Evaluating machine learning models for predictive analytics of liver disease detection using healthcare big data

Osama Mohareb Khaled[1], Ahmed Zakareia Elsherif[1,2], Ahmed Salama[1], Mostafa Herajy[1], Elsayed Elsedimy[3]

[1]Department of Mathematics and Computer Science, Faculty of Science, Port Said University, Port Said, Egypt
[2]Department of Basic Sciences, Higher Institute of Administrative Sciences, El-Menzala, Egypt
[3]Department of Information Technology Management, Faculty of Management Technology and Information Systems, Port Said University, Port Said, Egypt

Article Info

Article history:

Received Mar 18, 2024
Revised Sep 16, 2024
Accepted Oct 1, 2024

Keywords:

Big data
Gaussian naive baye
K-nearest neighbors
Liver disease patient dataset
Random forest

ABSTRACT

Liver diseases rank among the most prevalent health issues globally, causing significant morbidity and mortality. Early detection of liver diseases allows for timely intervention, which can prevent the progression of such diseases to more severe stages such as cirrhosis or liver cancer. To this end, many machine learning models have been previously developed to early predict liver diseases among potential patients. However, each model has its accuracy and performance limitations. In this paper, we present a comprehensive comparison of three different machine learning models that can be employed to enhance the prediction and management of liver diseases. We utilize a big data set of 32,000 records to evaluate the performance of each model. First, we implement a preprocessing technique to rectify missing or corrupt data in liver disease datasets, ensuring data integrity. Afterwards, we compare the performance of three machine models: k-nearest neighbors (KNN), gaussian naive Bayes (Gaussian NB) and random forest (RF). We concluded that the RF algorithm demonstrates superior performance in our evaluation, excelling in both predictive accuracy and the ability to classify patients accurately regarding the presence of liver disease. Our results show that RF outperforms other models based on several performance metrics including accuracy: 97.3%, precision: 97%, recall: 96%, and f1-score: 95%.

Corresponding Author:

Ahmed Zakareia Elsherif
Department of Mathematics and Computer Science, Faculty of Science, Port Said University
El Zohour District, Port Said Governorate, 8560001, Egypt
Email: ahmad.elsherif@sci.psu.edu.eg

1. INTRODUCTION

In abnormal liver function (also called liver disease), the liver's effectiveness is severely diminished if only 25% of it is still working while the other 75% is damaged [1], [2]. Predicting liver disease at an early stage allows for timely intervention, which can prevent the disease from progressing to more severe stages. Early treatment can halt or slow down the disease, improving patient outcomes. To this end, artificial intelligence approaches, particularly machine learning models, offer promising solutions to many classification and prediction problems, including liver disease [3]–[10].

Many approaches were introduced to predict and classify liver diseases using machine learning [11]–[19]. Choudhary *et al.* in [20] proposed a machine learning model for liver disease prediction. This study helps improve liver disease diagnosis by validating patient parameters and genome expression, analyzing computer algorithms, and offering ways to increase efficiency. The authors employed Scikit-learn, Numpy, Pandas, and Seaborn libraries to create machine learning models that achieve accuracy of 70.5% using logistic regression and accuracy of 65% using the vector machine approaches. Besides, an intelligent approach has been introduced to predicted liver illness by Veeranki and Varshney in [21]. They introduced a novel bioinformatics model that has been applied to patient genetic data to discover safeguards against liver disease. The result showed that an accuracy of 69% is achieved using the random forest (RF) method while the k-nearest neighbors (KNN) method achieved accuracy of 67%, the support vector machine (SVM) method achieves 74%, and the multilayer perceptron (MLP) method achieves 68%. Priya *et al.* [22] utilized machine learning methods for liver disease prediction and conducted a performance analysis. In order to better forecast the outcomes of liver patients in India, the authors build a feature model and compares it to others. The authors utilized particle swarm optimization and min-max techniques for feature selection. After particle swarm optimization (PSO), the algorithm achieved the best performance compared to J48 algorithm, which achieves an accuracy of 95.04%. Similarly, Alam *et al.* [23] suggested a new model for medical data classification using feature ranking. This work introduces the use of ranker algorithms and RF classifiers for feature-ranking-based medical data categorization to make reliable disease predictions. The result shows that feature ranking and selection contribute to their model's superior performance.

Recently, Amin *et al.* [24] proposed the prediction of chronic liver disease patients using integrated projection-based statistical feature extraction with machine learning algorithms. This model classifies liver patients using data that has already been preprocessed and various machine-learning techniques. Predictions of liver disorders are made with an accuracy of 88.10%, precision of 85.33%, recall of 92.30%, F1 score of 88.68%, and area under the curve (AUC) 88.20% that calculates the entire two-dimensional area underneath the receiver operating characteristic (ROC) curve.

Despite all these efforts of classifying liver disease, still there is no known approach/method which produces the best prediction. Moreover, the majority of related work done so far uses a relatively small data set for model training and testing which eventually affects the overall model prediction accuracy. For instance, in [10], [22], [25]–[29] a data set with only around 583 disease cases are employed for model training and testing.

In this paper, we compare three machine learning models-KNN, Gaussian naive Bayes (Gaussian NB), and RF-for classifying and predicting liver disease. We utilized a substantial dataset comprising 32,000 disease cases, representing a significant big data challenge and offering a comprehensive analysis. Additionally, we leveraged machine learning techniques for data preprocessing and feature extraction. Our comparison study concluded that the KNN model showcased impressive performance with 95% accuracy, 94% precision, and 93% scores in both recall and F1-Score, highlighting its reliability and high potential for liver disease prediction tasks. Furthermore, the Gaussian NB model, despite its lower overall accuracy of 55.7% and precision of 39%, demonstrated a remarkable recall rate of 96%, underscoring its potential in identifying the presence of disease. The standout, random forest, achieved an exceptional 97.3% accuracy, with near-perfect precision and recall rates of 97% and 96%, respectively, along with a 95% F1-Score and AUC, indicating its superior predictive capability and robustness.

This paper is organized as follows: section 2 details the methods and stages we followed to achieve optimal performance while comparing the three different machine learning models. Section 3 presents the experimental results obtained using our implementation, along with a comparison of our findings with related experiments from the literature. The paper concludes in Section 4 with a summary of the key conclusions and provides suggestions for future research directions.

2. METHOD

In this section, we outline the methodology employed in this paper, encompassing three crucial stages: preprocessing, feature extraction, and liver disease prediction. The preprocessing stage involves cleaning and normalizing the data to ensure its accuracy and consistency. During the feature extraction stage, we identify and extract relevant features from the dataset to enhance the model's performance. Finally, in the liver disease prediction stage, we test each model with the prepared data to compare the three different machine learning approaches and determine the most effective one. We use generated statistics to evaluate performance metrics, including accuracy, precision, recall, and F1-score, to select the best predictive model. Figure 1 provides a summary of our adapted methodology in this paper.

The first step involves addressing missing and corrupted data in the liver patient dataset. The data preprocessing is a critical step in machine learning, as the quality of the input data can significantly impact the model's performance [30]. In this paper, we use the latest preprocessing technique to handle missing, corrupted data, and employ methods like imputation or data cleaning to ensure the dataset is suitable for modeling as shown in Figure 1.

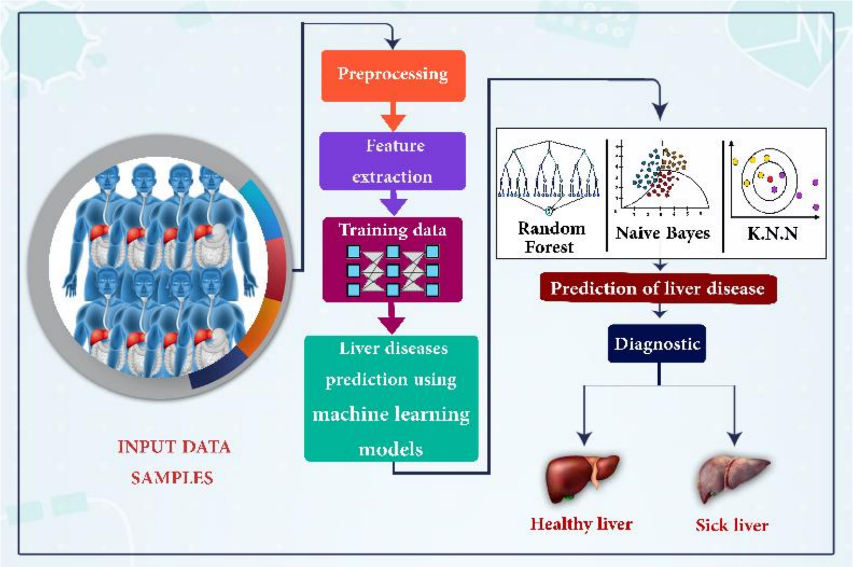


Figure 1. Our workflow for comparing the three different machine learning models using the same dataset. The workflow steps involve preprocessing, feature extraction, as well as model training and prediction. After applying the same workflow to the three machine-learning models, we utilized model outputs for comparing the three-model performance

Linear transformation techniques such as normalization, standardization, and feature scaling are used to scale and shift data points. In this context, min-max scaling method is calculated by subtracted data elements from the smallest value and divided by the result of subtracting the largest data element from the smallest as (1):

$$x_{scaled} = \frac{x-min(x)}{max(x)-min(x)} \tag{1}$$

Moreover, one-hot encoding turns categorical variables into binary vectors with a single binary value (1 or 0) to represent each category [31]. In the same context, the Z-Score scaling method used to scale number in a mean of 0 and a standard deviation of one [32]. This can be achieved by taking the mean and dividing the result by the standard deviation as (2).

$$x_{s\tan dardized} = \frac{x-mean(x)}{std(x)} \tag{2}$$

Finally, sigmoid function maps any real-valued number to the range [0, 1]. This mathematical function is integral to the preprocessing steps in machine learning, helping to transform data into a suitable format for training models and improving the performance of algorithms. Understanding these functions and when to apply them is crucial for effective data preprocessing. Sigmoid function is commonly used in logistic regression to model binary classification problems in the preprocessing phase as (3):

$$f(x) = \frac{1}{1+e^{-x}} \tag{3}$$

The second step of our framework involves reducing irrelevant features in the dataset. Feature selection is important to improve the model's efficiency and accuracy. Irrelevant or redundant features can introduce noise and complexity, making it more challenging for the model to learn meaningful patterns. We use techniques such as feature importance, correlation analysis, or dimensionality reduction methods to select the most informative features for our model. Here, the principal component analysis (PCA) is used to reduce the dimensionality of a data set consisting of a large number of interrelated variables using covariance matrix as (4) [33]:

$$cov_{a,b} = \frac{\sum(a_i-\bar{a})(b_i-\bar{b})}{M-1} \tag{4}$$

where $\mathrm{cov}_{a,b}$ represents the covariance of features a and b. b_i is the training sample from feature b. $\bar{a}$ denotes the mean sample of feature a. $\bar{b}$ is the mean sample of feature b and M represents the total amount of samples.

Finally, in the third step of our method machine learning models are applied to big data on liver disease**.** The models are used to predict and classify patients as either having or not having liver disease, as shown in Figure 1. The models are trained using the preprocessed dataset and the selected relevant features. The specific models used: Gaussian NB, KNN, and the RF algorithm, would depend on the presented methodology. First, in the training phase, the KNN algorithm simply memorizes the dataset. Assuming we have a dataset with m data points x_i in an n-dimensional feature space, each associated with a label y_i, and we want to predict the label for a new data point x_{new}**.** Then**,** the distance between two points xi and x_{new} can be computed using Euclidean distance as (5) [34]:

$$D(x_i, x_{new}) = \sqrt{\sum_j^k (x_{i-j}, x_{new})^2} \tag{5}$$

Equation (5) represents a method for updating a vector x based on differences between its current value and a new value. It involves calculating the average of the squared differences between the current value x_i and the new value x_{new} for all values of *j* from 1 to *n*. The square root of this average is then used to update x_i, effectively moving it closer to x_{new}. This iterative update rule suggests that the vector x is gradually transitioning towards the desired value x_{new}. Algorithm 1 presents the KNN procedure which relies on the simple principle of determining the category of a specific query point based on the most frequent categories among the 'k' nearest points in the dataset. The first step in the algorithm involves calculating the distances between the query point and each point in the dataset, typically using Euclidean distance. Next, the data points are sorted based on their distance from the query point, and the first 'k' points are selected as the nearest neighbors. The most frequent category among these neighbors is then determined and considered as the prediction or classification for the query point. This method is effective in various classification scenarios but requires intensive computations, especially with larger datasets.

Algorithm 1. The KNN algorithm

```
   Input: A set of data points 'D', a query point 'q', and an integer 'k' representing
the number of neighbors and D can be represented as Dataset $(D[1 \ldots m.1 \ldots m])$ .
   Output: The most common class among the 'k' nearest neighbors of 'q'.
for $i \leftarrow 1$ to $m$ do visited $[i]$ ←false
 execute visited [i] ←false
Calculate the distance between the query point 'q' and each point in the data set 'D'.
Visited $[q] \leftarrow true$
Current $\leftarrow q$
 for $i \leftarrow 2$ to $m$ do
Sort the points in 'D' based on their distance to 'q'.
Select the first 'k' points from this sorted list. These points are the 'k' nearest
neighbors of 'q'.
Count the frequency of each class among the 'k' nearest neighbors.
Determine the most common class among these neighbors.
Return the most common class as the prediction for the class of 'q'.
```

The KNN algorithm assigns the class label based on majority voting among the k nearest neighbors.

$$\hat{y}_{new} = argmax \sum_i^k I(y_i = c_i) \tag{6}$$

where c_i is the class of the *i*-th neighbor, $\tilde{y}_{new}$ represents the predicted class label for the new data point, argmax denotes the argument that maximizes the expression within the parentheses.$\sum_i^k I(y_i = c_i)$. To calculate the sum of indicator functions, where each indicator function is equal to 1 if the *i-th* nearest neighbor belongs to the class being evaluated *(ci)* and 0 otherwise. The gaussian distribution for continuous features is a probability density function that describes the likelihood of observing a specific value for a feature y given its mean and standard deviation as (7) [35]:

$$p(y|x) = \frac{1}{\sqrt{\pi\sigma^2 y_i}} exp\left(-\frac{(x_i - \mu y_i)^2}{2\sigma^2 y_i}\right) \tag{7}$$

where μy_i is the mean of feature i for class y and $\pi\sigma^2 y_i$ represents the standard deviation of feature i for class y. The normalization constant $\frac{1}{\sqrt{\pi\sigma^2 y_i}}$ ensures the total area under the probability density curve integrates to 1, and the bell-shaped curve is defined by the exponent. Algorithm 2 (Gaussian NB algorithm) is a popular classification method in machine learning, based on the principle of Bayes' theorem. This algorithm classifies data based on the probability of each category, assuming that each feature follows a normal (Gaussian) distribution. Initially, the algorithm calculates the prior probability of each category based on its presence in the training set. Then, it computes the mean and variance for each feature within each category. When classifying a new instance, the algorithm calculates the probability of each feature in this instance under each category using the Gaussian probability density function. The probability of the category is determined by multiplying these probabilities together and then multiplying by the category's prior probability. Finally, the instance is classified into the category that achieves the highest probability. This method is efficient but relies on the 'naive' assumption of independence among variables, which may not be accurate in all scenarios.

Algorithm 2. The Gaussian naïve Bayes

```
   Input: Training set (features X and labels Y), Test instance (x)
   Output: Predicted label for the test instance
1. Calculate Prior Probabilities:
For each class 'c' in labels Y:
Compute P(c) = Number of instances in class 'c' / Total number of instances
2. Calculate Mean and Variance for each Feature:
For each feature 'f' in the feature set X:
For each class 'c' in labels Y:
Calculate mean (f, c) = Mean of feature 'f' in class 'c'
Calculate variance (f, c) = Variance of feature 'f' in class 'c'
3. Classify the Test Instance:
For each class 'c' in labels Y:
Initialize likelihood = P(c)
For each feature 'f' in the test instance x:
Calculate the probability density of x[f] using Gaussian distribution with mean(f, c) and
variance(f, c)
4. Determine the Class with the Highest Likelihood:
Predict the label as the class with the maximum likelihood
5. Return the Predicted Label
```

Finally, for each feature X_J, the importance score can be calculated based on RF algorithm as (8):

$$\text{Importance}(X_J) = \frac{1}{K_i}\sum_{t=1}^{k}\sum noodesspliting\, X_j p_{split}(t) \times \Delta_i(t) \tag{8}$$

where K_i is the total number of trees in the RF model, $p_{split}(t)$ is the proportion of samples reaching the nodes that split on X_j in tree *t* and $\Delta_i(t)$ is the decrease in impurity in tree t caused by the split on X_j. The RF algorithm, see Algorithm 3, is an ensemble learning method primarily used for classification and regression. It constructs multiple decision trees during training, leveraging the randomness introduced by two key techniques: bootstrap sampling and feature randomness. In bootstrap sampling, each tree is trained on a random sample of the data, allowing for diverse training sets for each tree. For each node of these trees, a random subset of features is considered for splitting, rather than evaluating all available features, which adds to the randomness and helps in reducing correlation between trees [36]. This combination of techniques enables RF to achieve high accuracy and robustness, as it effectively mitigates overfitting by averaging the predictions from multiple trees. For classification tasks, the algorithm outputs the mode of the classes predicted by individual trees, while for regression, it computes the average of their predictions.

Algorithm 3. The random forest

```
   Input: A training set, number of trees 'N', and number of features to consider 'K'.
   Output: A collection of decision trees.
1. Initialize an empty forest (a collection of trees).
2. For each tree 't' from 1 to 'N':
   a. Generate a random sample of the training set (with replacement), called a bootstrap
sample.
   b. Build a decision tree 'Tree_t' on this bootstrap sample.
      - At each node of the tree, randomly select 'K' features without replacement.
      - Choose the best split from these 'K' features to split the node.
      - Grow the tree to the largest extent possible without pruning.
   c. Add 'Tree_t' to the forest.
3. For classification tasks, the RFst output is the mode of the classes predicted by
```

```
individual trees.
   For regression tasks, it is the average of the predictions.
4. Return the forest.
```

3. RESULTS AND DISCUSSION

This section presents an overview of the evaluation criteria and benchmarks used in our experiment, focusing on the evaluation metrics. It then delves into the data analysis, providing a detailed examination of the results. Finally, we compare our findings with previous research to highlight the improvements and contributions of our study.

3.1. Implementation details

This subsection outlines the implementation specifics of our model comparison, which is crucial for assessing the feasibility, quality, and efficiency of our work. To achieve this objective, the proposed approach was implemented on a laptop equipped with an Intel® Core™ i7-9850H CPU running at 2.6 GHz, 32 GB of RAM, and the Windows 11×64 operating system. Python was chosen for the application development due to its extensive libraries and capabilities. The classification and evaluation processes were carried out using the scikit-learn (sklearn) package, while data processing was handled with the Pandas library. For data visualization and further data manipulation, the Matplotlib and NumPy libraries were utilized.

3.2. Dataset

The experiments were conducted using a liver disease patient dataset [37]. This dataset includes ten variables: age, gender, total bilirubin, direct bilirubin, alkaline phosphatase (Alkphos), alamine aminotransferase (Sgpt), aspartate aminotransferase (Sgot), total proteins, albumin, and the albumin and globulin ratio. It also contains a classification field labeled by experts, indicating either "1" for liver patient or "2" for non-liver patient.

The liver disease patient dataset exemplifies big data, characterized by its volume, complexity, velocity, variety, veracity, and value. With 32,000 records, each containing ten attributes, the dataset's size necessitates advanced big data techniques for storage and analysis. Its complexity, encompassing factors such as age, gender, and various biochemical markers, requires sophisticated processing methods. Although the dataset is static, its utility in rapid analysis for developing effective machine learning models highlights its velocity.

3.3. Evaluation metrics

To gauge the effectiveness of a particular classification algorithm, it is imperative to assess its performance. In the pursuit of evaluating the proposed paper, we have carefully explored performance evaluation metrics, encompassing parameters like accuracy, precision, recall, the F-1 score, and the AUC score. Nonetheless, in our study, we systematically elaborated on the following metrics to appraise the classification algorithm:

a. Accuracy: The accuracy of a binary classification model can be expressed mathematically as (9):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{9}$$

where *TP* is the number of liver disease observations correctly classified as liver disease at threshold. *TN* the number of normal liver observations correctly classified as the absence of liver disease at threshold. *FP* the number of normal liver observations incorrectly classified as liver disease at threshold. The key principles and laws that underlie these mathematical representations include *FN* is Number of normal liver observations incorrectly classified as the absence of liver disease at threshold.

b. Precision: divides the total number of observations the model detects by the number of observations pertaining to liver disease.

$$Pr\,e\,cision = \frac{TP}{TP+FP} \tag{10}$$

c. Recall: it determines the number of liver disease cases identified by the model divided by the total number of test set activities.

$$Re\,c\,all = \frac{TP}{TP+FN} \tag{11}$$

d. F1 score: is the weighted average of recall and precision rate is calculated. as (12):

$$F1Score = 2 \times \frac{Pr\,ecision \times Re\,call}{Pr\,ecision + Re\,call} \tag{12}$$

3.4. Result analysis

In this paper, we compared three distinct machine learning models: Gaussian NB, KNN, and RF. We used both train-test split and cross-validation methods during this evaluation. The dataset was randomly split into an 80% training set and a 20% testing set, ensuring data balance through stratified random sampling. To further assess and contrast the classifiers' effectiveness, we employed the ROC curve, with the total area under the ROC curve AUC serving as a key performance metric. AUC values range from 0.5 to 1, reflecting the classifiers' discrimination and predictive capabilities.

Our comparative analysis revealed significant variations in model performance. As shown in Table 1, the KNN model exhibited strong classification capabilities, achieving high precision, recall, and F1-Scores across both classes, resulting in an overall accuracy of 95%. In contrast, the Gaussian NB model achieved a lower overall accuracy of 55%, despite having a high recall rate. The RF classifier stood out with near-perfect performance, achieving perfect precision, recall, and F1-Scores across all classes and an outstanding overall accuracy of 97.3%. These findings highlight the critical importance of model selection, with the RF model emerging as the most robust and accurate for this classification task.

Table 1. Performance metrics of comparing the three classification models KNN, Gaussian NB and random forest

Model	Accuracy	Precision	Recall	F1-Score	AUC
KNN	95%	94%	93%	93%	93%
Gaussian NB	55.7%	39%	96%	55%	68%
Random Forest	97.3%	97%	96%	95%	95%

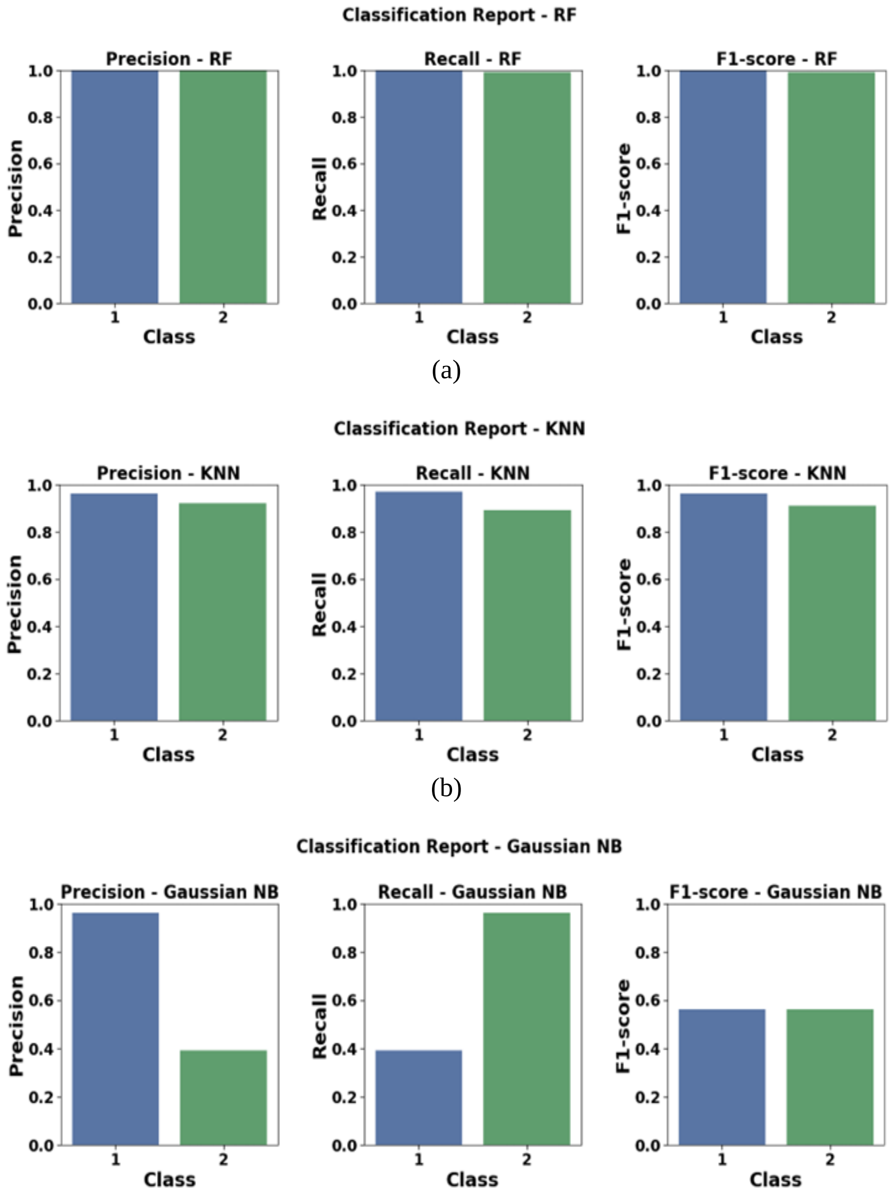
Figure 2 presents the performance metrics of the three models, including precision, recall, F1-score, and support. The RF model, shown in Figure 2(a), achieves the highest accuracy at 97.3% and an F1-Score of 95%. The KNN model, depicted in Figure 2(b), follows with a precision of approximately 94%, recall of about 93%, and an F1-Score of 93%. Although the Gaussian NB model, illustrated in Figure 2(c), has a lower accuracy of 55.7% and an F1-Score of 55%, it excels in recall with a value of 96%, highlighting its strength in identifying positive instances.

Figure 3 presents the confusion matrices for the three classification models. The KNN model in Figure 3(a) correctly identified 8,439 instances as true positives but also incorrectly labeled 333 instances as false positives and missed 430 true positives, classifying them as false negatives. The Gaussian NB model in Figure 3(b), however, exhibited a higher rate of false positives at 5,358, while correctly identifying 3,371 true positives. The balance between false positives and true positives is crucial as it affects model decisions depending on the application context. The RF model Figure 3(c) demonstrated exceptional performance, with only 7 false positives and 351 false negatives, resulting in a high count of 8,765 true positives. The choice of model depends on the specific task requirements and the acceptable trade-offs between false positives and false negatives, which vary across different domains and applications. The confusion matrix is essential for evaluating these trade-offs and making informed model selection decisions, as illustrated in Figure 3. While KNN and Gaussian NB have higher false positive rates, RF shows minimal false positives and negatives, resulting in a superior true positive count.

Figure 4 displays the ROC curves for the KNN, Gaussian NB, and RF models. In this study, we evaluated these models based on their discriminative power, as depicted by their ROC curve values. The ROC curve illustrates a model's capacity to differentiate between positive and negative classes across various threshold settings. The area under the ROC curve AUC quantifies overall model performance, with values nearing 1.0 indicating high accuracy and values around 0.5 suggesting random classification.

Our findings show that the RF model Figure 4(a) achieves the highest AUC value at 95%, demonstrating exceptional class differentiation ability. The KNN model Figure 4(b) follows with an AUC of 93%, indicating high but slightly lower performance. The Gaussian NB model Figure 4(c) has an AUC of 86%, reflecting a less robust classification ability. The ROC curves and their AUC values provide crucial insights into the models' effectiveness, helping to inform decisions about their appropriateness for specific applications or tasks. These values highlight RF's superior class distinction capability compared to KNN and Gaussian NB, offering valuable guidance for classification task selection.

Figure 5 presents the correlation matrix for the KNN, Gaussian NB, and RF models, providing insights into the linear relationships between various liver function tests and demographic data. The matrix reveals that Age has a negligible correlation with other variables, indicating its limited linear impact on liver-related tests. Total and Direct Bilirubin features show a strong positive correlation (0.887), suggesting a direct relationship in liver function. Liver enzymes, such as Alkphos, Sgpt, and Sgot, exhibit moderate to high correlations, particularly between Sgpt and Sgot (0.783), reflecting their interconnected roles in liver health. Total proteins and albumin (ALB) also display a strong correlation (0.776), underscoring their combined importance in liver function assessments. The A/G ratio shows a significant positive correlation with albumin (0.683), which aligns with its composition. Gender, represented as binary variables, shows a strong negative correlation between its categories (-0.928), as expected from binary data. This matrix is crucial for understanding the interdependencies among liver-related variables and can guide in-depth analysis and model development, particularly in identifying potentially redundant or highly predictive variables. The correlations among liver enzymes and proteins are notably strong, with gender exhibiting a significant negative correlation between its binary categories, offering critical insights for precise liver function analysis.

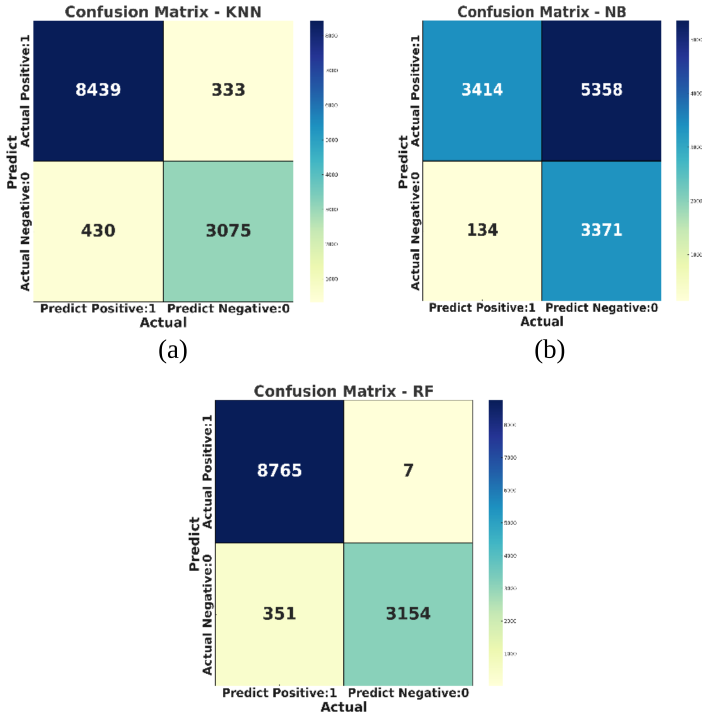


(a)

(b)

(c)

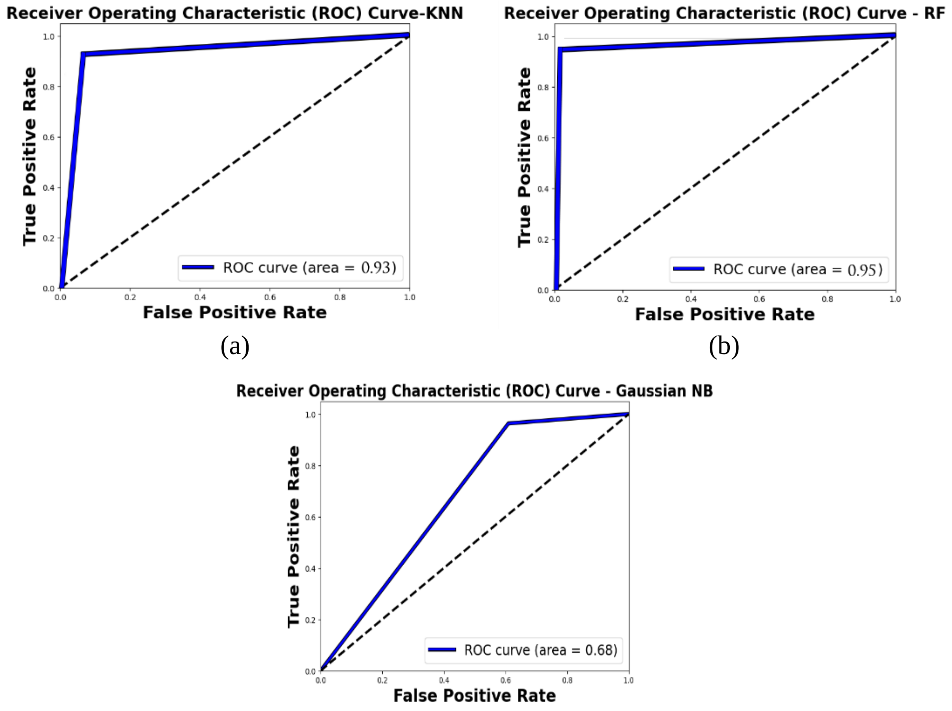
Figure 2. A classification report of the three models (a) KNN, (b) Gaussian NB, and (c) RF. RF demonstrates the highest accuracy and F1-Score, followed by KNN, while Gaussian NB lags behind in accuracy and F1-Score but excels in recall



(a) (b)

(c)

Figure 3. Confusion matrix for (a) KNN, (b) Gaussian NB, and (c) RF. KNN and Gaussian NB has higher false positive rates, while RF exhibits minimal false positives and negatives, resulting in a higher true positive count



(a) (b)

(c)

Figure 4. ROC curves for (a) RF, (b) KNN, and (c) Gaussian NB, showcasing their discriminative power. The AUC values indicate RF's superior ability to distinguish between classes compared to KNN and Gaussian NB. These curves provide valuable insights into model performance, aiding in informed decision-making for classification tasks

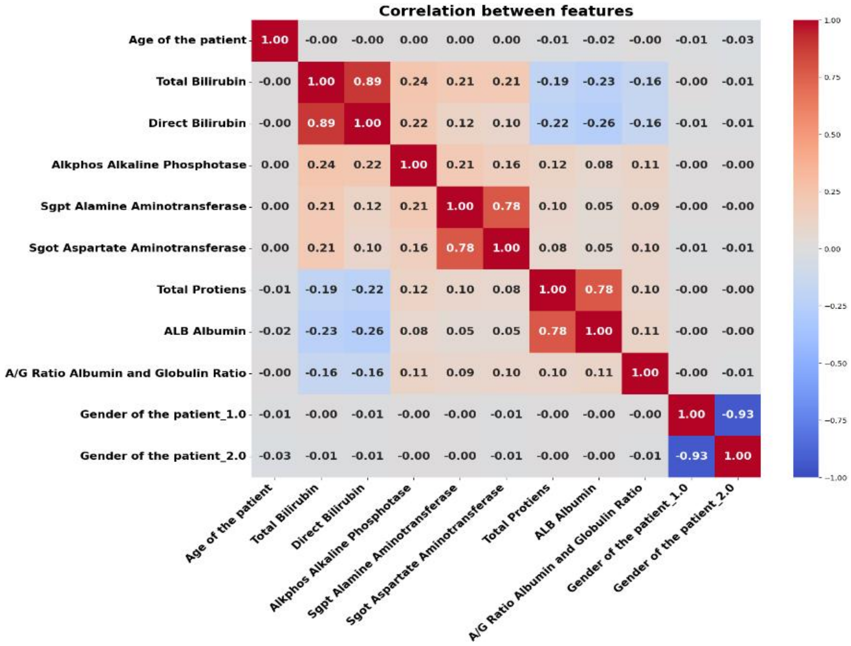


Figure 5. Correlation matrix between variables, highlighting relationships between liver function tests and demographic data. Strong correlations are evident among liver enzymes and proteins, with gender displaying a notable negative correlation between its binary categories. This matrix offers critical insights for precise liver function analysis

3.5. Comparison with previous works

Table 2 provides a summary of previous research that employed different machine learning models to predict liver disease. Each model has a different level of accuracy, precision, recall, F1-score, and AUC. Singh *et al.* [10] implemented logistic regression (LR) with an accuracy of 74.36%. Priya *et al.* [22] used SVM, achieving an accuracy of 71.35%. Ghosh *et al.* [25] applied back propagation, reporting an accuracy of 73.2% and precision of 65.7%. Bahramirad *et al.* [26] also used logistic regression, achieving an accuracy of 73.39% and a precision of 57.69%. Thirunavukkarasu *et al.* [27] employed decision trees (DT), achieving a notable accuracy of 81%. Nahar *et al.* [28] used logistic regression, with an accuracy of 73.97%. Vijayarani and Dhayanand [29] utilized AdaBoost, achieving an accuracy of 70.25%. In another study, they used SVM and achieved a higher accuracy of 79.66% with 76.6% precision. In this study, the KNN model achieved a remarkable accuracy of 95%, with precision, recall, and F1-score values of 94%, 93%, and 93%, respectively, demonstrating its effectiveness in classification. The Gaussian NB model displayed lower performance, with an accuracy of 55.7%, precision of 39%, recall of 96%, and F1-score of 55%, indicating less effectiveness compared to other models. The random forest model emerged as the top performer, boasting an accuracy of 97.3% and well-balanced precision, recall, and F1-score values of 97%, 96%, and 95%, respectively. By leveraging a large and complex dataset, this study not only maintained but also improved the prediction accuracy for liver diseases. This can contribute positively to enhancing the ability to predict liver diseases effectively, thus facilitating early diagnosis and timely intervention.

Table 2 compares previous papers on liver disease prediction using machine learning models

Paper	Model	Accuracy	Precision	Recall	F1-Score	AUC
Singh *et al.* [10]	LR	74.36%	-	-	-	-
Priya *et al.* [22]	SVM	71.35%	-	-	-	-
Sontakke *et al.* [1]	Back Propagation	73.2%	65.7%	-	-	-
Bahramirad *et al.* [26]	Logistic	73.39%	57.69%	-	-	-
Thirunavukkarasu *et al.* [27]	DT	81%	-	-	-	-
Nahar *et al.* [28]	LR	73.97%	-	-	-	-
Vijayarani *et al.* [29]	AdaBoost	70.25 %	-	-	-	-
	SVM	79.66%	76.6%	-	-	-
Results of this study	KNN	95%	94%	93%	93%	93%
	Gaussian NB	55.7%	39%	96%	55%	68%
	RF	97.3%	97%	96%	95%	95%

4. CONCLUSION

This paper explores the prediction of liver diseases by comparing various machine learning models using a big data liver patient dataset, which contains over 32,000 patient records and includes 10 distinct variables. By employing advanced machine learning techniques, including data processing, classification, and prediction, our aim was to enhance the early detection and accurate assessment of liver diseases. We addressed the complexities inherent in big data through sophisticated preprocessing methods, ultimately contributing to improved healthcare outcomes in liver disease diagnosis.

This study utilized and compared three distinct machine learning models: KNN, Gaussian NB, and RF. The models were rigorously evaluated based on key performance metrics, including accuracy, precision, recall, F1-score, and the AUC. The results indicated that the RF model consistently outperformed the others, demonstrating superior performance across all metrics. KNN was the second-best model, while Gaussian NB showed comparatively lower results. Despite the valuable contributions this paper makes to liver disease prediction, there is still room for improvement. Future work should focus on developing more refined models that can not only detect the presence of liver disease but also accurately classify the specific type, such as hepatitis A, B, C, or fatty liver disease.

Furthermore, future research should aim to assess how close individuals are to developing liver diseases, focusing on their risk proximity and the progression timeline. This can be achieved by integrating more interpretable data and enhancing the effectiveness of machine learning models. In summary, our study highlights the crucial role of big data and machine learning techniques in the early diagnosis and prediction of liver diseases. The adoption of advanced and interpretable models will deepen our understanding of these conditions, leading to more effective healthcare interventions that could save lives and improve patient care quality.

REFERENCES

[1] S. Sontakke, J. Lohokare, and R. Dani, “Diagnosis of liver diseases using machine learning,” in *2017 International Conference on Emerging Trends {\&} Innovation in ICT (ICEI)*, Feb. 2017, pp. 129–133, doi: 10.1109/ETIICT.2017.7977023.

[2] F. Himmah, R. Sigit, and T. Harsono, “Segmentation of liver using abdominal CT scan to detection liver desease area,” in *2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, Oct. 2018, pp. 225–228, doi: 10.1109/KCIC.2018.8628561.

[3] W. Ji, M. Xue, Y. Zhang, H. Yao, and Y. Wang, “A machine learning based framework to identify and classify non-alcoholic fatty liver disease in a large-scale population,” *Frontiers in Public Health*, vol. 10, pp. 1–10, Apr. 2022, doi: 10.3389/fpubh.2022.846118.

[4] S. Perveen, M. Shahbaz, K. Keshavjee, and A. Guergachi, “A systematic machine learning-based approach for the diagnosis of non-alcoholic fatty liver disease risk and progression,” *Scientific Reports*, vol. 8, no. 1, pp. 1–12, Feb. 2018, doi: 10.1038/s41598-018-20166-x.

[5] H. Ma, C. Xu, Z. Shen, C. Yu, and Y. Li, “Application of machine learning techniques for clinical predictive modeling: a cross-sectional study on nonalcoholic fatty liver disease in china,” *BioMed Research International*, vol. 2018, pp. 1–9, Oct. 2018, doi: 10.1155/2018/4304376.

[6] G. S. Harshpreet Kaur, “The diagnosis of chronic liver disease using machine learning techniques,” *Information Technology in Industry*, vol. 9, no. 2, pp. 554–564, Mar. 2021, doi: 10.17762/itii.v9i2.382.

[7] A. Q. Md, S. Kulkarni, C. J. Joshua, T. Vaichole, S. Mohan, and C. Iwendi, “Enhanced preprocessing approach using ensemble machine learning algorithms for detecting liver disease,” *Biomedicines*, vol. 11, no. 2, pp. 1–23, Feb. 2023, doi: 10.3390/biomedicines11020581.

[8] C.-C. Wu *et al.*, “Prediction of fatty liver disease using machine learning algorithms,” *Computer Methods and Programs in Biomedicine*, vol. 170, pp. 23–29, Mar. 2019, doi: 10.1016/j.cmpb.2018.12.032.

[9] G. L. Wong, P. Yuen, A. J. Ma, A. W. Chan, H. H. Leung, and V. W. Wong, “Artificial intelligence in prediction of non-alcoholic fatty liver disease and fibrosis,” *Journal of Gastroenterology and Hepatology*, vol. 36, no. 3, pp. 543–550, Mar. 2021, doi: 10.1111/jgh.15385.

[10] J. Singh, S. Bagga, and R. Kaur, “Software-based Prediction of liver disease with feature selection and classification techniques,” *Procedia Computer Science*, vol. 167, pp. 1970–1980, 2020, doi: 10.1016/j.procs.2020.03.226.

[11] C. Geetha and A. R. Arunachalam, “Evaluation based approaches for liver disease prediction using machine learning algorithms,” in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2021, pp. 1–4, doi: 10.1109/ICCCI50826.2021.9402463.

[12] S. Tokala *et al.*, “Liver disease prediction and classification using machine learning techniques,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 2, pp. 871–878, 2023, doi: 10.14569/IJACSA.2023.0140299.

[13] M. S. Devi *et al.*, “Feature predominance ensemble inquisition towards liver disease prediction using machine learning,” *SSRN Electronic Journal*, pp. 1–6, 2021, doi: 10.2139/ssrn.3842561.

[14] E. Dritsas and M. Trigka, “Supervised machine learning models for liver disease risk prediction,” *Computers*, vol. 12, no. 1, pp. 1–15, Jan. 2023, doi: 10.3390/computers12010019.

[15] M. O. Edeh *et al.*, “Artificial intelligence-based ensemble learning model for prediction of hepatitis C disease,” *Frontiers in Public Health*, vol. 10, Apr. 2022, doi: 10.3389/fpubh.2022.892371.

[16] E. A. El-Shafeiy, A. I. El-Desouky, and S. M. Elghamrawy, “Prediction of liver diseases based on machine learning technique for big data,” in *in The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018)*, 2018, pp. 362–374, doi: 10.1007/978-3-319-74690-6_36.

[17] L. R. Guarneros-Nolasco, G. Alor-Hernández, G. Prieto-Avalos, and J. L. Sánchez-Cervantes, “Early identification of risk factors in non-alcoholic fatty liver disease (NAFLD) using machine learning,” *Mathematics*, vol. 11, no. 13, Jul. 2023, doi: 10.3390/math11133026.

[18] M. M. Islam, C.-C. Wu, T. N. Poly, H.-C. Yang, and Y.-C. J. Li, “Applications of machine learning in fatty live disease prediction,” in *Building Continents of Knowledge in Oceans of Data: The Future of Co-Created eHealth*, 2018, pp. 166–170, doi: 10.3233/978-1-61499-852-5-166.

[19] M. Zhao, C. Song, T. Luo, T. Huang, and S. Lin, “Fatty liver disease prediction model based on big data of electronic physical examination records,” *Frontiers in Public Health*, vol. 9, Apr. 2021, doi: 10.3389/fpubh.2021.668351.

[20] R. Choudhary, T. Gopalakrishnan, D. Ruby, A. Gayathri, V. S. Murthy, and R. Shekhar, “An efficient model for predicting liver disease using machine learning,” in *Data Analytics in Bioinformatics: A Machine Learning Perspective*, Wiley, 2021, pp. 443–457.

[21] S. R. Veeranki and M. Varshney, “Intelligent techniques and comparative performance analysis of liver disease prediction,” *International Journal of Mechanical Engineering*, vol. 7, no. 1, pp. 7115–7124, 2022.

[22] M. B. Priya, P. L. Juliet, and P. R. Tamilselvi, “Performance analysis of liver disease prediction using machine learning algorithms,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 5, no. 1, pp. 206–211, 2018.

[23] M. Z. Alam, M. S. Rahman, and M. S. Rahman, “A random forest based predictor for medical data classification using feature ranking,” *Informatics in Medicine Unlocked*, vol. 15, 2019, doi: 10.1016/j.imu.2019.100180.

[24] R. Amin, R. Yasmin, S. Ruhi, M. H. Rahman, and M. S. Reza, “Prediction of chronic liver disease patients using integrated projection based statistical feature extraction with machine learning algorithms,” *Informatics in Medicine Unlocked*, vol. 36, 2023, doi: 10.1016/j.imu.2022.101155.

[25] M. Ghosh *et al.*, “A comparative analysis of machine learning algorithms to predict liver disease,” *Intelligent Automation & Soft Computing*, vol. 30, no. 3, pp. 917–928, Feb. 2021, doi: 10.32604/iasc.2021.017989.

[26] S. Bahramirad, A. Mustapha, and M. Eshraghi, “Classification of liver disease diagnosis: a comparative study,” in *2013 Second International Conference on Informatics {\&} Applications (ICIA)*, Sep. 2013, pp. 42–46, doi: 10.1109/ICoIA.2013.6650227.

[27] K. Thirunavukkarasu, A. S. Singh, M. Irfan, and A. Chowdhury, “Prediction of liver disease using classification algorithms,” in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, Dec. 2018, pp. 1–3, doi: 10.1109/CCAA.2018.8777655.

[28] N. Nahar, F. Ara, M. A. I. Neloy, V. Barua, M. S. Hossain, and K. Andersson, “A comparative analysis of the ensemble method for liver disease prediction,” in *2019 2nd International Conference on Innovation in Engineering and Technology (ICIET)*, Dec. 2019, pp. 1–6, doi: 10.1109/ICIET48527.2019.9290507.

[29] S. Vijayarani and S. Dhayanand, “Liver disease prediction using SVM and Naïve bayes algorithms,” *International Journal of Science, Engineering and Technology Research (IJSETR)*, vol. 4, no. 4, pp. 816–820, 2015.

[30] M. Masum and H. Shahriar, “TL-NID: deep neural network with transfer learning for network intrusion detection,” in *2020 15th International Conference for Internet Technology and Secured Transactions (ICITST)*, Dec. 2020, pp. 1–7, doi: 10.23919/ICITST51030.2020.9351317.

[31] J. Li, Y. Si, T. Xu, and S. Jiang, “Deep convolutional neural network based ECG classification system using information fusion and one-hot encoding techniques,” *Mathematical Problems in Engineering*, vol. 2018, pp. 1–10, Dec. 2018, doi: 10.1155/2018/7354081.

[32] G. R. DeVore, “Computing the Z score and centiles for cross-sectional analysis: a practical approach,” *Journal of Ultrasound in Medicine*, vol. 36, no. 3, pp. 459–473, Mar. 2017, doi: 10.7863/ultra.16.03025.

[33] J. P. Bharadiya, “A tutorial on principal component analysis for dimensionality reduction in machine learning,” *International Journal of Innovative Science and Research Technology*, vol. 8, no. 5, pp. 2028–2032, 2023.

[34] R. Ullah, A. H. Khan, and S. M. Emaduddin, “ck-NN: a clustered k-nearest neighbours approach for large-scale classification,” *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 8, no. 3, pp. 67–77, Aug. 2019, doi: 10.14201/ADCAIJ2019836777.

[35] K. P. Murphy, *A probabilistic perspective*. MIT press, 2012.

[36] J. Wang *et al.*, “A descriptive study of random forest algorithm for predicting COVID-19 patients outcome,” *PeerJ*, vol. 8, Sep. 2020, doi: 10.7717/peerj.9945.

[37] B. D. Jesty, “Machine learning for liver disease classification,” M.S. thesis, Department of Electronics and Computer Science, University of Southampton, Southampton, UK, 2019.

BIOGRAPHIES OF AUTHORS

Osama Mohareb Khaled is a professor at Port Said University. In 2012, he received his Ph.D. in mathematical statistics from Zagazig University. He has published research articles in reputable international journals of mathematics and statistics. His research interests are in the areas of statistical modeling and data science and its applications. He can be contacted at email: osama_mohareb@sci.psu.edu.eg.

Ahmed Zakareia Elsherif is a researcher in the Faculty of Science, Port Said University, Department of Mathematics and Computer Science. He works as a teaching assistant at the Higher Institute of Administrative Sciences in Manzala. He obtained a bachelor’s degree in engineering in 2015 and obtained a diploma in Computer Science in 2019. He can be contacted ahmad.elsherif@sci.psu.edu.eg.

Ahmed Salama Professor of Mathematics and Computer Science - Port Said University - Egypt, Champion of Neutrosophic Systems for over 20 years, leads the field as President of the International Neutrosophic Systems Society. He pioneered Crisp's neutrosophic theory and laid the foundation for important areas of research, influencing fields such as fuzzy logic systems. His over 285 publications and over 30 books demonstrate his dedication, reinforced by prestigious awards such as Africa's greatest scientist and best research papers award. As editor-in-chief and collaborating researcher, Dr. Salama inspires future generations and shapes the future of neutrospheric systems. He can be contacted at email: ahmed_salama_2000@sci.psu.edu.eg or drsalama44@gmail.com.

Mostafa Herajy is an Associate Professor of Computer Science at Port Said university, Egypt. His research interests are hybrid simulation, hybrid Petri nets and their applications to computational biology as well as intelligent systems. He can be contacted at email: mherajy@sci.psu.edu.eg.

Elsayed Elsedimy has joined the Port Said University, Egypt, in December 2012 as assistant lecturer after 8 years being senior lecturer (associate professor) at Ministry of Higher Education. Currently, he is Assistant Professor of Information Technology at Faculty of Technology and Information System, Port Said University, Egypt. He obtained his Ph.D. degree in Computer Science (mainly in cloud computing and artificial modeling) in 2018 from the University of Mansoura, Egypt, and the M.Sc. and B.Sc. degrees in rough set theory and decision support systems applications, from the University of Mansoura, Egypt. He can be contacted at email: elsodamey_sayed@himc.psu.edu.eg.