LIVER DISEASE PREDICTION AND ANALYSIS USING KNN

Submitted in partial fulfillment of the requirement
For
The award of degree of

Bachelor of Technology
in
Computer Science & Engineering

2021-2025

Submitted By

Batch no: 56

ABHINAY MARUPAKULA	–	2103A51527
VANCHANAGIRI YASHWANTH	–	2103A51439
BHARATH SIMHA REDDY KOLANU	–	2103A51524
VAMSHI PANAKA	–	2103A51375
DAYAKAR RAO SUKINE	–	2103A51211

Under the Guidance of

Dr. Ratnesh Ranjan

Assistant Professor

Department of CSE & AI
SR University
Anathasagar, Hasanparthy, Hanumakonda 506371

AY – 2024-2025

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE

CERTIFICATE

This is to certify that this Project entitled "**Liver Disease Prediction and Analysis Using K-Nearest Neighbor** ",is the bonfide work carried out by **M.Abhinay (2103A51527), K.Bharath (2103A51524) , P.Vamshi (2103A51375) , V.Yashwanth (2103A51439) ,S.Dayakar Rao (2103A51211)** B. Tech **IV year I semester**, as a Capstone Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE** during the academic year 2024-2025 under our guidance and Supervision.

Internal Guide

Dr. RATNESH RANJAN

Asst. Professor

HOD-CSE

Dr. M .SHESHIKALA

Professor

External Examiner

DECLARATION

I declare that this project report titled "**Liver Disease Prediction and Analysis using K - Nearest Neighbor "** submitted in partial fulfillment of the degree of B. Tech in CSE is a record of original work I carried out under the supervision of **Dr. Ratnesh Ranjan Asst. Professor,** and has not formed the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice of reporting scientific information, due acknowledgments have been made wherever the findings of others have been cited.

M.Abhinay	2103A51527
K.Bharath Reddy	2103A51524
P. Vamshi	2103A51375
V. Yashwanth	2103A51439
S. Dayakar Rao	2103A51211

ACKNOWLEDGEMENT

We owe an enormous debt of gratitude to our project guide Dr. Ratnesh Ranjan , Assistant Professor as well as Head of the CSE Department Dr. M.Sheshikala, Associate Professor for guiding us from the beginning through the end of the Minor Project with their intellectual advices and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We express our thanks to the project co-ordinators Dr. MD Sallauddin, Assoc. Prof for their encouragement and support.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, Dr.Indrajeet Gupta,Professor for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

M.Abhinay	2103A51527
K.Bharath Reddy	2103A51524
P. Vamshi	2103A51375
V. Yashwanth	2103A51439
S. Dayakar Rao	2103A51211

ABSTRACT

Liver diseases are significant health concern globally, attributed to various factors including alcohol consumption, exposure to environmental toxins, and poor dietary habits. Early detection of liver disease is crucial as it allows for timely intervention and management, which can significantly improve patient outcomes and reduce the burden on healthcare systems. The main aim of our project is to detect the liver disease accurately using parameters like age ,gender, total Bilirubin, Alkaline phosphatase, Aspartate aminotransferase, Albumin, Globulin ranges. The Existing model used to predict the liver cancer by using the random forest algorithm. The drawback of existing system is that the accuracy of model is less. The proposed model used to detect liver disease using K-Nearest Neighbor (K-NN) technique. The steps we follow to implement our model include collecting and preprocessing relevant datasets, identifying informative features, and training machine learning algorithms, the system can accurately detect the likelihood of liver cancer development or progression.

TOOLS USED : **:** PYTHON, JUPYTER NOTEBOOK, HTML, CSS

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER – 1

INTRODUCTION

1.1 Introduction

With a growing trend of sedentary and lack of bodily activities, diseases associated with liver have become a common come upon these days. Viral hepatitis alone reasons 1.34 million deaths every year. An early diagnosis of liver problems will boom affected person's survival rate. Liver screw ups are at high fee of danger amongst Indians. It's far expected that by way of 2025 India may emerge as the sector capital for liver sicknesses. The sizeable occurrence of liver contamination in India is contributed due to deskbound life-style, extended alcohol consumption and smoking. There are approximately a hundred varieties of liver infections. With such alarming figures, it is essential to have a subject in the direction of tackling those sicknesses. Afterall, we cannot assume an advanced and wealthy state, with unhealthy youths. Symptoms of liver disease include: Pain in the abdomen ,prone to bruising ,indigestion or vomiting, swelling in your legs or arms. Your risk of developing liver cancer can be increased by certain types of liver disease. Others continue to harm your liver if left untreated. Scarring, or cirrhosis, develops. Over the long haul, a harmed liver will not have sufficient sound tissue to work. Failure of the liver can result from untreated liver disease.

1.2 Problem Statement

The Problem Statement of this project revolves around the need to build a model which can accurately identify the liver disease by using certain parameters. The existing system used the random forest and decision tree algorithm to build a model for liver disease prediction. The models achieved about 60 -70% accuracy for predicting the liver disease. The Proposed system uses KNN algorithm to build the model and it aims to achieve the better prediction and accuracy compared to the previous models.

1.3 Objective of Project

The main objective of this project is:

1. Increased convenience for predicting a liver disease: As of now there is no accurate model that can tell you if you are in danger of having a liver disease from home just by giving some attributes. and the prediction comes to be danger or probability of having the liver disease you may consult our doctor as soon as possible.

2. Reduction in number of deaths due to liver diseases: Viral hepatitis alone causes 1.34 million deaths every year, Our main aim is to save as many life as we can Just by making people conscious about their liver.
3. More accurate diagnosis of liver disease by the doctors: As people are now concern about their liver now. After using our Machine Learning model they will contact their respective doctors about there Concern and that diagnosis will be more accurate.

1.4 Goal of the project

The goal of the liver disease prediction and analysis project using a k-nearest neighbors (KNN) model is to develop a reliable and accurate tool for early detection of chances of having a liver disease based on data clinical and biochemical data Age , Gender , Total_Bilirubin , Direct_Bilirubin , Alkaline_Phosph otase ,Alamine_Aminotransferase, Aspartate_Aminotransferase ,Total_Protiens , Albumin , Albumin_ and_Globulin_Ratio. This involves training a KNN model to differentiate between healthy individuals and those with liver disease, analyzing the importance of various risk factors, and evaluating the model's performance using metrics such as accuracy, precision, and also to share the model we have built with others. No matter how many models we create, if they remain offline, very few people will be able to see what we are achieving. That's why we should deploy our templates, so that anyone can play with them through a nice User Interface (UI). For this system, we build a single page web application with Flask as the UI of our system.

CHAPTER 2

PROBLEM IDENTIFICATION

2.1 Existing System

The Existing liver disease diagnosis system involves doctors or medical professionals using various medical tests, such as blood tests, biopsy, and imaging techniques like ultrasound, MRI, or CT scans to identify liver disease in patients. The interpretation of the test results and diagnosis is done by medical professionals based on their experience and knowledge. This approach can be time-consuming and costly, and the accuracy of diagnosis may depend on the skills and experience of the medical professionals. Liver disease prediction systems have played a vital part in people's lives, and many scholars believe it to be an important topic. Although the results of the forecast are promising, these old methods are still far from being highly precise and efficient.

Existing systems are straightforward and effective, but they are extremely sensitive to disruption. Furthermore, state-of-the-art methods only use one algorithm, resulting in erroneous findings. This could lead to erroneous assumptions and incorrect diagnoses and treatments for patients. In recent years, automated prediction systems have incorporated machine learning models like decision trees and random forests. Decision trees provide a straightforward approach by splitting the data into branches based on feature values, which helps in understanding the decision-making process but can suffer from overfitting and instability with small variations in the data. Random forests, which are ensembles of multiple decision trees, improve upon this by reducing overfitting and increasing prediction accuracy through the aggregation of multiple trees' results. However, these models still have limitations, such as requiring extensive computational resources and sometimes lacking interpretability. Despite these advances, there remains a need for even more refined, accurate, and efficient prediction models to enhance early detection and treatment of liver disease.

2.2 Proposed System

In the Proposed work the liver disease prediction model is build using k-nearest neighbor. There are many factors which causes the liver disease. Some of them which influence to detect the liver disease are Total Bilirubin (Total amount of bilirubin when old red blood cell breaks down inside the human body), Direct Bilirubin (It is a substance is 9 made when the body breakdowns the old red blood cells. It is also part of bile, which your liver makes to help digest the food we eat), Alkaline Phosphatase (It is part of protein which release the enzymes to act as a catalyst which help the bile juice which produce by the

liver), Alamine Aminotransferase (It is found in the plasma and various body part mostly in the liver), Aspartate Aminotransferase (It is a part of metabolism which help to digest the food we eat and keep the liver healthy), Total Proteins (Total Protein value present in the body), Albumin (Albumin is a kind of protein which is found inside human body and a major part for participation in total protein value), Albumin and Globulin Ratio (It is the ratio of both the proteins inside human body i.e. the ratio of albumin and Globulin).

The models could be trained on various features, such as liver enzymes, blood tests, imaging, and other clinical data. Once the models are trained, they can be used to diagnose patients with liver disease with high accuracy. The proposed system has several advantages over the existing system, including Accuracy, speed, cost- effective, consistency and scalability. We are using k-nearest neighbors (KNN) model to enhance early detection and classification accuracy. KNN is a simple yet powerful machine learning algorithm that classifies data points based on their proximity to other labeled data points in the feature space. The KNN model can identify patterns and similarities that indicate the presence or absence of liver disease. This system aims to overcome the limitations of traditional methods and existing models . The proposed system has several advantages over the existing system, including Accuracy, speed, cost-effective, consistency and scalability.

CHAPTER 3

SOFTWARE AND HARDWARE REQUIREMENTS

3.1 Software Requirements

- **Operating System** **:** Windows XP & higher
- **IDE** **:** Jupyter Notebook
- **Programming Language** : Python , html
- **Libraries** :
  - pandas
  - numpy
  - sci-kit-learn
  - seaborn
  - pickle
- **Development Tools:** Jupyter Notebook, VS Code

3.2 Hardware Requirements

Device name : PC

Processor: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz

Installed RAM: 8.00 GB

Device ID: AF922660-7000-4475-AE79-CD2EFCE01E25

System type: 64-bit operating system, x64-based processor

CHAPTER 4

DESIGN AND IMPLEMENTATION

4.1 Design

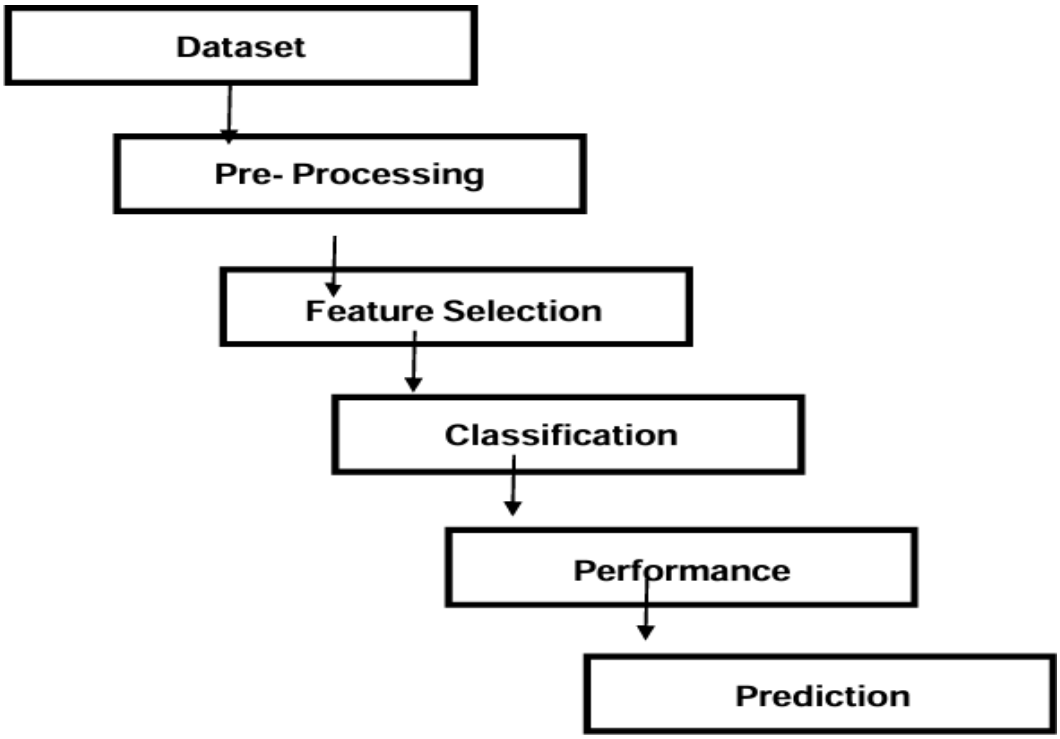

Fig 4.1 Proposed System Flowchart

Fig 4.1 describes - Dataset :We obtained the Indian Liver Patient Dataset (ILPD) . There are 416 records for liver patients and 167 for non-liver patients in this dataset. The Selector is a class label that is used to divide people into groups (liver patients or not). There are 441 male and 142 female patient records in this dataset, and have attributes Age , Gender , Total_Bilirubin , Direct_Bilirubin , Alkaline_Phosphotase ,Alamine_ Am ino transferase, Aspartate_Aminotransferase ,Total_Protiens , Albumin , Albumin_and_Globulin_Ratio .

Data pre-processing :At first, the preprocessing of data is carried out once it is collected. In this step, several tasks are performed. The collected data include many records that may have missing data or values consider age features. In general, the missing values are replaced with the nearest or closest value to their feature. And the liver disease target data are categorized into two groups i.e. group 1 represent the presence of liver disease and group 2 represent the absence of liver disease patient records. The values of the target label in a classification model are converted into non- numeric. Afterward, the division of the dataset is carried out into two groups: training and testing.

Feature Selection : One of the main segments in liver disease prediction is the selection of important features of liver disorder. In this step, several features such as age, gender that represent the personal information of each patient is selected. Some other clinical features are also collected from different medical tests.

Data Classification : Classification is an important process and function in data mining. The function of the collected items assigns to the target class or category. The classification aims to get the target class to predict accurately for all case data. After data pre- processing, features are inserted in a classification knn model.

Performance:Different classification criteria including accuracy, precision are computed for performance evaluation of the classification and performance of the model is analyzed.

Prediction **:** After completion of building the model we build a website .It predicts the output by using input values from the dataset or values given by user.

4.2 Implementation

Import Required Libraries :The first step is to import the necessary libraries for data manipulation, model building, and evaluation. We will be using pandas for data handling, scikit-learn for machine learning tasks, and numpy for numerical operations.

Load the Dataset Next, we need to load the Indian liver patient disease dataset into a pandas DataFrame.

Separate Features and Target Variable : After loading the dataset, we need to separate the features (independent variables) and the target variable (dependent variable). In this case, the target variable represents whether a person has liver disease or not.

Split the Data into Training and Testing Sets : To evaluate the performance of the KNN model, we need to split the dataset into training and testing sets. The model will be trained on the training set and evaluated on the testing set.

Scale the Data : KNN is a distance-based algorithm, and it is sensitive to the scale of the features. To ensure that all features are on the same scale, we need to perform feature scaling.

Create a KNN Classifier Instance and train the model: Next, we need to create an instance of the KNN classifier from scikit-learn, With the scaled training data we will train the KNN model.

KNN (K-Nearest Neighbor): KNN is a supervised machine learning algorithm used for classification and regression tasks. In classification, KNN predicts the class of a data point by looking at the 'k' nearest data points and determining the majority class among them. In regression, it predicts the value of a data point by averaging the values of its 'k' nearest neighbors.

Make Predictions on the Test Set : After training the model, we can use it to make predictions on the test set .

Evaluate the Model's Performance : Finally, we can evaluate the performance of the trained KNN model by calculating the accuracy and precision score.

Analysis and Interpretation : After evaluating the model's performance, you can analyze the results to gain insights into the effectiveness of the KNN algorithm for liver disease prediction. Based on the analysis, you can make informed decisions about whether the KNN algorithm is suitable for liver disease prediction or if you need to explore other machine learning algorithms or techniques to improve the model's performance.

CHAPTER – 5

SAMPLE SOURCE CODE

5.1 Source Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
dataset = pd.read_csv('indian_liver_patient.csv')
dataset.head()
dataset.describe()
dataset.shape
dataset.columns
#Data Cleaning
dataset.duplicated()
dataset.duplicated().sum()
dataset = dataset.drop_duplicates()
print( dataset.shape )
# Checking Missing Values
dataset.isna().sum()
sns.boxplot(data = dataset, x= 'Albumin_and_Globulin_Ratio' )
dataset['Albumin_and_Globulin_Ratio'].mode()
dataset['Albumin_and_Globulin_Ratio'].median()
dataset['Albumin_and_Globulin_Ratio'].mean()
dataset['Albumin_and_Globulin_Ratio'] =
dataset['Albumin_and_Globulin_Ratio'].fillna(dataset['Albumin_and_Globulin_Ratio'].median())
dataset.isna().sum()
import seaborn as sns
sns.countplot(data = dataset, x='Gender', label='count')
Male, Female = dataset['Gender'].value_counts()
print('Number of patients that are male: ',Male)
print('Number of patients that are female: ',Female)
def partition(x):
    if x == 'Male':
```

```
        return 1
    return 0
dataset['Gender'] = dataset['Gender'].map(partition)
dataset
def partition(x):
    if x == 2:
        return 0
    return 1
dataset['Dataset'] = dataset['Dataset'].map(partition)
dataset['Dataset']
plt.figure(figsize=(10,10))
sns.heatmap(dataset.corr())
x = dataset.iloc[: , :-1].values
y = dataset.iloc[: , -1].values
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x , y , test_size = 0.25, random_state = 42)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
x_train.shape
from sklearn.neighbors import KNeighborsClassifier
knn_classifier = KNeighborsClassifier(n_neighbors =21, metric = 'minkowski')
knn_classifier.fit(x_train, y_train)
knn_y_pred = knn_classifier.predict(x_test)
from sklearn.metrics import confusion_matrix
knn_cm = confusion_matrix(y_test, knn_y_pred)
sns.heatmap(knn_cm , annot=True)
from sklearn.metrics import accuracy_score, precision_score
print(accuracy_score(y_test,knn_y_pred))
print(precision_score(y_test , knn_y_pred))
pip install --upgrade scikit-learn
import pickle
pickle.dump(knn_classifier, open('model.pkl', 'wb'))
```

```
pickle.dump(sc, open('sc.pkl', 'wb'))
```

index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>E-Hospital</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<style>
body {
  font-family: Arial, Helvetica, sans-serif;}
.navbar {
  overflow: hidden;
  background-color: rgba(252, 17, 107, 0.74);}
.navbar a {
  float: left;
  font-size: 16px;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;}
.dropdown {
  float: left;
  overflow: hidden;}
.dropdown .dropbtn {
  font-size: 16px;
  border: none;
  outline: none;
  color: white;
  padding: 14px 16px;
  background-color: inherit;
  font-family: inherit;
  margin: 0;}
```

```
.navbar a:hover, .dropdown:hover .dropbtn {
  background-color: rgb(56, 15, 238);}
.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2); z-index: 1;}
.dropdown-content a {
  float: none;
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
  text-align: left;}
.dropdown-content a:hover {
  background-color: #ddd;}
.dropdown:hover .dropdown-content {
  display: block;}
input[type=text], select {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;}
.registerbtn {
  background-color: #4CAF50;
  color: white;
  padding: 16px 20px;
  margin: 8px 0;
  border:  none;
  cursor: pointer;
```

```
  width: 100%;
  opacity: 0.9;}
.registerbtn:hover {
opacity:1;}
input[type=submit]:hover {
background-color: #45a049;}
.center {
  margin: auto;
  width: 60%;
  border: 3px solid #4d00dbc5;
  padding: 10px;}
</style>
</head>
<body>
<h2><a href = '/'> <p style="text-align:center;">Liver Disease Prediction</p></a></h2>
   <div class="center">
  <form action="/predict"method="post">
    <input type="text" id="Age" placeholder="Age " name="Age">
    <br>
    <input type="text" id="Gender" placeholder="Gender (Male:1 , female:0)" name="Gender"><br>
    <input type="text" id="Total_Bilirubin" placeholder="Total_Bilirubin" name="Total_Bilirubin">
    <br>
    <input type="text" id="Direct_Bilirubin" placeholder="Direct Bilirubin" name="Direct_Bilirubin">
    <br>
    <input type="text" id="Alkaline_Phosphotase" placeholder="Alkaline Phosphotase"
name="Alkaline_Phosphotase">
    <br>
    <input type="text" id="Alamine_Aminotransferase" placeholder="Alamine Aminotransferase"
name="Alamine_Aminotransferase">
    <br>
    <input type="text" id="Aspartate_Aminotransferase" placeholder="Aspartate_Aminotransferase"
name="Aspartate_Aminotransferase">
    <br>
    <input type="text" id="Total_Protiens" placeholder="Total Protiens" name="Total_Protiens">
```

```
    <br>
    <input type="text" id="Albumin" placeholder="Albumin " name="Albumin">
    <br>
    <input type="text" id="Albumin_and_Globulin_Ratio" placeholder="Albumin and Globulin_Ratio"
name="Albumin_and_Globulin_Ratio">
    <br>
    <br>
    <button type="submit" class="registerbtn">Predict</button>
</form></div> </body> </html>
```

Result.html

```
<!DOCTYPE html>
<html>
<head>
  <title>E-Hospital</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css">
<style>
body {
  font-family: Arial, Helvetica, sans-serif;}
.navbar {
  overflow: hidden;
  background-color: rgba(252, 17, 107, 0.74);}
.navbar a {
  float: left;
  font-size: 16px;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;}
.dropdown {
  float: left;
  overflow: hidden;}
```

```
.dropdown .dropbtn {
  font-size: 16px;
  border: none;
  outline: none;
  color: white;
  padding: 14px 16px;
  background-color: inherit;
  font-family: inherit;
  margin: 0;}
.navbar a:hover, .dropdown:hover .dropbtn {
  background-color: rgb(56, 15, 238);}
.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;}
.dropdown-content a {
  float: none;
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
  text-align: left;}
.dropdown-content a:hover {
  background-color: #ddd;}
.dropdown:hover .dropdown-content {
  display: block;}
input[type=text], select {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
```

```
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;}
.registerbtn {
  background-color: #4CAF50;
  color: white;
  padding: 16px 20px;
  margin: 8px 0;
  border:  none;
  cursor: pointer;
  width: 100%;
  opacity: 0.9;}
.registerbtn:hover {
opacity:1;}
input[type=submit]:hover {
background-color: #45a049;}
.center {
  margin: auto;
  width: 60%;
  border: 3px solid #4d00dbc5;
  padding: 10px;}
</style></head><body>
      <h2 ><p style="text-align: center;">Results:</p> </h2>
    {% if prediction == 1%}
    <h2 style="color: red;">Chances of having Liver Disease is more, please consult a Doctor.</h2>
    <h3>Symptoms</h3>
    <p>Classic symptoms of liver disease include:</p>
     <ul> <li>nausea</li>
      <li>vomiting</li>
      <li>right upper quadrant abdominal pain, and </li>
      <li>jaundice (a yellow discoloration of the skin due to elevated bilirubin concentrations in the
bloodstream).</li>
    </ul>
    <br>
```

```
    <p style="text-align:center;"><img src="https://img.collegedekhocdn.com/media/img/careers/doctor-clinic.jpg" alt="HTML" width="350" height="250"></p>
    {% elif prediction == 0%}
    <h2 style="color: blue;">No Worries!!! You don't have Liver Disease.</h2>
    <p style="text-align: center;"><img src="https://img.collegedekhocdn.com/media/img/careers/doctor-clinic.jpg" alt="HTML" width="350" height="250"></p>
    {% endif %}
    </body>
</html>
```

App.py

```
from flask import Flask , request ,url_for , render_template
import numpy as np
import pickle
sc = pickle.load(open('sc.pkl' , 'rb'))
model = pickle.load(open('model.pkl' , 'rb'))
app = Flask(_name_)
@app.route('/')
def home():
    return render_template('index.html')
@app.route('/predict' , methods=['POST'])
def predict():
    inputs = [float(x) for x in request.form.values()]
    inputs = np.array([inputs])
    inputs = sc.transform(inputs)
    output = model.predict(inputs)
    if output < 0.5:
        output = 0
    else:
        output = 1
    return render_template('result.html' , prediction = output)
if __name__=='_main_':
    app.run(debug=True)
```

5.2 Screenshots of the application:

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Dataset
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90	1
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	1
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	1
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	1.00	1
4	72	Male	3.9	2.0	195	27	59	7.3	2.4	0.40	1

Fig.5.1 Printing first 5 rows

Fig 5.1 shows the top five records by using the pandas method head() of Indian_liver_patient dataset withattributes Age , Gender , Total_Bilirubin,Direct_Bilirubin, Alkaline_Phosphotase ,Alamine_Aminotransferase , Aspartate_ Amino transferase ,Total_Protiens , Albumin , Albumin_and_Globulin_Ratio , Dataset.

	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Dataset
count	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	579.000000	583.000000
mean	44.746141	3.298799	1.486106	290.576329	80.713551	109.910806	6.483190	3.141852	0.947064	1.286449
std	16.189833	6.209522	2.808498	242.937989	182.620356	288.918529	1.085451	0.795519	0.319592	0.452490
min	4.000000	0.400000	0.100000	63.000000	10.000000	10.000000	2.700000	0.900000	0.300000	1.000000
25%	33.000000	0.800000	0.200000	175.500000	23.000000	25.000000	5.800000	2.600000	0.700000	1.000000
50%	45.000000	1.000000	0.300000	208.000000	35.000000	42.000000	6.600000	3.100000	0.930000	1.000000
75%	58.000000	2.600000	1.300000	298.500000	60.500000	87.000000	7.200000	3.800000	1.100000	2.000000
max	90.000000	75.000000	19.700000	2110.000000	2000.000000	4929.000000	9.600000	5.500000	2.800000	2.000000

Fig.5.2 Describing the dataset

Fig 5.2 describes the total records of the datsaset by using describe() method , It generates descriptive statistics that summarize the central tendency, dispersion, and shape of the dataset, The method returns important statistics such as count, mean, standard deviation, minimum, maximum, and the quartiles (25th, 50th, and 75th percentiles).of the records for each attribute. It gives a quick statistical overview of your data, helping to understand its distribution and identify potential outliers.

```
Index(['Age', 'Gender', 'Total_Bilirubin', 'Direct_Bilirubin',
       'Alkaline_Phosphotase', 'Alamine_Aminotransferase',
       'Aspartate_Aminotransferase', 'Total_Protiens', 'Albumin',
       'Albumin_and_Globulin_Ratio', 'Dataset'],
      dtype='object')
```

Fig.5.3 Printing the colums

Fig 5.3 shows the column names in the dataset Age , Gender , Total_Bilirubin , Direct_Bilirubin , Alkaline_Phosphotase ,Alamine_Aminotransferase , Aspartate_Aminotransferase ,Total_Protiens , Albumin , Albumin_and_Globulin_Ratio , Dataset, using column() attribute in Pandas that lists all the column names of a DataFrame. It returns an Index object containing the names of the columns.

```
0       False
1       False
2       False
3       False
4       False
        ...
578     False
579     False
580     False
581     False
582     False
Length: 583, dtype: bool
```

Fig.5.4 checking the duplicate values

Fig 5.4 decribes the duplicate values dataset for that we used duplicated() method it identifies duplicate rows in a DataFrame, returning a Series of True for duplicates and False for unique rows. It helps in data cleaning by indicating which rows are redundant. You can customize the check to specific columns and control whether to mark all duplicates or only those after the first occurrence. This method is useful for maintaining data integrity by spotting and removing duplicate entries.

```
Age                          0
Gender                       0
Total_Bilirubin              0
Direct_Bilirubin             0
Alkaline_Phosphotase         0
Alamine_Aminotransferase     0
Aspartate_Aminotransferase   0
Total_Protiens               0
Albumin                      0
Albumin_and_Globulin_Ratio   4
Dataset                      0
dtype: int64
```

Fig.5.5 checking missing values

Fig 5.5 shows the sum of missing values in each attribute Age , Gender , Total_Bilirubin , Direct_Bilirubin , Alkaline_Phosphotase ,Alamine_Aminotransferase , Aspartate_Aminotransferase ,Total_Protiens , Albumin , Albumin_and_Globulin_Ratio , Dataset. We used isnull().sum() method that helps identify missing values in a Dataset. This method is useful for quickly assessing the extent of missing data in each column, facilitating data cleaning and preprocessing.

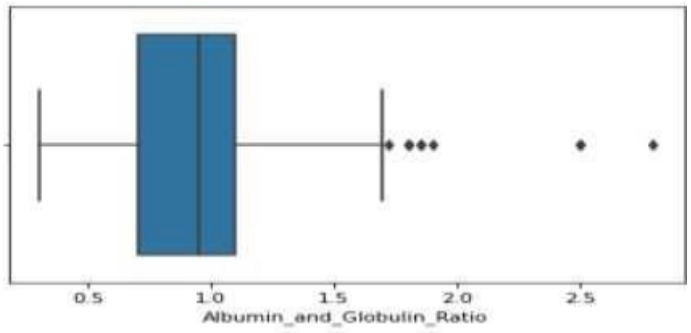

Fig.5.6 Boxplot of albumin and globulin ranges

Fig 5.6 describes a box plot that visualize the distribution of the 'Albumin_and_Globulin_Ratio' column in the dataset. which explains distribution of data based on a five-number summary: minimum, first quartile (Q1), median, third quartile (Q3), and maximum. This plot helps identify outliers, detect skewness, and compare the distribution of data in the 'Albumin_and_Globulin_Ratio' column.

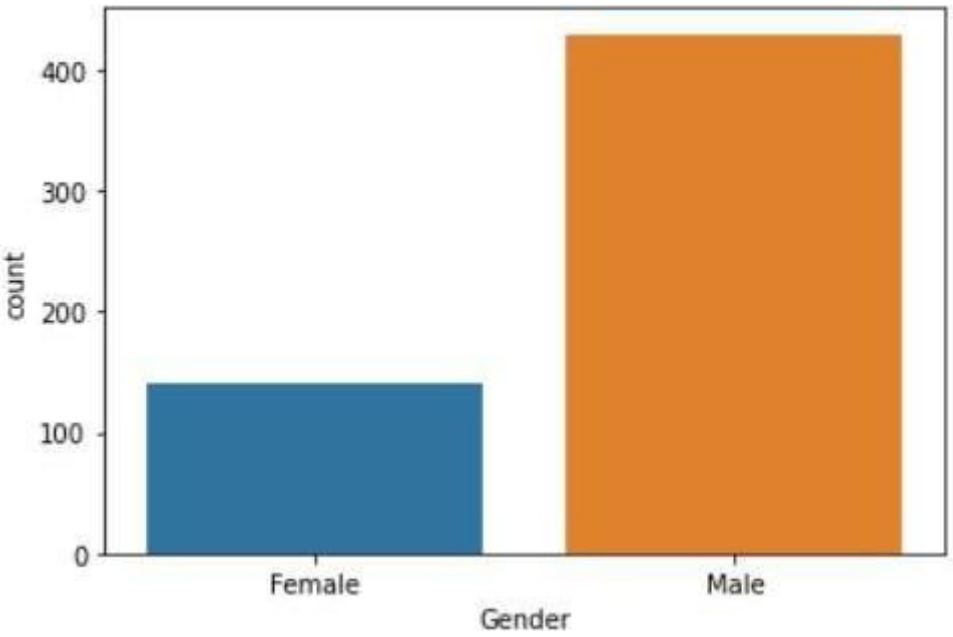

Fig.5.7 Countplot of number of male and female

Fig 5.7 describes a bar plot that visualize the count of each unique value in the 'Gender' column with the height of each bar representing the frequency of that value. of the dataset. It shows the count of female records are between 100-200 and the count of male records are above 400. This plot helps to quickly understand the distribution and proportion of different gender categories in the dataset.

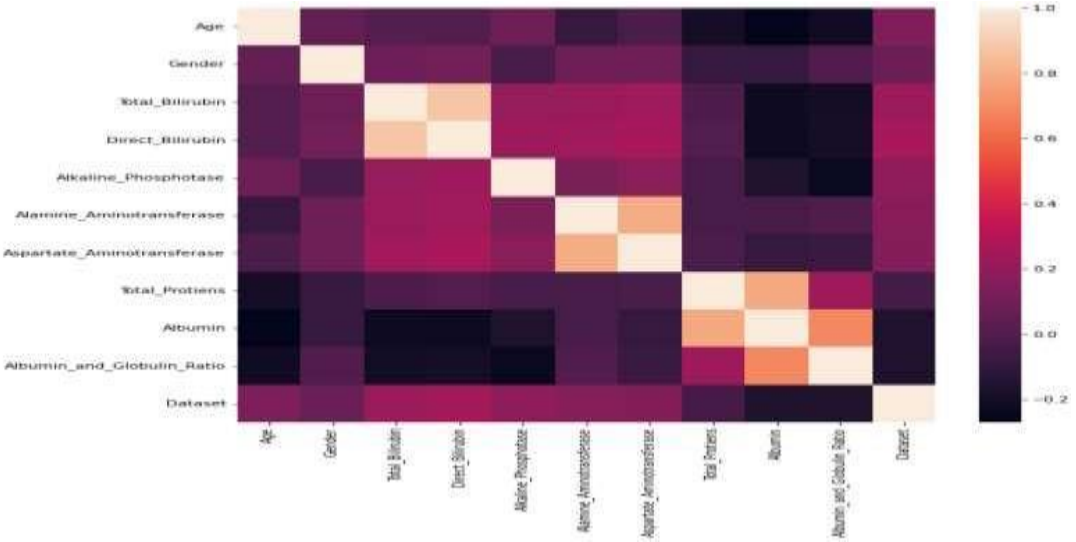

Fig.5.8 Correlation matrix

Fig 5.8 displays the correlations between different numerical variables in the dataset. It also calculates the correlation coefficients between all pairs of numerical columns in the dataset. It helps in identifying relationships between variables; positive correlations are indicated by warmer colors (red), negative correlations by cooler colors , and no correlation by neutral colors (e.g., white).

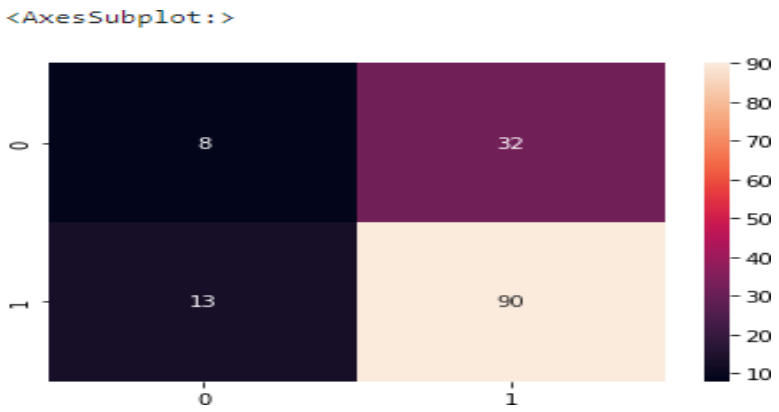

Fig.5.9 Confusion matrix

Fig 5.9 displays the confusion matrix, a tabular representation of model predictions versus ground truth labels. where each cell represents the count of true positives, true negatives, false positives, and false negatives. visually highlights the performance of the classification model, helping to identify areas of correct and incorrect predictions across different classes.

```
0.6853146853146853
0.7377049180327869
```

Fig.5.10 Accuracy and Precision score

Fig 5.10 shows the accuracy and precision score of the k-nn model. It says that the accuracy of k-nn model is 68% and precision score of the model is 73%. The accuracy score computes the proportion of correctly classified samples out of the total samples. It's a measure of overall correctness of the model's predictions. The precision score calculates the ratio of correctly predicted positive observations to the total predicted positives.

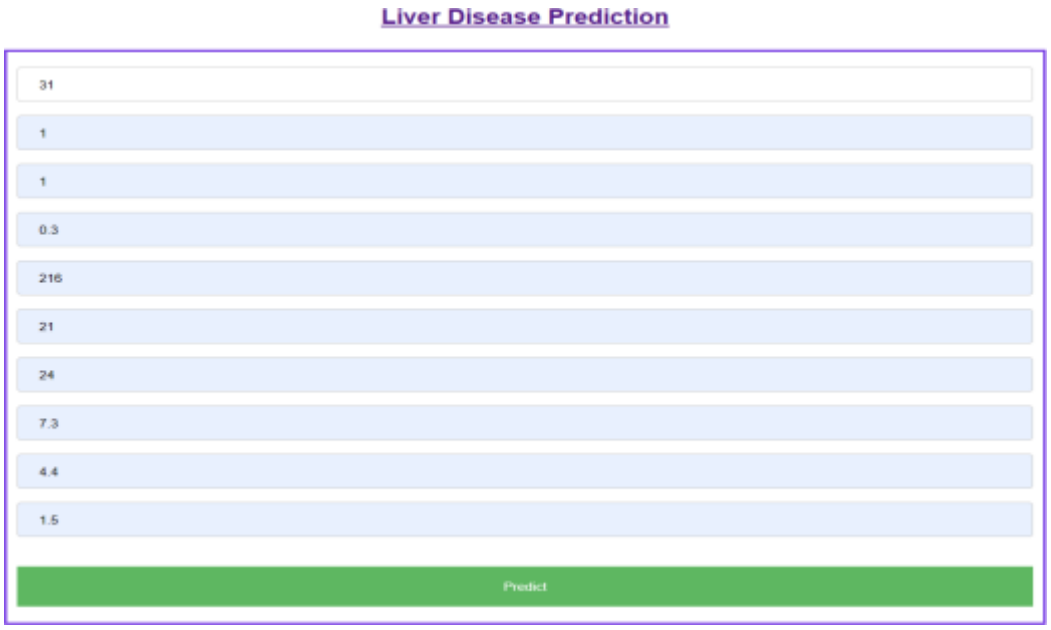

Fig.5.11 Getting input from user

Fig 5.11 displays a form with input fields for various medical features like Age, Gender, Total Bilirubin, etc., which users can fill out. Upon submission, the form sends a POST request to the "/predict" endpoint for disease prediction. The page is styled using CSS, featuring a centered layout, input fields with placeholders, and a Predict button styled as a green button. The application likely utilizes a backend server to handle the prediction logic and serve this HTML page to users.

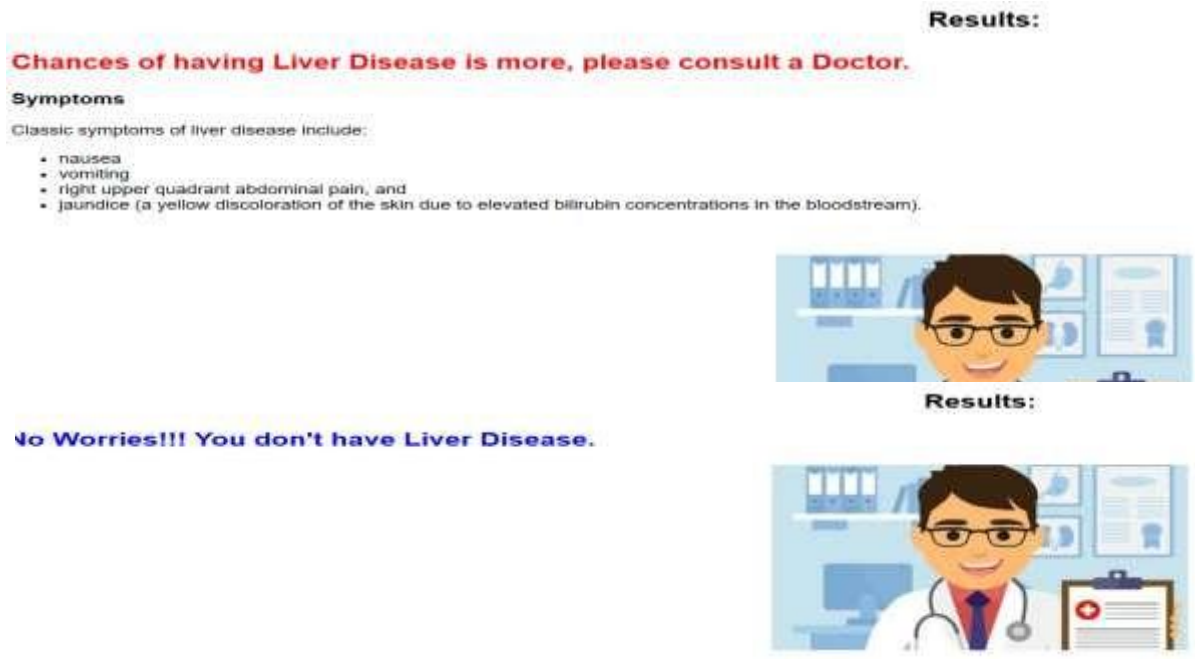

Fig 5.12 Predicted output

Fig 5.12 displays the output of the liver disease prediction it gives different messages and images based on the prediction outcome, indicating whether the user is predicted to have liver disease or not. The page is styled using CSS for visual appeal, with conditional rendering implemented to display content based on the prediction result. Images of a doctor's clinic are included to provide visual context. Overall, it effectively communicates the prediction results to the user in a clear and visually appealing manner.

CHAPTER 6

CONCLUSIONS & FUTURE SCOPE

6.1 Conclusion

Machine learning has shown promising results in the field of liver disease diagnosis. Various studies have been conducted to develop models that can accurately predict the presence of liver diseases based on patient data. In Conclusion the accuracy of the K-NN model we developed ranges from 60 percent to 70 percent. We have used a specific dataset Indian liver patient dataset where we have 11 attributes and 583 patient's data.In future we would try to enhance the algorithm's performance, particularly in handling large-scale datasets and improving its predictive accuracy.

6.2 Future Scope

One area for improvement is the use of more advanced algorithms and models for data processing and analysis. This can lead to even more accurate predictions, as well as improve efficiency and speed. Furthermore, the use of machine learning techniques in combination with other medical technologies, such as wearable devices and telemedicine, can provide more comprehensive and continuous monitoring of patients with liver disorders. Development of user-friendly interfaces and decision support systems to help medical professionals interpret and act on the results generated by machine learning models. Overall, future enhancements for liver disorder diagnosis using machine learning techniques will likely focus on improving accuracy, efficiency, and accessibility, as well as integrating multiple data sources and medical technologies to provide more comprehensive and personalized patient care.

REFERENCES

[1] Gupta, A., Biwal, R., Joshi, S., & Singh, P. A COMPARATIVE STUDY ON LIVER DISEASE PREDICTION USING SUPPORT VECTOR MACHINE ALGORITHM.

[2] W. Yu, T. Liu, R. Valdez, M. Gwinn, and M. J. Khoury, ‘‘Application of support vector Machine modeling for prediction of common diseases: The case of diabetes and pre-diabetes,’’ BMC Med. Inform. Decis. Making, vol. 10, no. 1, p. 16, 2010.

[3] Q. Zou, K. Qu, Y. Luo, D. Yin, Y. Ju, and H. Tang, ‘‘Predicting diabetes mellitus with machine learning techniques,’’ Frontiers Genet., vol. 9, p. 515, Nov. 2018.

[4] Z.-H. Zhou, Y. Jiang, Y.-B. Yang, and S.-F. Chen, ‘‘Lung cancer cell identification based on artificial neural network ensembles,’’ Artif. Intell. Med., vol. 24, no. 1, pp. 25–36, 2002.

[5] S. F. Dessai, ‘‘Intelligent heart disease prediction system using probabilistic neural network,’’ Int. J. Adv. Comput. Theory Eng., vol. 2, no. 3, pp. 2319–2526, 2013

[6] B. E. Bejnordi et al., ‘‘Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer,’’ JAMA, vol. 318, no. 22, pp. 2199–2210, Dec. 2017

[7] Y. Cao, Z.-D. Hu, X.-F. Liu, A.-M. Deng, and C.-J. Hu, ‘‘An MLP classifier for prediction of HBV-induced liver cirrhosis using routinely available clinical parameters,’’ Disease Markers, vol. 35, no. 6, pp. 653– 660, 2013.

[8] K. Hornik, M. Stinchcombe, and H. White, ‘‘Multilayer feedforward networks are universal approximators,’’ Neural Netw., vol. 2, no. 5, pp. 359–366, 1989.

[9] G. E. Hinton, S. Osindero, and Y.-W. Teh, ‘‘A fast learning algorithm for deep belief nets,’’ Neural Comput., vol. 18, no. 7, pp. 1527–1554, 2006.

[10] A. L. Samuel, ‘‘Some studies in machine learning using the game of checkers,’’ IBM J. Res. Develop., vol. 3, no. 3, pp. 29–210, 1959.

[11] K. Tretyakov, ‘‘Machine learning techniques in spam filtering,’’ Data Mining Problem-Ori

[12] D. West and V. West, ‘‘Improving diagnostic accuracy using a hierarchical neural network to model decision subtasks,’’ Int. J. Med. Inform., vol. 57, no. 1, pp. 41–55, 2000. ented Seminar, MTAT, vol. 3, no. 177, pp. 60–79, 2004

[13] Divya, B, Kalaiselvi, R, Review on Confidentiality of the Outsourced Data, Research Journal of Science and Engineering Systems, vol.1, pp.1-7, 2017.

[14] P. Lakhani and B. Sundaram, ‘‘Deep learning at chest radiography: Automated classification of pulmonary tuberculosis by using convolutional neural networks,’’ Radiology, vol. 284, no. 2, pp. 574 582, Aug. 2017.

[15] S. L. Oh, Y. Hagiwara, U. Raghavendra, R. Yuvaraj, N. Arunkumar, M. Murugappan, and U. R. Acharya, ‘‘A deep learning approach for Parkinson’s disease diagnosis from EEG signals,’’ Neural Comput. Appl., Aug. 2018. doi: 10.1007/s00521-018-3689-5.