahyono, "A study of Hold-Out and K-Fold Cross Validation for accuracy of groundwater modeling in Tidal Lowland Reclamation using Extreme Learning Machine," in *2014 2nd International Conference on Technology, Informatics, Management, Engineering & Environment*, IEEE, Aug. 2014, pp. 228–233. doi: 10.1109/TIME-E.2014.7011623.