Rajiv Gandhi University of Knowledge Technologies

R.K Valley, Y.S.R Kadapa (Dist)-516330

A Report On

HearSmoking Smoking Detection

Submitted by

M.Venkata Teja	R170395
C.Indu	R170470
K.Jaya Prakash	R170319
C.Suneel	R170315
N.Mani Kumar	R170324

Under the guidance of

A.Mahendra sir

(Assistant Professor)

Department of Computer Science Engineering

This project report has been submitted in fulfillment of the requirements for the Degree of Bachelor of Technology in software Engineering

May2023

Rajiv Gandhi University of Knowledge Technologies

IIIT,R.K.Valley,YSR Kadapa(Dist)-516330

CERTIFICATE

This is to certify that report entitled **“HearSmoking Smoking Detection”** Submitted by M.Venkata Teja (R170395) , C.Indu (R170470) , K.Jaya Prakash (R170319) , N.Mani Kumar (R170324) , C.Suneel (R170315) in partial fulfillment of the requirements of the award of bachelor of technology in Computer Science Engineering is a Bonafide work carried out by them under the Supervision and Guidance.

The Report has been not submitted previously in part or full to this or any other university or institute for the award of any degree or diploma.

Guide	**HOD**
Mr.A.MAHENDRA	Mr.N.SATYANANDARAM
Assistant Professor	HOD of CSE
RGUKT, RK VALLEY	RGUKT, RK VALLEY

Submitted On...

ACKNOWLEDGEMENT

The satisfaction that accompanies the succeful completion of any task would be incomplete without the mention of the people who made it possible and who’s constant guidance and encouragement crown all the efforts success.

I would like to express my sincere gratitude to **Mr.A.Mahendra sir** , my project guide for valuable suggestions and keen interest throughout the progress of my project.

I am grateful to **Mr.N.Satyanandaram sir HOD CSE** ,for providing excellent Computing facilities and congenial atmosphere for progressing my project.

At the outset,I would like to thank **Rajiv Gandhi Of University of Knowledge Technologies(RGUKT),**for providing all the necessary resources and support for the successful completion of my course work.

DECLARATION

We hereby declare that this report entitled **“HearSmoking Smoking Detection”** Submitted by us under the guidance of and Supervision of **Mr.Mahendra** , is a bonafide work we also declare that it has not been submitted previously in part or in full to this University or other institution for the award of any degree or diploma.

Date:- 06-05-2023
Place:-RK VALLEY

N.Mani Kumar(R170324)
K.Jaya Prakash(R170319)
C.Indu(R170470)
M.Venkata Teja(R170395)
C.Suneel(R170315)

INDEX

ABSTRACT

Driving safety has drawn much public attention in recent years due to the fast- growing number of cars. Smoking is one of the threats to driving safety but is often ignored by drivers. Existing works on smoking detection either work in contact manner or need additional devices. This motivates us to explore the practicability of using smartphones to detect smoking events in driving environment. In this paper, we propose a cigarette smoking detection system, named HearSmoking, which only uses acoustic sensors on smartphones to improve driving safety. After investigating typical smoking habits of drivers, including hand movement and chest fluctuation,
we design an acoustic signal to be emitted by the speaker and received by the microphone.

We calculate Relative Correlation Coefficient of received signals to obtain movement patterns of hands and chest. The processed data is sent into a trained Convolutional Neural Network for classification of hand movement. We also design a method to detect respiration at the same time. To improve system performance, we further analyse the periodicity of the composite smoking motion.Through extensive experiments in real driving environments, HearSmoking detects smoking events with an average total accuracy of 93:44% in real-time.

INTRODUCTION :

1.1 Introduction

With rapid development and great success of automotive industry, more and more vehicles have been put into use. On one hand, these modern means of transports bring people with much convenience in daily life. On the other hand, various issues related to road safety increase and arouse wide public attention. A series of efforts have been undertaken by traffic departments or government organizations to improve road safety, such as installing surveillance cameras and making traffic rules, which can in some way regulate driving behavior.

Among different undesirable driving habits, smoking is a kind of behavior that can easily distract a driver’s attention and cause danger.Unfortunately, most of drivers fail to realize the risk of smoking.

However, to the best of our knowledge, a ubiquitous smoking detection system designed for driving environment is still absent. Smart phones, with their powerful capability and usability in driving, is highly ideal to act as the platform of a smoking detection system. In view of the aforementioned situations and motivations, we take the first attempt to build a novel smoking detection system, which uses acoustic sensors in smart phones, to detect drivers’ smoking behaviors
in real driving environment. The basic idea is that the smart phone emits acoustic signals by its speaker and receives reflected signals by its microphone, and then analyses the received signals to detect whether the driver is smoking or not. The system naturally has two advantages: smart phones are widely available and low- cost to use. In addition, leveraging acoustic sensors is a non-contact way that does not require any device to put on.

1.2 Project Objective :

The main objective of this project were to find the results effectively and quickly.

SOFTWARE REQUIREMENTS:

Operating system : Windows 7 Ultimate.
Coding Language : Python.
Front-End : Python.
Back-End : Django-ORM
Designing : Html, css, javascript.
Data Base : MySQL (WAMP Server).

SYSTEM REQUIREMENTS :

➢ H/W System Configuration:-
➢Processor- Pentium –IV
➢ RAM
➢ Hard Disk - 4 GB (min)- 20 GB
➢ Key Board - Standard Windows Keyboard
➢ Mouse - Two or Three Button Mouse
➢ Monitor – SVG

Purpose:

A self-determined mechanism is proposed to analyse smoking related events directly from videos by combining color re-projection techniques, Gaussian mixture models and hierarchical holographic modeling framework. HLSDA is a smoking detection algorithm, which collects various sensor data from a smartwatch and recognizes smoking behavior.

Advantages :

- The system proposes a smoking detection system, named HearSmoking, which only uses acoustic sensors on smartphones in driving environments.
- The system analyses the amplitude and period of the waveform to determine whether there is a breath similar to smoking breath. If both hand movement and breath pattern fit the characteristic of those in a smoking event, we then analyse the periodicity of the detected composite motion to improve system performance.

Disadvantages :

- An existing system that doesn't focus on improving the quality of daily driving by using smartphones emerge in quantity.
- An existing system doesn't focus on multiple body movements when a driver is smoking during driving, e.g., steering with one hand, holding cigarette with another hand, putting up and down the cigarette, inhaling and exhaling smoke with chest expanding and shrinking.

METHODS AND ALGORITHMS:

Decision tree classifiers:

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision making knowledge from the supplied data. Decision tree can be generated from training sets. The procedure for such generation based on the set of objects (S), each belonging to one of the classes C1, C2, …, Ck is as follows:

Step 1. If all the objects in S belong to the same class, for example Ci, the decision tree for S consists of a leaf labeled with this class

Step 2. Otherwise, let T be some test with possible outcomes O1, O2,…, On. Each object in S has one outcome for T so the test partitions S into subsets S1, S2,… Sn where each object in Si has outcome Oi for T. T becomes the root of the decision tree and for each outcome Oi we build a subsidiary decision tree by invoking the same procedure recursively on the set Si.

K-Nearest Neighbors (KNN)

- Simple, but a very powerful classification algorithm

- Classifies based on a similarity measure
- Non-parametric
- Lazy learning
- Does not “learn” until the test example is given

- Whenever we have a new data to classify, we find its K-nearest neighbors from the training data

Logistic regression Classifiers:

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name *logistic regression* is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name *multinomial logistic regression* is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

Naïve Bayes

The naive bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature .

Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques. Various reasons have been advanced in the literature. In this tutorial, we highlight an explanation based on the representation bias. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM (support vector machine). The difference lies on the method of estimating the parameters of the classifier (the learning bias).

SVM

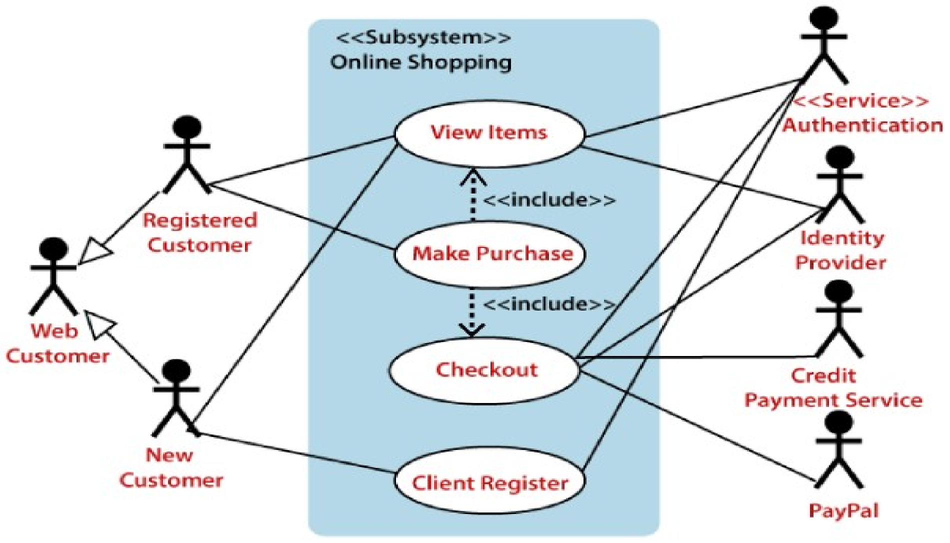
In classification tasks a discriminant machine learning technique aims at finding, based on an *independent and identically distributed* (*iid*) training dataset, a discriminant function that can correctly predict labels for newly acquired instances. Unlike generative machine learning approaches, which require computations of conditional probability distributions, a discriminant classification function takes a data point *x* and assigns it to one of the different classes that are a part of the classification task.

Less powerful than generative approaches, which are mostly used when prediction involves outlier detection, discriminant approaches require fewer computational resources and less training data, especially for a multidimensional feature space and when only posterior probabilities are needed. From a geometric perspective, learning a classifier is equivalent to finding the equation for a multidimensional surface that best separates the different classes in the feature space.

Uml Diagrams:

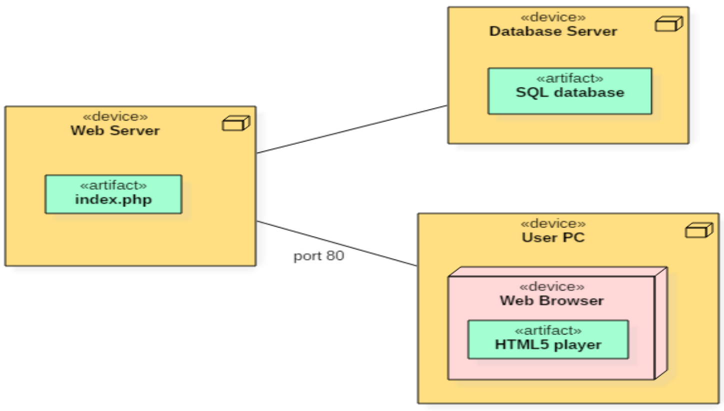
Use Case Diagram :

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system.



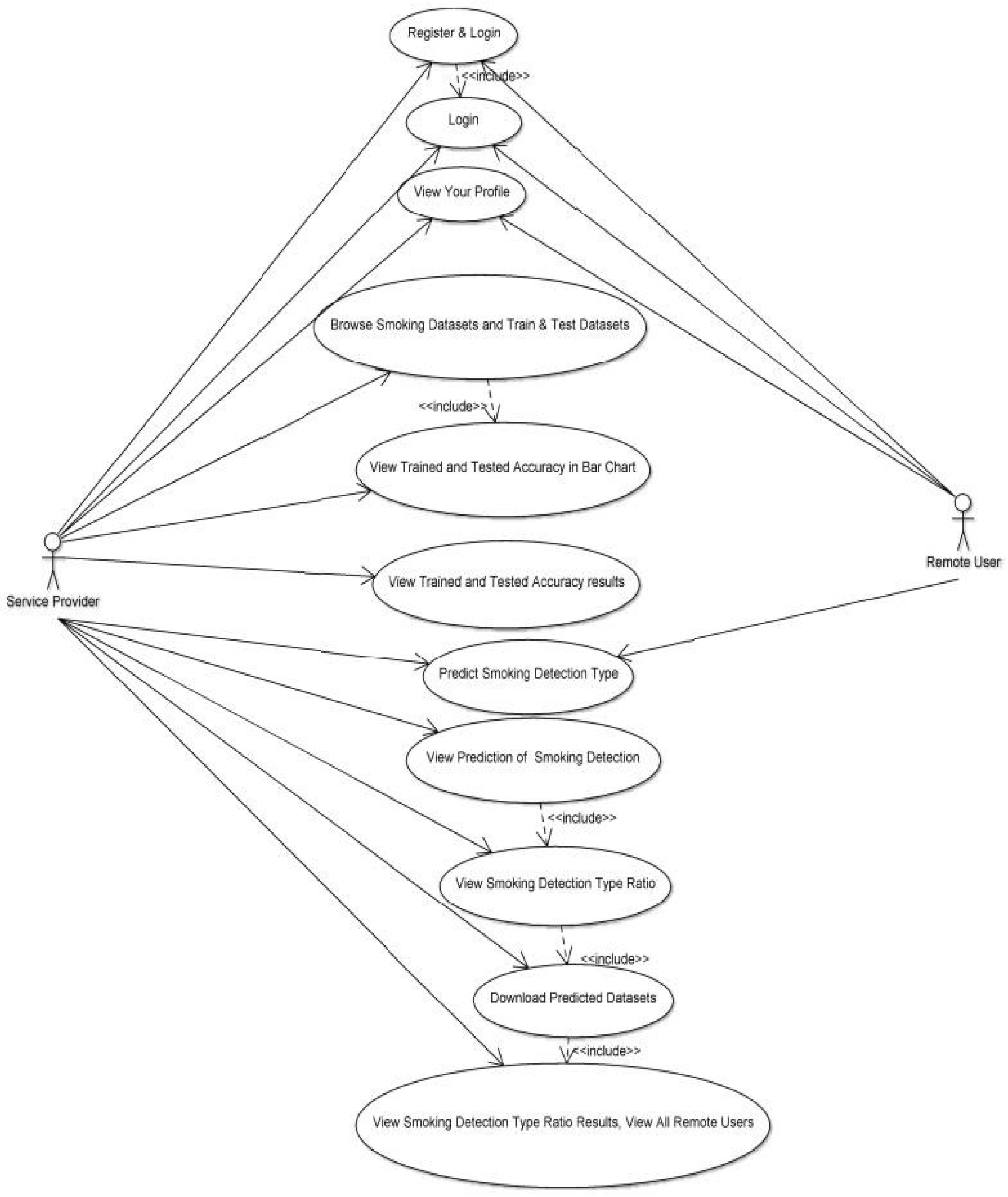
Deployment Diagram :

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. These diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.



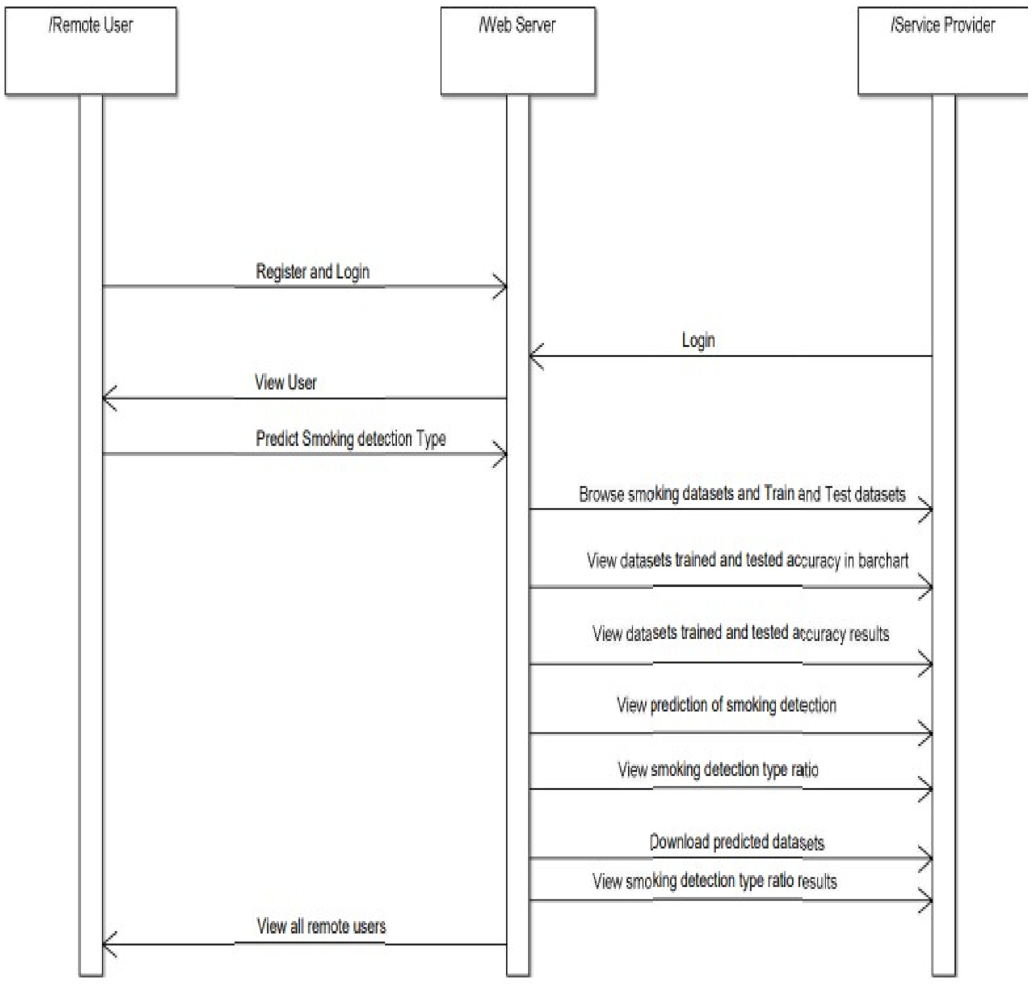
USE CASE DIAGRAM:

A Use case Diagram shows a set of use cases and actors and their relationships.Use case diagrams address the static Use case view of a system. These diagrams are especially important in organizing and modelling the behaviour's of a system.


Register & Login
<<include>>
Login
View Your Profile
Browse Smoking Datasets and Train & Test Datasets
<<include>>
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy results
Service Provider
Remote User
Predict Smoking Detection Type
View Prediction of Smoking Detection
<<include>>
View Smoking Detection Type Ratio
<<include>>
Download Predicted Datasets
<<include>>
View Smoking Detection Type Ratio Results, View All Remote Users

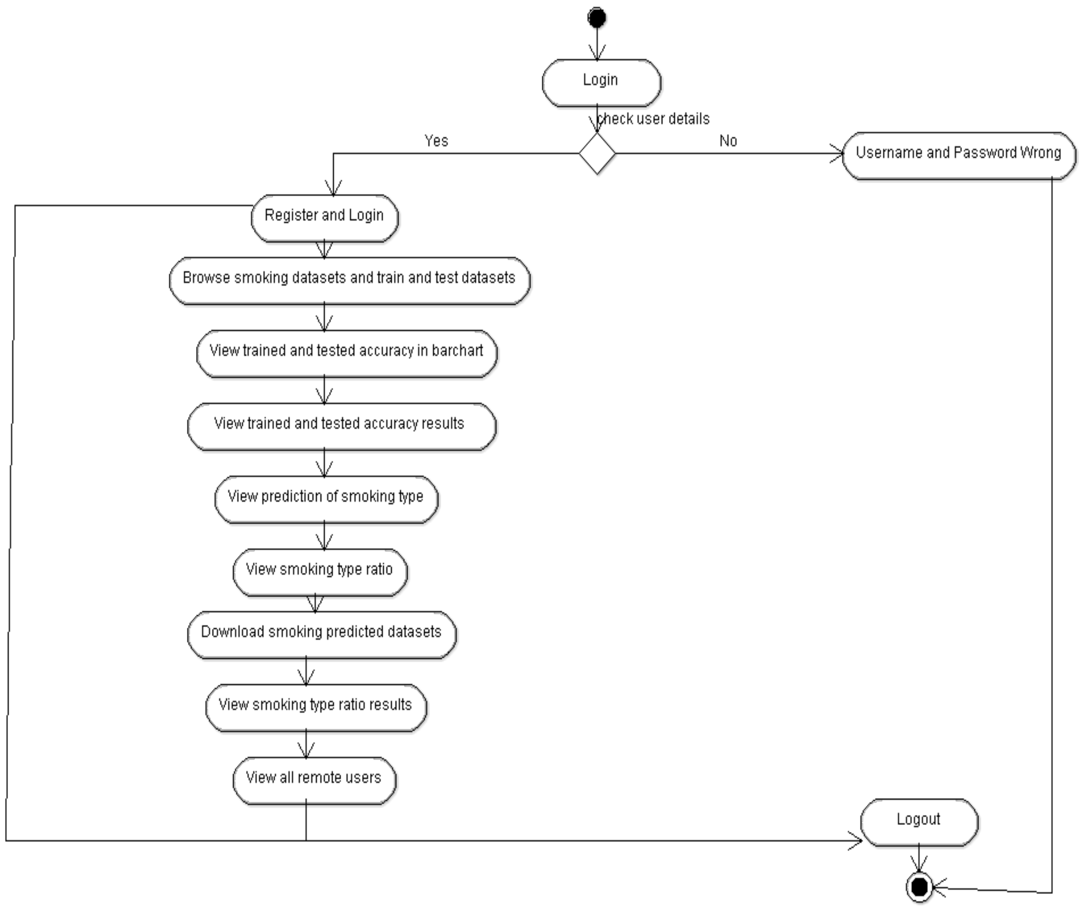
SEQUENCE DIAGRAM :

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.


/Remote User
/Web Server
/Service Provider
Register and Login
Login
View User
Predict Smoking detection Type
Browse smoking datasets and Train and Test datasets
View datasets trained and tested accuracy in barchart
View datasets trained and tested accuracy results
View prediction of smoking detection
View smoking detection type ratio
Download predicted datasets
View smoking detection type ratio results
View all remote users

ACTIVITY DIAGRAM :

Activity diagram is another important diagram in UML to describe th dynamic aspects of the system. It is basically a flowchart to represent the flow from one activity to another. The activity can be described as an operation of a system. The control flow can be drawn from one operation to another.


Login
check user details
Yes
No
Username and Password Wrong
Register and Login
Browse smoking datasets and train and test datasets
View trained and tested accuracy in barchart
View trained and tested accuracy results
View prediction of smoking type
View smoking type ratio
Download smoking predicted datasets
View smoking type ratio results
View all remote users
Logout

SYSTEM CODING AND IMPLEMENTATION :

SOURCE CODE :

```
from django.db.models import  Count, Avg

from django.shortcuts import render, redirect

from django.db.models import Count

from django.db.models import Q

import datetime

import xlwt

from django.http import HttpResponse

import numpy as np

import pandas as pd

from sklearn.feature_extraction.text import CountVectorizer

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

from sklearn.metrics import accuracy_score

from sklearn.tree import DecisionTreeClassifier

# Create your views here.

From Remote User. Models import ClientRegister Model, smoking detection,

Detection ratio,detection_accuracy

def serviceproviderlogin(request):

    if request.method  == "POST":

        admin = request.POST.get('username')

         password = request.POST.get('password')

        if admin == "Admin" and password =="Admin":

            detection_accuracy.objects.all().delete()
```

```
return redirect('View_Remote_Users')

return render(request,'SProvider/serviceproviderlogin.html')

def  View_Smoking_Detection_Ratio(request):

    detection_ratio.objects.all().delete()

    ratio = ""

    kword = 'No Smoking Detection'

    print(kword)

    obj = smoking_detection.objects.all().filter(Q(Prediction=kword))

    obj1 = smoking_detection.objects.all()

    count = obj.count();

    count1 = obj1.count();

    ratio = (count / count1) * 100

    if ratio != 0:

        detection_ratio.objects.create(names=kword, ratio=ratio)

    ratio12 = ""

    kword12 = 'Smoking Detection'

    print(kword12)

    obj12 = smoking_detection.objects.all().filter(Q(Prediction=kword12))

    obj112 = smoking_detection.objects.all()

    count12 = obj12.count();

    count112 = obj112.count();

    ratio12 = (count12 / count112) * 100

ifratio12!=0:

        detection_ratio.objects.create(names=kword12, ratio=ratio12
obj = detection_ratio.objects.all()

    return  render(request,  'SProvider/View_Smoking_Detection_Ratio.html',  {'objs':
obj})

def  View_Remote_Users(request):

    obj=ClientRegister_Model.objects.all()

    return render(request,'SProvider/View_Remote_Users.html',{'objects':obj})

def charts(request,chart_type):

    chart1 = detection_ratio.objects.values('names').annotate(dcount=Avg('ratio'))

 return render(request,"SProvider/charts.html", {'form':chart1, 'chart_type':chart_type})

def charts1(request,chart_type):

    chart1 = detection_accuracy.objects.values('names').annotate(dcount=Avg('ratio'))

return render(request,"SProvider/charts1.html", {'form':chart1, 'chart_type':chart_type})

def  View_Prediction_Of_Smoking_Detection(request):

    obj =smoking_detection.objects.all()

    return  render(request,  'SProvider/View_Prediction_Of_Smoking_Detection.html',
{'list_objects': obj})

def  likeschart(request,like_chart):

     charts   =detection_accuracy.objects.values('names').annotate(dcount=Avg('ratio'))
return
render(request,"SProvider/likeschart.html",{'form':charts,'like_chart':like_chart})

def Download_Predicted_DataSets(request):

 response = HttpResponse(content_type='application/ms-excel')

    # decide file name

    response['Content-Disposition'] = 'attachment; filename="Predicted_Datasets.xls"'

# creating workbook
```

```
 wb = xlwt.Workbook(encoding='utf-8')

  # adding sheet

   ws = wb.add_sheet("sheet1")

   # Sheet header, first row

   row_num = 0

   font_style = xlwt.XFStyle()

   # headers are bold

   font_style.font.bold = True

   # writer = csv.writer(response)

   obj = smoking_detection.objects.all()

   data = obj  # dummy method to fetch data.

   for my_row in data:

       row_num = row_num + 1

      ws.write(row_num, 0, my_row.CHASSIS_NO, font_style)

       ws.write(row_num, 1, my_row.MODEL_YEAR, font_style)

       ws.write(row_num, 2, my_row.USE_OF_VEHICLE, font_style)

       ws.write(row_num, 3, my_row.MODEL, font_style)

       ws.write(row_num, 4, my_row.MAKE, font_style)

       ws.write(row_num, 5, my_row.gender, font_style)

       ws.write(row_num, 6, my_row.age, font_style)

       ws.write(row_num, 7, my_row.height_cm, font_style)

   ws.write(row_num, 8, my_row.weight_kg, font_style)

    ws.write(row_num, 9, my_row.waist_cm, font_style)

ws.write(row_num, 10, my_row.eyesight_left, font_style)
```

```
       ws.write(row_num, 11, my_row.eyesight_right, font_style)
       ws.write(row_num, 12, my_row.hearing_left, font_style)
        ws.write(row_num, 13, my_row.hearing_right, font_style)
        ws.write(row_num, 14, my_row.Prediction, font_style)
        wb.save(response)
    return response
def train_model(request):
    detection_accuracy.objects.all().delete()
   df = pd.read_csv('Smoking_Datasets.csv',encoding='latin-1')
def apply_response(label):
        if (label==0):
            return 0  # No Smoking
        elif (label==1):
            return 1  # Smoking
df['Label'] = df['smoking'].apply(apply_response)
 cv = CountVectorizer()
 X = df['CHASSIS_NO'].apply(str)
  y = df['Label']
  print("CHASSIS_NO")
  print(X)
  print("Results")
  print(y)
   X = cv.fit_transform(X)
```

```
#X = cv.fit_transform(df['CHASSIS_NO'].apply(lambda x: np.str_(X)))

 models = []

from sklearn.model_selection import train_test_split

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)

    X_train.shape, X_test.shape, y_train.shape

print(X_test)

print("Naive Bayes")

from sklearn.naive_bayes import MultinomialNB

    NB = MultinomialNB()

    NB.fit(X_train, y_train)

    predict_nb = NB.predict(X_test)

    naivebayes = accuracy_score(y_test, predict_nb) * 100

    print(naivebayes)

    print(confusion_matrix(y_test, predict_nb))

    print(classification_report(y_test, predict_nb))

    models.append(('naive_bayes', NB))

    detection_accuracy.objects.create(names="Naive Bayes", ratio=naivebayes)

 # SVM Model

    print("SVM")

    from sklearn import svm

    lin_clf = svm.LinearSVC()

    lin_clf.fit(X_train, y_train)

  predict_svm = lin_clf.predict(X_test) svm_

  acc = accuracy_score(y_test, predict_svm) * 100
```

```
print(svm_acc)

   print("CLASSIFICATION REPORT")

    print(classification_report(y_test, predict_svm))

    print("CONFUSION MATRIX")

    print(confusion_matrix(y_test, predict_svm))

    models.append(('svm', lin_clf))

    detection_accuracy.objects.create(names="SVM", ratio=svm_acc)

print("Logistic Regression")

from sklearn.linear_model import LogisticRegression

    reg = LogisticRegression(random_state=0, solver='lbfgs').fit(X_train, y_train)

    y_pred = reg.predict(X_test)

    print("ACCURACY")

    print(accuracy_score(y_test, y_pred) * 100)

    print("CLASSIFICATION REPORT")

    print(classification_report(y_test, y_pred))

    print("CONFUSION MATRIX")

print(confusion_matrix(y_test, y_pred))

models.append(('logistic',reg))detection_accuracy.objects.create(names="Logistic
Regression", ratio=accuracy_score(y_test, y_pred) * 100)

print("Gradient Boosting Classifier")

from sklearn.ensemble import GradientBoostingClassifier

clf=GradientBoostingClassifier(n_estimators=100,learning_rate=1.0,max_depth=1,rand
om_state=0).fit(   X_train,  y_train)

clfpredict = clf.predict(X_test)

print("ACCURACY")
```

```
    print(accuracy_score(y_test, clfpredict) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, clfpredict))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, clfpredict))
    models.append(('GradientBoostingClassifier', clf))
    detection_accuracy.objects.create(names="Gradient Boosting Classifier",
    ratio=accuracy_score(y_test, clfpredict) * 100)
print("KNeighborsClassifier")
from sklearn.neighbors import KNeighborsClassifier
    kn = KNeighborsClassifier()
    kn.fit(X_train, y_train)
    knpredict = kn.predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, knpredict) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, knpredict))
    print("CONFUSION MATRIX")
 print(confusion_matrix(y_test,knpredict))
detection_accuracy.objects.create(names="KNeighborsClassifier",
ratio=accuracy_score(y_test, knpredict) * 100)
 csv_format = 'Results.csv'
f.to_csv(csv_format, index=False)  df.to_markdown
    obj = detection_accuracy.objects.all()
    return render(request,'SProvider/train_model.html', {'objs': obj})
```

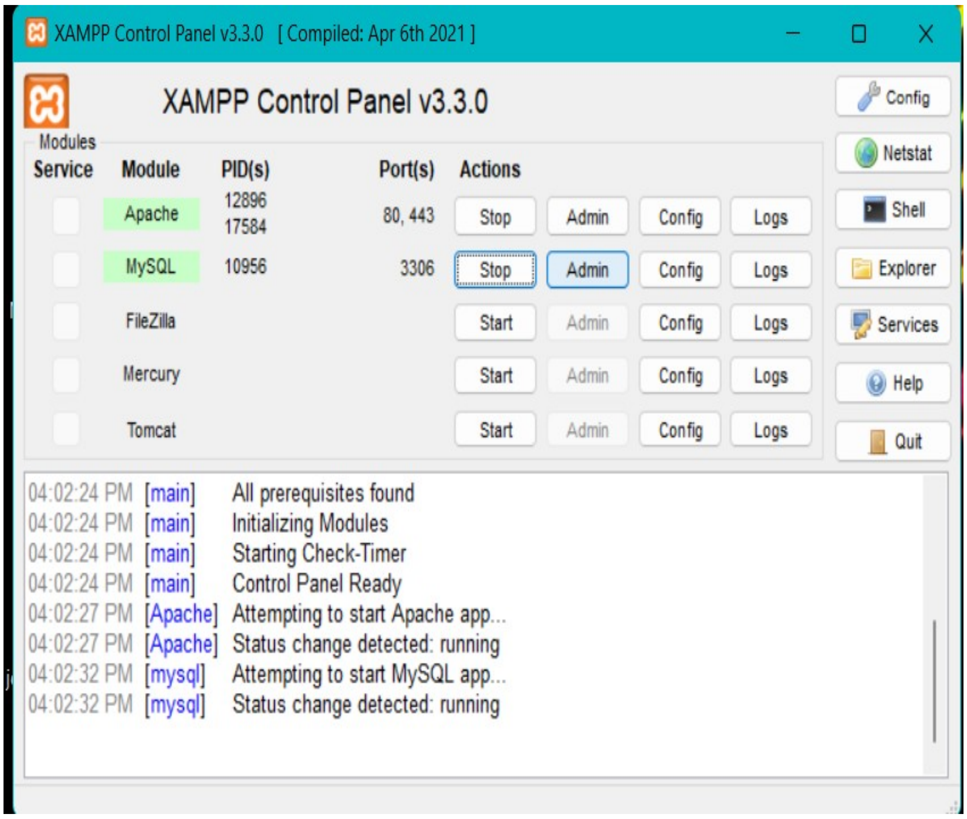
Project Implementation and Output:

Step-1: Open the XAMPP server.

Step-2: Start Apache and MySQL.

Apache- It is responsible for accepting directory (HTTP) requests from Internet users and sending them their desired information in the form of files and Webpages.

MySQL- It uses standalone clients that allow users to interact with MySQL and also to use it with other programs for applications that need relational database capabilities.



XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

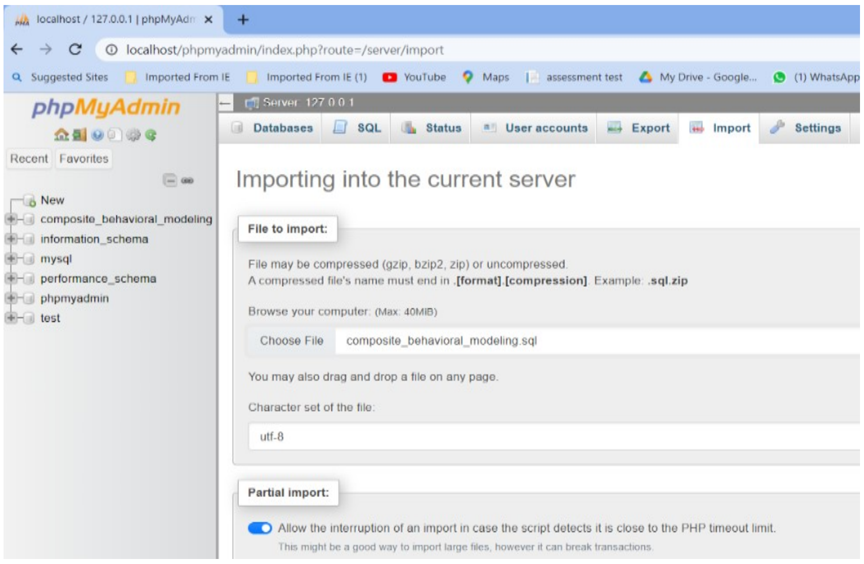
XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions			
	Apache	12896 17584	80, 443	Stop	Admin	Config	Logs
	MySQL	10956	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
04:02:24 PM  [main]     All prerequisites found
04:02:24 PM  [main]     Initializing Modules
04:02:24 PM  [main]     Starting Check-Timer
04:02:24 PM  [main]     Control Panel Ready
04:02:27 PM  [Apache]   Attempting to start Apache app...
04:02:27 PM  [Apache]   Status change detected: running
04:02:32 PM  [mysql]    Attempting to start MySQL app...
04:02:32 PM  [mysql]    Status change detected: running
```

Start Apache, MySQL in XAMPP control panel

Step-3: Next, go to Admin part of MySQL and import the datasets that contain Chassis_No, Use of Vehicle, Make, Model, Username, Gender, Pin Code, Email, Address etc for identity smoking detection.



localhost/phpmyadmin/index.php?route=/server/import

Importing into the current server

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: (Max: 40MiB)

Choose File composite_behavioral_modeling.sql

You may also drag and drop a file on any page.

Character set of the file:

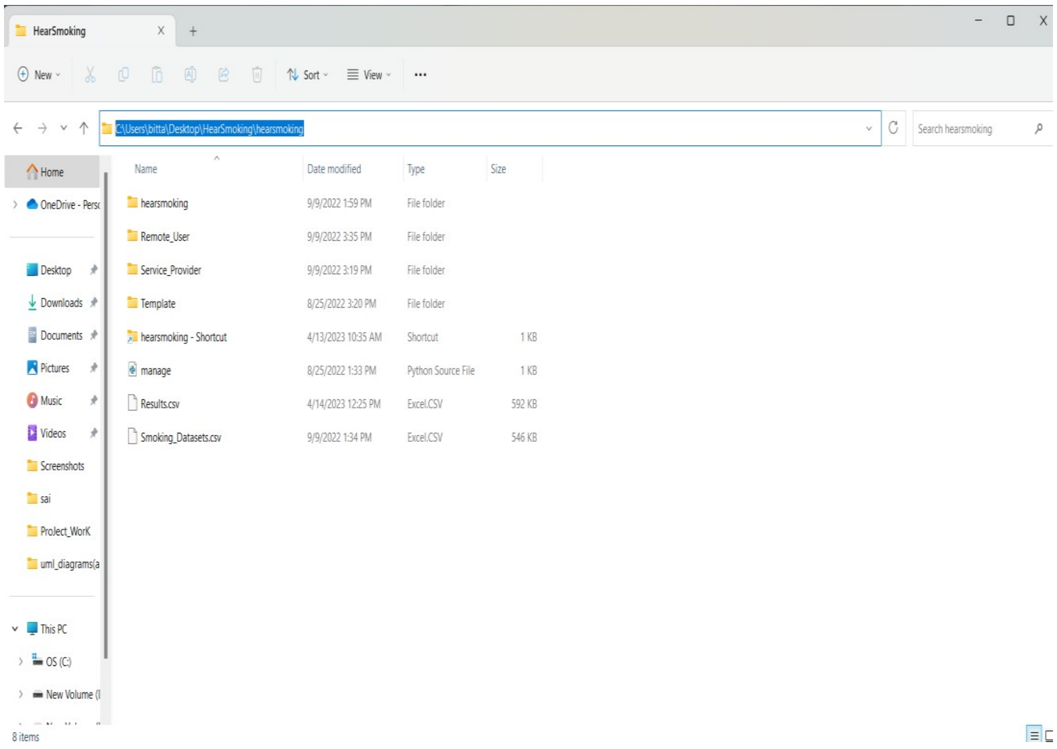
utf-8

Partial import:

Allow the interruption of an import in case the script detects it is close to the PHP timeout limit.
This might be a good way to import large files, however it can break transactions.

Importing Data Sets

Step-4: Go to the project code location, type cmd at file path it opens the command prompt.



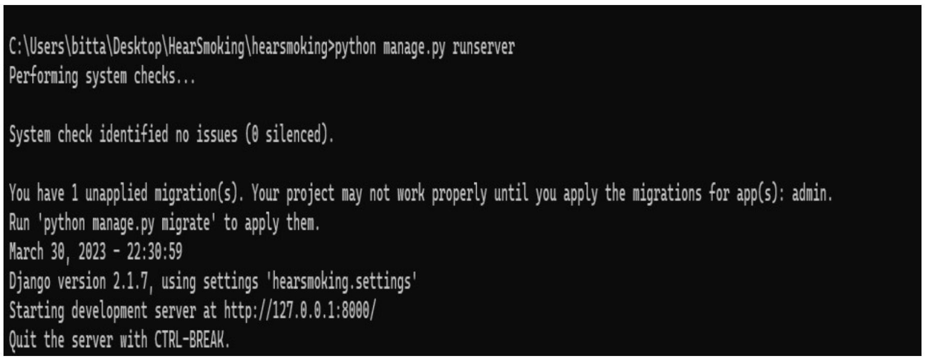
C:\Users\bitta\Desktop\HearSmoking\hearsmoking

Name	Date modified	Type	Size
hearsmoking	9/9/2022 1:59 PM	File folder	
Remote_User	9/9/2022 3:35 PM	File folder	
Service_Provider	9/9/2022 3:19 PM	File folder	
Template	8/25/2022 3:20 PM	File folder	
hearsmoking - Shortcut	4/13/2023 10:35 AM	Shortcut	1 KB
manage	8/25/2022 1:33 PM	Python Source File	1 KB
Results.csv	4/14/2023 12:25 PM	Excel.CSV	592 KB
Smoking_Datasets.csv	9/9/2022 1:34 PM	Excel.CSV	546 KB

8 items

Opening command prompt by using storage location

Step-5: Allow the XAMPP server at background then run the project code in command prompt using the command manage.py run server, it generates the output link as shown below in figure.

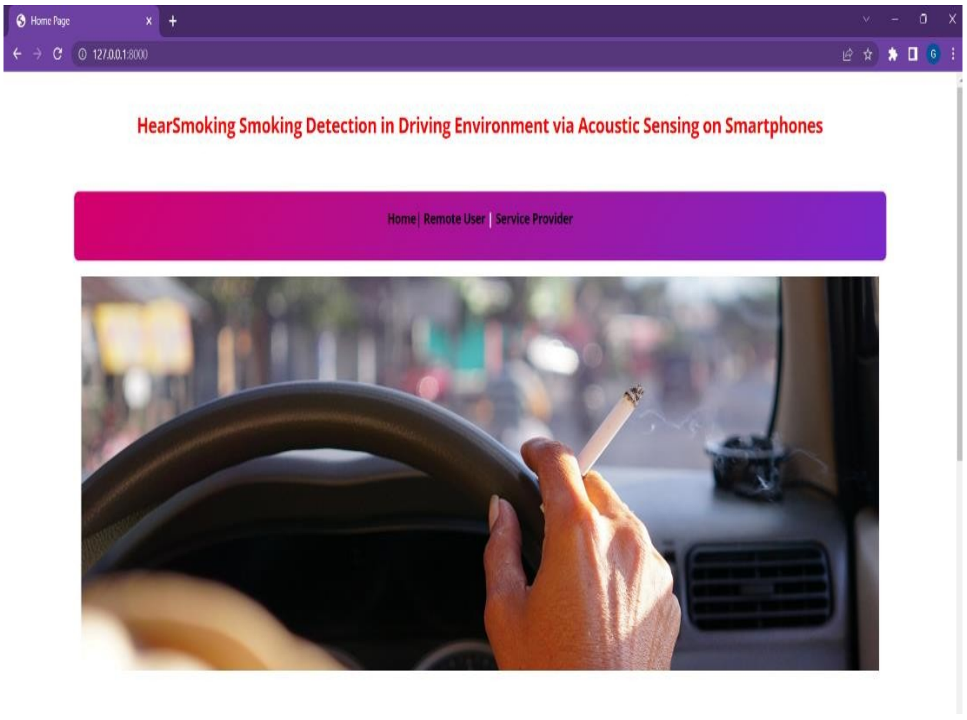


```
C:\Users\bitta\Desktop\HearSmoking\hearsmoking>python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).

You have 1 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin.
Run 'python manage.py migrate' to apply them.
March 30, 2023 - 22:30:59
Django version 2.1.7, using settings 'hearsmoking.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Output in form of HTTP request

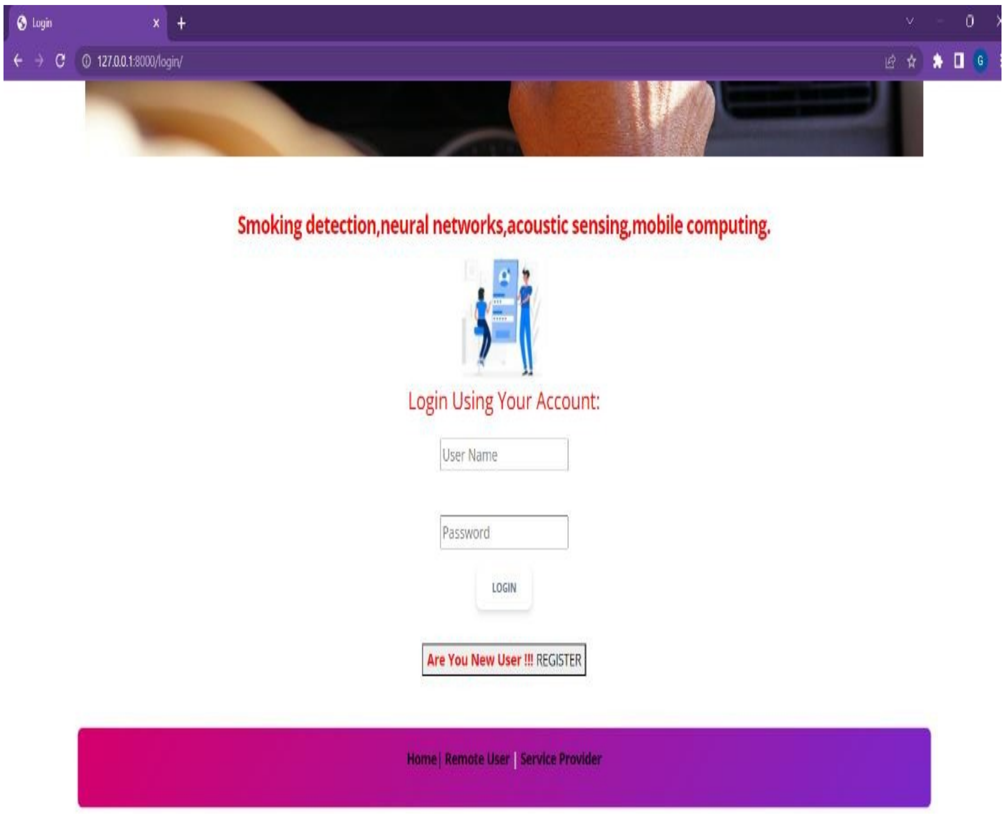
When we follow the output link in any browser then we get the following screenshots representing output.

Description for Home Page: Below screen chart is the home page of HearSmoking for identify Smoking Detection Type which contains the remote user, service provider modules where user and admin can login.


Home Page
127.0.0.1:8000
HearSmoking Smoking Detection in Driving Environment via Acoustic Sensing on Smartphones
Home | Remote User | Service Provider

Home page of HearSmoking

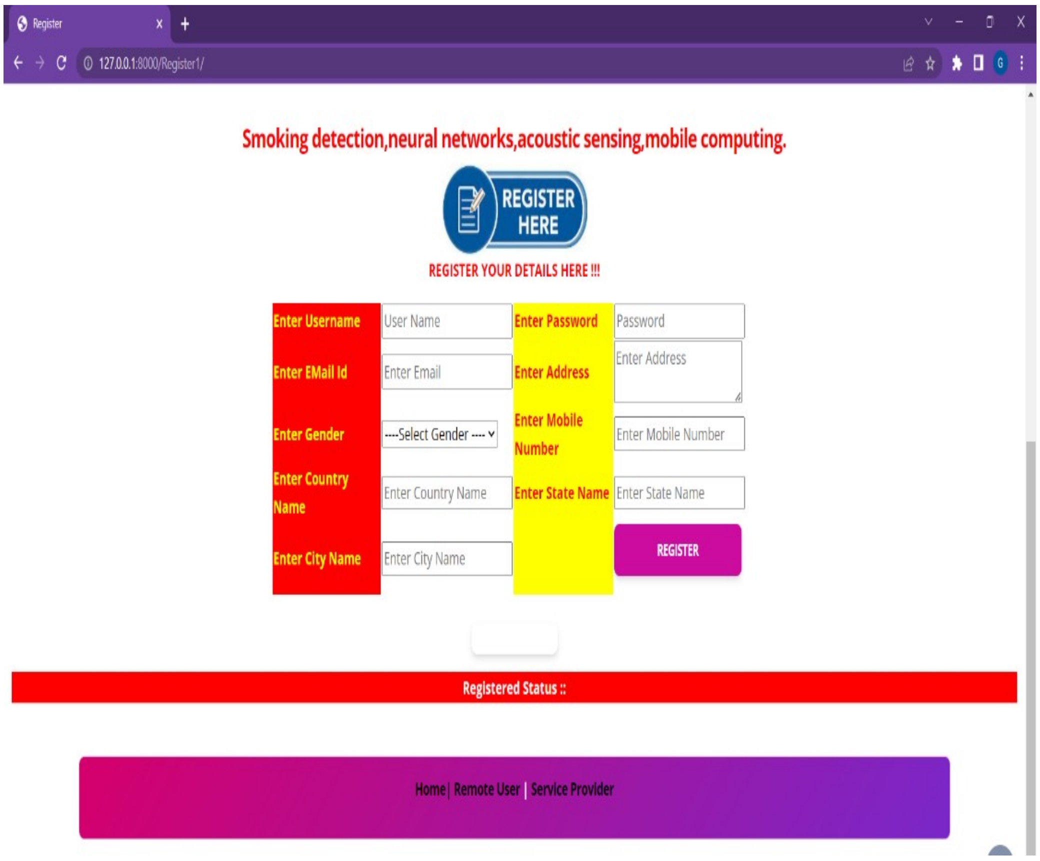
Description for User Login: The screen chart shows the remote user login page which enables the user to provide the data to identify smoking detection. Remote user page requires username and password to login.


Login
127.0.0.1:8000/login/
Smoking detection,neural networks,acoustic sensing,mobile computing.
Login Using Your Account:
User Name
Password
LOGIN
Are You New User !!! REGISTER
Home | Remote User | Service Provider

User Login

Description for User Registration: The screen chart shows the user registration web page. If the user is new then he/she has to register by providing the user credentials such as

- Username
- email-id
- gender
- country name
- city name
- password
- address
- mobile number


Smoking detection,neural networks,acoustic sensing,mobile computing.

REGISTER YOUR DETAILS HERE !!!

Enter Username	User Name	Enter Password	Password
Enter EMail Id	Enter Email	Enter Address	Enter Address
Enter Gender	----Select Gender ----	Enter Mobile Number	Enter Mobile Number
Enter Country Name	Enter Country Name	Enter State Name	Enter State Name
Enter City Name	Enter City Name		REGISTER

Registered Status ::

Home| Remote User | Service Provider

User Registration

Description for Predict Smoking Detection Type: After successful registration user can login with help of remote user. Then the user will be allowed to provide smoking data which is used to identify a genuine account. For each user there will be a behavioral record which contains the credentials of user account.

The datasets which are used for identity smoking detection such as

CHASSIS_NO

Model_year

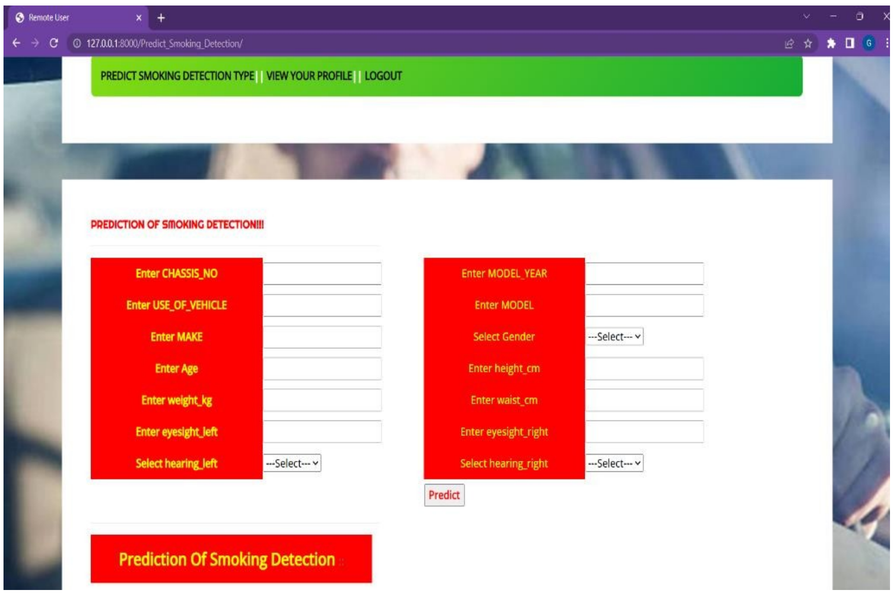
Make

Use of Vehicle

Model

Gender

Age



Remote User

127.0.0.1:8000/Predict_Smoking_Detection/

PREDICT SMOKING DETECTION TYPE || VIEW YOUR PROFILE || LOGOUT

PREDICTION OF SMOKING DETECTION!!!

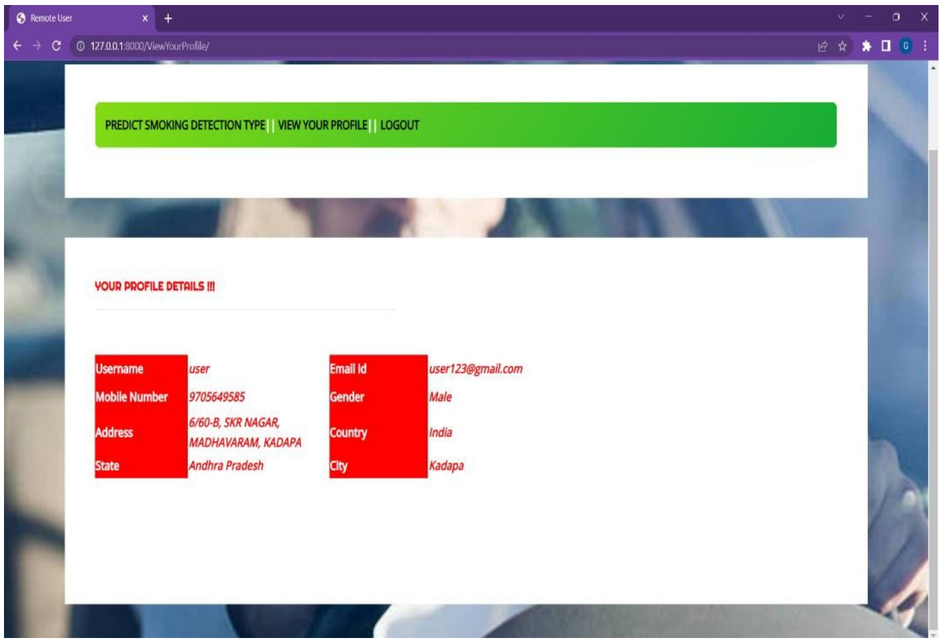
Enter CHASSIS_NO		Enter MODEL_YEAR	
Enter USE_OF_VEHICLE		Enter MODEL	
Enter MAKE		Select Gender	---Select---
Enter Age		Enter height_cm	
Enter weight_kg		Enter waist_cm	
Enter eyesight_left		Enter eyesight_right	
Select hearing_left	---Select---	Select hearing_right	---Select---

Predict

Prediction Of Smoking Detection :

Prediction of Smoking Detection Type

Description for Viewing profile: The below screen chart shows the user data which was given at registration process in remote user web page.



Remote User

127.0.0.1:8000/ViewYourProfile/

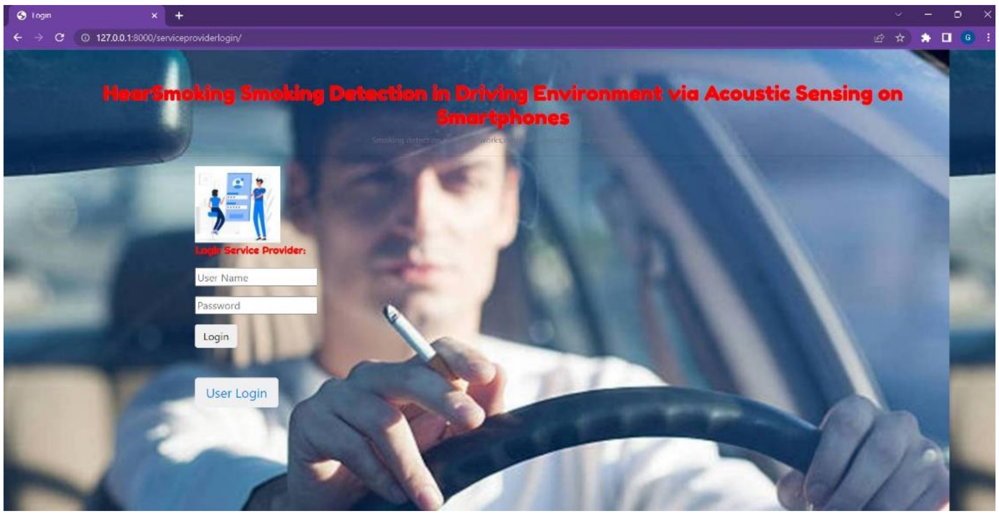
PREDICT SMOKING DETECTION TYPE || VIEW YOUR PROFILE || LOGOUT

YOUR PROFILE DETAILS !!!

Username	user	Email Id	user123@gmail.com
Mobile Number	9705649585	Gender	Male
Address	6/60-B, SKR NAGAR, MADHAVARAM, KADAPA	Country	India
State	Andhra Pradesh	City	Kadapa

View User Profile

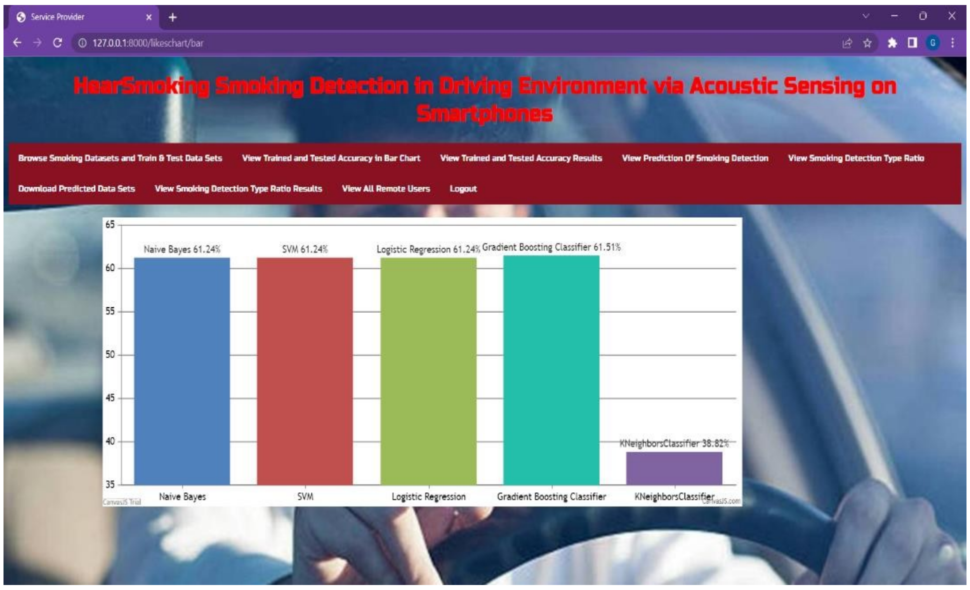
Description for Admin login: Admin can login to Service provider to view the results predicted by remote user. Service provider page shows the results including Register and Login operations of user and identity smoking type.


Login
127.0.0.1:8000/serviceproviderlogin/
HearSmoking Smoking Detection in Driving Environment via Acoustic Sensing on Smartphones
Login Service Provider:
User Name
Password
Login
User Login

Admin Login

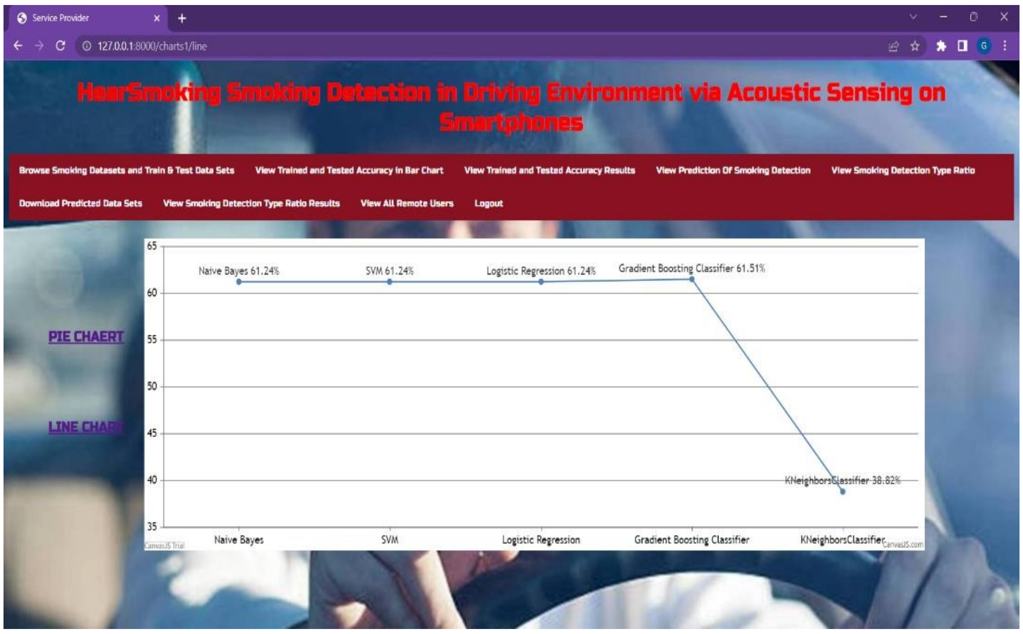
Description for Viewing trained and tested accuracy results:

Bar Chart: This below figure shows the bar chart representing the accuracy results of attack type by using machine learning algorithms. The bar chart can be obtained after browsing the tested and trained datasets in Service provider web page.


Service Provider
127.0.0.1:8000/likeschart/bar
HearSmoking Smoking Detection in Driving Environment via Acoustic Sensing on Smartphones
Browse Smoking Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Smoking Detection
View Smoking Detection Type Ratio
Download Predicted Data Sets
View Smoking Detection Type Ratio Results
View All Remote Users
Logout
65
60
55
50
45
40
35
Naive Bayes 61.24%
SVM 61.24%
Logistic Regression 61.24%
Gradient Boosting Classifier 61.51%
KNeighborsClassifier 38.82%
Naive Bayes
SVM
Logistic Regression
Gradient Boosting Classifier
KNeighborsClassifier
CanvasJS Trial
CanvasJS.com

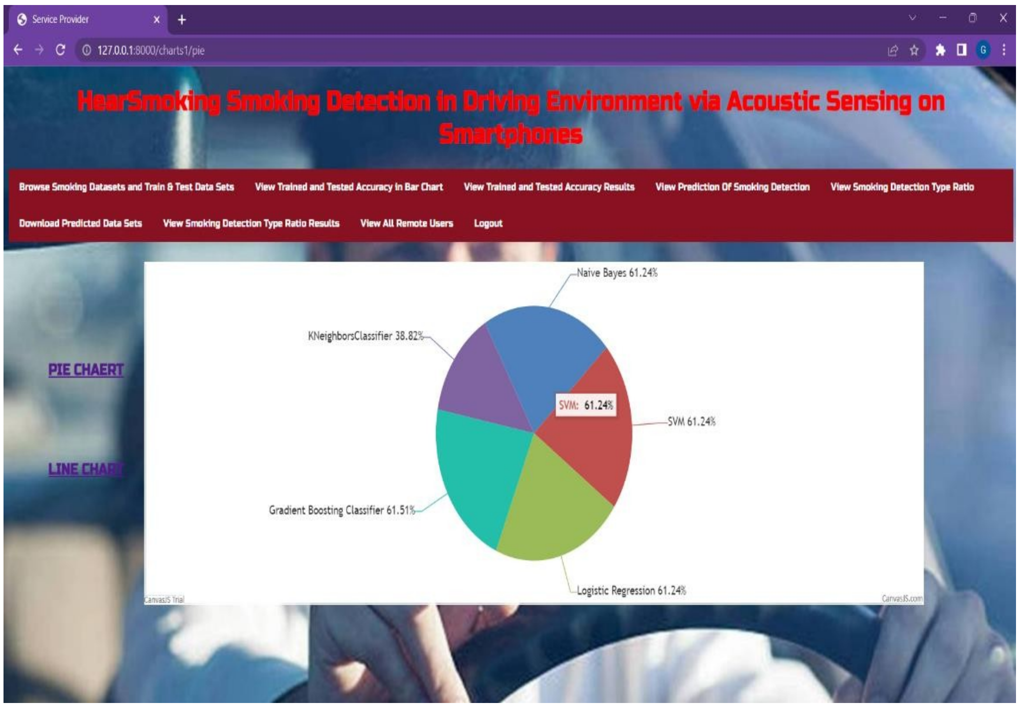
Trained and Tested accuracy results in form of Bar Chart

Line Chart: This below figure shows line chart which representing the accuracy results of smoking type.


Service Provider
127.0.0.1:8000/charts1/line
HearSmoking Smoking Detection in Driving Environment via Acoustic Sensing on Smartphones
Browse Smoking Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Smoking Detection
View Smoking Detection Type Ratio
Download Predicted Data Sets
View Smoking Detection Type Ratio Results
View All Remote Users
Logout
PIE CHAERT
LINE CHART
65
60
55
50
45
40
35
Naive Bayes 61.24%
SVM 61.24%
Logistic Regression 61.24%
Gradient Boosting Classifier 61.51%
KNeighborsClassifier 38.82%
Naive Bayes
SVM
Logistic Regression
Gradient Boosting Classifier
KNeighborsClassifier
CanvasJS Trial
CanvasJS.com

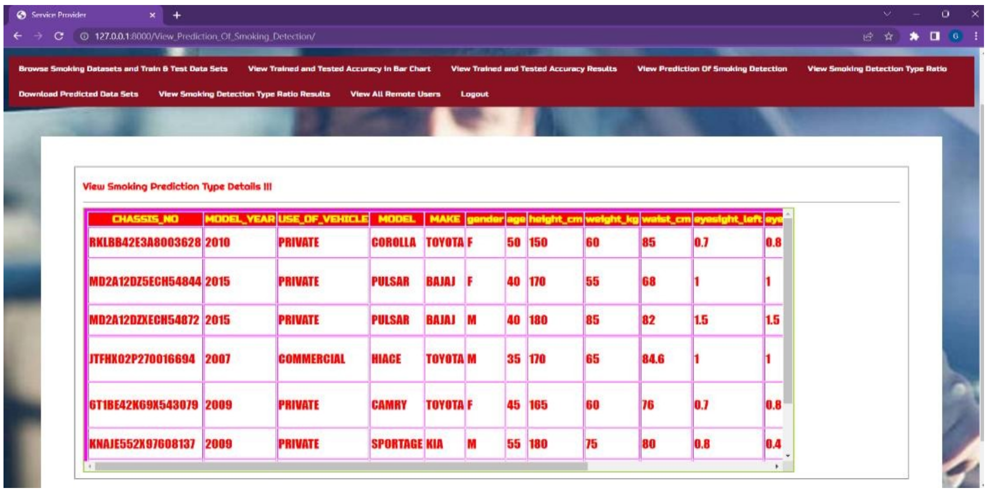
Trained and Tested accuracy results in form of Line Chart

Pie Chart: This below figure shows line chart which representing the accuracy results of smoking type.


Service Provider
127.0.0.1:8000/charts1/pie
HearSmoking Smoking Detection in Driving Environment via Acoustic Sensing on Smartphones
Browse Smoking Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Smoking Detection
View Smoking Detection Type Ratio
Download Predicted Data Sets
View Smoking Detection Type Ratio Results
View All Remote Users
Logout
PIE CHAERT
LINE CHART
Naive Bayes 61.24%
KNeighborsClassifier 38.82%
SVM: 61.24%
SVM 61.24%
Gradient Boosting Classifier 61.51%
Logistic Regression 61.24%
CanvasJS Trial
CanvasJS.com

Trained and Tested accuracy results in form of Pie Chart

Description for viewing smoking detection type status: The below screenshot shows the smoking detection type status. It shows the status of chassis no, model year, use of vehicle, model, make, gender and so on.

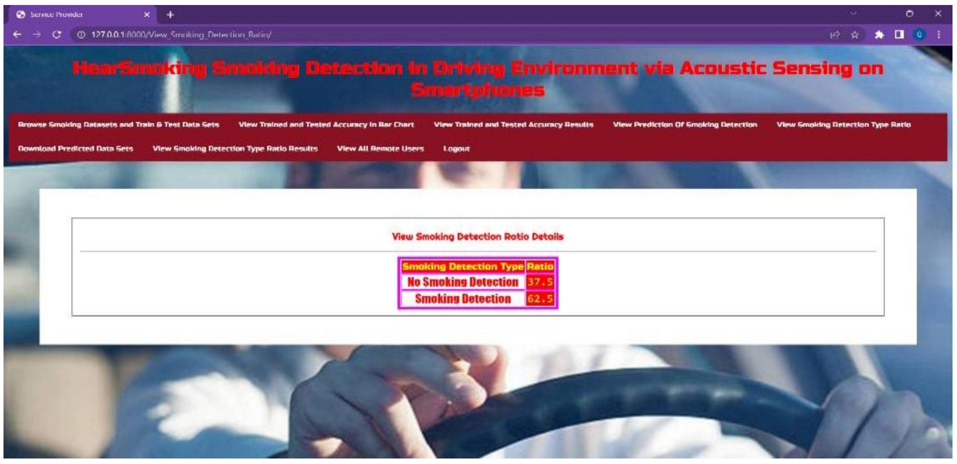

Service Provider
127.0.0.1:8000/View_Prediction_Of_Smoking_Detection/
Browse Smoking Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Smoking Detection
View Smoking Detection Type Ratio
Download Predicted Data Sets
View Smoking Detection Type Ratio Results
View All Remote Users
Logout

View Smoking Prediction Type Details !!!

CHASSIS_NO	MODEL_YEAR	USE_OF_VEHICLE	MODEL	MAKE	gender	age	height_cm	weight_kg	waist_cm	eyesight_left	eye
RKLBB42E3A8003628	2010	PRIVATE	COROLLA	TOYOTA	F	50	150	60	85	0.7	0.8
MD2A12DZ5ECH54844	2015	PRIVATE	PULSAR	BAJAJ	F	40	170	55	68	1	1
MD2A12DZXECH54872	2015	PRIVATE	PULSAR	BAJAJ	M	40	180	85	82	1.5	1.5
JTFHX02P270016694	2007	COMMERCIAL	HIACE	TOYOTA	M	35	170	65	84.6	1	1
6T1BE42K69X543079	2009	PRIVATE	CAMRY	TOYOTA	F	45	165	60	76	0.7	0.8
KNAJE552X97608137	2009	PRIVATE	SPORTAGE	KIA	M	55	180	75	80	0.8	0.4

View Smoking Prediction Type Status

Description for smoking type status ratio: The screen chart shows the identity smoking type status ratio. The ratio between the No smoking or smoking based on accuracy results.


Service Provider
127.0.0.1:8000/View_Smoking_Detection_Ratio/
HearSmoking Smoking Detection in Driving Environment via Acoustic Sensing on Smartphones
Browse Smoking Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Smoking Detection
View Smoking Detection Type Ratio
Download Predicted Data Sets
View Smoking Detection Type Ratio Results
View All Remote Users
Logout

View Smoking Detection Ratio Details

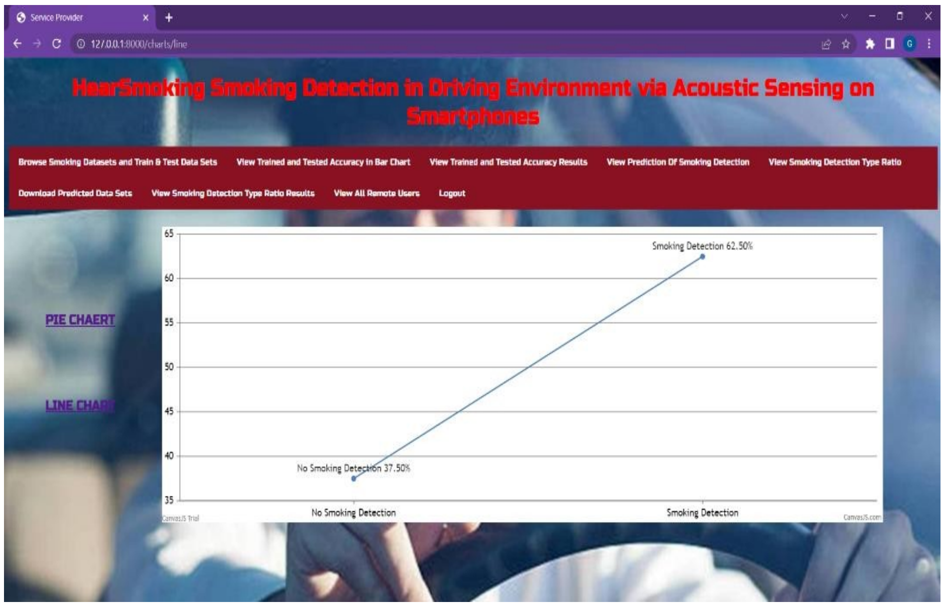
Smoking Detection Type	Ratio
No Smoking Detection	37.5
Smoking Detection	62.5

Smoking Detection Type ratio results

Description for smoking type ratio results:

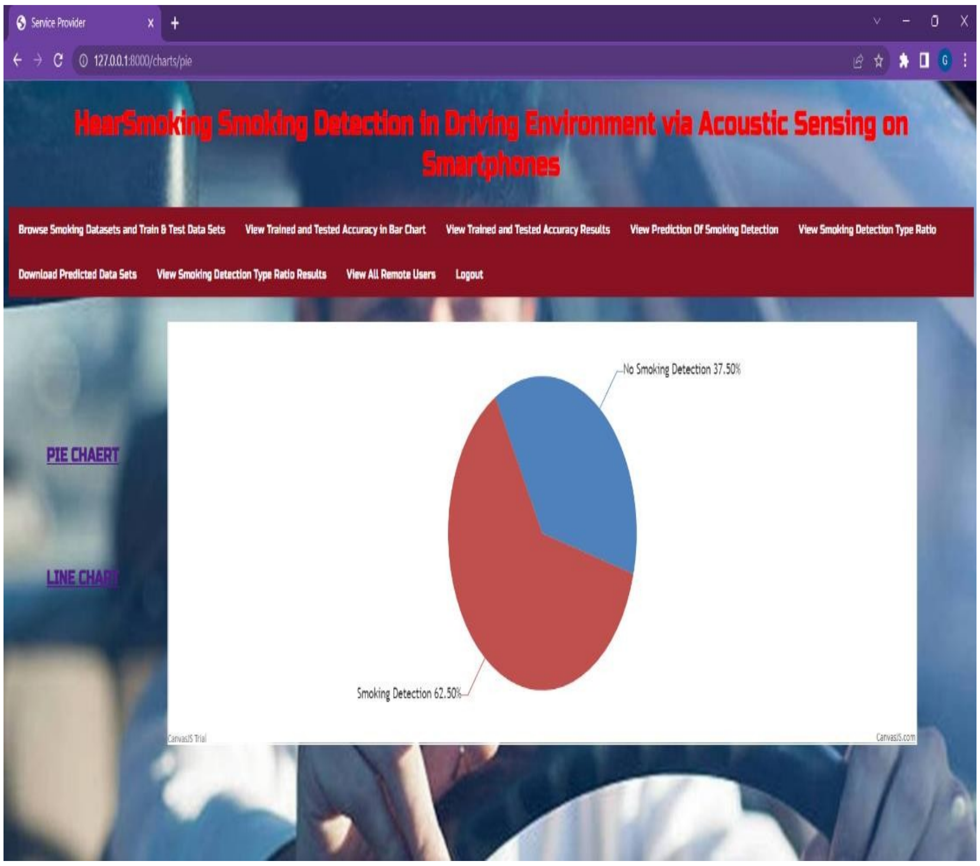
Line Chart:

The Below screen chart representing line chart which shows the smoking type accuracy ratio (i.e., smoking/ no smoking).


Service Provider
127.0.0.1:8000/charts/line
HearSmoking Smoking Detection in Driving Environment via Acoustic Sensing on Smartphones
Browse Smoking Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Smoking Detection
View Smoking Detection Type Ratio
Download Predicted Data Sets
View Smoking Detection Type Ratio Results
View All Remote Users
Logout
PIE CHAERT
LINE CHART
Smoking Detection 62.50%
No Smoking Detection 37.50%
No Smoking Detection
Smoking Detection

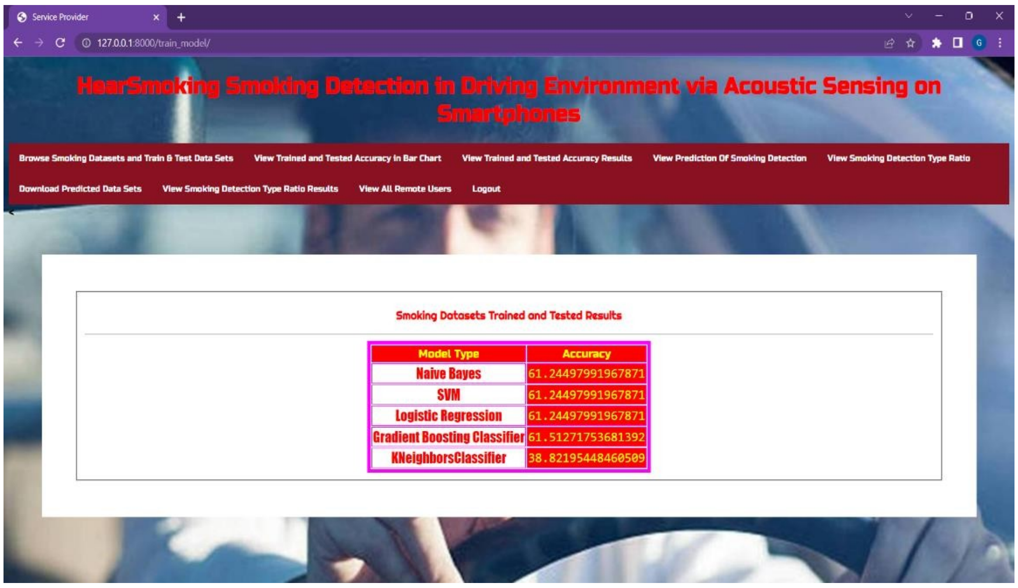
Smoking type ratio results in form of line chart

Pie Chart: Same as Line Chart below screen chart representing the pie chart shows the smoking type ratio .


Service Provider
127.0.0.1:8000/charts/pie
HearSmoking Smoking Detection in Driving Environment via Acoustic Sensing on Smartphones
Browse Smoking Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Smoking Detection
View Smoking Detection Type Ratio
Download Predicted Data Sets
View Smoking Detection Type Ratio Results
View All Remote Users
Logout
PIE CHAERT
LINE CHART
No Smoking Detection 37.50%
Smoking Detection 62.50%

Smoking type ratio results in form of Pie Chart

Description for Smoking datasets trained and tested results: The below screenshot shows the smoking datasets and trained and tested accuracy results.

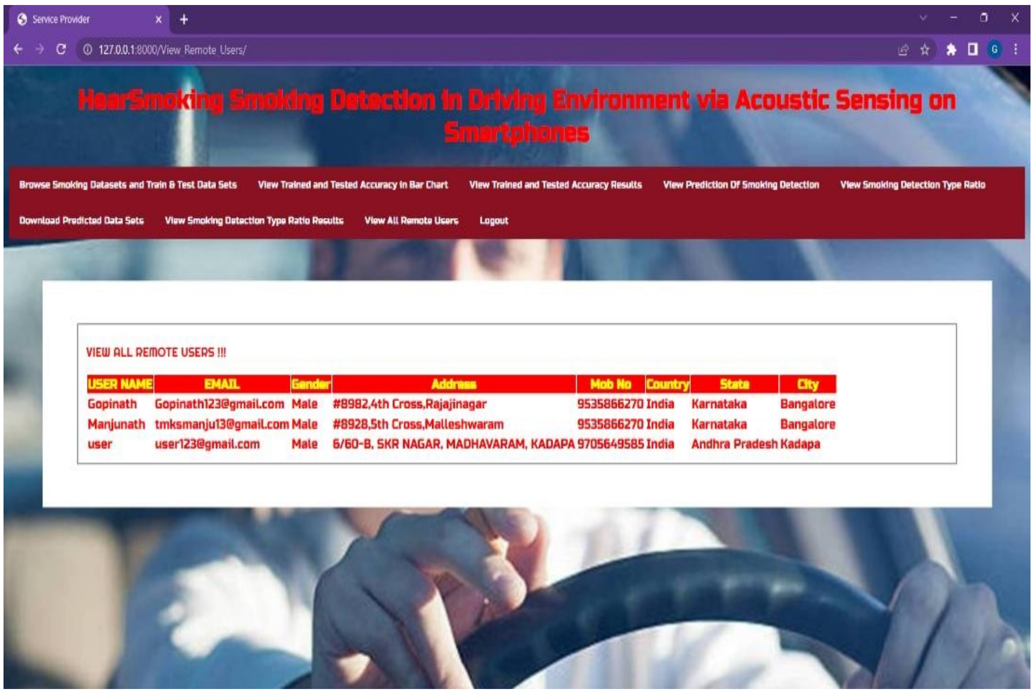

Service Provider
127.0.0.1:8000/train_model/
HearSmoking Smoking Detection in Driving Environment via Acoustic Sensing on Smartphones
Browse Smoking Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Smoking Detection
View Smoking Detection Type Ratio
Download Predicted Data Sets
View Smoking Detection Type Ratio Results
View All Remote Users
Logout

Smoking Datasets Trained and Tested Results

Model Type	Accuracy
Naive Bayes	61.24497991967871
SVM	61.24497991967871
Logistic Regression	61.24497991967871
Gradient Boosting Classifier	61.51271753681392
KNeighborsClassifier	38.82195448460509

Smoking datasets trained and tested results

Description for view all remote users: Below screen chart show the all users who are registered to identify a smoking type performed by a genuine account. Admin can view all users data such as username, email, gender, address, mobile number.


Service Provider
127.0.0.1:8000/View_Remote_Users/
HearSmoking Smoking Detection in Driving Environment via Acoustic Sensing on Smartphones
Browse Smoking Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Smoking Detection
View Smoking Detection Type Ratio
Download Predicted Data Sets
View Smoking Detection Type Ratio Results
View All Remote Users
Logout

VIEW ALL REMOTE USERS !!!

USER NAME	EMAIL	Gender	Address	Mob No	Country	State	City
Gopinath	Gopinath123@gmail.com	Male	#8982,4th Cross,Rajajinagar	9535866270	India	Karnataka	Bangalore
Manjunath	tmksmanju13@gmail.com	Male	#8928,5th Cross,Malleshwaram	9535866270	India	Karnataka	Bangalore
user	user123@gmail.com	Male	6/60-B, SKR NAGAR, MADHAVARAM, KADAPA	9705649585	India	Andhra Pradesh	Kadapa

View All Remote Users

CONCLUSION:

In this paper, we address how to detect cigarette smoking activity during driving to improve road safety. Through literature survey and experimental verification, we find some characteristic smoking patterns in driving environment. We propose a smoking detection system, named Hear Smoking, which leverages acoustic sensors on smart phones to detect cigarette smoking events of drivers when they are driving. Hear Smoking takes advantages of RCC and CNN to detect both hand movements and respirations of the driver. Methods of composite analysis and periodicity analysis are designed to improve system performance. We conduct extensive experiments in different driving environments. Hear Smoking can detect smoking events with an average accuracy of 93:44% in real-time, which indicates that it works efficiently and reliably.

REFERENCES:

1) D. L. Group, “Smoking while driving causes accidents in clearwater,” https://www.dolmanlaw.com/smoking-driving-causes-distracted-driving-accidents-clearwater-fl, 2019.

2) S. Gupta and V. Kumar, “A study on effects of smoking on society: a case study,” MOJ Public Health, vol. 7, no. 4, pp. 192–194, 2018.

3) “Smoke free law and vehicles,” http://www.smokefreeengland.co.uk/faq/vehicles/, 2007.

4) Lyft, “Safety policies,” https://help.lyft.com/hc/en-us/articles/115012923127-Safety-policies#nosmoking, 2018.

5) W.Wu and C. Chen, “Detection system of smoking behavior based on face analysis,” in ICGEC, 2010, pp. 184–187.