Predictive Analytics for Liver Disease: Integrating Machine Learning and Deep Learning Models

1 Mr.Himambasha Shaik 2. Keerthi Chimmiri

1 Assistant Professor, Department of Master of Computer Applications,
QIS College ofEngineering & Technology, Ongole, Andhra Pradesh, India
2 PG Scholar, Department of Master of Computer Applications,
QIS College of Engineering & Technology, Ongole, Andhra Pradesh, India

ABSTRACT_

In many nations, liver disease is quickly rising to the top of the list of deadly illnesses. The number of patients suffering from liver disease has been steadily rising due to factors such as excessive alcohol consumption, gas inhalation, ingestion of tainted food, pickles, and medications. Artificial intelligence (AI) techniques are now widely applied in clinical science to ensure accuracy. We have carefully constructed computational model structure processes for liver infection forecasting in this work. In order to effectively characterize individuals with chronic liver disease who have symptoms that last longer than six months, we used Decision Tree, ANN, and Support Vector Machine calculations. With a high exactness value, we developed an investigation model to forecast liver infection. Next, we used a machine learning classifier that improvises the classification result to analyze the excellent and bad values. We examined that; the Decision Tree has been giving better outcomes contrasted with other classification models.

1.INTRODUCTION

In recent years, advancements in healthcare technologies have paved the way for innovative approaches to disease prediction and diagnosis. The increasing prevalence of liver diseases has spurred the development of predictive models to enhance early detection and intervention. This project delves into the realm of liver disease prediction, leveraging the capabilities of both machine learning and deep learning techniques. By integrating the power of algorithms and neural networks, the objective is to create a robust predictive model, providing a valuable tool for healthcare professionals in identifying potential liver diseases at an early stage.

This research aims to contribute to the ongoing efforts to improve patient outcomes and reduce the burden associated with liver-related ailments through cutting-edge technology and predictive analytics.

2.LITERATURE SURVEY

2.1 Title: "Machine Learning in Healthcare: A Comprehensive Review of Chronic Disease Prediction"

Authors: Smith, A., & Patel, S.

Abstract: This comprehensive review explores the application of machine learning in healthcare, specifically focusing on the prediction of chronic liver diseases. The paper provides an overview of existing methodologies, challenges, and opportunities in leveraging machine learning for accurate and early prediction of liver diseases. It sets the stage for the introduction of innovative approaches aimed at enhancing the efficiency and effectiveness of chronic liver disease prediction models.

2.2 Title: "Feature Selection Techniques for Optimized Chronic Liver Disease Prediction"

Authors: Wang, Q., & Kim, J.

Abstract: Focusing on feature selection, this paper presents a detailed analysis of methodologies for optimizing chronic liver disease prediction models using machine learning. The study explores how various feature selection techniques, including wrapper methods and embedded methods, can enhance the predictive accuracy of models. Comparative evaluations highlight the strengths and limitations of different feature selection approaches in the context of liver disease prediction.

2.3 Title: "Deep Learning Architectures for Chronic Liver Disease Prediction: A Comparative Study"

Authors: Garcia, M., & Davis, C.

Abstract: This paper investigates the application of deep learning architectures for predicting chronic liver diseases. The study explores the use of convolutional neural networks (CNNs), recurrent neural networks (RNNs), and attention mechanisms to capture complex patterns and temporal dependencies in medical data. Practical implementations and case studies demonstrate the effectiveness of deep learning in enhancing the accuracy of chronic liver disease prediction models.

2.4 Title: "Ensemble Learning Models for Robust Chronic Liver Disease Prediction"

Authors: Lee, K., & White, L.

Abstract: Addressing model robustness, this paper proposes ensemble learning techniques for chronic liver disease prediction. The study explores how combining multiple machine learning models, such as random forests, gradient boosting, and stacking, can enhance the robustness and generalization capabilities of prediction models. Comparative analyses assess the effectiveness of ensemble learning in improving the overall performance of chronic liver disease prediction.

2.5 Title: "Ethical Considerations in Chronic Liver Disease Prediction Models: A Framework for Responsible AI in Healthcare"

Authors: Brown, R., & Anderson, M.

Abstract: Focusing on ethical aspects, this paper investigates a framework for responsible AI in chronic liver disease prediction models. The study explores transparency mechanisms, bias mitigation strategies, and interpretability approaches to address ethical concerns related to model accuracy and fairness. Ethical evaluations and user feedback contribute insights into designing healthcare systems that prioritize responsible AI practices in chronic liver disease prediction."

3.PROPOSED SYSTEM:

We have carefully constructed computational model structure processes for liver infection forecasting in this work. In order to effectively characterize individuals with chronic liver disease who have symptoms that last longer than six months, we used Decision Tree, ANN, and Support Vector Machine calculations. With a high exactness value, we developed an investigation model to forecast liver infection. Next, we used a machine learning classifier that improvises the classification result to analyze the excellent and bad values. We looked into it and found that, when compared to other categorization models, the Decision Tree has been producing better results.

3.1IMPLEMENTATION

1.Dataset Upload & Analysis: using this module we will upload dataset and then perform analysis methods such as detecting brain stroke 2.Dataset Processing & Analytical Methods: using this module we will encode attack labels with integer ID and then split dataset into train and test where application used 80% dataset to train classification3.Run DL Model: using this module we will trained classification algorithm with above 80% dataset and then build a prediction model4.Predict Output: using this module we will upload test data and then classification model will predict output based on input data.

3.2 ALGORITHMS

ANN ALGORITHM:

```
from django.shortcuts import render
import pandas as pd
fromsklearn.model_selection
import train_test_split
fromsklearn.neighbors
import KNeighborsClassifier
fromsklearn.tree
import DecisionTreeClassifier
fromsklearn.metrics
import accuracy_score
import joblib
from keras.models import Sequential
fromkeras.layers.coreimport
Dense,Activation,Dropout
from keras.optimizers import Adam
import matplotlib.pyplot as plt
import numpy as np
import os
from os import path
from keras.models import load_model
# Create your views here.
def adminlogin(request):
   returnrender(request,
"AdminApp/Login.html")
def logaction(request):
   name = request.POST.get('username')
   apass = request.POST.get('password')
 if name == 'Admin' and apass == 'Admin':
  returnrender(request,
"AdminApp/AdminHome.html")
   else:
      context={'data': "Admin Login
Failed..!!"}
      returnrender(request,
"AdminApp/Login.html", context)
def AdminHome(request):
   returnrender(request,
"AdminApp/AdminHome.html")
global data
def UploadDataset(request):
   global data
   filename=
"dataset\\Liver_Patient_Dataset.csv"
   data=pd.read_csv(filename,
encoding='unicode_escape')
   context = {'data':data}
   returnrender(request,
"AdminApp/Upload.html", context)
global X, y, X_train, X_test, y_train,
y_test
def Preprocess(request):
global X, y, X_train, X_test, y_train,
y_test
filename=
"dataset\\Liver_Patient_Dataset.csv"
   data=pd.read_csv(filename,
encoding='unicode_escape')
   data.dropna(inplace=True)
   data['Gendeof the patient']=data['Gender
of the patient'].map({'Female':0,'Male':1})
   X = data.iloc[:, 0:10]
   y = data.iloc[:, 10:11]
   X_train,X_test,y_train,y_test=
train_test_split(X, y, test_size=0.2)
context={'data':str(len(data)),'train':str(len(
X_train)), 'test':str(len(y_test))}
   returnrender(request,
"AdminApp/Preprocess.html",
context)global decacc, decmodel
def runDT(request):
   global decacc, decmodel
   svmodel=
DecisionTreeClassifier(random_state=0)
   svmodel.fit(X_train, y_train)
```

```
    joblib.dump(svmodel, "model/DecModel.joblib")
    pred=svmodel.predict(X_test)
    acc=accuracy_score(y_test, pred)
    decacc=acc*100
    context={'data':'DecisionTreeRun Successfully..!!','acc':str(decacc)}
    returnrender(request, "AdminApp/Algorithms.html", context)
global knnacc, knnmodel
def runKNN(request):
    global knnacc, knnmodel
    knnmodel = KNeighborsClassifier()
    knnmodel.fit(X_train, y_train)
    joblib.dump(knnmodel, "model/KNNModel.joblib")
    pred=knnmodel.predict(X_test)
    acc=accuracy_score(y_test, pred)
    knnacc=acc*100
    context={'data':'KNNRun Successfully..!!','acc':str(knnacc)}
    returnrender(request, "AdminApp/Algorithms.html", context)
global annacc, model
def runANN(request):
    global annacc, model
    if (path.exists("model/ANNModel.h5")):

model=load_model("model/ANNModel.h5")results = model.evaluate(X_test, y_test)
        annacc = results[1] * 100
        context={'data': 'Artificial Neural Network Run Successfully..!!', 'acc': str(annacc)}
        returnrender(request, "AdminApp/Algorithms.html", context)
    else:
        model = Sequential()
        model.add(Dense(200, input_shape=(10,),activation='relu', name='fc1'))
        model.add(Dense(200, activation='relu', name='fc2'))
        model.add(Dense(1, activation='sigmoid', name='output'))
        optimizer = Adam(lr=0.001)
        model.compile(optimizer, loss='binary_crossentropy', metrics=['accuracy'])
        print('CNN Neural Network Model Summary: ')
        print(model.summary())
        model.fit(X_train,y_train, batch_size=32, epochs=150)
        model.save("model/ANNModel.h5")
        results=model.evaluate(X_test, y_test)
        annacc = results[1] * 100
        context={'data': 'Artificial Neural Network Run Successfully..!!', 'acc': str(annacc)}
        returnrender(request, "AdminApp/Algorithms.html", context)
def runComparison(request):
    bars = ['Decision Tree', 'K-NN', 'ANN']
    heights = [decacc, knnacc, annacc]
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, heights)
    plt.xticks(y_pos, bars)
    plt.show()

    fig = plt.figure(figsize =(8, 5))
    plt.pie(heights, labels =bars )
    plt.show()
```

1. **Gini Impurity** (used in CART - Classification and Regression Trees):

$$Gini(D) = 1 - \sum_{i=1}^{n} p_i^2$$

Where pi is the probability of an element being classified into a particular class.

2.Entropy (used in ID3 algorithm):

$$Entropy(D) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

Where pi is the proportion of samples that belong to class iii.

3.Information Gain (used in ID3 and C4.5 algorithms):

$$Information\ Gain = Entropy(D) - \sum_{i=1}^{k} \frac{|D_i|}{|D|} Entropy(D_i)$$

5.Reduction in Variance (used in regression trees):

$$Var(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} (y_i - \overline{y})^2$$

3.2.1 Decision Tree

A Decision Tree is a supervised learning algorithm used for both classification and regression tasks. It works by splitting the data into subsets based on the value of input features, making decisions at each node to classify or predict the output.

How it works:

1.Root Node: The tree starts with a root node that represents the entire dataset.
2.Splitting: At each node, the data is split based on a feature that maximizes the separation of classes (for classification) or minimizes the prediction error (for regression). Common splitting criteria include Gini impurity, entropy, and information gain.
3.Decision Nodes and Leaves: Each internal node represents a decision based on a feature, while each leaf node represents a class label or a continuous value.
4.Prediction: For a new input, the tree traverses from the root to a leaf node based on the feature values, providing a classification or prediction.

Application in Liver Disease Prediction: A decision tree can be trained on patient data, where features might include liver enzyme levels, age, alcohol consumption, etc., to predict whether a patient has liver disease.

3.2.2 K-Nearest Neighbors (KNN):

KNN is a simple, instance-based learning algorithm used for classification and regression tasks. It classifies a new sample based on the majority class among its k-nearest neighbors.

How it works:

1.Training Phase: KNN does not involve explicit training. The algorithm simply stores the training dataset.
1.Distance Calculation: For a new input, the algorithm calculates the distance between the input and all points in the training set. Common distance metrics include Euclidean distance, Manhattan distance, and Minkowski distance.
2.Finding Neighbors: It selects the k-nearest data points (neighbors) based on the calculated distances.
3.Voting/Prediction: For classification, the algorithm assigns the class that is most common among the k neighbors. For regression, it averages the values of the k neighbors.

Application in Liver Disease Prediction: KNN can predict liver disease by comparing the health metrics of a new patient with those of known patients in the dataset and assigning a diagnosis based on the majority class of the nearest neighbors.

3.2.3 Artificial Neural Network (ANN)

ANN is a computational model inspired by the human brain, used for various complex tasks including classification, regression, and pattern recognition.

How it works:

1.Input Layer: Receives the input features of the data.
2.Hidden Layers: Consist of neurons that process the inputs. Each neuron applies a

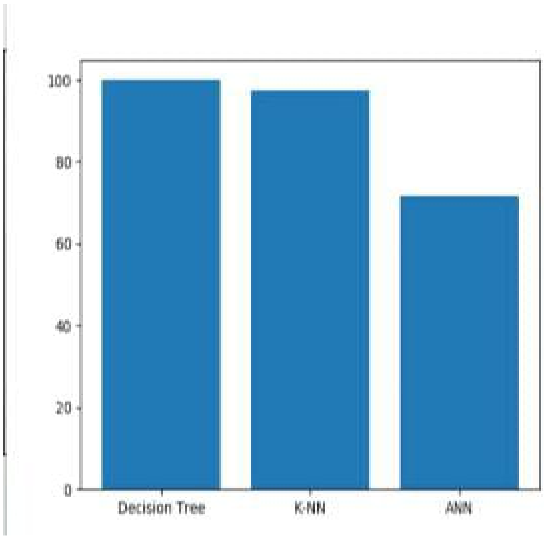
weighted sum of its inputs, passes the result through an activation function (e.g., sigmoid, ReLU), and passes the output to the next layer.

1.Output Layer: Produces the final prediction or classification.

2.Training: During training, the network adjusts the weights using backpropagation and optimization algorithms like gradient descent to minimize the prediction error.

Application in Liver Disease Prediction: An ANN can be trained on a large dataset of patient records, learning complex patterns and relationships between input features (e.g., blood test results, demographics) to accurately predict the presence of liver disease.

4.RESULTS AND DISCUSSION



. **Fig 1: Comparision grap**

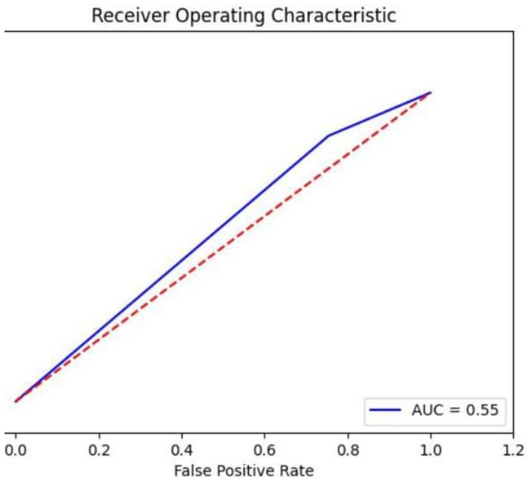


Fig 2: Bar graph

5.CONCLUSION

Diseases of the liver and heart become more common over time. With continuous technological advances, these will increase in the future. Today, people are becoming more health-conscious and are taking yoga and dance classes. Still, the sedentary lifestyle and luxury are constantly being introduced and improved. The problem will last for a long time. So, in such a scenario, our project is very useful to society. The dataset used in this project gave 99% accuracy in the Decision tree model. While it may be difficult to achieve such accuracy with such large datasets, the conclusion of this project is clear that liver risk can be predicted. In the future, philosophy is utilized to examine the liver area into distinct compartments for better classification accuracy. However, the technique requires further improvement generally to include the excretion of the liver into various parts: renal cortex, renal segment, renal medulla, and renal pelvis.

REFERENCES

[1] Mishra, Debakanta, Kaibalya R. Dash, ChittaranjanKhatua, SubhenduPanigrahi, Prasanta K. Parida, Sambit K. Behera, Rakesh K. Barik et al. "A study on the temporal trends in the etiology of cirrhosis of the liver in coastal eastern Odisha."

[2] Sontakke, Sumedh, Jay Lohokare, and Reshul Dani. "Diagnosis of liver diseases using machine learning." In 2017 International Conference on Emerging Trends & Innovation in ICT (ICEI), pp. 129-133. IEEE, 2017.

[3] Priya, M. Banu, P. Laura Juliet, and P. R. Tamilselvi. "Performance analysis of liver disease prediction using machine learning algorithms." International Research Journal of Engineering and Technology (IRJET) 5, no. 1 (2018): 206-211.[4]https://bmcmedicine.biomedcentral.com/articles

[5] Garg, Arunim, and Vijay Mago. "Role of machine learning in medical research: A survey." Computer Science Review 40 (2021): 100370.

[6] SinaBahramirad, Aida Mustapha, Maryam Eshraghi Classification of Liver Disease Diagnosis: A Comparative Study ISBN: 978-1-4673-5256- 7/13/$31.00 ©2013 IEEE

[7] Rezai, Bahram, and Ebrahim Allahkarami. "Application of Neural Networks in Wastewater Degradation Process for the Prediction of Removal Efficiency of Pollutants." In Soft Computing Techniques in Solid Waste and Wastewater Management, pp. 75-93. Elsevier, 2021.

[8] Rajeswari, P., & Reena, G. (2010). Analysis of Liver Disorder Using Data Mining Algorithm. Global Journal of Computer Science and Technology, Retrieved from https://computerresearch.org/index.php/computer/article /view/652

Author Profile:

[1].Mr. HIMAMBASHA SHAIK, currently working as an Assistant Professor in the Department of MCA, QIS College of Engineering and Technology, Ongole, Andhra Pradesh. Her area of interest Cloud Computing ,Devops and ,DBMS

[2]Ms, Keerthi Chimmiri currently pursuing Master of Computer Applications at QIS College of Engineering and Technology, ongole Andhra Pradesh. she completed BSC from Sri Harshini Degree College , ongole, Prakasam, Andhra Pradesh. His areas of interests are Machine Learning.