A

Major Project

On

LiSA-G: AUTHENTICATE AND IDENTIFY USERS ON THE WIDELY AVAILABLE COMMERCIAL SMARTWATCHES

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

NEELAM PRIYANSHA	(197R1A0539)
PASUMARTHI PRASHANTHI	(197R1A0542)
ZEBA UNNISSA	(197R1A0560)

Under the Guidance of

Dr. BAGAM LAXMAIAH

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2019-2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"LiSA-G: AUTHENTICATE AND IDENTIFY USERS ON THE WIDELY AVAILABLE COMMERCIAL SMARTWATCHES"** being submitted by **NEELAM PRIYANSHA(197R1A0539), PASUMARTHI PRASHANTHI(197R1A0542), and ZEBA UNNISSA(197R1A0560)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr. Bagam Laxmaiah
(Associate Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on _____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express our profound gratitude and deep regard to our guide **Dr.Bagam Laxmaiah,** Associate Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. Punyaban Patel, Ms. Shilpa, Dr.T. Subha Mastan Rao** and **J. Narasimharao** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering, **Dr. Ashuthosh Saxena**, Dean R&D, and **Dr. D T V Dharmajee Rao**, Dean Academics for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

NEELAM PRIYANSHA (197R1A0539)

PASUMARTHI PRASHANTHI (197R1A0542)

ZEBA UNNISSA (197R1A0560)

ABSTRACT

On top of the predominant smartphones, there is a rapid increase in the number of wearable IoT devices such as smart glasses, smart watches and so on. The latest wearable devices are equipped with various communication modules such as WiFi and a variety of sensors. However, such multiple connectives on wearable devices can expose a variety of personal information and further increase the risk of security attacks due to lack of security measures which necessitates robust security measures. Wearable IoT devices available today monitor human activities, many of which are unconscious or subconscious. Interestingly, some of these activities exhibit distinct patterns for each individual. Among those activities, walking is one of the most basic activities. Considering each individual's unique walking pattern, gait, which is the pattern of limb movements during locomotion, can be utilized as a biometric feature for user authentication. In this project, we propose a lightweight seamless authentication framework based on gait (LiSA-G) that can authenticate and identify users on the widely available commercial smartwatches. Unlike the existing works, our proposed framework extracts not only the statistical features but also the human-action-related features from the collected sensor data in order to more accurately and efficiently reveal distinct patterns. Our experimental results show that our framework achieves a higher authentication accuracy in comparison with the existing works while requiring fewer features and less amount of sensor data. This makes our framework more practical and rapidly deployable in wearable IoT systems with limited computing power and energy capacity.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT INTRODUCTION

The emergence of the Internet of Things (IoT) has revolutionized numerous systems and the way we interact with computing and communication systems. On top of the predominant smartphones, there is a rapid increase in the number of wearable IoT devices such as smartwatches, smart glasses, and so on. Although initial wearable devices were equipped with limited connectivity such as Bluetooth, the latest wearable devices are equipped with various communication modules such as WiFi and a variety of sensors. However, such multiple connectives on wearable devices can expose a variety of personal information and further increase the risk of security breaches which necessitates robust security measures. However, the attention to security aspects of wearable IoT devices has not kept pace with that of quantitative growth. Compared to the precedent IoT devices, such as smartphones, wearable IoT devices are more prone to various security attacks due to the lack of security measures(e.g., insufficient user authentication) and limited resources (e.g., computing power and energy capacity). For example, in 2013, hackers were able to remotely infiltrate Google Glass systems to watch and hear everything wearers did. In 2015, a study conducted by HP demonstrated that all smartwatches can be vulnerable to security attacks.

1.2 PROJECT SCOPE

This project is titled "LiSA-G: Authenticate and Identify Users on Widely Available Commercial Smartwatches." Research works have been proposed to authenticate wearable or hand-held device users, using subconscious activities. However, most of the prior research conducted their experiments on custom made devices. Even though there are some research projects conducted on commercial devices, they involve a long period of the authentication process mostly incurred by their walking detection algorithm.

Consequently, they require a relatively large volume of data, which can be burdensome to wearable IoT devices with limited computing resources and energy capacity.

1.3 PROJECT PURPOSE

Thus, to address the vulnerability of the current wearable IoT security system, we consider user authentication which is one of the most principal security measures. We utilize subconscious activities to authenticate wearable IoT device users. In interactions with users, wearable IoT devices generate various sensor data such as accelerometer or gyroscope data, and the pattern in such sensor data can be distinct as each user performs subconscious activities in their own way, which is not required to remember as well as difficult to hack or copy. Among various subconscious activities, We consider gait, which is the pattern of limb movements during locomotion as it satisfies the goal of our authentication framework to provide reliable and user-friendly authentication. Specifically, gait is shown to provide unique patterns even between people having similar physical attributes, and walking is one of the most rudimentary and mundane activities.

1.4 PROJECT FEATURES

The main features of this project are that this authentication framework based on gait (LiSA-G) securely authenticates users on commercial smartwatches in a light manner without typing a password or providing biometric measures. Unlike the existing works that extract only the statistical features from sensor data, LiSA-G additionally considers mechanical traits that are bound by the physical attribute of individuals while using less number of features. This framework provides reliable and user-friendly authentication. Specifically, gait is shown to provide unique patterns even between people having similar physical attributes, and walking is one of the most rudimentary and mundane activities that requires significant effort to copy or mimic.

2. LITERATURE SURVEY

2. LITERATURE SURVEY

The emergence of smartwatches poses new challenges to information security. Recent studies further raised the security concern for smartwatches. Florencio et al. conducted a large-scale password reuse study in 2006 by instrumenting Microsoft Windows Live Toolbar. The study included half a million users monitored over a three-month period. They found that each user had about 25 accounts and 6.5 passwords, each shared across 3.9 sites. This study provides an updated estimate of 80 online accounts per user and potentially 25 accounts per password, but it is conducted over a much smaller and less diverse user sample. Wash et al. examined the types of passwords that are more frequently reused. They developed a Web browser plugin to collect user passwords, and conducted a user study with 134 participants. They found that strong and more frequently used passwords were reused more often. Ur et al. investigated the relationship between users' perceptions of the strength of specific passwords and their actual strength, and found that users had serious misconceptions. In theory, good password hygiene and risk management are straightforward: strong, unique passwords for all accounts, but especially for more important ones. However, the proliferation of accounts, weak password policies, and difficulty remembering all of these passwords make good password behaviors hard to implement in practice. Throughout the research, it is observed that users' security perceptions and intent rarely match their security realities. Some reasons for this lie in misconceptions about risk and a desire for convenience, identified by other researchers. Also, they showed that users do not really understand how password attacks work. So, the Gait authentication method is introduced. It is a kind of biometric authentication, based on wearable sensors. Gait movement is both unique and unavoidable in everyday life. The pattern of acceleration values created by walking differs from one person to another.

3. SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 INTRODUCTION

Analysis of System is the important section in the system development procedure. The system is studied to the minute details and analyzed. The system analyst plays an vital function of an interrogator, and dwells deep into the operating of the existing gadget. In analysis, an in depth take a look at of these operations achieved by using the system and their relationships within and outdoors the device is executed. A key question considered right here is, “what should be carried out to solve the problem?” The system is regarded as a whole and the input to the machine is identified. Once analysis is finished the analyst has a firm expertise of what is to be carried out.

3.2 PROBLEM DEFINITION

Compared to the precedent IoT devices, such as smartphones, wearable IoT devices are more prone to various security attacks due to the lack of security measures (e.g., insufficient user authentication) and limited resources (e.g., computing power and energy capacity). Thus, to address the vulnerability of the current wearable IoT security system, we consider user authentication which is one of the most principal security measures.

3.3 EXISTING SYSTEM

To address the vulnerability of the current wearable IoT security system, we consider user authentication which is one of the most principal security measures. In user authentication, one of the most commonly used methods is using passwords due to its simplicity. Thus, people often have difficulty in remembering a correct password. According to the survey conducted by Centrify at Infosecurity Europe 2015, 33% of participants in the study suffered from password rage. Recently, Hanamsagar *et al.* showed there is significant password reuse over multiple different online accounts.

Besides, the PIN/Pattern-based authentication on smartwatches can be also prone to shoulder surfing and social engineering attacks. Fingerprint-based authentication can be deceived by high quality fingerprint images or counterfeit fingers created using a 3D printer. Even commercial iris-based authentication systems were breached by high-quality iris images. Various research works have been proposed to authenticate wearable or hand-held device users, using subconscious activities. However, most of the prior research conducted their experiments on their own custom-made devices. Even though there are some research projects conducted on commercial devices, they involve a long period of the authentication process mostly incurred by their walking detection algorithm.

3.3.1 DISADVANTAGES OF EXISTING SYSTEM

- Security limitations in wearable IoT devices.

- They require a relatively large volume of data, which can be burdensome to wearable IoT devices with limited computing resources and energy Capacity.

3.4 PROPOSED SYSTEM

We utilize subconscious activities to authenticate wearable IoT device users. In interactions with users, wearable IoT devices generate various sensor data and the pattern in such sensor data can be distinct as each user performs subconscious activities in their own way, which is not required to remember as well as difficult to hack or copy. Among various subconscious activities, we consider gait, which is the pattern of limb movements during locomotion as it provides a reliable and user-friendly authentication. So, we propose a lightweight seamless authentication framework based on gait (LiSA-G) that securely authenticates users on commercial smartwatches in a light manner.

3.4.1 ADVANTAGES OF PROPOSED SYSTEM

- Users do not need to remember their passwords, as our proposed system is both reliable and user-friendly.

- Increased the authentication accuracy by using human-action-related features as well as statistical ones and less amount of sensor data.

- Laborious for attackers to hack.

3.5 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During System Analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

3.5.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. The criteria to ensure that effort is concentrated on a project, which will give best, return at the earliest. Oneofthe factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication that the system is economically possible for development.

3.5.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client.The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.5.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.6 HARDWARE & SOFTWARE REQUIREMENTS

3.6.1 HARDWARE REQUIREMENTS

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements,

- System : Intel Core i3 and above
- Hard Disk : 20 GB and above
- Ram : 4 GB and above

3.6.2 SOFTWARE REQUIREMENTS

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

- Operating system : Windows
- Coding Language : Python

4. ARCHITECTURE

4. ARCHITECTURE

4.1 PROJECT ARCHITECTURE

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system.

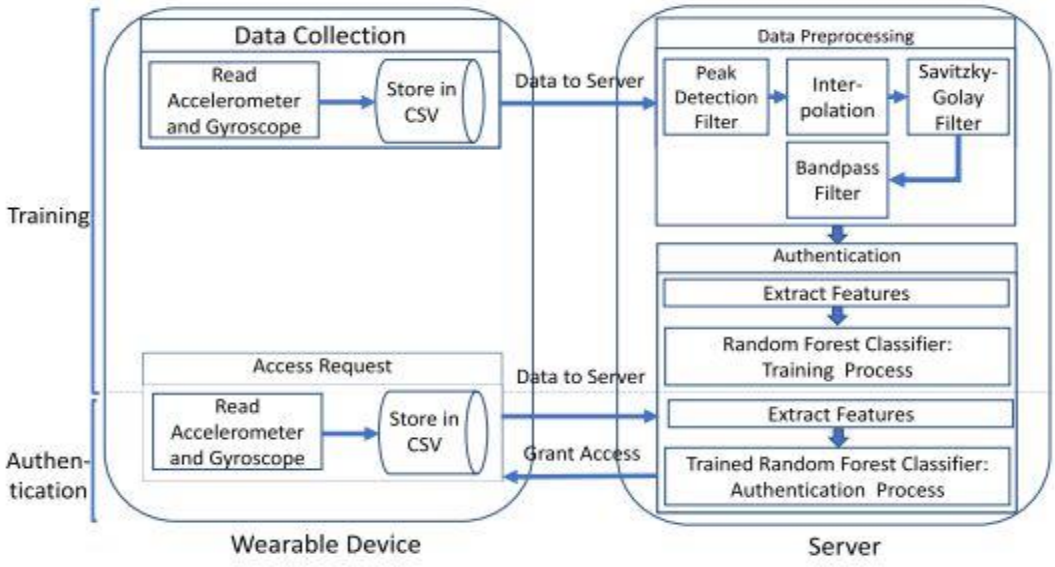


Figure 4.1: Architecture for LiSA-G: Authenticate and Identify Users on Widely Available Commercial Smartwatches

4.2 DESCRIPTION

In the conventional authentication system, a user needs to authenticate himself/herself by typing a password or providing biometric measures. In other words, the existing authentication system requires direct user interaction, which may hinder seamless authentication. However, when enabled to automatically analyze users' sensor data collected in real-time via wearable IoT devices, we can authenticate users without requiring their direct or active interaction.

Aiming such seamless authentication, we propose a gait-based authentication framework for wearable IoT devices that automatically authenticates users by analyzing their sensor data. As shown in Figure 1, the workflow of our framework mainly consists of three steps: 1) data collection, 2) data preprocessing, and 3) authentication.

4.3 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram to present a graphical overview of the functionality provided by a system in terms of actors, their goals which are represented as use cases, and any dependencies between those use cases. It is defined by and created from a Use-case analysis. The main purpose of a use case diagram is to show what system functions are performed for which actor.

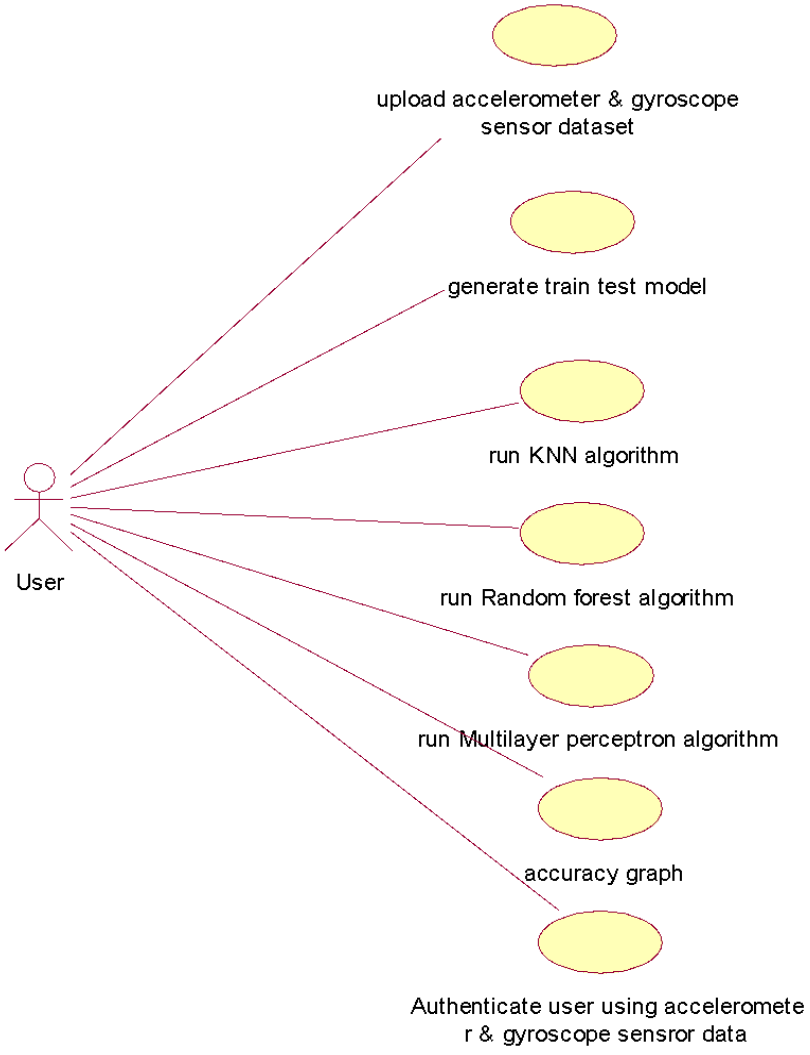


Figure 4.2: Use Case Diagram for LiSA-G: Authenticate and Identify Users on Widely Available Commercial Smartwatches

4.4 CLASS DIAGRAM

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.

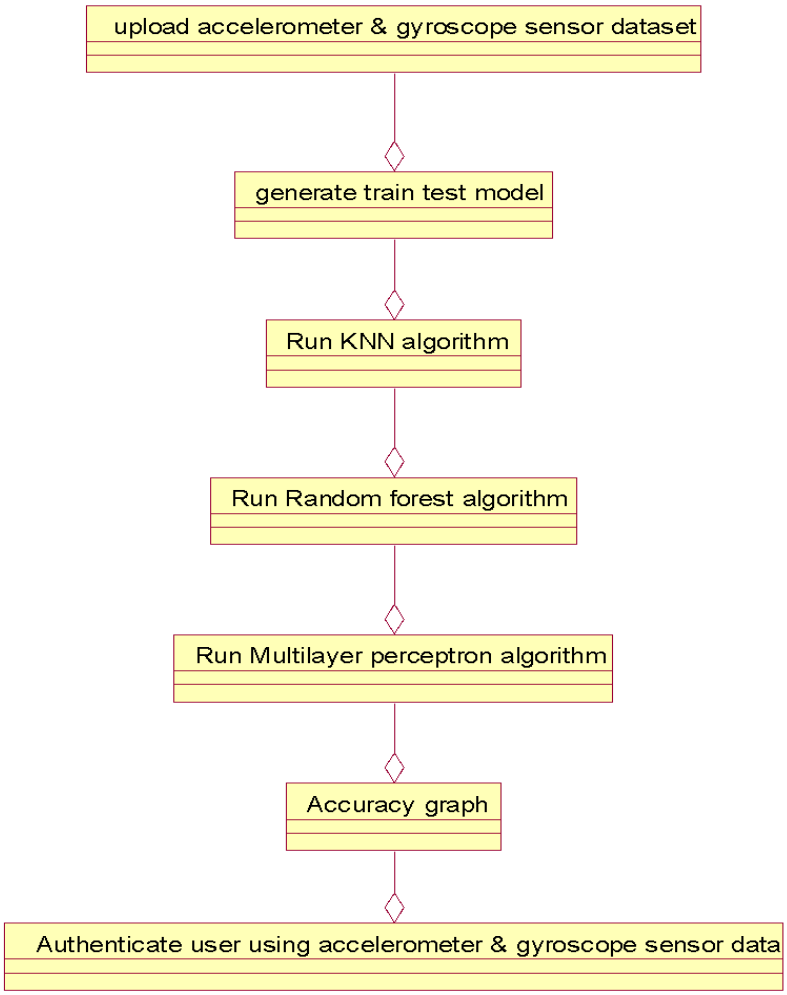


Figure 4.3: Class Diagram for LiSA-G: Authenticate and Identify Users on Widely Available Commercial Smartwatches

4.5 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is also called Event Diagrams, and Timing Diagrams. It is a kind of interaction diagram showing how processes operate with one another and in what order.

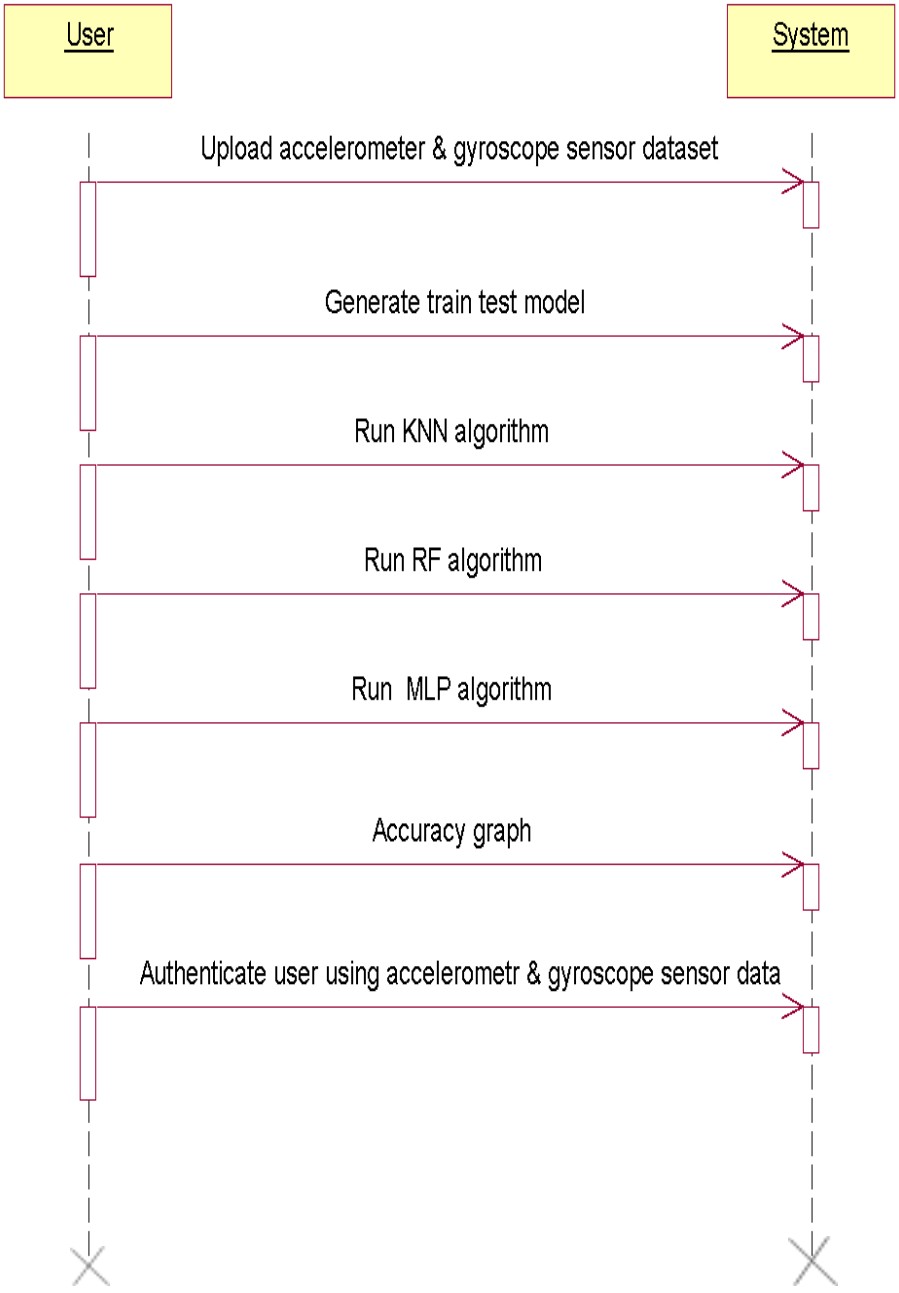


Figure 4.4: Sequence Diagram for LiSA-G: Authenticate and Identify Users on Widely Available Commercial Smartwatches

4.6 ACTIVITY DIAGRAM

Activity diagrams within the Unified Modeling Language, describe the commercial enterprise and operational step-by-step workflows of components in a system. It represents workflows of stepwise activities and moves with assist for preference, iteration and concurrency. An activity diagram suggests the overall flow of control.

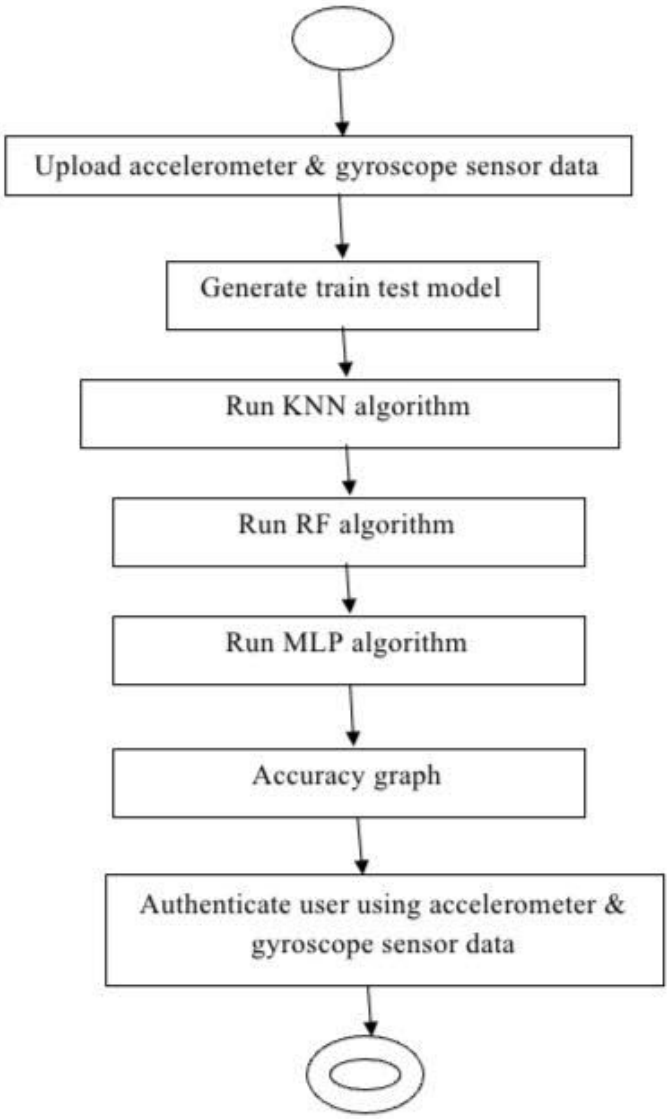


Figure 4.5: Activity Diagram for LiSA-G: Authenticate and Identify Users on Widely Commercial Smartwatches

5. IMPLEMENTATION

5. IMPLEMENTATION

5.1 SAMPLE CODE

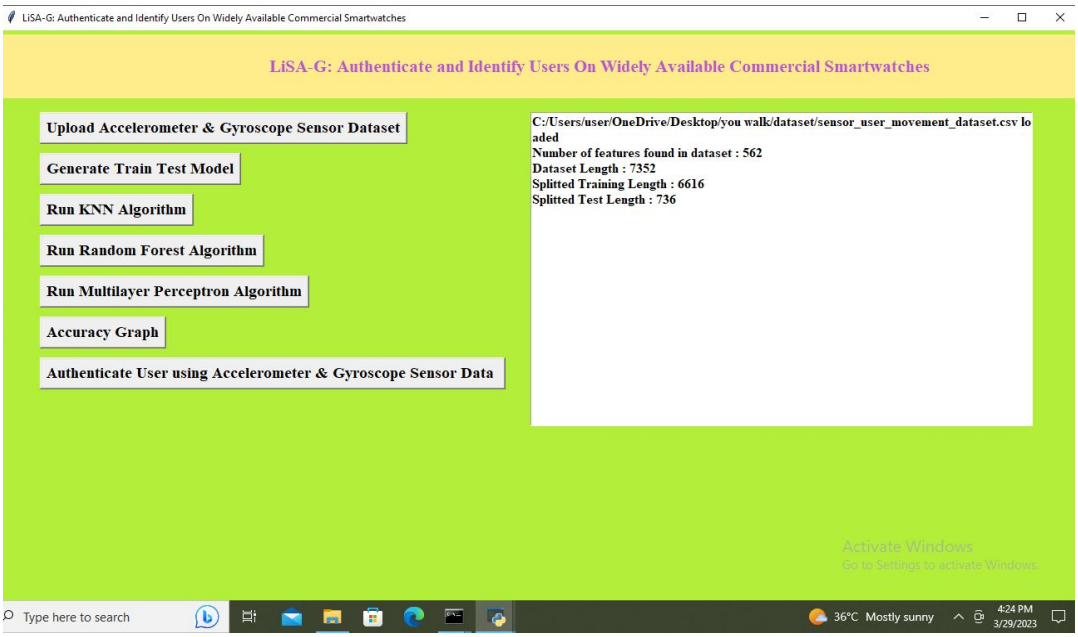
```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from tkinter import simpledialog
from tkinter import filedialog
from sklearn.metrics import accuracy_score
from sklearn_extensions.extreme_learning_machines.elm import GenELMClassifier
from sklearn_extensions.extreme_learning_machines.random_layer import
RBFRandomLayer, MLPRandomLayer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.neighbors import KNeighborsClassifier
main = tkinter.Tk()
main.title("You Walk, We Authenticate") #designing main screen
main.geometry("1300x1200")
global filename
global random_acc,knn_acc,mlp_acc
global X, Y, X_train, X_test, y_train, y_test
global data
global features
global classifier
def upload():
    global filename
    global data
    global features
```

```
    filename = filedialog.askopenfilename(initialdir="dataset")
    data = pd.read_csv(filename)
    rows = data.shape[0]  # gives number of row count
  cols = data.shape[1]  # gives number of col count
    features = cols - 1
    text.delete('1.0', END)
    text.insert(END, filename+" loaded\n");
    text.insert(END,"Number of features found in dataset : "+str(cols)+"\n");
  def splitdataset(balance_data):
      global X, Y, X_train, X_test, y_train, y_test
      X = balance_data.values[:, 0:features]
      Y = balance_data.values[:, features]
      print(X)
      print(Y)
      X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.1,
  random_state = 0)
      return X, Y, X_train, X_test, y_train, y_test
  def generateModel():
      X, Y, X_train, y_train, X_test, y_test = splitdataset(data)
      text.insert(END,"Dataset Length : "+str(len(X))+"\n");
      text.insert(END,"Splitted Training Length : "+str(len(X_train))+"\n");
      text.insert(END,"Splitted Test Length : "+str(len(y_train))+"\n\n");
  def prediction(X_test, cls):
    y_pred = cls.predict(X_test)
      for i in range(len(X_test)):
        print("X=%s, Predicted=%s" % (X_test[i], y_pred[i]))
      return y_pred
  # Function to calculate accuracy
  def cal_accuracy(y_test, y_pred, details):
      accuracy = accuracy_score(y_test,y_pred)*100
      text.insert(END,details+"\n")
      text.insert(END,"Accuracy : "+str(accuracy)+"\n\n")
      return accuracy
  def knnAlgorithm():
      global knn_acc
      text.delete('1.0', END)
      cls = BaggingClassifier(KNeighborsClassifier(),max_samples=0.5,
  max_features=0.5)
      cls.fit(X_train, y_train)
  prediction_data = prediction(X_test, cls)
```

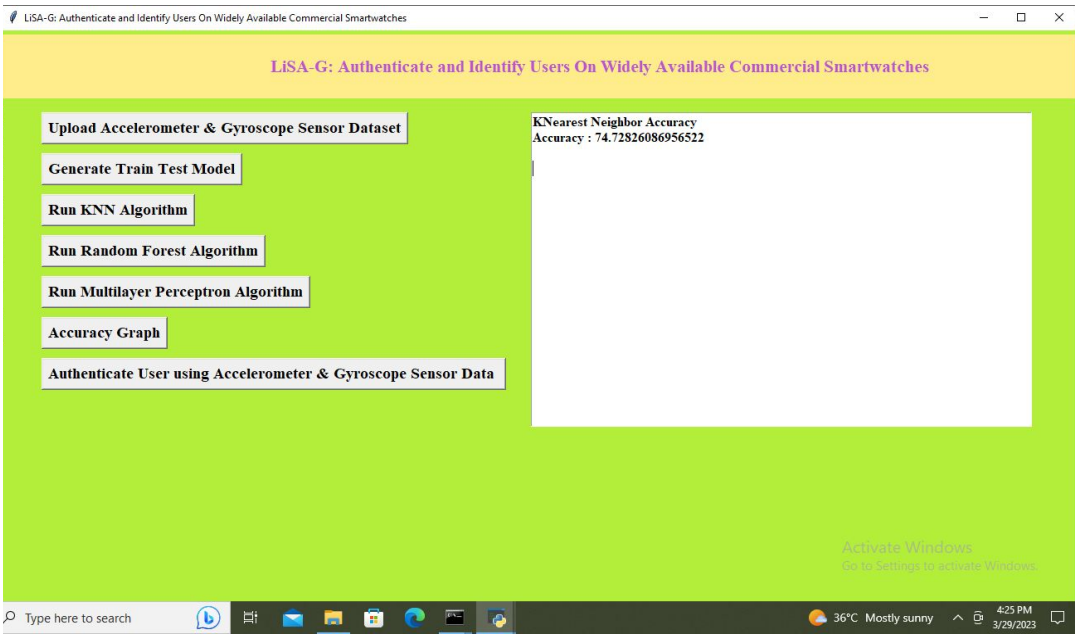
```
knn_acc = cal_accuracy(y_test, prediction_data,'KNearest Neighbor Accuracy')
def randomAlgorithm():
    global random_acc
    global classifier
text.delete('1.0', END)
    cls = RandomForestClassifier(max_depth=50, random_state=0)
    cls.fit(X_train, y_train)
    prediction_data = prediction(X_test, cls)
    random_acc = cal_accuracy(y_test, prediction_data,'Random Forest Accuracy')
    classifier = cls
def MLPAlgorithm():
    global mlp_acc
    text.delete('1.0', END)
    srhl_tanh = MLPRandomLayer(n_hidden=30, activation_func='tanh')
    cls = GenELMClassifier(hidden_layer=srhl_tanh)
    cls.fit(X_train, y_train)
    prediction_data = prediction(X_test, cls)
    mlp_acc = cal_accuracy(y_test, prediction_data,'Multilayer Perceptron Algorithm
Accuracy')
def graph():
    height = [knn_acc,random_acc,mlp_acc]
    bars = ('KNN Accuracy','Random Forest Accuracy','Multilayer Perceptron
Accuracy')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
  plt.show()
def Authentication():
    text.delete('1.0', END)
    filename = filedialog.askopenfilename(initialdir="dataset")
    test = pd.read_csv(filename)
    test = test.values[:, 0:features]
    text.insert(END,filename+" test file loaded\n");
    y_pred = classifier.predict(test)
    print(y_pred)
    for i in range(len(test)):
        text.insert(END,str(test[i])+" Authenticated as user "+str(y_pred[i])+" from
Accelerometer & Gyroscope Sensor Data\n\n");
font = ('times', 16, 'bold')
title = Label(main, text='You Walk, We Authenticate: Lightweight Seamless
```

```
Authentication Based on Gait in Wearable IoT Systems')
title.config(bg='LightGoldenrod1', fg='medium orchid')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)
font1 = ('times', 12, 'bold')
text=Text(main,height=20,width=75)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=640,y=100)
text.config(font=font1)
font1 = ('times', 14, 'bold')
uploadButton = Button(main, text="Upload Accelerometer & Gyroscope Sensor
Dataset", command=upload)
uploadButton.place(x=50,y=100)
uploadButton.config(font=font1)
generateButton = Button(main, text="Generate Train Test Model",
command=generateModel)
generateButton.place(x=50,y=150)
generateButton.config(font=font1)
knnButton = Button(main, text="Run KNN Algorithm", command=knnAlgorithm)
knnButton.place(x=50,y=200)
knnButton.config(font=font1)
randomButton = Button(main, text="Run Random Forest Algorithm",
command=randomAlgorithm)
randomButton.place(x=50,y=250)
randomButton.config(font=font1)
mlpButton = Button(main, text="Run Multilayer Perceptron Algorithm",
command=MLPAlgorithm)
mlpButton.place(x=50,y=300)
mlpButton.config(font=font1)
graphButton = Button(main, text="Accuracy Graph", command=graph)
graphButton.place(x=50,y=350)
graphButton.config(font=font1)
authButton = Button(main, text="Authenticate User using Accelerometer &
Gyroscope Sensor Data ", command=Authentication)
authButton.place(x=50,y=400)
authButton.config(font=font1)
main.config(bg='OliveDrab2')
main.mainloop()
```

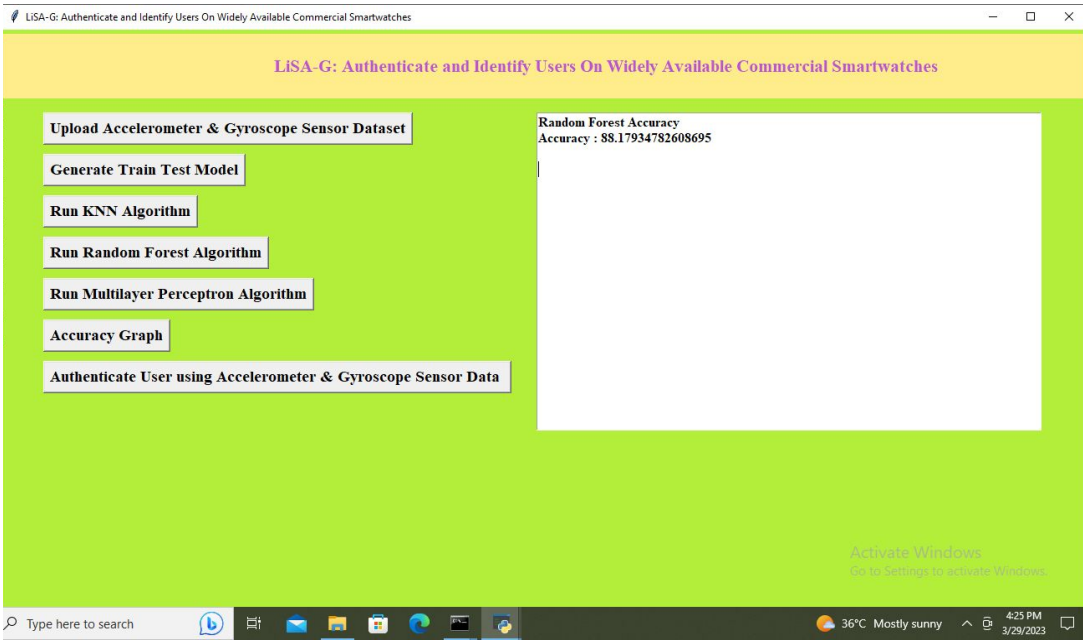
6. SCREENSHOTS



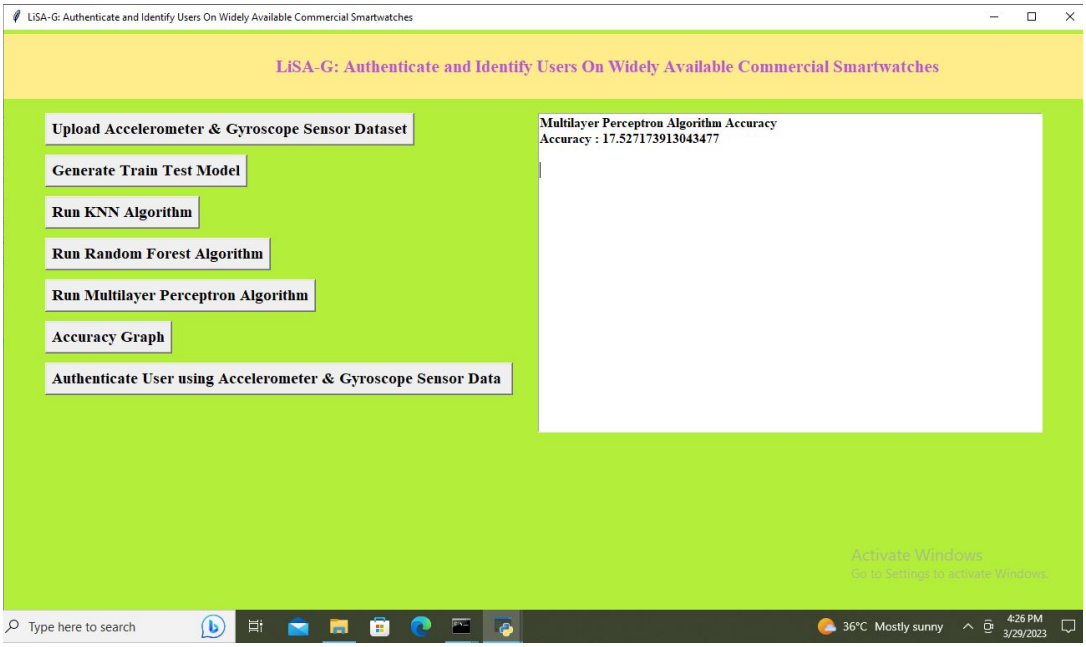
Screenshot 6.1: Uploading Sensor Dataset and Generating Train Test Model



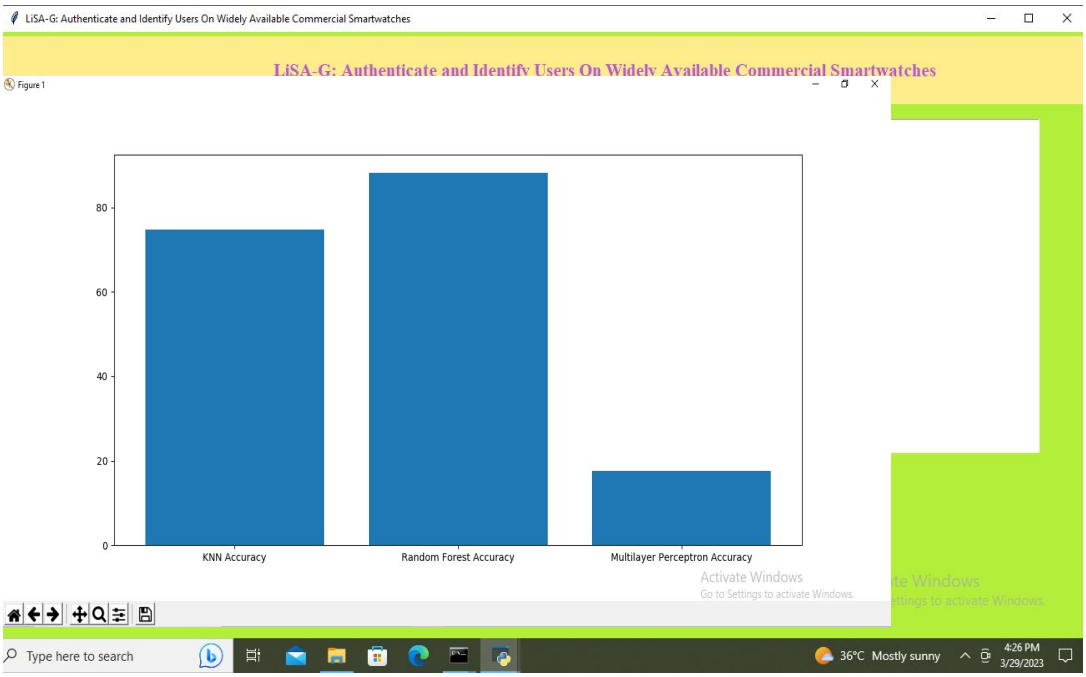
Screenshot 6.2: Running KNN Algorithm



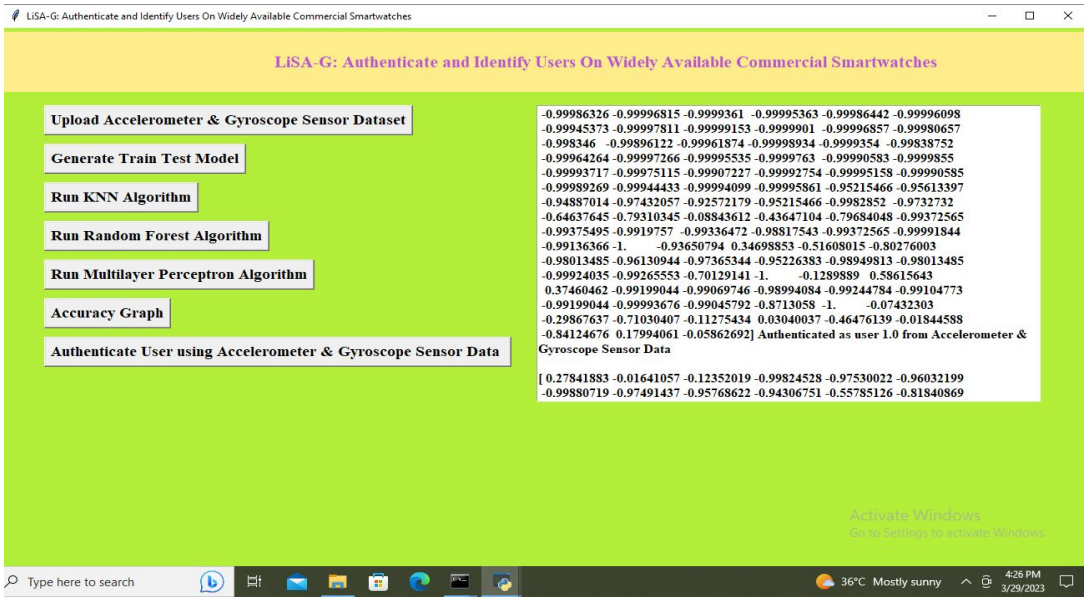
Screenshot 6.3: Running Random Forest Algorithm



Screenshot 6.4: Running Multilayer Perceptron Algorithm



Screenshot 6.5: Accuracy Graph



Screenshot 6.6: Authenticate User Using Accelerometer and Gyroscope Sensor Data

7. TESTING

7. SYSTEM TESTING

7.1 INTRODUCTION

System testing, also referred to as system-level tests or system-integration testing, is the process in which a quality assurance (QA) team evaluates how the various components of an application interact together in the full, integrated system or application. System testing verifies that an application performs tasks as designed. This step, a kind of black box testing, focuses on the functionality of an application. System testing, for example, might check that every kind of user input produces the intended output across the application.

Phases of system testing:

System testing examines every component of an application to make sure that they work as a complete and unified whole. A QA team typically conducts system testing after it checks individual modules with functional or user-story testing and then each component through integration testing.

If a software build achieves the desired results in system testing, it gets a final check via acceptance testing before it goes to production, where users consume the software. An app-dev team logs all defects, and establishes what kinds and amount of defects are tolerable.

7.2 LEVELS OF TESTING

Code Testing:

It is the examination of the logic of the program.

Speciation Testing:

Executing specification is beginning with what the program is ought to do, and how it should perform under diverse situations. Test cases for various situations, and combinations of conditions in all the modules are tested.

Unit Testing:

This is additionally called Module Testing. In unit testing every module is tested individually and integrated with the overall system. It particularly focuses on verification efforts which are accomplished on the smallest unit of the software layout within the module. The module of the system is tested one at a time. This testing is performed during the programming level itself. In the testing step each module is observed to work satisfactorily as regard to expected output from the module. There are few validation checks for fields additionally.

Every module can be tested using the following two strategies:

- Black Box Testing
- White Box Testing

7.2.1 BLACK BOX TESTING

Black Box Testing is a technique for software testing wherein the capability of Software Under Test(SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software program. This sort of testing is based totally on the software requirements and specifications.

Black Box Testing mainly focuses on the inputs and outputs of the software system without considering the inner knowledge of the software program. The "black box" in "black box testing" symbolizes not being able to see the internal workings of the software, so that only the end-user experience can be tested.

Types of Black Box Testing

The following are the prominent ones among many

Functional testing: This type of black box testing is performed by the software testers. It is related to the functional requirements of a system.

Non-functional testing: This type of black box testing is related to non-functional requirements such as performance, scalability, usability however not related to testing of a particular functionality.

Regression testing: Regression testing is done after code fixes , upgrades or any other system maintenance to check the new code has not affected the existing code.

7.2.2 WHITE BOX TESTING

White Box Testing is also known as clear, open, structural, and glass box testing. It is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and enhancing layout and usability. The clear box or whitebox name symbolizes the ability to see via the software's outer shell (or "box") into its inner workings.

Integration Testing:

The main motive of integration testing is to expose faults in interaction between integrated units. Integration testing of software testing is a level of testing in which individual units are combined and are tested as a group. Test drivers and test stubs are used to assist in Integration Testing. Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. It occurs after unit testing and before validation testing. Integration testing can be done in two ways: Bottom Up integration testing and Top Down integration testing.

8. CONCLUSION

8. CONCLUSION & FUTURE SCOPE

8.1 CONCLUSION

Walking pattern, e.g., arm swings in a walk can be utilized to effectively authenticate users. In this work, we proposed a new gait based authentication framework, LiSA-G, that is reliable, user-friendly, and easily deployable. Our framework can classify users with a higher accuracy (91.8% success rate) than other existing works while using less number of features by extracting a new combination of features that are related to the human behavioral traits. The proposed framework is user friendly as it capitalizes on users' mundane activities, and easily deployable as commercially available smartwatches are used.

Considering its application to the IoT ecosystem with limited resources, the proposed framework is designed lightweight by eliminating the gait cycle detection process and using much less amount of data. We expect that the proposed framework can help to provide seamless authentication and can be readily integrated with other systems to provide multi-factor authentication.

8.2 FUTURE SCOPE

Given its appropriateness to the IOT ecosystem's limited assets, the proposed structure is intended to be lightweight by eliminating the step cycle identification method and utilizing considerably less information. We guess that the proposed structure will support the arrangement of consistent validation and will be effectively associated with different frameworks to empower multifaceted confirmation.

9. BIBLIOGRAPHY

9. BIBLIOGRAPHY

[1] T. Micro. (Mar. 2018). Are your Wearables Fit to Secure You? Researchers Outline 3 Attack Surfaces.

[2] M. Prigg. (May 2013). Google Glass Hacked to Transmit Everything You See and Hear: Experts Warn ’the Only Thing it Doesn’t Know are Your Thoughts.

[3] J. Chatzky. (May 2017). Password Rage, it’s a Thing.

[4] A. Hanamsagar, S. S. Woo, C. Kanich, and J. Mirkovic, ‘‘Leveraging semantic transformation to investigate password habits and their causes,’’ in Proc. CHI Conf. Hum. Factors Comput. Syst., 2018, p. 570.

[5] Y. Zhao, Z. Qiu, Y. Yang, W. Li and M. Fan, "An empirical study of touch-based authentication methods on smartwatches", Proc. ACM Int. Symp. Wearable Comput. (ISWC), pp. 122-125

[6] J. Myerson, How to Fool a Fingerprint Sensor, Mar. 2017

[7] S. Khandelwal, Hacker Finds a Simple Way to Fool Iris Biometric Security Systems, Mar. 2015

[8] D. Gafurov, E. Snekkenes and P. Bours, "Spoof attacks on gait authentication system", IEEE Trans. Inf. Forensics Security, vol. 2, pp. 491-502, Sep. 2007.

[9] G. Cola, M. Avvenuti, F. Musso and A. Vecchio, "Gait-based authentication using a wrist-worn device", Proc. 13th Int. Conf. Mobile Ubiquitous Syst. Comput. Netw. Services (MOBIQUITOUS), pp. 208-217, Nov. 2016.

[10] A. Mahfouz, T. M. Mahmoud and A. S. Eldin, "Poster: A behavioral biometric authentication framework on smartphones", Proc. ACM Asia Conf. Comput. Commun. Secur. (ASIA CCS), pp. 923-925, Apr. 2017

9.1 GITHUB LINK

https://github.com/zebaunnissa/LISA-G-AUTHENTICATE-AND-IDENTIFY-USERS-ON-THE-WIDELY-AVAILABLE-COMMERCIAL-SMARTWATCHES.git