



Liver Cirrhosis Prediction System Using Random Forest

¹AMITHA S, ²PRATHEEKSHA B R, ³SUSHMITHA C K, ⁴UMME HUSNAIN Z M, ⁵YASHAS N U

¹Assistant Professor, ²Student, ³ Student, ⁴ Student, ⁵ Student

¹Department of Computer Science and Engineering,

¹ATME College of Engineering, Mysuru, India

Abstract: -- Liver cirrhosis is a severe and chronic liver condition that often goes undiagnosed until advanced stages, leading to significant morbidity and mortality. Early detection is critical for effective treatment and improved patient outcomes. This project focuses on the development of a predictive model for liver cirrhosis using the Random Forest algorithm, a robust machine learning technique known for its high accuracy and interpretability. The system utilizes a dataset of patient information, including demographic details, liver function test results, and other relevant clinical factors. After preprocessing the data by handling missing values and encoding categorical variables, the Random Forest classifier is trained to predict whether a patient is at risk of liver cirrhosis. The model demonstrates its reliability in identifying potential cirrhosis cases. A Flask-based web application is integrated with the predictive model to provide an accessible interface for medical professionals and researchers.

Keywords: Liver Cirrhosis, Random Forest Classifier, Predictive Model, Flask Web Application

I. INTRODUCTION

Liver cirrhosis is a significant global health concern, often diagnosed in its advanced stages, which reduces the chances of early intervention and treatment [1]. Traditional medical diagnostic techniques play a crucial role in detecting cirrhosis; however, integrating predictive models can enhance early diagnosis and support healthcare professionals in making informed decisions [2]. By identifying cirrhosis risk at an early stage, timely medical intervention can be implemented to slow disease progression, improve treatment outcomes, and reduce healthcare costs [2-3].

Accurate prediction of cirrhosis risk can also help optimize the distribution of medical resources, ensuring that patients receive timely and effective care. This project aims to develop a predictive system that utilizes machine learning techniques to assess cirrhosis risk. The solution includes a trained predictive model and a user-friendly Flask-based web application [3]. The web interface is designed to provide medical professionals and patients with a secure and accessible tool for accurate liver cirrhosis prediction, aiding in better disease

management and personalized treatment plans.

II. RELATED WORK

Research conducted at the University of Michigan highlights the challenge of reducing disease management expenses for liver cirrhosis patients. To address this, researchers developed a predictive analytics-based method to identify individuals at high risk. Their findings suggest that early identification could enable timely and effective treatment, particularly for high-risk patients. Compared to previous methods, their approach demonstrated improved accuracy.

Additionally, researchers in [6] explored how factors such as gender and obesity contribute to the prevalence of liver cirrhosis in different populations. Their study emphasized the importance of incorporating these variables—such as sex, body mass index (BMI), bilirubin levels, and alanine aminotransferase (ALT)—into predictive models to enhance accuracy and aid healthcare professionals in formulating effective treatment strategies. The study reported notable differences in contributing factors among patients. For instance, it was observed that males over 60 years old tend to have a lower mean BMI than females under 60. Furthermore, individuals with higher BMI values were found to be at greater risk of early-stage complications related to liver cirrhosis [7].

In another study [8], a comparative analysis was conducted on three data mining techniques—decision trees, Naïve Bayes, and neural networks—for predicting liver cirrhosis virus infection. Similarly, research in [9] focused on advancements in artificial intelligence and machine learning for predicting esophageal varices, a common complication in chronic liver cirrhosis patients. Their findings indicated that among 24 examined factors, nine were identified as the most significant for predictive analysis using their proposed approach.

III. PROCEDURE AND METADODOLOGY

In today's complex systems, a well-designed methodology is crucial for ensuring the efficiency, scalability, and maintainability of the system. A system architecture provides a blueprint for the system's components, interactions, and data flows, enabling developers to design and implement the system in a structured and organized manner.

The methodology of our Liver Cirrhosis Prediction System Using Random Forest is described below. The system consists of the following steps:

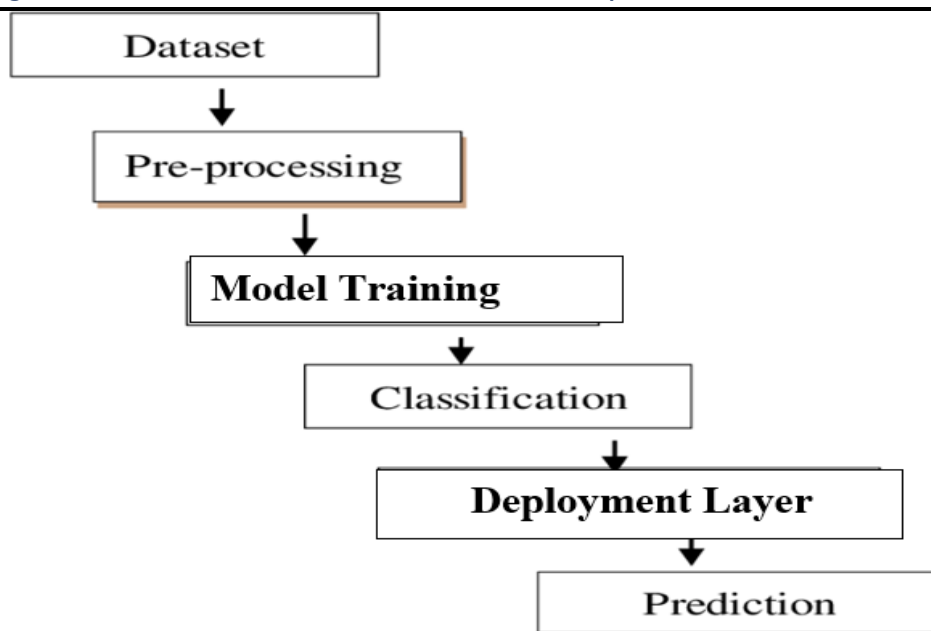


Fig 1. Methodology

A. Data Collection

This step involves gathering the dataset containing relevant medical parameters for liver cirrhosis diagnosis. The dataset is collected from Kaggle it includes features like age, total bilirubin, albumin, platelet count, prothrombin etc[].

B. Dataset Preprocessing

In this step, raw data is cleaned and prepared for the model. The first step involves reading and preparing the dataset to ensure data quality and consistency. Dataset Loading: The dataset is loaded using the Pandas library

1). Handling missing values

It is the crucial step in data preprocessing to ensure the dataset's integrity and improve model performance. Missing values can occur due to various reasons such as incomplete data collection or errors in data entry. One common approach is imputation, where missing values are replaced with statistical measures. The feature containing missing (NaN) values, are filled with the median of that feature using fillnan(), ensuring that the dataset remains consistent and complete for effective machine learning model training..

2). Statistical Description:

It provides insights into the central tendencies, variability, and distribution of the dataset features, serving as a critical step in data analysis. A statistical summary of the dataset is generated using dataset.describe(), which provides metrics such as mean, median, standard deviation, min, max, and quartiles to understand the data distribution.

3). Multicollinearity Check:

A correlation matrix (heatmap) is generated to assess multicollinearity between features. Features with high correlation (greater than 0.85) are dropped. For example, 'Total_Bilirubin' and 'Direct_Bilirubin' are highly correlated (0.87), so 'Direct_Bilirubin' is dropped to avoid Redundancy.

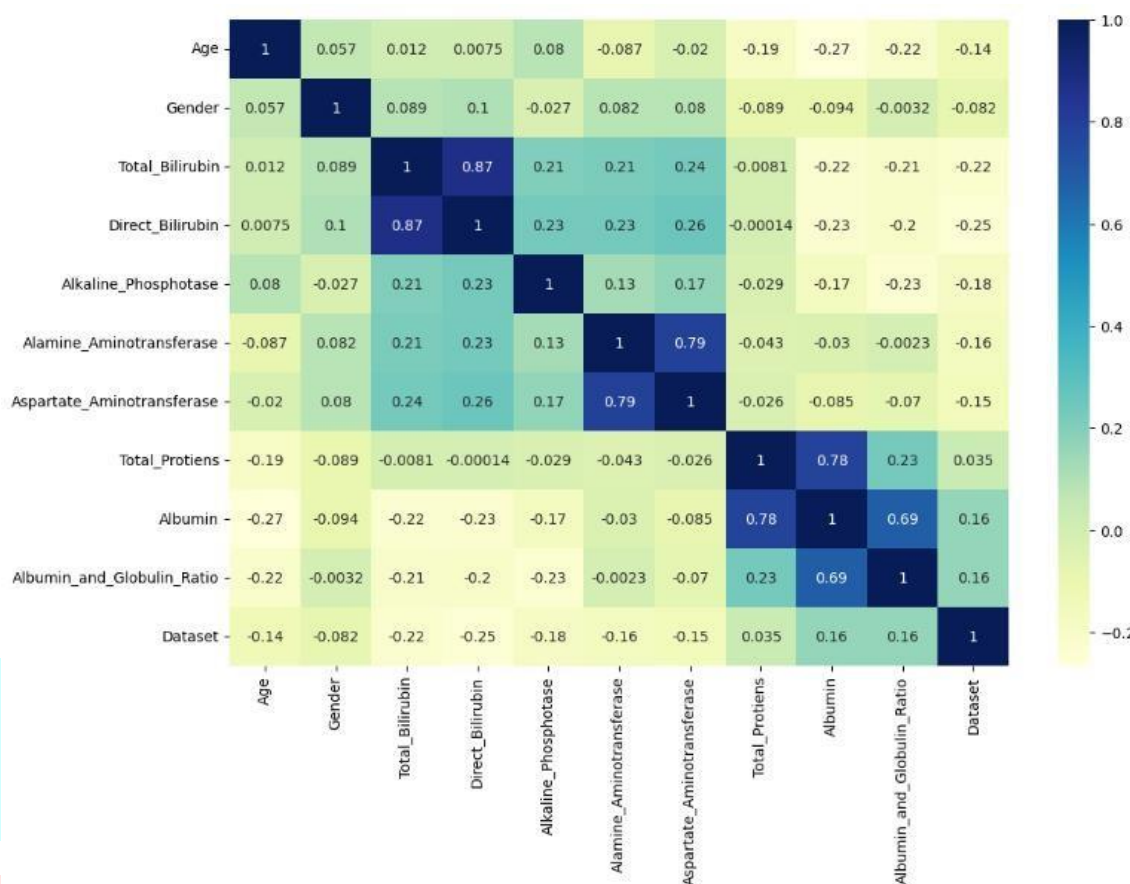


Fig 2. Correlation Matrix

4). Outlier Detection and Treatment:

Interquartile Range (IQR): The IQR is used to calculate the lower and upper boundaries for several features like 'Total_Bilirubin', 'Albumin', etc. Values beyond these boundaries are considered extreme outliers and are capped to the upper boundary.

5). Handling Class Imbalance:

The dataset has an imbalance in the target variable (liver disease vs non-liver disease). This is addressed using SMOTETomek: SMOTE generates synthetic samples for the minority class to balance the dataset. Tomek links remove noisy or redundant examples to ensure better model performance.

```
Before SMOTE : Counter({1: 416, 2: 167})
After SMOTE : Counter({1: 396, 2: 396})
```

Fig 3. Before and after applying SMOTE

C. Train-Test Split

The dataset is split into training and testing sets using an 80:20 ratio for model training and evaluation.

D. Feature Selection

The SelectKBest method is applied to rank features based on their importance using the Chi-Square score. The top 9 features are selected for training the model.

E. Model Training and Evaluation

The Random Forest Classifier is used as the primary prediction model. Three ensemble techniques are tested for comparison: Random Forest Classifier, A baseline model is trained and evaluated. AdaBoost Classifier, An additional boosting algorithm is implemented. Gradient Boosting Classifier, A gradient-based boosting algorithm is compared. The Random Forest Classifier was employed as the primary prediction model, with additional comparison to ensemble techniques such as AdaBoost and Gradient Boosting Classifiers. To enhance model performance, hyperparameter optimization was conducted using RandomizedSearchCV for a broad exploration of hyperparameter combinations, followed by a refined search with GridSearchCV, confirming Random Forest as the best-performing model. The model's performance was evaluated using standard metrics, including accuracy, confusion matrix, and classification report (precision, recall, and F1-score). For seamless deployment, the trained Random Forest model was serialized and saved as a .pkl file using the pickle library. This approach enabled efficient reuse of the model without retraining and was integrated into a Flask-based web application, allowing real-time predictions through user inputs while ensuring scalability, reduced computational overhead, and smooth production deployment.

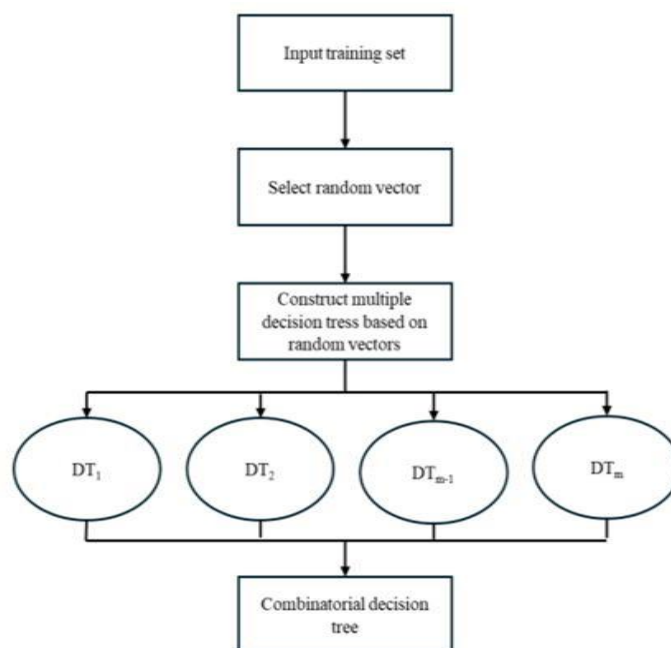
F. Applied Algorithms

One of the most often found illnesses in medical research is the ever-increasing occurrence of Liver cirrhosis. Following algorithms were used in this paper.

- Random Forest Classifier
- Ada boost Classifier
- Gradient Boost Classifier

1). Random Forest Classifier

Fig 4. Random Forest Classifier



The Random Forest Classifier is a powerful and versatile machine learning algorithm widely applied to classification and regression tasks. It operates by constructing multiple decision trees during training and combining their outputs to make predictions. The process begins with bootstrap sampling, where random subsets of the training data are created with replacement, and each subset is used to train a different decision tree—an approach known as bagging (Bootstrap Aggregating). To reduce correlation between trees and enhance robustness, random feature selection is applied at each split in a tree, selecting only a subset of features for consideration. Each decision tree independently predicts the outcome, and the results are aggregated through majority voting for classification tasks or by averaging for regression tasks. This ensemble approach improves generalization and model accuracy while reducing the risk of overfitting.

2). AdaBoost Classifier

The AdaBoost Classifier (Adaptive Boosting) is an ensemble learning technique that combines multiple weak learners to form a strong and accurate classifier. It operates by sequentially training weak models, such as decision stumps, and adjusting their focus on difficult-to-classify instances. Initially, equal weights are assigned to all training samples. After training a weak learner on the weighted dataset, the weighted error rate is computed. Weights of misclassified samples are increased to ensure the next weak learner pays more attention to these challenging examples, while weights for correctly classified samples are decreased. Each weak learner is assigned a weight based on its accuracy, and their predictions are combined through weighted voting. This process is repeated for a predefined number of iterations or until the error rate converges, resulting in a robust and efficient predictive model.

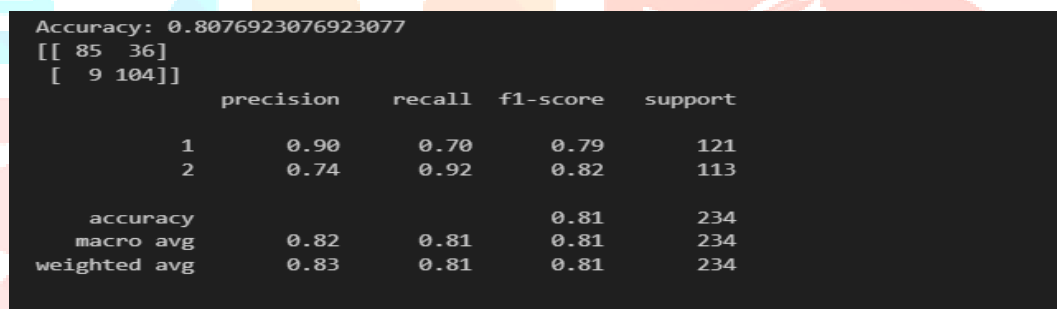
3). Gradient Boost Classifier

The Gradient Boosting Classifier is an advanced ensemble learning technique that builds a strong predictive model by sequentially adding weak learners, typically decision trees, and optimizing their performance using gradient descent. The process begins by initializing the model with a base prediction, such as the mean or median for regression tasks or log odds for classification. At each step, residuals, which represent the errors between the true target values and the current model's predictions, are computed. A weak learner is then trained to predict these residuals, effectively learning from the model's mistakes. The predictions of the weak learner are added to the current model's predictions, scaled by a learning rate to control the contribution of each learner. This process is repeated iteratively, with each new learner correcting the errors of the combined model so far, until convergence or a specified number of iterations is reached. The final model aggregates the base prediction and the weighted contributions of all weak learners, providing a powerful and accurate classification tool.

IV. Model Evaluation

1). Random Forest Classifier

Figure 5 shows the Random Forest model's assessment report.



```

Accuracy: 0.8076923076923077
[[ 85 36]
 [  9 104]]

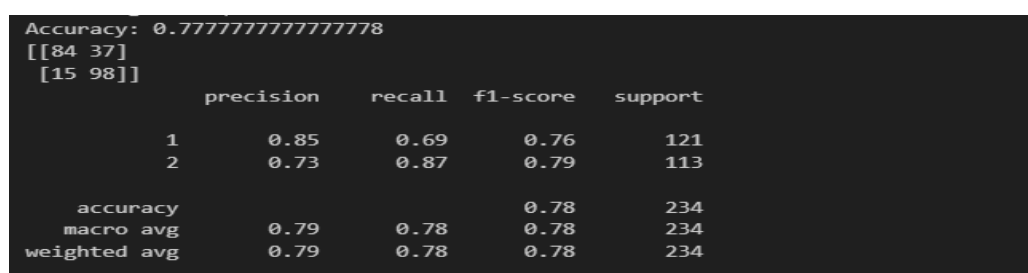
```

	precision	recall	f1-score	support
1	0.90	0.70	0.79	121
2	0.74	0.92	0.82	113
accuracy			0.81	234
macro avg	0.82	0.81	0.81	234
weighted avg	0.83	0.81	0.81	234

Fig 5. Random Forest Classification Report

This text displays various metrics evaluating a classification model's performance: Accuracy: 0.80 - This represents the overall percentage of correct predictions. Confusion Matrix: [[85 36], [9 104]] - This matrix summarizes the model's predictions. 85: True Positives (correctly classified as class 1), 36: False Positives (incorrectly classified as class 1), 9: FalseNegatives (incorrectly classified as class 2), 104: True Negatives (correctly classified as class 2)

2). AdaBoost Classifier



```

Accuracy: 0.7777777777777778
[[84 37]
 [15 98]]

```

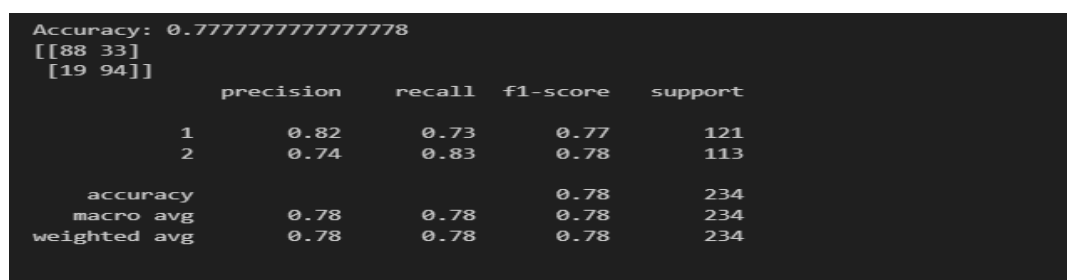
	precision	recall	f1-score	support
1	0.85	0.69	0.76	121
2	0.73	0.87	0.79	113
accuracy			0.78	234
macro avg	0.79	0.78	0.78	234
weighted avg	0.79	0.78	0.78	234

Fig 6. AdaBoost Classification report

Accuracy: 0.77, this represents the overall percentage of correctly classified instances.

The precision for class 1 is 0.85, meaning 85% of the instances predicted as class 1 were actually class 1. The precision for class 2 is 0.73. The model has a recall of 0.69 for class 1 and 0.87 for class 2. The model has an f1-score of 0.76 for class 1 and 0.79 for class 2. The model has a support of 121 for class 1 and 113 for class 2. The model has a macro average of 0.79 for precision, recall and f1-score. The model has a weighted average of 0.79 for precision, recall and f1-score

3). Gradient Boost Classifier



```

Accuracy: 0.7777777777777778
[[88 33]
 [19 94]]

```

	precision	recall	f1-score	support
1	0.82	0.73	0.77	121
2	0.74	0.83	0.78	113
accuracy			0.78	234
macro avg	0.78	0.78	0.78	234
weighted avg	0.78	0.78	0.78	234

Fig 7. Gradient Boost Classification

The accuracy obtained in Gradient Boost is 0.78, Precision in class 1 is 0.82 , Recall is 0.73 F1-score is 0.77 and Support is 121 in class 2 we have precision , recall , F1-score and support as 0.74, 0.83, 0.78, 113 respectively.

V. IMPLEMENTATION

The implementation phase of the project focuses on translating the planned structure and functionalities into executable code, ensuring that the system performs the required computations effectively. During this phase, key decisions are made regarding platform selection, programming languages, and tools to be used, considering factors such as the operational environment, required processing speed, security, and system-specific needs.

A. Data and Model Preparation

Data preprocessing involved handling multicollinearity by dropping redundant features and filling missing values using the median for "Albumin_and_Globulin_Ratio." Outliers were addressed using the interquartile range (IQR), and class imbalance was mitigated using SMOTE. Exploratory Data Analysis (EDA) was conducted using histograms and heatmaps to visualize correlations and inform feature selection decisions. SelectKBest with a chi-square test identified the top features for model training, reducing computation time. Random Forest, AdaBoost, and Gradient Boosting models were trained and evaluated using a balanced dataset (70% training, 30% testing). Hyperparameter tuning was performed using RandomizedSearchCV and GridSearchCV, with Random Forest emerging as the best-performing model based on accuracy, precision, recall, and F1-score. The optimized Random Forest model was serialized into a .pkl file using Python's pickle library for seamless integration into real-world applications.

B. User Interface

A user-friendly interface was developed with role-based access control:

Admin and Receptionist Login/Registration: Admins manage system operations, while receptionists handle patient record management. **Doctor Login and Prediction Functionality:** Doctors log in to access a prediction interface where patient data (age, gender, and test results) are inputted for liver cirrhosis prediction using the trained Random Forest model.

The model used for prediction is loaded using `joblib.load()` from the specified file path. The trained Random Forest model (Liver2.pkl) is used to make predictions based on the user's input

VI. CONCLUSION

The liver cirrhosis detection system developed in this project combines machine learning and web application technology to deliver an efficient, user-friendly solution for healthcare professionals. The Random Forest model, achieving an accuracy of 81%, effectively predicts liver cirrhosis based on patient health parameters. The web application features a secure, role-based interface for admins, receptionists, and doctors, enabling seamless user authentication, patient management, and prediction result visualization. Robust preprocessing techniques, including handling missing values, outliers, and class imbalances, ensured the reliability and accuracy of the predictions.

The project showcases the successful integration of backend technologies such as Flask and MySQL with a responsive frontend design, enhancing user experience and system functionality. This application provides a scalable and cost-effective solution for early liver cirrhosis detection, reducing human error and aiding healthcare professionals in decision-making. Despite challenges such as optimizing hyperparameters and addressing class imbalances, the project highlights significant technical achievements and learning opportunities. Future enhancements could involve integrating real-time data processing, exploring advanced models like deep learning for improved accuracy, and implementing stricter security measures to comply with healthcare regulations. This system demonstrates the potential of technology to address critical healthcare challenges, paving the way for improved diagnostics and patient care.

REFERENCES

- [1] Hanif, Ishtiaque, and Mohammad Monirujjaman Khan. "Liver cirrhosis prediction using machine learning approaches." *2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2022
- [2] Khan, S., Haider, A. (2020). Handling Imbalanced Datasets in Liver Disease Prediction using Random Forest with SMOTE. *International Conference on Data Science and Applications (ICDSA)*, September 2020.
- [3] Zhai, Yinping, et al. "Artificial intelligence-based evaluation of prognosis in cirrhosis." *Journal of*

Translational Medicine

22.1 (2024): 933.

- [4] Bala, Bindu, and Sunny Behal. "A Brief Survey of Data Preprocessing in Machine Learning and Deep Learning Techniques." 2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC). IEEE, 2024.
- [5] S. C. R. Nandipati, C. XinYing and K. K. Wah, "Liver cirrhosis virus (LD) prediction by machine learning techniques," Applications of Modelling and Simulation, vol. 4, pp. 89–100, 2020.
- [6] T. M. K. Motawi, N. A. H. Sadik, D. Sabry, N. N. Shahin and A. S., "Fahim, rs2267531, a promoter SNP within glypican-3 gene in the X chromosome, is associated with hepatocellular carcinoma in Egyptians," Scientific Reports, vol. 9, no. 1, pp. 1–10, 2019.
- [7] M. Reiser, B. Wiebner and J. Hirsch, "Neural-network analysis of socio-medical data to identify predictors of undiagnosed Liver cirrhosis virus infections in Germany (DETECT)," Journal of Translational Medicine, vol. 17, no. 1, pp. 1–7, 2019.
- [8] J. Bresnick, "Predictive analytics identify high risk Liver cirrhosis patients," Health IT Analytics, 2015. [Online]. Available: <https://healthitanalytics.com/news/predictive-analytics-identify-high-risk-hepatitis-cpatients/> [Accessed: 15-Oct-2020].
- [9] Y. C. Tsao, J. Y. Chen, W. C. Yeh, Y. S. Peng and W. C. Li, "Association between visceral obesity and Liver cirrhosis infection stratified by gender: A cross-sectional study in Taiwan," BMJ Open, vol. 7, no. 11, pp. e017117, 2017.
- [10] Singh, P., Patel, N. (2021). Comparative Analysis of Machine Learning Models for Liver Disease Diagnosis. IEEEAccess, 9, 54328-54335.doi:10.1109/ACCESS.2021.3056602.
- [11] Choudhury, A., Mallick, P. (2019). Prediction of Liver Diseases using Random Forest. International Journal of Scientific & Technology Research (IJSTR), 8(12), 754-760. ISSN: 2277- 8616.
- [12] Lurie, Yoav, et al. "Non-invasive diagnosis of liver fibrosis and cirrhosis." World journal of gastroenterology 21.41 (2015): 11567.
- [13] Wieczorek, Mikolaj, et al. "A deep learning approach for detecting liver cirrhosis from volatolomic analysis of exhaled breath." Frontiers in Medicine 9 (2022): 992703.
- [14] Mallikharjuna Rao, K., Ghanta Saikrishna, and Kundrapu Supriya. "Data preprocessing techniques: emergence and selection towards machine learning models-a practical review using HPA dataset." Multimedia Tools and Applications 82.24 (2023): 37177-37196.
- [15] Kosolwattana, Tanapol, et al. "A self-inspected adaptive SMOTE algorithm (SASMOTE) for highly imbalanced data classification in healthcare." BioData Mining 16.1 (2023): 15.
- [16] Kosolwattana T, Liu C, Hu R, Han S, Chen H, Lin Y. A self- inspected adaptive SMOTE algorithm (SASMOTE) for highly imbalanced data classification in healthcare. BioData Mining. 2023 Apr 25;16(1):15
- [17] Tufail, Shahid, et al. "Advancements and challenges in machine learning: A comprehensive review of models, libraries, applications, and algorithms." Electronics 12.8 (2023): 1789.

- [18] Brownlee, Jason. Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python. Machine Learning Mastery, 2020.
- [19] R. Huang, H. Rao, M. Yang, Y. Gao, J. Wang et al., “Noninvasive measurements predict liver fibrosis well in Liver cirrhosis virus patients after direct-acting antiviral therapy,” Digestive Diseases and Sciences, vol. 65, no. 5, pp. 1491–1500, 2020

