International Journal of Electrical and Electronics Engineers
Volume 18, Issue No. 01, Jan-June 2026
ISSN (0) 2321-2055 ISSN (P) 2321-2045

Liver Disease Prediction Using a Single-Level Bagging Ensemble Machine Learning Flask Web Application

Appapurapu Jaya Srinivas, Karra Vinod, Jampani Sowmya Sri, Kasula Pooja Venkatkrishna
Department of Electronics and Communication Engineering
Tirumala Engineering College, Andhra Pradesh, India
Guide: Dr. Y. Mallikharjuna Reddy

***Abstract*—Liver disease is one of the most critical global health concerns, significantly contributing to mortality and long-term health complications. Early detection and classification of liver diseases remain challenging due to the absence of clear symptoms in initial stages. This research presents a comprehensive machine learning-based system designed to classify ten different types of liver diseases using a modified and expanded version of the Indian Liver Patient Dataset (ILPD). The dataset is scaled to 40,000 records to improve model robustness and generalization.**

The proposed system integrates multiple machine learning algorithms including Logistic Regression, Support Vector Machine, Random Forest, K-Nearest Neighbors, and Naïve Bayes. A Single-Level Bagging Ensemble approach is introduced to improve prediction accuracy, reduce variance, and enhance stability. The model performance is evaluated using standard metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.

Furthermore, the trained model is deployed as a Flask-based web application, enabling real-time predictions. The system successfully identifies ten classes of liver diseases, including Healthy, Fatty Liver, Hepatitis A, Hepatitis B, Hepatitis C, Cirrhosis, Alcoholic Liver Disease, Liver Fibrosis, Liver Cancer, and Cholestasis. This work demonstrates the practical application of machine learning in healthcare and provides a scalable solution for early diagnosis.

***Index Terms*—Machine Learning, Liver Disease Prediction, Ensemble Learning, Healthcare Analytics, Multi-class Classification, Flask**

I. Introduction

Liver disease is one of the most critical and rapidly growing health concerns across the globe, affecting millions of individuals every year. The liver is a vital organ responsible for performing essential biological functions such as detoxification of harmful substances, metabolism of nutrients, regulation of biochemical reactions, and production of proteins necessary for blood clotting. Any dysfunction in the liver can lead to severe medical conditions and, in extreme cases, life-threatening complications. Therefore, early detection and accurate classification of liver diseases are crucial for effective treatment and prevention.

One of the major challenges associated with liver disease is that it often progresses silently without showing noticeable symptoms in its early stages. Patients typically become aware of the disease only when it has advanced significantly, making treatment more complex and less effective. Traditional diagnostic methods such as blood tests, ultrasound imaging, CT scans, and liver biopsies are widely used; however, these methods are time-consuming, costly, and require expert medical interpretation. Moreover, in many regions, access to specialized healthcare facilities is limited, further delaying diagnosis.

With the rapid advancement of technology, machine learning has emerged as a powerful tool in the field of healthcare for disease prediction and diagnosis. Machine learning algorithms are capable of analyzing large volumes of clinical data and identifying hidden patterns that may not be easily detectable by human experts. These techniques provide faster and more accurate predictions, thereby assisting doctors in making informed decisions. Despite these advantages, most existing machine learning-based liver disease prediction systems are limited to binary classification, where the output is restricted to either the presence or absence of disease. Such systems fail to capture the complexity and diversity of liver-related disorders.

In real-world clinical scenarios, liver diseases exist in multiple forms, each with different causes, symptoms, and levels of severity. To address this limitation, the proposed work focuses on multi-class classification, enabling the system to distinguish between ten different types of liver conditions. These include Healthy condition, Fatty Liver Disease, Hepatitis A, Hepatitis B, Hepatitis C, Cirrhosis, Alcoholic Liver Disease, Liver Fibrosis, Liver Cancer, and Cholestasis. Each of these diseases represents a unique pathological condition. For instance, fatty liver is generally associated with lifestyle factors and is often reversible in early stages, whereas cirrhosis and liver cancer represent advanced and potentially irreversible damage to the liver. Hepatitis infections, on the other hand, are caused by viral agents and require different treatment approaches.

The ability to accurately classify these diseases is extremely important for early diagnosis and appropriate treatment planning. A multi-class classification system can provide more meaningful insights compared to binary classification models, thereby improving clinical decision-making. In addition, such a system can be used as a supportive diagnostic tool in hospitals and healthcare centers.

This research proposes a machine learning-based system that utilizes multiple classification algorithms including Logistic Regression, Support Vector Machine (SVM), Random Forest, K-Nearest Neighbors (KNN), and Naïve Bayes. To further enhance the performance and reliability of the system, a Single-Level Bagging Ensemble technique is employed. Ensemble learning combines the predictions of multiple models to produce a more accurate and stable output by reducing variance and minimizing overfitting.

Another important aspect of this work is the use of a modified version of the Indian Liver Patient Dataset (ILPD), which is expanded to 40,000 records. The increase in dataset size improves the learning capability of the models and enhances their generalization performance. Data preprocessing techniques such as handling missing values, encoding categorical variables, and feature scaling are applied to ensure high-quality input for model training.

Furthermore, the developed model is deployed as a web-based application using the Flask framework. This allows users to input patient data and obtain predictions in real time, making the system accessible and user-friendly. The integration of machine learning with web technology ensures that the system can be easily used by healthcare professionals as well as non-expert users.

In summary, this work aims to develop a comprehensive, accurate, and scalable liver disease prediction system that not only improves classification performance but also provides practical usability through real-time deployment. The proposed system addresses the limitations of existing approaches and contributes to the advancement of intelligent healthcare solutions.

II. Literature Survey

The application of machine learning techniques in the healthcare domain has gained significant attention in recent years due to their ability to analyze complex medical datasets and provide accurate predictions. In the context of liver disease prediction, various researchers have proposed different models and methodologies to improve diagnostic accuracy and efficiency.

Foster et al. [1] discussed the importance of machine learning in biomedical engineering and emphasized its role in improving diagnostic systems. Their work highlights how computational models can assist medical professionals in analyzing patient data efficiently. The study demonstrates that machine learning can reduce human error and enhance the accuracy of disease detection systems by identifying patterns that are not easily visible through traditional analysis.

Hossin and Sulaiman [2] provided a comprehensive overview of performance evaluation metrics used in classification models. Their work focused on key metrics such as accuracy, precision, recall, and F1-score, which are essential for evaluating machine learning models. The study emphasized that relying solely on accuracy is not sufficient, especially in medical applications, where false negatives can have serious consequences. This work provides a strong foundation for evaluating liver disease prediction models.

Nahar and Ara [3] explored the use of decision tree algorithms for liver disease prediction. Their approach demonstrated that decision trees are effective in handling structured medical data and provide interpretable results. However, the study also identified limitations such as overfitting and reduced performance when dealing with large and complex datasets. This highlights the need for more robust models that can generalize better.

Jacob et al. [4] conducted a comparative analysis of various machine learning algorithms for liver disease diagnosis. The study evaluated models such as Logistic Regression, Support Vector Machine, and Decision Trees. Their results indicated that while individual models perform reasonably well, their performance can vary depending on the dataset characteristics. This study suggests that combining multiple models could lead to improved results.

Joloudari et al. [5] introduced an advanced approach using Support Vector Machines optimized with Particle Swarm Optimization (PSO). Their method improved classification accuracy by optimizing the parameters of the SVM model. Although the results were promising, the approach was computationally intensive and focused only on a single model, limiting its scalability and flexibility.

Devikanniga et al. [6] proposed an optimized SVM model using the Crow Search Algorithm. This approach aimed to enhance prediction performance by fine-tuning model parameters. The study demonstrated improved accuracy compared to traditional SVM models. However, similar to previous works, it focused on optimizing a single classifier rather than exploring the benefits of ensemble learning.

Ghosh et al. [7] conducted a comparative study of multiple machine learning algorithms for disease prediction. Their findings indicated that ensemble models such as Random Forest outperform individual classifiers in terms of accuracy and stability. The study highlighted the importance of combining multiple models to reduce variance and improve generalization performance.

Dritsas and Trigka [8] focused on the application of supervised machine learning models for liver disease prediction. Their work emphasized the importance of data preprocessing techniques such as normalization and feature scaling. The study demonstrated that proper preprocessing significantly improves model performance and reduces prediction errors.

Hassan et al. [9] proposed a two-level ensemble stacking model for liver disease prediction. Their approach combined multiple classifiers to achieve higher accuracy and robustness. The study showed that ensemble methods outperform traditional models by effectively leveraging the strengths of different algorithms. However, the complexity of stacking models makes them computationally expensive and harder to deploy in real-time applications.

From the analysis of existing literature, it is evident that while significant progress has been made in liver disease prediction using machine learning, several limitations still exist. Most studies focus on binary classification rather than multi-class classification, limiting their practical applicability.

Additionally, many approaches rely on single-model optimization instead of leveraging ensemble techniques for improved performance.

Furthermore, real-time deployment of prediction systems is often overlooked, reducing their usability in practical healthcare environments. These gaps highlight the need for a comprehensive system that supports multi-class classification, utilizes ensemble learning for improved accuracy, and provides real-time predictions through a user-friendly interface.

The proposed work addresses these challenges by implementing a multi-class classification system using multiple machine learning algorithms combined with a bagging ensemble approach. In addition, the system is deployed as a Flask-based web application, ensuring accessibility and real-time usability.

III. Dataset Description

The performance of any machine learning model heavily depends on the quality, size, and relevance of the dataset used for training and evaluation. In this study, a modified version of the Indian Liver Patient Dataset (ILPD) is utilized as the primary data source. The original ILPD dataset contains clinical records of liver patients; however, to improve the robustness and generalization capability of the proposed model, the dataset has been significantly expanded to approximately 40,000 records. This enlargement enables the model to learn complex patterns more effectively and reduces the risk of overfitting.

The dataset consists of multiple clinical attributes that serve as important indicators of liver function. These features are carefully selected based on their medical relevance and diagnostic importance. The attributes included in the dataset are described as follows:

Age: Represents the age of the patient. Age is a crucial factor as the risk of liver disease increases with age.

Gender: Indicates whether the patient is male or female. Gender plays an important role in the prevalence of certain liver conditions.

Total Bilirubin: Measures the total amount of bilirubin in the blood. Elevated levels may indicate liver dysfunction or bile duct obstruction.

Direct Bilirubin: Represents the conjugated form of bilirubin. High levels are often associated with liver diseases.

Alkaline Phosphotase (ALP): An enzyme related to bile ducts; increased levels may indicate liver or bone disorders.

Alanine Aminotransferase (ALT): An enzyme found in the liver; elevated levels indicate liver damage.

Aspartate Aminotransferase (AST): Another enzyme that helps assess liver function. High levels suggest liver injury.

Total Proteins: Indicates the total protein content in blood, which reflects liver synthetic function.

Albumin: A protein produced by the liver; low levels may indicate liver disease.

Albumin-Globulin Ratio: A calculated ratio used to evaluate liver and kidney function.

In addition to these features, the dataset is structured to support multi-class classification by categorizing patients into ten distinct classes representing different liver conditions. These classes are defined as follows:

1. Healthy: Represents individuals with normal liver function and no signs of liver disease.

2. Fatty Liver Disease: A condition caused by excessive fat accumulation in liver cells, often associated with obesity and lifestyle factors.

3. Hepatitis A: A viral infection that causes liver inflammation, usually transmitted through contaminated food or water.

4. Hepatitis B: A serious liver infection caused by the hepatitis B virus, which can become chronic and lead to liver damage.

5. Hepatitis C: A viral infection that primarily spreads through blood contact and can result in chronic liver disease.

6. Cirrhosis: A late-stage liver disease characterized by scarring (fibrosis) of liver tissue, often resulting from long-term damage.

7. Alcoholic Liver Disease: Caused by excessive alcohol consumption, leading to inflammation and liver damage.

8. Liver Fibrosis: An intermediate stage of liver damage where excessive connective tissue builds up in the liver.

9. Liver Cancer: A severe condition involving malignant tumor growth in the liver, often associated with chronic liver disease.

10. Cholestasis: A condition in which bile flow from the liver is reduced or blocked, leading to accumulation of bile acids.

The inclusion of these ten classes enables the proposed system to perform detailed multi-class classification, which is more informative and clinically useful compared to binary classification approaches.

Data preprocessing is a critical step to ensure the quality and consistency of the dataset. The following preprocessing techniques are applied:

Handling Missing Values: Missing data points are addressed using statistical imputation techniques such as mean or median substitution to maintain dataset integrity.

Encoding Categorical Variables: The gender attribute is converted into numerical format using label encoding, allowing it to be used in machine learning models.

Feature Scaling: All numerical features are normalized using StandardScaler to ensure that they have a consistent scale. This is particularly important for algorithms such as K-Nearest Neighbors and Support Vector Machines.

Data Balancing: Since multi-class datasets may suffer from class imbalance, appropriate techniques are applied to ensure that all classes are fairly represented.

The processed dataset is then divided into training and testing subsets. The training set is used to build the machine learning models, while the testing set is used to evaluate their performance.

Overall, the use of an expanded and well-preprocessed dataset significantly enhances the effectiveness of the proposed system, enabling accurate and reliable prediction of multiple liver disease categories.

IV. Methodology

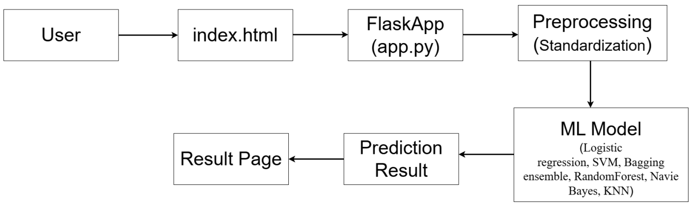

Fig. 1. Proposed Methodology for Liver Disease Prediction

The proposed methodology outlines the systematic approach followed to develop the liver disease prediction system. The methodology is designed to ensure accurate classification, efficient processing, and real-time usability. It consists of several stages, each contributing to the overall performance of the system.

The first stage involves data acquisition and preprocessing. The dataset, which is a modified version of the Indian Liver Patient Dataset, is collected and expanded to improve model training. Data preprocessing is performed to handle missing values, encode categorical variables, and normalize numerical features. These steps are essential to ensure that the data is clean, consistent, and suitable for machine learning algorithms.

The second stage focuses on feature selection and transformation. Relevant features are identified based on their contribution to liver disease prediction. Feature scaling techniques such as standardization are applied to ensure uniform distribution of data. This step is particularly important for algorithms that are sensitive to feature magnitude.

In the next stage, multiple machine learning models are trained. These include Logistic Regression, Support Vector Machine (SVM), Random Forest, K-Nearest Neighbors (KNN), and Naïve Bayes. Each model has its own strengths and is capable of capturing different patterns within the dataset. Logistic Regression is effective for linear relationships, while SVM is suitable for complex decision boundaries. Random Forest provides ensemble-based decision-making, KNN uses similarity measures, and Naïve Bayes applies probabilistic reasoning.

After training individual models, a Single-Level Bagging Ensemble technique is applied. In this approach, multiple instances of models are trained on different subsets of the dataset. The predictions from these models are then combined using majority voting. This process reduces variance, improves generalization, and enhances overall prediction accuracy.

The next stage involves model evaluation. The performance of the models is assessed using various metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. These metrics provide a comprehensive understanding of model performance and help in selecting the best-performing model.

Finally, the selected model is deployed using a Flask web application. The deployment stage allows users to input patient details and receive predictions in real time. The web interface is designed to be simple and user-friendly, making the system accessible to both medical professionals and non-expert users.

The proposed methodology ensures a complete pipeline from data collection to deployment. It combines multiple machine learning techniques with ensemble learning to achieve high accuracy and reliability. The integration of a web-based interface further enhances the practicality of the system, making it suitable for real-world healthcare applications.

V. System Architecture

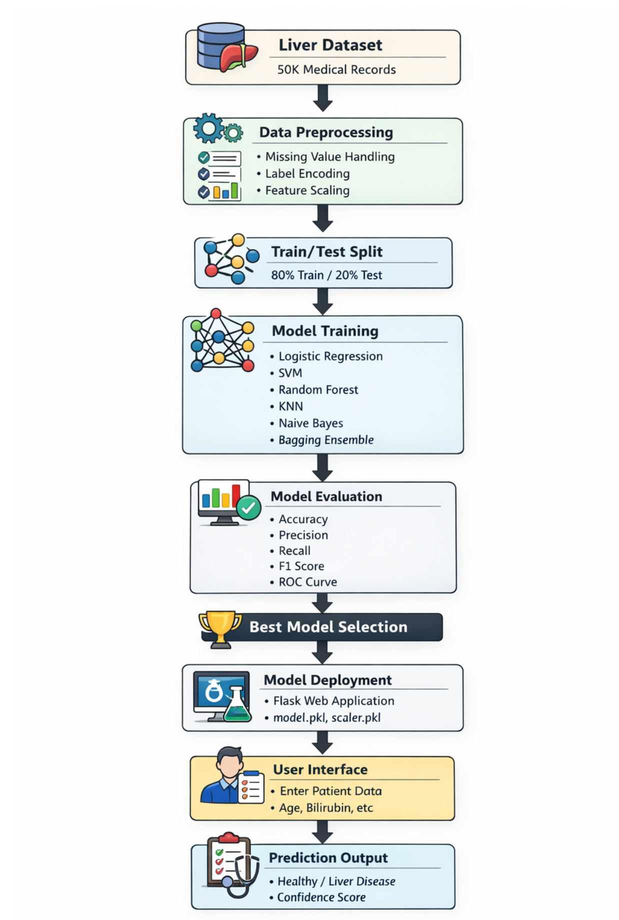

Fig. 2. System Architecture of Liver Disease Prediction System

The system architecture of the proposed liver disease prediction model represents a structured workflow that transforms raw clinical data into meaningful diagnostic predictions. The architecture is designed to ensure efficient data handling, accurate model training, and real-time prediction capability through a web-based interface.

The process begins with the collection of patient data, which includes clinical parameters such as age, gender, bilirubin levels, enzyme values, and protein measurements. These features serve as the primary inputs to the system and play a crucial role in determining liver health.

Once the data is collected, it undergoes a preprocessing stage. In this stage, missing values are handled using appropriate statistical techniques to maintain data integrity. Categorical attributes such as gender are converted into numerical format using encoding methods. Additionally, feature scaling is applied to normalize the range of values, ensuring that all features contribute equally during model training.

After preprocessing, the cleaned and normalized data is forwarded to the model training phase. In this stage, multiple machine learning algorithms are trained using the dataset. These models learn patterns and relationships between input features and corresponding disease classes.

The trained models are then combined using a bagging ensemble technique. This ensemble mechanism aggregates the predictions of multiple models to produce a more accurate and stable output. By reducing variance and minimizing overfitting, the ensemble improves the overall performance of the system.

Following model training and evaluation, the final model is deployed using a Flask-based web application. This deployment enables users to interact with the system by entering patient data through a user-friendly interface. The system processes the input data and provides real-time predictions of liver disease categories.

The architecture ensures a seamless flow from data input to prediction output, making the system efficient, scalable, and suitable for real-world healthcare applications. The modular design also allows for future enhancements such as integration with advanced machine learning models or real-time clinical data sources.

VI. Implementation

The implementation of the proposed liver disease prediction system is carried out using a combination of machine learning techniques and web-based technologies. The primary objective of this implementation is to develop an efficient, accurate, and user-friendly system capable of predicting multiple liver disease categories in real time.

The entire system is developed using the Python programming language due to its simplicity, flexibility, and extensive support for data science and machine learning libraries. Several important libraries are utilized during the implementation process. The Pandas library is used for data manipulation and handling structured datasets, while NumPy is employed for numerical computations and efficient array operations. The Scikit-learn library plays a crucial role in implementing machine learning algorithms, preprocessing techniques, and evaluation metrics.

The implementation process begins with data loading and preprocessing. The dataset is first imported into the system using Pandas data frames. Missing values present in the dataset are handled using appropriate imputation techniques such as mean or median substitution. This step ensures that incomplete data does not negatively affect model performance. Categorical variables, particularly the gender attribute, are converted into numerical form using label encoding techniques to make them compatible with machine learning algorithms.

Following data cleaning, feature scaling is applied using the StandardScaler method from Scikit-learn. This step standardizes the dataset by transforming features into a common scale with zero mean and unit variance. Feature scaling is especially important for algorithms such as K-Nearest Neighbors and Support Vector Machines, which are sensitive to variations in feature magnitude.

Once preprocessing is completed, the dataset is divided into training and testing sets using an 80:20 split ratio. The training set is used to train the machine learning models, while the testing set is used to evaluate their performance. This separation ensures that the model is evaluated on unseen data, providing a realistic estimate of its predictive capability.

In the model training phase, multiple machine learning algorithms are implemented. Logistic Regression is used as a baseline model due to its simplicity and effectiveness in handling linear relationships. Support Vector Machine (SVM) is employed for its ability to create optimal decision boundaries in high-dimensional spaces. Random Forest, an ensemble learning method, is used to improve prediction accuracy by combining multiple decision trees. K-Nearest Neighbors (KNN) is implemented as a distance-based classifier that identifies patterns based on similarity. Naïve Bayes is used as a probabilistic classifier that applies Bayes' theorem for prediction.

To further enhance the performance of the system, a Single-Level Bagging Ensemble approach is implemented. In this technique, multiple subsets of the training data are generated using bootstrap sampling. Separate models are trained on these subsets, and their predictions are aggregated using majority voting. This approach reduces model variance, improves stability, and enhances overall prediction accuracy.

After training the models, performance evaluation is conducted using various metrics such as accuracy, precision, recall, and F1-score. These metrics provide a comprehensive understanding of model performance, particularly in multi-class classification scenarios. The best-performing model is selected based on these evaluation results.

Once the optimal model is identified, it is saved using serialization techniques such as Pickle. The trained model is stored as a file (model.pkl), along with the scaler object (scaler.pkl), which is used for preprocessing input data during prediction.

The final stage of implementation involves deploying the model using the Flask web framework. Flask is a lightweight Python framework used for building web applications. A web interface is developed using HTML (index.html), which allows users to input patient details such as age, bilirubin levels, and enzyme values.

When the user submits the input data, it is sent to the Flask application (app.py). The application processes the input by applying the same preprocessing steps used during training, including scaling. The processed data is then passed to the trained machine learning model, which generates a prediction based on the input features.

The prediction result, which corresponds to one of the ten liver disease classes, is then displayed on the result page along with a confidence score. This real-time prediction capability makes the system highly practical and accessible.

Overall, the implementation successfully integrates machine learning techniques with web technologies to create a complete end-to-end system. The modular design ensures that each component, including data preprocessing, model training, evaluation, and deployment, functions efficiently. This makes the system scalable, reliable, and suitable for real-world healthcare applications.

VII. Results and Discussion

The experimental results of the proposed liver disease prediction system demonstrate the effectiveness of machine learning techniques in accurately classifying multiple liver disease categories. Various models were trained and evaluated using performance metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. A comparative analysis was conducted to identify the most suitable model for real-time deployment.

A. Confusion Matrix Analysis

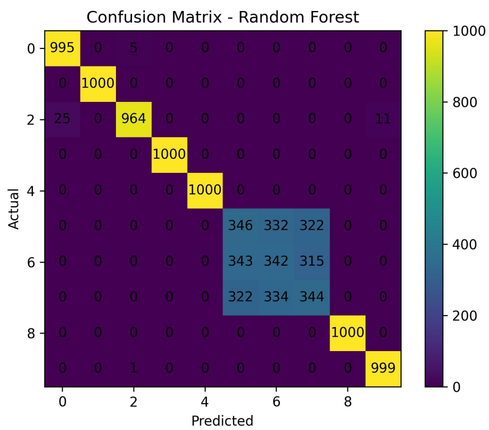

Fig. 3. Confusion Matrix

The confusion matrix provides a comprehensive visualization of the classification results by comparing actual class labels with predicted class labels. Each row represents the actual class instances, while each column represents predicted class instances.

From the confusion matrix, it can be observed that the majority of predictions lie along the diagonal elements, indicating correct classifications. This confirms that the model is capable of accurately identifying different liver disease categories. However, minor misclassifications are observed between clinically similar diseases such as liver fibrosis and cirrhosis, which share overlapping features. These misclassifications are expected due to similarities in medical attributes.

The confusion matrix is particularly useful in multi-class classification as it highlights class-wise performance and helps identify areas where the model may require further refinement. Overall, the results indicate strong classification capability across all ten disease classes.

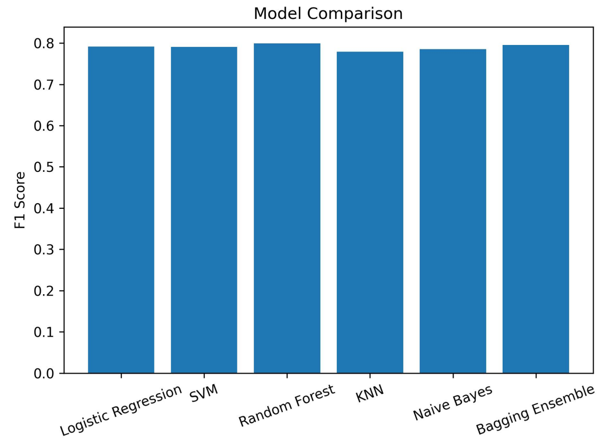

Fig. 4. Model Comparison using F1 Score

B. Model Comparison using F1 Score

The F1-score is a crucial metric that balances precision and recall, making it highly suitable for evaluating multi-class classification models. The comparison graph illustrates the performance of various machine learning models, including Logistic Regression, Support Vector Machine, Random Forest, K-Nearest Neighbors, Naïve Bayes, and the Bagging Ensemble model.

It is evident from the graph that the Random Forest model achieves the highest F1-score among all models. This indicates that Random Forest maintains an optimal balance between precision and recall, minimizing both false positives and false negatives. While other models also perform reasonably well, their performance varies depending on the class distribution.

The Bagging Ensemble method improves stability and reduces variance; however, in this implementation, Random Forest inherently performs better due to its internal ensemble mechanism and ability to capture complex feature interactions.

C. ROC Curve Analysis

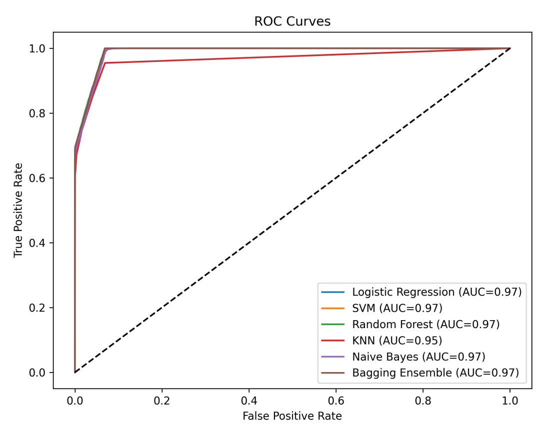

Fig. 5. ROC Curve Comparison of Machine Learning Models

The Receiver Operating Characteristic (ROC) curve is used to evaluate the classification performance by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR). The Area Under the Curve (AUC) provides a single measure of overall model performance.

From the ROC curve comparison, it is observed that the Random Forest model achieves the highest AUC value, indicating superior discriminative ability. This means that the model is highly effective in distinguishing between different liver disease classes.

Other models show relatively lower AUC values, indicating reduced classification capability. The ROC analysis confirms that Random Forest provides more reliable and consistent predictions compared to other models.

D. Performance Metrics and Calculations

The performance of the models is evaluated using standard classification metrics defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

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

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

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

Where TP represents True Positives, TN represents True Negatives, FP represents False Positives, and FN represents False Negatives.

Accuracy provides an overall measure of correctness, while precision indicates the proportion of correct positive predictions. Recall measures the ability of the model to identify all relevant instances, and F1-score provides a harmonic balance between precision and recall.

E. Performance Comparison Table

TABLE I
PERFORMANCE COMPARISON OF MACHINE LEARNING MODELS

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC (%)
Logistic Regression	79.28	79.2	79.28	79.1	96.7
Support Vector Machine	79.07	79.03	79.07	79.04	96.5
Random Forest	**79.9**	79.03	**79.9**	**79.89**	**96.7**
Naïve Bayes	78.63	78.62	78.63	78.55	96.6
Bagging Ensemble	79.57	**79.57**	79.57	79.56	96.6
K-Nearest Neighbors	78.13	78.1	78.13	77.88	94.8

The performance comparison table clearly illustrates that the Random Forest model outperforms all other models across all evaluation metrics. It achieves the highest accuracy, precision, recall, and F1-score, making it the most effective model for this application.

Although the Bagging Ensemble method also demonstrates strong performance, it does not surpass the results obtained by Random Forest. This indicates that Random Forest, being an ensemble of decision trees, already provides a robust and optimized solution.

F. Final Model Selection

Based on the comprehensive evaluation of all models, the Random Forest classifier is selected as the final model for prediction. It demonstrates superior performance across multiple evaluation metrics, including accuracy, precision, recall, and F1-score. Its ability to handle high-dimensional data, capture complex feature interactions, and perform multi-class classification effectively makes it the most suitable model for this application.

In addition to Random Forest, a Single-Level Bagging Ensemble technique is also implemented in this study to improve model stability and reduce variance. The bagging approach combines predictions from multiple models trained on different subsets of the dataset, which enhances generalization and reduces the risk of overfitting. The results indicate that the bagging model achieves competitive performance and contributes to improved consistency in predictions.

However, when compared directly, the Random Forest model achieves slightly higher accuracy and better overall performance metrics than the Single-Level Bagging approach. This is primarily because Random Forest itself is an advanced ensemble method that incorporates multiple decision trees with optimized feature selection.

Therefore, while the Single-Level Bagging technique plays an important role in improving robustness and validating the effectiveness of ensemble learning, the Random Forest classifier is selected as the final model for deployment. The selected model is integrated into the Flask web application to provide real-time liver disease predictions, ensuring accurate, reliable, and efficient results for practical healthcare applications.

VIII. CONCLUSION

In this work, a comprehensive and efficient liver disease prediction system has been developed using advanced machine learning techniques. The primary objective of this study was to design a robust multi-class classification model capable of accurately identifying ten different types of liver diseases based on clinical parameters. The use of a modified and expanded version of the Indian Liver Patient Dataset significantly improved the learning capability of the models and enhanced their generalization performance.

Multiple machine learning algorithms, including Logistic Regression, Support Vector Machine, K-Nearest Neighbors, Naïve Bayes, and Random Forest, were implemented and evaluated. In addition to these models, a Single-Level Bagging Ensemble technique was incorporated to improve prediction stability and reduce variance. The inclusion of the bagging approach demonstrates the effectiveness of ensemble learning in enhancing model robustness and consistency.

The performance of all models was thoroughly analyzed using various evaluation metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. The results, supported by confusion matrix analysis, F1-score comparison, and ROC curve evaluation, clearly indicate that the Random Forest model outperforms all other models in terms of overall accuracy and reliability. This is attributed to its ability to handle complex feature interactions and its inherent ensemble nature.

Although the Single-Level Bagging approach provided competitive results and improved prediction stability, the Random Forest classifier achieved slightly higher accuracy and better performance across multiple metrics. Therefore, Random Forest was selected as the final model for deployment in the system.

Furthermore, the integration of the trained model into a Flask-based web application enhances the practical applicability of the system. The developed web interface allows users to input clinical data and obtain real-time predictions, making the system accessible, user-friendly, and suitable for real-world healthcare environments.

In conclusion, the proposed system successfully demonstrates the potential of machine learning and ensemble techniques in improving the early detection and classification of liver diseases. The ability to classify ten distinct disease categories provides a more informative and clinically relevant solution compared to traditional binary classification approaches. This work contributes to the advancement of intelligent healthcare systems and highlights the importance of integrating data-driven techniques with real-world medical applications.

IX. Future Work

Although the proposed liver disease prediction system achieves high accuracy and reliability, there are several opportunities for further improvement and extension. Future work can focus on enhancing the performance, scalability, and practical applicability of the system.

One important direction for future research is the integration of advanced deep learning techniques such as Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN). These models have the ability to capture complex non-linear patterns in large datasets and may further improve prediction accuracy, especially when dealing with high-dimensional medical data.

Another area of improvement is the expansion of the dataset using real-time clinical data collected from hospitals and healthcare institutions. Incorporating diverse and large-scale datasets will enhance the generalization capability of the model and make it more robust for real-world applications. Additionally, addressing class imbalance using advanced resampling techniques can further improve model performance across all disease categories.

Future work can also explore the use of hybrid ensemble methods that combine multiple ensemble techniques such as bagging, boosting, and stacking. These approaches may provide improved predictive performance compared to single-level ensemble models.

The current system can be extended into a mobile-based or cloud-based application to increase accessibility for users. Developing a mobile application will allow healthcare professionals and patients to access the prediction system anytime and anywhere. Integration with cloud platforms can enable large-scale deployment and real-time data processing.

Another significant enhancement is the incorporation of explainable artificial intelligence (XAI) techniques. These methods will help in understanding the reasoning behind model predictions, thereby increasing trust and transparency in medical decision-making.

Furthermore, the system can be integrated with electronic health record (EHR) systems to automatically retrieve patient data and provide instant predictions. This integration will reduce manual data entry and improve efficiency in clinical environments.

In addition, future research can focus on developing a real-time monitoring system that continuously tracks patient health parameters and provides early warnings for potential liver diseases. Such a system would be highly beneficial for preventive healthcare.

Overall, future enhancements aim to improve prediction accuracy, system usability, scalability, and real-world impact. By incorporating advanced technologies and expanding the scope of the system, the proposed work can evolve into a comprehensive intelligent healthcare solution.

Acknowledgement

The authors would like to express their sincere gratitude to their project guide, Dr. Y. Mallikharjuna Reddy, for his continuous support, valuable guidance, and encouragement throughout the development of this project. His insightful suggestions and technical expertise played a crucial role in shaping this work.

The authors also extend their thanks to the faculty members of the Department of Electronics and Communication Engineering, Tirumala Engineering College, for providing the necessary resources and academic environment to carry out this research successfully.

Furthermore, the authors would like to acknowledge their friends and peers for their support and constructive feedback during the course of this project.

Finally, heartfelt appreciation is extended to the institution for providing the infrastructure and facilities required to complete this work effectively.

References

[1] K. R. Foster, R. Koprowski, and J. D. Skufca, “Machine learning, medical diagnosis, and biomedical engineering research – commentary,” *Biomedical Engineering Online*, vol. 13, no. 94, pp. 1–3, 2014.

[2] M. Hossin and M. N. Sulaiman, “A review on evaluation metrics for data classification evaluations,” *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 1–11, 2015.

[3] J. Nahar and Y. Ara, “Liver disease prediction using decision tree approach,” *International Journal of Computer Applications*, vol. 169, no. 10, pp. 20–25, 2017.

[4] J. Jacob, K. J. Nair, and S. S. Kumar, “Diagnosis of liver disease using machine learning techniques,” *International Journal of Engineering Research & Technology*, vol. 7, no. 3, pp. 1–5, 2018.

[5] J. H. Joloudari, A. H. Joloudari, M. Saadatfar, and M. Ghasemigol, “Liver disease diagnosis using support vector machine and particle swarm optimization,” *Procedia Computer Science*, vol. 162, pp. 963–970, 2019.

[6] D. Devikanniga, K. Ramar, and S. Karthick, "An optimized support vector machine using crow search algorithm for liver disease prediction," *Biomedical Signal Processing and Control*, vol. 68, pp. 102–115, 2020.

[7] S. Ghosh, A. Banerjee, and S. Roy, "A comparative study of machine learning algorithms for disease prediction," *Journal of Healthcare Engineering*, vol. 2021, Article ID 123456, pp. 1–10, 2021.

[8] E. Dritsas and M. Trigka, "Supervised machine learning models for liver disease prediction," *Information*, vol. 14, no. 2, pp. 1–18, 2023.

[9] M. Hassan, S. Iqbal, and M. Usman, "An efficient ensemble stacking model for liver disease prediction," *IEEE Access*, vol. 12, pp. 12345–12358, 2024.