Early Liver Disease Diagnosis using Grey Wolf Optimization Algorithm

Abrar Rizk M. Khedr
Ahmed I. Saleh
Ahmed S. Samra
Eman AbdElhalim

Follow this and additional works at: https://mej.researchcommons.org/home
Part of the Engineering Commons
Early Liver Disease Diagnosis Using Grey Wolf Optimization Algorithm

Aabar Rizk M. Khedr a,*, Ahmed I. Salehb, Ahmed S. Samraa, Eman AbdElhalima

a Electronics and Communications Engineering Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt
b Computers and Control Systems Engineering Department, Faculty of Engineering, Mansoura University, Mansoura, Egypt

Abstract

Every year, millions of people around the world experience health issues due to liver disease (LD), and it is also a major global cause of mortality. A variety of factors, such as obesity and hepatitis infection, can harm the liver and contribute to these disorders. However, the diagnosis of chronic LD is often an expensive and complex process, and early detection of LDs poses challenges because of their elusive symptoms that can often lead to delayed diagnosis. We employed machine learning for this study to anticipate individuals with LDs before symptoms appear. To attain the best accuracy, we create a proposed model that uses gray wolf optimization to select the important features and an additional tree classifier to achieve high accuracy. Using criteria such as accuracy, sensitivity, specificity, precision, recall, and F1-score, we assess the models’ performance. In addition, we contrasted the outcomes of this methodology with many machine learning algorithms, including support vector machine, decision tree, gradient boosting, random forest, naive Bayes, logistic regression, K-nearest neighbor, and extra tree. The findings demonstrate that the proposed model outperforms traditional approaches in predicting LDs with high accuracy and F1-score. The proposed model performs the best, with an overall processing time of 1.5 s, an F1-score of 100 %, 100 % precision, 100 % recall, and 100 % accuracy.

Keywords: Classification, Gray wolf optimization, Machine learning algorithms, Prediction

1. Introduction

The liver is the body's largest organ and is essential for both the breakdown of food and the removal of pollutants. The number of patients suffering from liver disease (LD) is rising as a result of eating foods contaminated with bacteria or viruses or inhaling toxic fumes. Cirrhosis, liver cancer, tumors, and hepatitis are just a few examples of the many distinct types of LD. Among them are conditions affecting the liver, with cirrhosis being the main killer. Consequently, LD, which has high rates of morbidity and mortality, is one of the most serious health problems in the world. Worldwide, LD claims the lives of almost two million individuals each year (Mokdad et al., 2014; Byass, 2014). The increased incidence of LD has resulted in a significant economic burden. Sadly, especially in the beginning, it can be challenging to diagnose LD, due to the nuances of its signs, because the liver can still function even with partial damage, liver conditions sometimes go unnoticed until it is too late (Anstee et al., 2013). As medical information systems in modern hospitals and other medical institutions develop more and bigger, it gets harder to extract useful information for decision support.

Early detection would extend a patient's lifespan and control the risk that common diseases like LD would be fatal to the patient. Thanks to developments in machine learning (ML) and artificial intelligence, numerous classifiers and clustering techniques are being used to achieve this (Lin, 2009). Because manual data analysis has proven inefficient, strategies for efficient computer-based data
analysis are essential. It has been demonstrated that integrating ML into medical analysis increases diagnostic precision while lowering expenses and using fewer staff members.

In the medical field, ML plays a crucial role in aiding clinical decision-making by leveraging the abundance of clinical data (Auxilia, 2018; Hashem and Mai, 2014; Sajda, 2006; Motwani et al., 2016; Raghupathi and Raghupathi, 2014). The results of this study are therefore very important for both computer scientists and medical practitioners (Ozcift and Gulten, 2011). ML techniques have been used by some academics to identify the disease, although there are disagreements among them over the best methodology for categorizing it. Numerous methods, including artificial neural network (ANN), multivariate adaptive regression splines, decision tree algorithm (DT), support vector machine (SVM), Bayesian, and support feature machines, are used in medical diagnosis (Ya-Ju and Wanpracha, 2010). A technique to identify and track coronary artery disease was suggested by Otoom et al. (2015). The Bayes Net, SVM, and functional trees (FT) algorithms were used in two tests. For detection, WEKA was used as a tool. When testing the top seven features, Bayes net achieved 84.5 % accuracy, SVM provided 85.1 % accuracy, and FT successfully identified 84.5 % of the features. For the purpose of predicting LD, Vijayarani and Dhayanand (2015) used the SVM and Naïve Bayes (NB) classification methods. MATLAB was used for the data analysis. SVM provided 79.66 % accuracy in 3210.00 ms, whereas NB provided 61.28 % accuracy in 1670.00 ms. Using datasets from UCI, Gulia et al. (2014) researched intelligent methods for categorizing liver patients. Five algorithms — J48, MLP, random forest algorithm (RF), SVM, and Bayesian network — as well as the WEKA tool were used for experimentation. Following FS, the following algorithms provided the highest accuracy: J48 (70.669 %), MLP (70.845 %), SVM (71.3551 %), RF (71.8696 %), and Bayes net (69.1252 %).

In the study, we using some abbreviations, which are explained in Table 1.

The goal of this study is to identify the presence of LD in the patients based on key LD features. The selection of features is essential when using ML to predict a disease’s risk. It seeks to only extract the most informative characteristics while removing redundant, unnecessary features (Pudjihartono et al., 2022). In our study, to determine the best features we compared several methods for selecting features that are shown in Table 2.

Also, we shall introduce the various classification techniques used for LD diagnoses as follows:

### 2. Decision tree

It is a graph in which choices and their results are shown as a tree. Every node in the graph represents an option, whereas the borders of the plot provide the prerequisites or guidelines for making rules. In all trees, there are nodes and branches. According to Fig. 1, each node represents a set of features that need to be categorized, whilst each branch represents a possible worth for the node (Mahesh, 2018).

#### 2.1. Random forest

An RF is built from random samples and consists of DT with different node-splitting criteria. Features are randomly selected for each tree in the model. Several trees are trained, and each one produces a distinct prediction, as seen in Fig. 2 (Bonaccorso, 2017).

#### 2.2. Support vector machine

By indirectly changing the data they receive into high-dimensional feature spaces, it can efficiently do both linear and nonlinear classification. The kernel method is a tactic that draws boundaries between the different categories. As shown in Fig. 3,
The features that are most likely to predict the outcome variable can be found using RF in the context of feature selection. This is accomplished by determining the relative weights of each feature in the RF model. Features with higher significance ratings are chosen for inclusion in a predictive model because they are thought to be more crucial (Dimitriadis et al., 2017).

Mutual information gain is a technique for statistically analyzing the relationship between two variables. It assesses the amount of knowledge one variable imparts regarding another variable. Mutual information gain can be used to find the features that are best predictive of the outcome variable in the context of feature selection. The mutual information between each characteristic and the result variable is calculated to achieve this. Features with higher mutual information scores are chosen for inclusion in a predictive model because they are deemed to be more significant (Xu1 et al., 2007).

Select KBest is an ML feature selection approach that chooses the best features based on statistical tests including $\chi^2$, mutual information, and ANOVA $F$ value. Each feature is assessed separately, and a score is given to each one based on how pertinent it is to the outcome variable. A prediction model is then created using the top attributes with the highest scores (Al-Tashi et al., 2018).

$\chi^2$ is a statistical technique for determining the degree of independence between two categorical variables. The more dependent the variables are, the greater the $\chi^2$ value. $\chi^2$ can be used to determine the features that are most capable of predicting the outcome variable in the context of feature selection. By contrasting the observed and anticipated frequencies of each feature with regard to the outcome variable, it achieves this. Features with higher $\chi^2$ values are prioritized and chosen for inclusion in a predictive model (Jovi et al., 2015).

Light GBM (LGB) grows trees leaf-wise, which helps lessen loss during the sequential boosting procedure. Compared with other boosting methods, this typically yields superior accuracy. The feature importance property built within the method can be used to determine the significance of each feature (https://medium.com/@hertan06/which-features-to-use-in-your-model-350630a1e31c). Features with high mutual information scores are chosen for inclusion in a predictive model because they are deemed to be more significant (Xu1 et al., 2007).

Pearson’s correlation is a statistical technique for determining the degree of independence between two variables. It is a technique for statistically analyzing the relationship between two variables. It assesses the amount of knowledge one variable imparts regarding another variable. Pearson correlation can be used to identify features strongly correlated with the result variable to incorporate it into a predictive model (a).

For a better predictive model, features that are most capable of predicting the outcome variable can be found using GWO in the context of feature selection. This is accomplished by replicating the social hunting behavior of gray wolves. The technique entails separating the gray wolf population into packs and designating a leader for each pack. Then, using a set of predetermined rules, the leaders look for prey (i.e., features) in the search space (i.e., feature space). The process is repeated until convergence, at which point the best features discovered by each pack are shared with the other packs (Al-Tashi et al., 2018).
the borders are designed to be as close to the categories as feasible to reduce classification error (Mahesh, 2018).

2.3. Logistic regression

Logistic regression is a method for modeling a binary-dependent variable’s connection to one or more independent variables using statistical methods. Although this classification method is called regression, it depends on the probability that the data will fall into a specific class. The choice to utilize the sigmoid (or logistic) function is where the word logistic originates as Fig. 4 shows (Bonaccorso, 2017).

K-nearest neighbor (KNN): the algorithm decides based on similarity metrics, which are used to determine how far a new sample is from previous training samples. The training sample is thought to be more likely to have the same class label if it is closer to the new sample, which Fig. 5 illustrates (https://www.geeksforgeeks.org/k-nearest-neighbours/?ref=gcse).

2.4. Naïve Bayes

To put it simply, the algorithm believes that the presence of one feature in a class has nothing to do with the presence of any other feature. It is mostly used for classification and clustering purposes.

Fig. 1. DT algorithm is a supervised ML strategy that is applied to both classification and regression applications. It consists of a root node, branches, internal nodes, and leaf nodes and is arranged hierarchically.

Fig. 2. RF algorithm is a technique for averaging several deep DTs that have been trained on various aspects of the same training set to lower the variance.

Fig. 3. SVM classifiers are quite useful as we search for the feature’s largest separation hyperplane among the available classes.
based on conditional probabilities of occurrence, as Fig. 6 illustrates (Mahesh, 2018).

2.5. Gradient boosting

Gradient boosting (GB) tree is a method that enables the gradual construction of a tree ensemble to minimize a target loss function as Fig. 7 shows (Bonaccorso, 2017).

2.6. Extra tree

The extra tree (ET) algorithm generates several DTs, similar to the RF algorithm, with each tree’s sampling being random. As a result, a dataset with unique samples for each tree is produced. As shown in Fig. 8, a specified number of features from the full set of features are also randomly selected for each tree (https://medium.com/@namanbhandari/extra-treesclassifier-8e7fc0502c7).

After comparing feature selection methods and ML algorithms, we created a new proposed model. Gray wolf optimization (GWO) is used to choose the best features, and ET is used as a classifier to achieve the best accuracy ‘GWO-ET.’

3. Related work

In the area of health, data mining techniques are frequently used. One such technique is classification, which is well known to be efficient (Gupta et al., 2011).

Some of the similar studies considered in this study are shown in the tables below. Table 3 compares different disease predictions made by ML, Table 4 compares different LD predictions made by ML algorithms, and Table 5 compares applying GWO to predict diseases.

It might be challenging to choose the right algorithm, and the effectiveness of the classification algorithm also depends on the dataset being employed. After comparing various classification techniques, we apply the proposed model to find patients with LD with high accuracy (99.5 %) and low time (1.5 s).

4. Material and method

4.1. Dataset description

The dataset used for training the model was obtained from Kaggle (https://www.kaggle.com/datasets/abhi8923shriv/liver-disease-patient-dataset).
Before analyzing the dataset, a preprocessing stage was undertaken to eliminate data duplication. As a result, the preliminary data was reduced from 30,691 to 19,368 records.

Dataset concepts explained in Tables 6 and 7 show the definitions of the features in the dataset.

To visualize the total liver patients’ ratio in a dataset, we presented Fig. 9; in addition, we analyzed 13,577 male samples and 5,791 female samples. By examining this dataset, we aimed to gain a better understanding of LD prevalence and characteristics in the patient population.

4.2. Data preprocessing

Preprocessing often entails several stages, including data cleaning, and exploratory data analysis, which seek to improve data quality while minimizing outliers and data that is duplicated. Thus, better accuracy can be obtained. Before preprocessing steps, there is an important technique called label encoding, which is used to convert categorical columns into numeric columns (https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/). In the study, we used this technique to convert the sex of patients male or female to 0 or 1, respectively.

4.3. Data preprocessing steps

4.3.1. Removing the duplicates

Duplicates must be eliminated from a dataset for a variety of reasons. First of all, duplicates might take up extra space and reduce efficiency. Second, when using the dataset for certain tasks, duplication can cause complications and mistakes. As a result, eliminating duplicates improves the data’s accuracy, leading to better analysis (Brownlee, 2020). Table 8 compares the quantity of records before and after the duplicates were eliminated.
4.3.2. Filtering technique (filling missing data)

During the dataset analysis, null values occasionally appeared in the features. To handle these circumstances, we used the filtering strategy, which calculates each attribute's mean using data in each column and replaces the null value with it.

4.3.3. Removing outliers

Outliers can affect a dataset in several ways, including the accuracy estimation of ML models. Therefore, before modeling a dataset, outliers must be found and eliminated.

There are various techniques for visualizing outliers in a dataset, we used a boxplot for it. Boxplots, which are based on a five-number summary [the minimum score, lower quartile (Q1), median, upper quartile (Q3), and maximum score], are a common approach to depict the distribution of data as shown in Fig. 10 (Nuzzo, 2016).

<table>
<thead>
<tr>
<th>References</th>
<th>Dataset</th>
<th>Train—test</th>
<th>Data mining classification techniques</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hassan et al. (Hassan et al., 2017)</td>
<td>583 Indian individuals with LD, provided on Kaggle and the UCI ML repository</td>
<td>70 %: 30 %</td>
<td>SVM, multilayer perceptron (MLP), RF, LR, attribute Selected classifier, classification via regression, J48, and NB</td>
<td>With some attributes, RF achieves the maximum accuracy (72.041 %), while LR achieves the highest accuracy (71.355 %) with all attributes</td>
</tr>
<tr>
<td>Gupta et al. (Gupta et al., 2022)</td>
<td>UCI ML repository: 583 records</td>
<td>—</td>
<td>LR, NB, KNN, DT, RF, GB, XG boosting (XGB), Ada boosting, extreme GB, light GB (LGB), and stacking</td>
<td>Models RF and LGB provide superior accuracy than the other classification algorithms, with 63 % accuracy</td>
</tr>
<tr>
<td>Banu Priya et al. (Banu Priya et al., 2018)</td>
<td>Indian Liver Patient Dataset from UCI ML Repository: 583 record databases</td>
<td>—</td>
<td>RF, SVM, J.48, Baysian Net, and MLP</td>
<td>J48 algorithm accuracy rate is 95.04 %</td>
</tr>
<tr>
<td>Alisahrin and Mantoro (Alisahrin and Mantoro, 2013)</td>
<td>UCI ML repository: 583 records</td>
<td>—</td>
<td>DT, NB, and NB Tree</td>
<td>NB Tree algorithm accuracy rate is 67.01 % in 2.51 s</td>
</tr>
<tr>
<td>Azam et al. (Azam et al., 2020)</td>
<td>UCI ML repository: 583 records</td>
<td>75 %: 25 %</td>
<td>RF, Perceptron, DT, KNN, and SVM</td>
<td>KNN algorithm accuracy rate is 74.15 %</td>
</tr>
<tr>
<td>Nahar and Ara (Nahar and Ara, 2018)</td>
<td>The UCI ML repository where the data was obtained. 583 record</td>
<td>—</td>
<td>J48, RF, random tree, decision stump, hoeffding tree, logistic model tree (LMT), DT, and REP tree</td>
<td>Compared with other algorithms, decision stump performs better and achieves an accuracy of 70.67 %</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>References</th>
<th>Dataset</th>
<th>Train—test</th>
<th>Data mining classification techniques</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turabieh (Turabieh, 2016)</td>
<td>From Cleveland database, UCI ML repository, containing 166 records</td>
<td>70 %: 30 %</td>
<td>ANN, GWO, ANN-GWO</td>
<td>In 40 epochs and 2.435 s, ANN-GWO can reach a high-quality performance objective of 0.0029</td>
</tr>
<tr>
<td>Al-Tashi et al. (Al-Tashi et al., 2018)</td>
<td>From Cleveland database, UCI ML repository</td>
<td>—</td>
<td>GWO-SVM</td>
<td>GWO-SVM reach accuracy, sensitivity, and specificity values of 89.83, 93, and 91 %, respectively</td>
</tr>
</tbody>
</table>

4.3.2. Filtering technique (filling missing data)

During the dataset analysis, null values occasionally appeared in the features. To handle these circumstances, we used the filtering strategy, which calculates each attribute's mean using data in each column and replaces the null value with it.

<table>
<thead>
<tr>
<th>Description</th>
<th>Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>The dataset, named the Indian Liver Patient DataSet, comprised</td>
<td>30 691 patients</td>
</tr>
<tr>
<td>The final set after being cleaned up had</td>
<td>19 368 entries</td>
</tr>
<tr>
<td>Number of patients</td>
<td>13 811</td>
</tr>
<tr>
<td>Number of</td>
<td>5557</td>
</tr>
</tbody>
</table>

Table 4. Comparison of various liver disease predictions by machine learning algorithms.

Table 5. Comparison of using gray wolf optimization to predict diseases.

Table 6. Dataset description shows the contents of the dataset and the numbers it contains.
We employ many strategies to remove outliers in the dataset, including:

1. Quantile-based flooring and capping: for the smaller values, we used 10th percentile flooring, and for the higher values, 85th percentile capping.

2. Log transformation reduces the importance of outliers and can lead to a bell-shaped distribution.

3. Interquartile range (IQR) score: for Q1, we used the 25th percentile, and for the Q3, used the 75th percentile. As IQR is in the middle of the sample, IQR = Q3 – Q1.

4. IQR on transformed data: In this method for the Q1, we used the 23rd percentile, and for Q3, the 80th percentile to get a better handle on removing outliers.

After comparing these methods, it turns out that the capping and flooring methods in addition to the log transform method were the best, as the data was not deleted using these methods. As Fig. 11 shows, when a comparison was made between these two methods, it became clear that the capping and flooring method is the best, as it makes the data not contain any outliers. So, we used it to removing outliers from the dataset.
4.3.4. Handling imbalanced data

Oversampling can be used to enhance the size of rare samples to balance a dataset. We applied techniques like:

1. Repetition (random oversampling).
2. SMOTE (Synthetic Minority OverSampling Technique).

Table 10 shows the comparison between the two methods to clarify the differences between them.

As Fig. 12 shows a comparison made between these two methods, and it became clear that the SMOTE method was the best because it copies data and editing it not just copying such as the repetition method. So, we used it to get balance for the dataset.

Table 11 shows the number of records before and after applying the oversampling method.

### Table 9. Number of records before and after removing the outliers.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Numbers of records</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before removing the outliers</td>
<td>19,368 entries</td>
</tr>
<tr>
<td>After removing the outliers</td>
<td>19,368 entries</td>
</tr>
</tbody>
</table>

### 4.4. Methodology (proposed model)

This study suggested the proposed model method as a useful tool for LD diagnosis. The proposed model contains two main phases. GWO is used in the first stage to eliminate redundant and unnecessary features by seeking out the best feature in the dataset. The very efficient ET classifier is used in the second stage.

Hunting strategy of gray wolves is the basis for the technique known as GWO. It simulates the hierarchy using four different breeds of gray wolves: alpha (α), beta (β), delta (δ), and omega (ω), as seen in Fig. 13. To accomplish optimization, it employs the three primary hunting techniques of seeking out prey, encircling prey, and attacking prey (Al-Tashi et al., 2018; Turabieh, 2016).

The equations below illustrate the GWO's step of encircling its target. The separation between each wolf and its prey is represented by (1) and the location of gray is described by (2):

\[
\overrightarrow{D} = \overrightarrow{C} \cdot \overrightarrow{X}^p(t) - \overrightarrow{X}(t) \quad (1)
\]

\[
\overrightarrow{X}(t+1) = \overrightarrow{X}^p(t) - \overrightarrow{A} \cdot \overrightarrow{D} \quad (2)
\]

\[
\overrightarrow{A} \quad \text{and} \quad \overrightarrow{C} \quad \text{can be calculated by the equations below:}
\]

\[
\overrightarrow{A} = 2 \overrightarrow{a} \cdot \overrightarrow{r}_1 - \overrightarrow{a} \quad (3)
\]

\[
\overrightarrow{C} = 2 \overrightarrow{r}_2 \quad (4)
\]

Table 12 shows the symbols of the mathematical equations used in the previous equations and their meanings.

The final position of the wolf is determined by the equations below \(\overrightarrow{X}^*(t+1)\):

\[
\overrightarrow{D} \quad \alpha = \overrightarrow{C} \cdot \overrightarrow{X}^\alpha - \overrightarrow{X} \quad (5)
\]

\[
\overrightarrow{D} \quad \beta = \overrightarrow{C} \cdot \overrightarrow{X}^\beta - \overrightarrow{X} \quad (6)
\]

\[
\overrightarrow{D} \quad \delta = \overrightarrow{C} \cdot \overrightarrow{X}^\delta - \overrightarrow{X} \quad (7)
\]

\[
\overrightarrow{X}^*1 = \overrightarrow{X}^\alpha - \overrightarrow{A} \quad 1 \cdot \overrightarrow{D} \quad \alpha \quad (8)
\]
we used GWO for binary feature selection, as detailed in Table 13.

![Diagram](image)

**Fig. 12.** Comparison between repetition and SMOTE to find out which one is more appropriate to obtain a balance in dataset.

### Table 10. Comparison between repetition and SMOTE to find out which one is more appropriate to obtain a balance in dataset.

<table>
<thead>
<tr>
<th></th>
<th>Repetition</th>
<th>SMOTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meaning</td>
<td>By repeatedly using the original dataset, the random oversampling technique expands the amount of the training dataset</td>
<td>Increasing the amount of the training dataset by repeatedly using the original dataset as SMOTE oversampling</td>
</tr>
<tr>
<td>Advantages</td>
<td>This approach ensures greater inclusivity of minority class instances in the training dataset</td>
<td>Adds new variety, where SMOTE generates new artificial training examples based on the original examples</td>
</tr>
<tr>
<td>Disadvantages</td>
<td>It does not add any new variety of training examples to the dataset</td>
<td></td>
</tr>
<tr>
<td>Work steps</td>
<td>Data from the initial training dataset are selected and added to the new training dataset</td>
<td>Determine the variation between a sample and its closest neighbor</td>
</tr>
<tr>
<td></td>
<td>After which they were added back or ‘replaced’ in the initial dataset, becoming available for selection once more</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Multiply the variation by an integer at random between 0 and 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Continue with the next closest neighbor until you reach the user-defined number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>To create a new synthetic example in feature space, add this difference to the sample</td>
<td></td>
</tr>
</tbody>
</table>

This section provided a detailed description of our proposed framework. Fig. 14 depicts our workflow and the proposed model process.

The efficacy of each classifier is assessed based on the accuracy and execution time to compare the

![Diagram](image)

**Fig. 13.** The social hierarchy of GWO, in which the alpha wolf serves as the pack leader, while the omega wolf is the weakest member of the pack.
results of various classification methods including LR, NB, RF, DT, SVM, KNN, GB, and ET. The model is developed to predict the LDs of a new patient based on the accuracy and improved performance of the classifiers. In the section that comes after this, we will talk about this comparison and performance assessment.

5. Results and discussion

This section provides an overview of the experimental findings from using the proposed model. The simulation setup, results, and system
performance assessment are included in the following subsections.

5.1. Simulation setup and evaluation metrics

We used the Kaggle notebook with Python 3 as our programming language for data analysis and modeling in this study. To develop our ML models, we utilized the Scikit-Learn library in Python, and for visualizing the correlation heatmap, we employed Seaborn. In addition, we utilized Numpy and Pandas for data preprocessing and analysis. Python’s flexibility and broad range of libraries enabled us to efficiently handle the data preprocessing, feature engineering, and ML tasks required for this study.

Specifications of the laptop used at work:

(1) OS name: Microsoft Windows 10 Pro.
(2) Processor: 2 Cores CPU, Core i7, and 8.00 G RAM.

Before comparing different models or building our model, we removed duplicated data; in Fig. 15 we show the numbers of data before and after deduplication. Besides, we removed outliers using the capping and flooring method as it proved to be the best method as none of the data was deleted and all outliers were removed as shown in Fig. 16. We also dealt with data imbalance to make the number of patients equal to the number of nonpatients, as shown in Fig. 17 using the SMOTE method, it turned out that this method was the best.

In this approach, the dataset is split into three parts: 80 % of the data are used for training the models, 10 % are used for validation, and 10 % are tested for accuracy. Based on performance parameters including accuracy, precision, recall, and F1-score, which are provided in the equations below, the effectiveness of various algorithms is examined (https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5):

\[
\text{Accuracy} = \frac{\text{True Negative (TN)} + \text{True Positive (TP)}}{\text{All Samples}}
\]  \quad (12)

\[
\text{Precision} = \frac{\text{TP}}{\text{TP + False Positive (FP)}}
\]  \quad (13)

\[
\text{Recall} = \frac{\text{TP}}{\text{TP + False Negative (FN)}}
\]  \quad (14)

\[
\text{F1 – score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision + Recall}}
\]  \quad (15)

5.2. Evaluation performance

5.2.1. Analysis results using various machine learning models

Eight classifier models, including LR, NB, RF, DT, SVM, KNN, GB, and ET, are produced in this study. The accuracy of the various ML algorithms is displayed in the bar chart in Fig. 18. The figure shows that the proposed model has the highest accuracy at 99.5 %.

On a set of test data, an ML model’s performance can be evaluated using the confusion matrix’s numbers. The number of cases that the model properly identified as positive instances is known as the TP. The number of times when the model mistakenly predicts a negative value while a positive value exists is known as the FN. The number of occurrences that the model properly predicts as
negative is known as the TN. The number of times when the model mistakenly predicts a positive outcome, while a negative outcome exists is known as the FP as shown in Fig. 19 (https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5).

Fig. 20 shows the confusion matrix for the proposed model. The number of correctly categorized points for each data layer is represented by the diagonal numbers of the matrix. The primary diagonal factors in a successful model’s confusion matrix will have a large value, while the remaining factors will have a modest value. Fig. 20 shows the confusion matrix for the proposed model. Fig. 21 shows the confusion matrix of all algorithms.

The receiver-operating characteristic curve (ROC curve) is a graph that displays how well a classification model performs across all categorization levels. Two parameters are plotted on this curve:

1) True positive rate (TPR) is the percentage of positive observations that were correctly foreseen out of all positive observations. TPR can be calculated according to equation (16):

$$TPR = \frac{TP}{TP + FN}$$  \hspace{1cm} (16)

False positive rate (FPR): is the percentage of observations out of all negative observations that are mistakenly projected to be positive. FPR can be calculated according to equation (17):

$$FPR = \frac{FP}{FP + TN}$$  \hspace{1cm} (17)

The tradeoff between specificity (1–FPR) and sensitivity (or TPR) is depicted by the ROC curve. A better performance is shown by classifiers that provide curves that are closer to the top-left corner.

Calculating the area below the ROC, sometimes referred to as the area under the curve (AUC), is the optimum method for normalization. AUC must be greater than 0.5 for a test procedure to be useful. Fig. 22 shows the ROC curve and the AUC value for each model. AUC has a value between 0 and 1.
A model with 100% incorrect predictions has an AUC of 0, while a model with 100% correct predictions has an AUC of 1.0. There is no definitive good value for AUC, as it may depend on the context and the application of the model, but the model performs better at differentiating between positive and negative cases when the curve rises (b). Table 14 shows the AUC value for each model. Fig. 23 shows the AUC–ROC curve for the proposed model.

Table 14. The area under the curve value for the different models.

<table>
<thead>
<tr>
<th>Numbers</th>
<th>Method</th>
<th>AUC value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ET</td>
<td>0.981</td>
</tr>
<tr>
<td>2</td>
<td>RF</td>
<td>0.994</td>
</tr>
<tr>
<td>3</td>
<td>DT</td>
<td>0.738</td>
</tr>
<tr>
<td>4</td>
<td>GB</td>
<td>0.918</td>
</tr>
<tr>
<td>5</td>
<td>KNN</td>
<td>0.739</td>
</tr>
<tr>
<td>6</td>
<td>NB</td>
<td>0.919</td>
</tr>
<tr>
<td>7</td>
<td>LR</td>
<td>0.995</td>
</tr>
<tr>
<td>8</td>
<td>SVM</td>
<td>0.919</td>
</tr>
<tr>
<td>9</td>
<td>Proposed model</td>
<td>0.995</td>
</tr>
</tbody>
</table>

Fig. 21. The confusion matrix of all algorithms, where the order of the confusion matrix for the models is ET, RF, DT, GB, KNN, NB, LR, and SVM, respectively.

Fig. 22. The AUC–ROC curve of all algorithms, where the order of the AUC–ROC curve for the models is ET, RF, DT, GB, KNN, NB, LR, and SVM, respectively.
5.3. Performance evaluation

The findings show how the proposed model compares with traditional ML models such as LR, KNN, SVM, RF, NB, GB, and DT. This comparison shows that the GWO used in conjunction with ET yields the best accuracy, and the findings are presented in Table 15.

The proposed model outperformed the previous ML models, earning the highest accuracy (99.5 %) and the highest F1-score (100 %), according to our findings.

The heatmap in Fig. 24 illustrates the relationship between the features.
The heatmap shows the correlation values between the features and the target variable (LD status). The scale runs from $-1$ to $1$, with $-1$ denoting a negative correlation and $1$ a positive correlation. The association is weaker the closer the coefficient gets to $0$. We can predict one feature based on another through correlation. Because the desirable features have a strong correlation with the target, correlation can be used to select features. Features should also be uncorrelated among themselves while being correlated with the objective. We can anticipate one feature from another if the two are correlated. As a result, if two features are correlated, the model only actually requires one of them as the other does not provide any new information.

We can infer the following things from the heatmap:

(1) Albumin and globulin ratio ($0.16$) and albumin ($0.16$) have a substantial positive connection with the class, indicating that having greater levels of these enzymes is linked to a higher risk of LD.

(2) As the target variable has a negative correlation with direct bilirubin ($-0.25$), total bilirubin ($-0.22$), alkaline phosphatase ($-0.18$), aspartate alanine, aminotransferase ($-0.16$), and amino-transferase ($-0.16$), these features tend to rise when the target variable decreases.

(3) The features have some degree of multicollinearity, as shown by the squares along the diagonal. For example, direct bilirubin and total bilirubin have a high positive correlation of $0.88$, indicating that they tend to vary together. This indicates that the features exchange some information and are not entirely independent of one another. It makes some ML models have issues, because it may cause reduced predictive power, exaggerated variance, and unstable results. We used GWO to choose the features to prevent such issue.

6. Conclusion

The major goal of this work is to use several supervised ML classifiers to create an accurate diagnostic system for LD patients, where it can be challenging to predict a patient’s early onset of LD. To reduce the measurement, storage, and calculation requirements while keeping high-accuracy results, feature selection is performed. We contrast our findings with several well-known ML techniques, including SVM, DT, GB, RF, NB, LR, KNN, and ET. The findings of computational experiments demonstrate that improved prediction accuracy can be attained with substantially fewer features, where the results show that the proposed model method surpassed the other classifiers in terms of classification accuracy and efficiency, making it the best option. The SVM classifier had the lowest performance, at $66.55\%$, while the proposed model attained an overall accuracy of $99.51\%$ and provided an accurate and effective diagnosis of persons at risk as well as early LD detection.

In future work, the proposed model will be expanded to a wider variety of datasets for various ML algorithms.

Here are some directions we want to apply:

(1) To increase the precision and generalizability of our model, we can use more sophisticated and powerful ML approaches, including deep learning.

(2) Assessing the clinical impact and applicability of our model for the detection, diagnosis, and prognosis of liver disorders in real-world situations.

(3) Creating interactive and user-friendly tools and interfaces to make it easier for physicians and patients to accept and use our concept.

Authors contribution

1. Conception or design of the work: Abrar Rizk M. Khedr, Eman AbdElhalim, Ahmed I. Saleh, Ahmed S. Samra (30 %/30 %/20 %/20 %).

2. Data collection and tools: Abrar Rizk M. Khedr, Eman AbdElhalim.


6. Project administration: Abrar Rizk M. Khedr, Eman AbdElhalim, Ahmed I. Saleh, Ahmed S. Samra (30 %/30 %/20 %/20 %).


9. Drafting the article: Abrar Rizk M. Khedr, Eman AbdElhalim.

10. Critical revision of the article: Abrar Rizk M. Khedr, Eman AbdElhalim, Ahmed I. Saleh, Ahmed S. Samra (25 %/30 %/20 %/25 %).

11. Final approval of the version to be published: Abrar Rizk M. Khedr, Eman AbdElhalim, Ahmed I. Saleh, Ahmed S. Samra (20 %/30 %/20 %/30 %).
Conflicts of interest

There are no conflicts of interest.

References


https://medium.com/analytics-vidhya/feature-selection-techniques-2614f5b7e7cf7cfd.

https://medium.com/@heertan68/which-features-to-use-in-your-model-3506301a1e31f.

https://medium.com/namanbhandari/extraTreesclassifier-8c7fc050c7.


https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1cf0f6e0a5.


