

**CHIKKANNA GOVERNMENT ARTS  
COLLEGE**

**DEPARTMENT OF BACHELOR OF  
COMPUTER APPLICATION**

**TIRUPUR-641602**

**(AFFILIATED TO BHARATHIAR UNIVERSITY)**



**TEAM MEMBERS NAME :**

**DEVAKUMAR A (2022J0024)**

**RAGHUL K (2022J0053)**

**SURIYA S (2022J0057)**

**NAVEENKUMAR S (2022J0049)**

# **A Review of Liver Patient Analysis Methods Using Machine Learning**

## **1.INTRODUCTION**

### **1.1 OVERVIEW :**

Liver diseases averts the normal function of the liver. This disease is caused by an assortment of elements that harm the liver. Diagnosis of liver infection at the preliminary stage is important for better treatment. In today's scenario devices like sensors are used for detection of infections. Accurate classification techniques are required for automatic identification of disease samples. This disease diagnosis is very costly and complicated.

Therefore, the goal of this work is to evaluate the performance of different Machine Learning algorithms in order to reduce the high cost of liver disease diagnosis. Early prediction of liver disease using classification algorithms is an efficacious task that can help the doctors to diagnose the disease within a short duration of time.

In this project we will analyse the parameters of various classification algorithms and compare their predictive accuracies so as to find out the best classifier for determining the liver disease. This project compares various classification algorithms such as Random Forest, Logistic Regression, KNN and ANN Algorithm with an aim to identify the best technique.

Based on this study, Random Forest with the highest accuracy outperformed the other algorithms and can be further utilised in the prediction of liver disease and can be recommended to the user.

## **1.2. PURPOSE**

In India, delays in diagnosing diseases are a major problem due to a lack of medical professionals. The typical scenario, which is mainly in rural and slightly urban areas:

1. A patient who sees a doctor with certain symptoms.
2. The doctor will perform some tests, such as blood and urine tests, depending on the symptoms.
3. The patient undergoes the above tests in the analytical laboratory.
4. The patient takes the reports back to the hospital, where they are examined and diagnosed.

The goal of this project is to reduce some of the delays caused by unnecessary detours between the hospital and the pathology laboratory. Historically, work has been done to detect the onset of heart disease, such as Parkinson's, and machine learning algorithms have been developed to predict liver disease in patients based on a variety of characteristics.

## 2. PROBLEM DEFINITION & DESIGN THINKING

## 2.1. EMPATHY MAP

Template

## Empathy map

Use this framework to develop a deep, shared understanding and empathy for other people. An empathy map helps describe the aspects of a user's experience, needs and pain points, to quickly understand your users' experience and mindset.

[Share template feedback](#)

### Need some inspiration?

See a finished version of this template to kickstart your work.

[Open example](#)

### Build empathy

The information you add here should be representative of the observations and research you've done about your users.

#### Says

What have we heard them say?  
What can we imagine them saying?

#### Thinks

What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?

**Says**

- "I'm not sure if I can afford to buy this product."
- "I don't know how to use this feature."
- "I'm not sure if I can afford to buy this product."
- "I don't know how to use this feature."
- "I'm not sure if I can afford to buy this product."
- "I don't know how to use this feature."

**Thinks**

- "I'm not sure if I can afford to buy this product."
- "I don't know how to use this feature."
- "I'm not sure if I can afford to buy this product."
- "I don't know how to use this feature."
- "I'm not sure if I can afford to buy this product."
- "I don't know how to use this feature."

**Does**

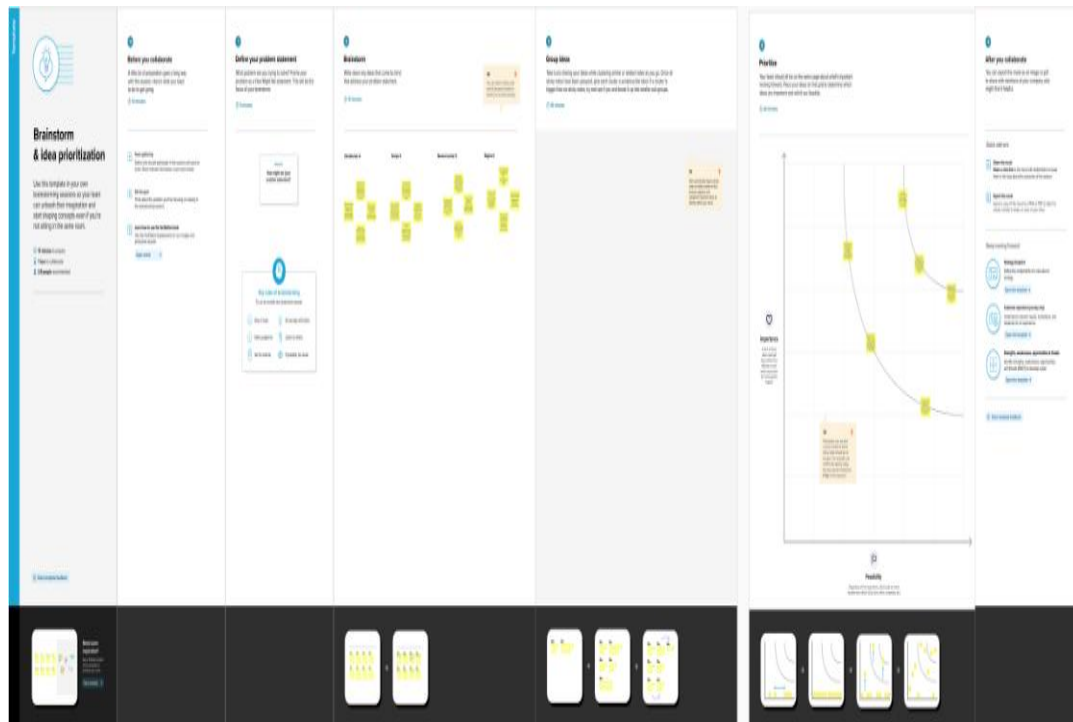
- "I'm not sure if I can afford to buy this product."
- "I don't know how to use this feature."
- "I'm not sure if I can afford to buy this product."
- "I don't know how to use this feature."
- "I'm not sure if I can afford to buy this product."
- "I don't know how to use this feature."

**Feels**

- "I'm not sure if I can afford to buy this product."
- "I don't know how to use this feature."
- "I'm not sure if I can afford to buy this product."
- "I don't know how to use this feature."
- "I'm not sure if I can afford to buy this product."
- "I don't know how to use this feature."

Give them a name and a portrait to empathize with your persona.

## 2.2 IDEATION & BRAINSTORMINGS MAP



### 3. RESULT

## Liver Patient Prediction

You have a liver disease problem, You must and should consult a doctor. Take care

## Liver Patient Prediction

Age:

Gender:

Enter 0 as male, 1 as female

Total\_Bilirubin:

Direct\_Bilirubin:

Alkaline\_Phosphatase:

Alamine\_Aminotransferase:

Aspartate\_Aminotransferase:

Total\_Protiens:

Albumin:

Albumin\_and\_Globulin\_Ratio:

Predict

# Liver Patient Prediction

You have a liver disease problem, You must and should consult a doctor. Take care

## **4.ADVANTAGES & DISADVANTAGES**

### **ADVANTAGES :**

- It is readily available, so the patient doesn't need to wait.
- No chance of rejection.
- No need for major surgery.

No need to take drugs, such as immuno-suppressants.

### **DISADVANTAGES :**

- Liver dialysis requires expensive machinery.
- The patient has to do dialysis at least 2-3 times a week for 4-6 hours at a time, so it is very time consuming.
- The patient has to monitor their diet carefully.
- It is painful for the patient
- Can cause infections

## **5. APPLICATION**



A function named `KNeighborsClassifier` is imported and train and test data are passed as the parameters. Inside the function, `KNeighborsClassifier` algorithm is initialised and training data is passed to the model with `.fit()` function. Test data is predicted with `.predict()` function and saved in new variable. For evaluating the model, confusion matrix and classification report is done.

Building and training an Artificial Neural Network (ANN) using the Keras library with TensorFlow as the backend. The ANN is initialised as an instance of the `Sequential` class, which is a linear stack of layers. Then, the input layer and two hidden layers are added to the model using the `Dense` class, where the number of units and activation function are specified. The output layer is also added using the `Dense` class with a sigmoid activation function. The model is then compiled with the Adam optimizer, binary cross-entropy loss function, and accuracy metric. Finally, the model is fit to the training data with a batch size of 100, 20% validation split, and 100 epochs.

## 6. CONCLUSION

The present study evaluated the studies associated with the diagnosis of chronic diseases. However, the implementation of correct methods or selection of the right models is a prerequisite to perform ideal decisions, as modern researchers are claiming that some ML models are compromised by enlarging contained datasets with malicious data that can have severe consequences. On the other hand, diagnosis limitations may lead to life-threatening attacks, and sometimes it might be a driving factor of fatality. In contrast, the wrong diagnosis prompts the skepticism in machine learning use, that can lead policy makers to avoid predictive model usage. Therefore, reviews on predictive models can provide evidence to propose excellent methods for the CDs diagnosis.

## 7. FUTURE SCOPE

The increasing prevalence of chronic liver disease is well known, as it is a fact that recorded data in all countries show continuing growth in the number of patients that need substitutive treatment for their renal function. The consequences from the social and economic viewpoint are very significant and we cannot be happy with morbidity and mortality rates in terminal stage renal patients that continue to be unacceptably high.<sup>1,2</sup> There are different reasons for such high mortality rates, amongst them significant increase in the age of patients undergoing treatment, restoration with haemodialysis and peritoneal dialysis liver function, and a significant associated co-morbidity. Despite the progress made in haemodialysis (membrane biocompatibility, high-flow membranes, increase frequency in sessions, water quality control, among others) and in peritoneal dialysis (infection risk reduction, introduction of a dialysis machine, etc.) no clear improvement has been shown in the evolution of patients.

Therefore, if so little improvement has been made after so many years, what is in store for the future for renal function replacement? This article aims at highlighting which are the future possibilities to face renal insufficiency, by substitutive techniques such as haemodialysis, peritoneal dialysis or liver transplant (or creation of new organs), as well as the possibility of regression of chronic liver disease before total loss of renal function.

## 8. APPENDIX

### A. SOURCE CODE :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
from google.colab import files
uploaded = files.upload()

data = pd.read_csv('indian_liver_patient (1).csv')

data.head()

data.tail()

data.describe()

data.info()

data.isnull().any()

data.isnull().sum()
```

```
data[data['Dataset']==1]
```

```
data['Dataset'].unique()
```

```
data.isnull().sum()
```

```
#mode imputation
```

```
#data['Albumin_and_Globulin_Ratio'] = data.fillna(data['Albumin_and_Globulin_Ratio'].mode()[0])
```

```
data_1 = data.dropna()
```

```
#checking for the missing data after cleaning data
```

```
#data['Albumin_and_Globulin_Ratio'] = data.fillna(data['Albumin_and_Globulin_Ratio'].mode()[0])
```

```
data_1.isnull().sum()
```

```
plt.figure(figsize=(15,10))
```

```
plt.subplot(3,3,1)
```

```
plt.scatter(data_1['Age'], data_1['Dataset'])
```

```
plt.ylabel('Dataset')
```

```
plt.xlabel('Age')
```

```
plt.subplot(3,3,2)
```

```
plt.scatter(data_1['Gender'], data_1['Dataset'],)
```

```
plt.ylabel('Dataset')
plt.xlabel('Gender')
```

```
plt.subplot(3,3,3)
plt.scatter(data_1['Total_Bilirubin'], data_1['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Total_Bilirubin')
```

```
plt.subplot(3,3,4)
plt.scatter(data_1['Direct_Bilirubin'], data_1['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Direct_Bilirubin')
```

```
plt.subplot(3,3,5)
plt.scatter(data_1['Alkaline_Phosphotase'], data_1['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Alkaline_Phosphotase')
```

```
plt.subplot(3,3,6)
plt.scatter(data_1['Alamine_Aminotransferase'], data_1['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Alamine_Aminotransferase')
```

```
plt.subplot(3,3,7)
plt.scatter(data_1['Aspartate_Aminotransferase'],data_1['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Aspartate_Aminotransferase')
```

```
plt.subplot(3,3,8)
plt.scatter(data_1['Total_Protiens'], data_1['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Total_Protiens')
```

```
plt.subplot(3,3,9)
plt.scatter(data_1['Albumin_and_Globulin_Ratio'], data_1['Dataset'])
plt.ylabel('Dataset')
plt.xlabel('Albumin_and_Globulin_Ratio')
```

```
sns.countplot(data=data_1, x = 'Dataset')
LD,NLD=data_1['Dataset'].value_counts()
print("liver disease patinet:",LD)
print("Non-liver disease patinets:",NLD)
```

```
sns.countplot(data+data_1, x = 'Gender', label='Count')
m,f=data_1['Gender'].value_counts()
print("No of Males:",m)
print("No of Females:",f)
```

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
data_1['Gender'] = le.fit_transform(data_1['Gender'])
data_1.head()
```

```
x=data_1.iloc[:,0:-1]
y=data_1.iloc[:, -1]
```

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3)
```

```
xtrain.shape
```

```
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
```

```
svc=SVC()
RFmodel=RandomForestClassifier()
```



```
KNNmodel=KNeighborsClassifier()
```

```
from sklearn.svm import SVC
```

```
svm=SVC()
```

```
svc.fit(xtrain,ytrain)
```

```
SVCpred=svc.predict(xtest)
```

```
from sklearn.metrics import accuracy_score,confusion_
matrix
```

```
SVCaccuracy=accuracy_score(SVCpred, ytest)
```

```
SVCaccuracy
```

```
SVCcm=confusion_matrix(SVCpred, ytest)
```

```
SVCcm
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
RFmodel=RandomForestClassifier()
```

```
RFmodel.fit(xtrain, ytrain)
```

```
RFpred=RFmodel.predict(xtest)
```

```
RFaccuracy=accuracy_score(RFpred,ytest)
```

```
RFaccuracy
```

```
RFcm=confusion_matrix(RFpred, ytest)
```

```
RFcm
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
KNN = KNeighborsClassifier()
```

```
KNN.fit(xtrain, ytrain)
```

```
KNNpred=KNN.predict(xtest)
```

```
KNNaccuracy=accuracy_score(KNNpred, ytest)
```

```
KNNaccuracy
```

```
KNNcm=confusion_matrix(KNNpred, ytest)
```

```
KNNcm
```

```
print("Support Vector Machine Algorithm accuracy score : {value:.2f} %" .format(value=SVCaccuracy*100))
```

```
print("Random Forest Algorithm accuracy score : {value:.2f} %" .format(value=RFaccuracy*100))
```

```
print("K-
```

```
Nearest Neighbors Algorithm accuracy score : {value:.2f} %" .format(value=KNNaccuracy*100))
```

```
import pickle
pickle.dump(svm, open('liver_analysis_1.pkl','wb'))
```