Optimized Machine Learning models for Hepatitis C prediction: Leveraging Optuna for Hyperparameter Tuning and Streamlit for Model deployment

Uriel Nguefack Yefou Author[1*†],
Pauline Ornela Megne Choudja Author[2*†], Binta Sow Author[2†]
and Abduljaleel Adejumo Author[2†]

[1*]African Institute for Mathematical Sciences Cameroon, Crystal Gardens, Limbe, 608, South West, Cameroon.
[2]African Institute for Mathematical Sciences Senegal, Km2 route de Joal (IRD Center), Mbour, 1418, Thies, Senegal.

*Corresponding author(s). E-mail(s): uriel.nguefack@aims-cameroon.org; pauline.o.m.choudja@aims-senegal.org;
Contributing authors: binta.sow@aims-senegal.org; abduljaleel.adejumo@aims-senegal.org;
†These authors contributed equally to this work.

Abstract

Machine learning techniques have gained significant attention for their potential in solving diverse real-world problems across various fields. This study focuses on leveraging machine learning algorithms to predict the stage of Hepatitis C, a prevalent liver disease affecting a substantial portion of the global population. By employing a dataset encompassing 615 patients and incorporating a multitude of factors associated with Hepatitis C, a comprehensive analysis was conducted to compare the performance of six prominent machine learning algorithms. The algorithms considered include categorical boosting (CatBoost), Extreme Gradient Boosting (XGBoost), Gaussian Naive Bayes (GNB), Light Gradient Boosting Machine (LGBM), Random Forest (RF), and ExtraTreeClassifier (ExtraT). To optimize the performance of these models, a hyperparameter optimization technique called Optuna was used to identify the optimal parameters for each algorithm. Subsequently, the performance of the models were

evaluated using the test dataset, comprising 20% of the overall patient data. The research findings revealed that the XGBoost algorithm emerged as the most effective approach, exhibiting a remarkable accuracy of 94.31%. Furthermore, the XGBoost model demonstrated exceptional F1-score, precision, and recall values, measuring 94.23%, 94.63%, and 94.31%, respectively. Building upon these promising results, we deployed the XGBoost model in a user-friendly web application leveraging Streamlit. This deployment ensures easy accessibility and usability of the model for the broader community.

Keywords: Machine Learning, Hepatitis C, Optuna, Streamlit, XGBoost, Hyperparameter

1 Introduction

The liver is one of the most vital organs with multiple functions including, secretion of bile, blood filtration, transformation and storage of substances absorbed from the digestive system [1]. Infections affecting the liver can have detrimental consequences, potentially leading to the development of liver cancer. Hepatitis C, caused by the hepatitis C virus (HCV), is an inflammatory condition of the liver. It is characterized by a slow progression of hepatic fibrosis, advancing from stage 0 (no fibrosis) to stage 4 (cirrhosis) [2]. Approximately 30% of individuals infected with HCV naturally clear the virus within 6 months without treatment, while the remaining 70% develop chronic HCV infection, which can progress to cirrhosis and liver cancer within 20 years [3]. As per the World Health Organization (WHO), HCV is a global health issue, with an estimated 71 million people worldwide living with chronic HCV infection in 2015, and around 1.75 million new infections occurring annually. The disease is prevalent across regions including the Americas, Europe, Asia, Africa, etc [4].
Unfortunately, there is currently no available vaccine for hepatitis C [5]. The diagnosis of HCV infection involves two steps. Firstly, a serological test is conducted to detect anti-HCV antibodies, identifying individuals who have been infected with the virus. Secondly, those who test positive for the antibody undergo a nucleic acid test to confirm chronic infection and assess the need for treatment by detecting HCV ribonucleic acid (RNA) [6]. Upon diagnosis of chronic HCV infection, an evaluation is performed to determine the extent of liver damage in terms of fibrosis and cirrhosis. However, this diagnostic approach is time-consuming, expensive, and limited in resource-constrained settings.

Early detection plays a pivotal role in mitigating the potential health consequences associated with infection and curbing the transmission of the virus. Leveraging the advancements in technology, the utilization of machine learning (ML) algorithms, in conjunction with medical expertise, holds the promise of facilitating rapid and efficient automated diagnosis. ML algorithms exhibit the capability to extract critical insights from data encompassing patient information, enabling the identification of

individuals who are most likely to derive benefits from diagnostic testing and subsequent treatment interventions. This approach not only enhances patient outcomes but also contributes to the reduction of healthcare expenditures. Overall, the integration of ML algorithms in the prediction and prevention of hepatitis C infection can yield substantial improvements in public health by enabling proactive interventions and curtailing the spread of the disease.

The main objective of this study is to optimize ML models for the accurate prediction of Hepatitis C. In addition, a key focus is to enhance the interpretability and transparency of these models, enabling healthcare professionals to comprehend the underlying rationale behind the predictions and validate the reliability of the models. This will help to empower physicians with actionable insights derived from ML algorithms, thereby facilitating informed decision-making and improving the overall effectiveness of Hepatitis C diagnosis and treatment.

This paper is organized as follows: Section 2 provides an overview of the relevant literature on machine learning techniques for classifying hepatitis C disease, and Section 3 describes the methodology implemented in this study, including a detailed explanation of the data used. Section 4 presents the results of the study as well as an analysis of the experimental tests and finally, in Section 5, the article concludes and offers suggestions for future research.

2 Related work

In [7], the authors used the XGBoost ML algorithm for predicting hepatitis C with data from blood donors and clinical data from patients with chronic hepatitis C. After evaluation of the model the experimental results show that the XGBoost algorithm is robust and performs better with an accuracy of 91.56% compared with the results obtained from SVM(Support Vector Machine), KNN(K Nearest Neighbor), DT(Decision Tree) and Adaboost algorithms. Similarly, [8] conducted a study using four ML techniques namely KNN, Support Vector Machine (SVM), Naive Bayes, and Decision Tree for the prediction, classification, and diagnosis of hepatitis C. Based on the analysis of 615 records, the Decision Tree method exhibited superior performance, achieving an accuracy of 93.44%, outperforming the other models in terms of the specific classification objectives.

[9],[10], conducted a study in which they developed an HCV predicting model using ML techniques, specifically the Random Forest (RF) and K-Nearest Neighbours algorithm (KNN) classifiers. The dataset used in their study was sourced from the UCI-ML repository and consisted of 668 instances of HCV. The result indicated that the Random Forest gave the best accuracy score of 94.88%. In [11], the authors addressed the imbalance problem in the dataset obtained from the UCI-ML Repository by applying the Synthetic Minority Oversampling Technique (SMOTE). Their results show that the random forest (RF) algorithm performed best across various evaluation metrics.

[12] used ensemble-based ML models to detect the presence of cirrhosis in hepatitis C patients. Four ML models, GBM(Gradient Boosting Machine), Random Forest,

ExtraT(Extra Tree) model, and Extreme Gradient Boosting were trained on the data, consisting of 28 attributes of 2038 Egyptian patients. The result shows that the Extra Trees model performed better than the other two models with an accuracy of 96.92%, a precision of 99.81%, and a recall of 94.00% using 16 features of the 28.

3 Methods

3.1 Dataset description

The Intelligent Systems and Machine Learning center at the California University, Irvine(UCI) provided the data for the UCI dataset, which was generated by [13] and contains information from 615 individuals. The target feature, bilirubin (BIL), Age, gender, blood levels of ALB, ALP(ALkaline Phosphatase), ALT(ALanine aminoTransferase), CHE(Choline Esterase), AST(ASpartate aminotransferase), CHOL(CHOLesterol), CREA(CREAtinine Blood test), GGT(Gamma-Glutamyl-Transferase) and PROT(Total protein test) are among the thirteen features that make up each individual's record. These features are used to classify each person as either a blood donor or as having hepatitis C disease, which includes its development to cirrhosis and fibrosis. The ML algorithms that will be used to determine the possibility of a person contracting the virus will be trained and test on this data.

Using Pearson's approach [14], this study determined the correlation coefficients between various features showed in Figure 1. Pearson's correlation coefficient, which has values ranging from -1 to 1, can be used to assess the linear link between two variables, with -1 signifying a perfect negative correlation, 0 indicating no correlation, and 1 representing a perfect positive correlation. It assesses the degree and direction of the link between two variables. The connection between features in the dataset used can provide insight on how different variables affect the development of hepatitis C virus (HCV) infection. It can, for example, assess whether older people are more susceptible to HCV infection by examining the link between the HCV infection rate and the age.

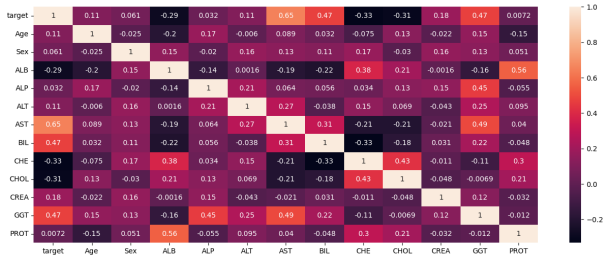


Fig. 1: Correlation plot HCV Disease

3.2 Data Preprocessing

The initial stage in the proposed system is data preparation or preprocessing, which removes noisy values and replaces missing values for specific attributes.

- Encode the sex and target column: The target column 'Category' which contains 5 classes was encoded as follow: '0=blood donor' by 0, '0s=suspect blood donor' by 1, '1=Hepatitis' by 2, '2=Fibrosis' by 3 and '3=cirrhosis' by 4. For the sex column, male was encoded as 1 and female as 0.
- Handle missing values: The empty values were filled with 10% trim_mean of the column using the librairy trim_mean of scipy.stats.
- Scaling: we have scaled our variables using the StandardScaler library of scikit-learn.
- Balance the data: Since our dataset was highly unbalanced, the 'SMOTE' [15] technique was employed to solve that imbalance problem.

3.3 Proposed framework

The framework used for this work is presented in this section. The suggested framework is divided into 06 stages, including (1) Data collection, (2) Data pre-processing, (3) data splitting, (4) model generation, (5) model optimization, (6) model evaluation, and (7) deployment of the best model. Figure 2 shows the proposed framework for predicting the HCV disease.

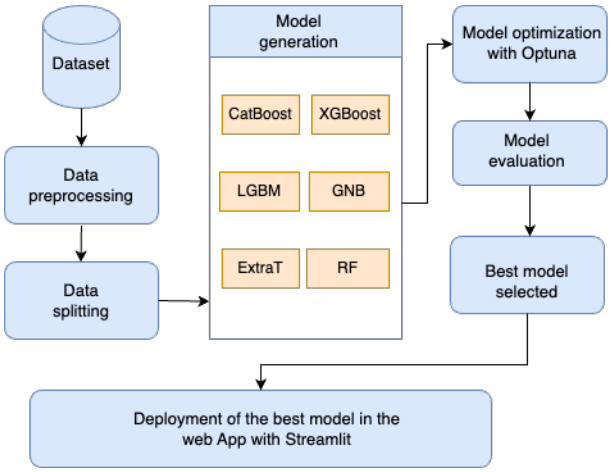


Fig. 2: Proposed framework for Hepatitis C detection

- Data collection: In this stage, our dataset is imported and will be used for the next phases.
- Data pre-processing: The techniques presented in Section 3.2 are used to preprocess the data before modeling.
- Data splitting: The data is splitted in 2 parts: test set(20%), and training set(80%).

- Model generation: We implemented six(06) ML algorithms: Catboost, XGBoost, LGBM, GNB, ExtraT and RF. These algorithms will be more explained in section 3.4.
- Model optimization: A technique called Optuna is used to discover the best parameters for each of the ML algorithms that we have developed. Optuna [16] is an autonomous hyperparameter optimisation software framework that is specifically built for machine learning.
- Model Evaluation: The models's performance are assessed using the Test set for future comparison. T
- Deployment of the best model: Following the evaluation phase, the best model is selected and implemented in our web app using Streamlit.

3.4 Description of utilized ML techniques

Ensemble algorithms combine the outputs of numerous models that have been trained. Boosting classifiers are iterative ensemble algorithms that adjust an observation's weight based on the most recent classification. Bagging classifiers use weighted averages or majorities to merge many independent variables. In this investigation, three boosting approaches (CatBoost, XGBoost, and LGBM), a straightforward ML algorithm (Gaussian Naive Bayes), and two bagging methods (ExtraT and RF) were employed.

1. **CatBoost**
   When utilized for Supervised ML problems, CatBoost, a component of the Gradient Boosted Decision Tree (GBDT) ensemble machine learning approaches, introduces two novelties: Ordered Boosting and Ordered Target Statistics [17]. It is an innovative algorithm used to process categorical features [18].
2. **XGBoost**
   Extreme Gradient Boosting is a Decision Tree-based machine learning algorithm. XGBoost addresses the issue of overfitting, which can be a serious concern for ensemble models, by incorporating additional regularisation in its goal function. This regularization component penalizes the model's complexity, improving generalizability and lowering the risk of overfitting [19]. The prediction of the model is given by:
   $$\hat{y_i} = \sum_{k=1}^{K} f_k(x_i), \qquad f_k \in \mathcal{F} \tag{1}$$
   where $f_k(x_i)$ is the prediction of the $k-th$ tree, $\mathcal{F}$ is the space of Regression Tree and K the total number of trees.
3. **LGBM**
   The LGBM, or Light Gradient Boosting Model is used when there are more variables in the data, which leads to a problem of scalability and effectiveness of the model. The primary reason for this behaviour is that it takes a significant amount of time and effort for each feature to scan through all of the different data examples and compute all of the probable split points. It uses two techniques: the Gradient-based On-Side Sampling (GOSS) and the Exclusive Feature Bundling (EFB). In reality, GOSS will just utilise the remaining data to compute the total information

gain, removing a large portion of the data section with weak gradients. In order to decrease the number of features, the EFB typically takes no non-zero value at the same time as the mutually exclusive features [20].

4. **Gaussian Naive Bayes**
   Gaussian Naive Bayes is a sophisticated Machine Learning technique for classification tasks which utilizes the rules of Bayes and supposes that each class follow a Gaussian distribution. Low variance, Incremental learning, Computational efficiency, Robustness in the face of absent value, Robustness in facing noise are important features of Gaussian Naive Bayes [21].
5. **Random Forest**
   Random Forest(RF) [22] is an approach to classification and regression using ensemble learning. By choosing random subsets of the characteristics and random subsets of the training data, Random Forest generates several decision trees. By averaging or tallying all of the different trees' forecasts, the ultimate prediction is made.
6. **ExtraT**
   The Extra Tree [23], also known as the Extremely Randomized Tree, generates predictive models for classification and regression problems. It is comparable to other algorithms like Random Forest and Decision Tree, but it provides superior predictions by including additional data facts. Additionally, the extra tree technique is faster and easier to use than earlier approaches. Therefore, it is a powerful data mining and predictive modelling tool.

3.5 Performance evaluation

The accuracy of the classification, precision, recall, and F1 score are well-known evaluation measures that can be used to gauge how well the proposed approach performs. Based on a "confusion matrix" that contains True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), these measures were developed. [24, 25]

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

3.6 Streamlit Framework

Streamlit lets you turn data scripts into shareable web apps in minutes. It is all Python, open-source, and free. And once you have created an app you can use the Community Cloud platform to deploy, manage, and share your app. The best model selected from the set of all our models is deployed using Streamlit. The model can be tested by everyone having the application.

4 Results

This section is divided into four parts: the first part will address the effectiveness of the optimization technique on the different models developed. The second part will focus on the feature importance of the best model which is the XGBoost model. The confusion matrix of the XGBoost model on the test data will be presented in the third part and the Web App designed with Streamlit in the last part.

4.1 Comparison of the models

The obtained results from the different models are evaluated based on accuracy, precision, recall, and the F1-score. Table 1 provides a comparison of the six models when using their default parameters. It can be observed that the CatBoost model outperforms the other models in terms of accuracy, recall, and the F1-score. On the other hand, when considering precision as the primary evaluation metric, the Gaussian Naive Bayes (GNB) model demonstrates superior performance.

Table 1: Performance of the model without optimization

Models	Accuracy	Precision	Recall	F1-score
CatBoost	**0.9186**	0.9155	**0.9187**	**0.9115**
XGBoost	0.9024	0.8979	0.9024	0.8973
LGBM	0.9024	0.9069	0.9024	0.8954
GNB	0.9106	**0.9169**	0.9106	0.9089
ExtraT	0.8780	0.8763	0.8780	0.8498
RF	0.8862	0.8889	0.8862	0.8687

Optuna framework was employed to perform a hyperparameter optimization process for each model. The best parameters were subsequently used to train the models and the outcomes are reported in Table 2. Table 2 shows that the XGBoost model outperforms the other models across all evaluated metrics and its best hyperparameter values are presented in Table A1.

Table 2: Performance of the models with optimization

Models	Accuracy	Precision	Recall	F1-score
CatBoost	0.9430	0.9434	0.9431	0.9409
XGBoost	**0.9431**	**0.9463**	**0.9431**	**0.9423**
LGBM	0.9350	0.9337	0.9350	0.9304
GNB	0.9106	0.9169	0.9106	0.9089
ExtraT	0.9268	0.9300	0.9268	0.9236
RF	0.8943	0.9124	0.8943	0.8854

4.2 Feature importance on the XGBoost model

Feature importance plot gives the visual representation on the features that contribute most to the model. The figure 3 shows that all the features had a significant contribution to the model, given data no weight is less than 0.04.

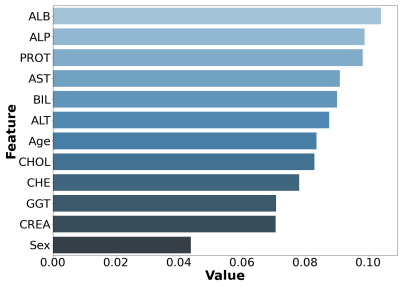


Fig. 3: XGBoost Feature Importance

The ALB had the highest contribution Figure 3 followed by the ALP and PROT respectively. Sex had the least contribution to the XGBoost model.

4.3 Confusion matrix plot

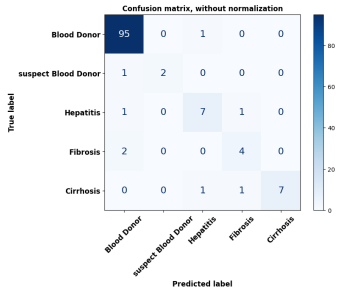
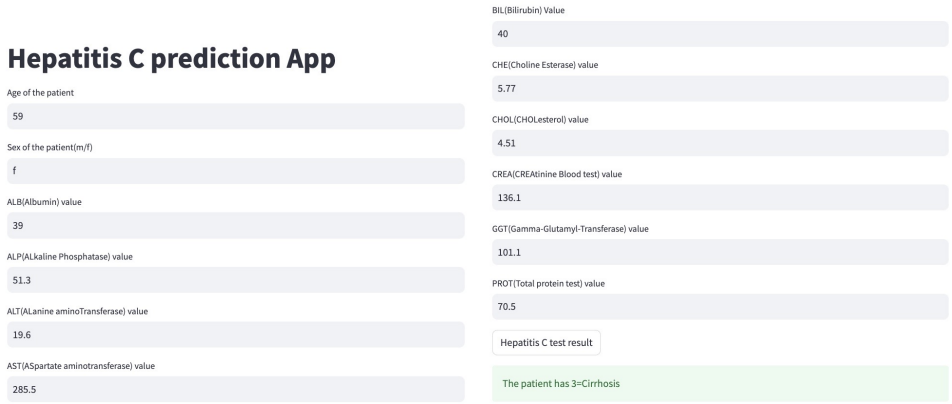


Fig. 4: XGBoost Confusion matrix

Figure 4 presents the confusion matrix obtained from the testing set using the XGBoost classifier. In this matrix, each row corresponds to the true label, representing the patients' status, while each column represents the predicted label. The model accurately predicted all 96 cases of Blood Donors, and correctly identified 2 (out of 3) suspected Blood Donors, 6 (out of 9) hepatitis cases, 4 (out of 6) Fibrosis cases, and 7 (out of 9) Cirrhosis cases.

4.4 Presentation of the Web App

Using the XGBoost classifier, a web App was designed with Streamlit and the result is presented in figure 5.



(a) First part of the Web App (b) Second part of the Web App

Fig. 5: Web App realised with Streamlit

Based on the given example, the model's prediction indicates that the patient has 'Cirrhosis'.

5 Conclusion

The integration of ML has significantly enhanced clinicians' diagnostic capabilities, leading to a notable reduction in the time required for disease diagnosis. This study proposes an ML framework based on CatBoost, XGBoost, LGBM, GNB, ExtraT, and RF algorithms, employing the Optuna optimization technique to classify and predict patients infected with Hepatitis C Virus (HCV). Through a comprehensive comparative analysis of the models using the test set, XGBoost outperformed all other models across all evaluation metrics. Furthermore, this best model was deployed using Streamlit, which facilitated smoother and easier testing of the application. Future research should focus on augmenting the model's efficiency by incorporating additional Hepatitis C-related variables. Additionally, gather more patient data to improve the model's performance and assess the robustness of the model by evaluating it on patient data collected from different regions.

Declarations

- Funding
  This research did not receive any specific grant from funding agencies in the public, commercial, or not-for profit sectors.
- Conflict of interest
  The authors declare no conflicts of interest.
- Consent for publication
  All authors have read and agreed to the published version of the manuscript.
- Availability of data and materials
  Publicly available datasets were analyzed in this study. This data can be found here: https://archive.ics.uci.edu/ml/datasets/HCV+data.
- Code availability
  The code is available upon request from the corresponding author.

Appendix A Best Hyperparameters for the XGBoost model

Table A1: Best Hyperparameters for XGBoost

Hyperparameter	Value
max_depth	4
learning_rate	0.1969097619840202
n_estimators	197
min_child_weight	1
gamma	0.09693404552771179
subsample	0.036768173724319515
colsample_bytree	0.19587409453980753
reg_alpha	0.41354766675033994
reg_lambda	0.009420448414595634

References

[1] Kalra, A., Yetiskul, E., Wehrle, C.J., Tuma, F.: Physiology, liver (2018)

[2] Marcellin, P., Asselah, T., Boyer, N.: Fibrosis and disease progression in hepatitis c. Hepatology **36**(S1), 47–56 (2002)

[3] Gerber, M.A.: Pathology of hepatitis c. FEMS microbiology reviews **14**(3), 205–210 (1994)

[4] Organization, W.H., et al.: Global hepatitis report, 2017: World health organization; 2017. PMID: None (2017)

[5] Zingaretti, C., De Francesco, R., Abrignani, S.: Why is it so difficult to develop a hepatitis c virus preventive vaccine? Clinical Microbiology and Infection **20**, 103–109 (2014)

[6] Organization, W.H., *et al.*: Hepatitis C Rapid Diagnostic Tests for Professional Use And/or Self-testing. World Health Organization, ??? (2022)

[7] Ma, L., Yang, Y., Ge, X., Wan, Y., Sang, X.: Prediction of disease progression of chronic hepatitis c based on xgboost algorithm. In: 2020 International Conference on Robots & Intelligent System (ICRIS), pp. 598–601 (2020). IEEE

[8] Oleiwi, A.: Development of diagnostic decision making for chronic hepatitis c virus patients by various supervised predictive model. Journal of Advanced Research in Dynamical and Control Systems **12**, 3113–3123 (2020)

[9] Nandipati, S.C., XinYing, C., Wah, K.K.: Hepatitis c virus (hcv) prediction by machine learning techniques. Applications of Modelling and Simulation **4**, 89–100 (2020)

[10] Safdari, R., Deghatipour, A., Gholamzadeh, M., Maghooli, K.: Applying data mining techniques to classify patients with suspected hepatitis c virus infection. Intelligent Medicine **2**(04), 193–198 (2022)

[11] Oladimeji, O.O., Oladimeji, A., Olayanju, O.: Machine learning models for diagnostic classification of hepatitis c tests. Frontiers in Health Informatics **10**(1), 70 (2021)

[12] Alotaibi, A., Alnajrani, L., Alsheikh, N., Alanazy, A., Alshammasi, S., Almusairii, M., Alrassan, S., Alansari, A.: Explainable ensemble-based machine learning models for detecting the presence of cirrhosis in hepatitis c patients. Computation **11**(6), 104 (2023)

[13] Lichtinghagen, R., Klawonn, F., Hoffmann, G.: Hcv data data set. Available online:(accessed on 19 march 2023), UCI Machine Learning Repository (2020)

[14] Cohen, I., Huang, Y., Chen, J., Benesty, J., Benesty, J., Chen, J., Huang, Y., Cohen, I.: Pearson correlation coefficient. Noise reduction in speech processing, 1–4 (2009)

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

[16] Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2623–2631 (2019)

[17] Hancock, J.T., Khoshgoftaar, T.M.: Catboost for big data: an interdisciplinary review. Journal of big data **7**(1), 1–45 (2020)

[18] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: Catboost: unbiased boosting with categorical features. Advances in neural information processing systems **31** (2018)

[19] Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)

[20] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y.: Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems **30** (2017)

[21] Anand, M.V., KiranBala, B., Srividhya, S., Younus, M., Rahman, H., et al.: Gaussian naïve bayes algorithm: A reliable technique involved in the assortment of the segregation in cancer. Mobile Information Systems **2022** (2022)

[22] Breiman, L.: Random forests. Machine learning **45**, 5–32 (2001)

[23] Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine learning **63**, 3–42 (2006)

[24] Sokolova, M., Japkowicz, N., Szpakowicz, S.: Beyond accuracy, f-score and roc: A family of discriminant measures for performance evaluation. In: Sattar, A., Kang, B.-h. (eds.) AI 2006: Advances in Artificial Intelligence, pp. 1015–1021. Springer, Berlin, Heidelberg (2006)

[25] Raschka, S.: An Overview of General Performance Metrics of Binary Classifier Systems (2014)