A

Project Phase II Report

On

Intrusion Detection System using Machine Learning

SUBMITTED BY

Name of Students	**Exam Seat Number**	**PRN**
Rahul Kumar	B150083069	71901749B
Tejas Bhagat	B150083019	71901269E
Rohit Bhandare	B150083021	71901765D

PROJECT GUIDE

Mrs. Sayali Mane

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION

D.Y.PATIL COLLEGE OF ENGINEERING

AKURDI, PUNE – 411044

2021-2022

D.Y.PATIL COLLEGE OF ENGINEERING

AKURDI, PUNE – 411044

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION

CERTIFICATE

This is to certify that **Rahul Kumar – B150083069, Tejas Bhagat – B150083019, Rohit Bhandare – B15003021.** of B.E. E&TC has completed the project phase II on Intrusion Detection System using Machine Learning satisfactorily under my supervision and guidance and submitted the project report in complete fulfillment of requirement for the award of Bachelors Degree of Engineering course under the Savitribai Phule Pune University during the academic year 2021-2022.

Name of Students	**Exam Seat Number**	**PRN**
Rahul Kumar	B150083069	71901749B
Tejas Bhagat	B150083019	71901269E
Rohit Bhandare	B150083021	71901765D

Name of Project Guide	**Dr. Mrs. R. Deshmukh**	**Dr. Mrs. P. Malathi**
Mrs Sayali Mane	**Administrative Coordinator**	**H.O.D. & I/C Principal**

Name and Signature of External Examiner

ACKNOWLEDGEMENT

We express our sincere gratitude towards the faculty members who makes this project a successful.We would like to express our thanks to our guide **Mrs Sayali Mane** for **her** whole hearted co-operation and valuable suggestions, technical guidance throughout the project work. Special thanks to our H.O.D. **Dr. Mrs. P. Malathi** for her kind official support and encouragement. We are also thankful to our project coordinators **Mrs. Sayali N. Mane** and **Mrs. Shailaja S.Yadav** for their valuable guidance.

Finally, we would like to thank all staff members and faculty members of E&TC Department who helped us directly or indirectly to complete this work successfully.

Students Name & Sign

Rahul Kumar

Tejas Bhagat

Rohit Bhandare

INDEX

LIST OF FIGURES:

Abstract:

The Intrusion network (INetwork) connect systems, applications, data storage, and services that may be a new gateway for cyber- attacks as they continuously offer services in the organization. Currently, software piracy and malware attacks are high risks to com- promise the security of Intrusion network. These Intrusion may steal important information that causes economic and reputational damages. In this Paper, we have proposed a combined Machinelearning approach to detect the pirated software and malware-infectedfiles across the Intrusion network. The Tensor Flow Machine neural network is proposed to identify pirated software using source code plagiarism. The tokenization and weighting feature methods are used to filter the noisy data and further, to zoom the importance of each token in terms of source code plagiarism. Then, the Machine learning approach is used to detect source code plagiarism. The dataset is collected from KDD cup 99 (KC99) to investigate software piracy. Apart from this, the Machine convolutional neural network is used to detect malicious infections in Intrusion network through color image visualization. The malware samples are obtained from Mailing dataset for experimentation. The experimental results indicate that the classification performance of the proposed solution to measure the cybersecurity Intrusion in Intrusion network are better than the state of the art methods.

CHAPTER 1

INTRODUCTION

1.1 Overview:

Intrusion network is the interconnection of physical moving objects “Things” through internet embedded with an electronic chip, sensors, and other forms of hardware. Each device is uniquely identified globally by Radio Frequency Identifier (RFID) tags. These smart objects communicate with other connected nodes and can be monitored and controlled remotely Intrusion network offers pervasive connectivity to a broad range of intelligent physical objects, service industries, cloud computing services, and applications. IBM stated that the number of connected devices through the internet is expected to increase up to 50 billion by It will increase the number of communication networks with the connection of smart objects as well asthe amount of big data that may be shared using cloud infrastructure. The Intrusion network enabled technologies can be used to develop smart cities, education system, e-shopping, e-banking, maintain our health, manage industry, and to entertain and protect human beings . The Intrusion network devices can be used for an open attackdue to always available on the network.

1.1.1 Motivation

The threat itself comes in many forms, including viruses, worms, distributed denial of services, electronic bombs, and derives many motives, including revenge, personal gains, hate, and joy rides, to name but a few.

1.1.2 Objective

To response is about utilizing big data analytics to find Intrusion across largeand disparate data sets. The objective is to find anomalies, analyze their threatlevel, and determine what mitigative action(s) may be required in response.

CHAPTER 2
LITERATURE SURVEY

2.1 Study Of Research Paper:

1. **Paper Name:** Cyber Threat Intelligence Model: An Evaluation of Taxonomies,Sharing Standards, and Ontologies within Cyber Threat Intelligence

   Author: Vasileios Mavroeidis

   Abstract ::- Threat intelligence is the provision of evidence-based knowledge about existing or potential Intrusion. Benefits of threat intelligence include improved efficiency and effectiveness in security operations in terms of detective and preventive capabilities. Successful threat intelligence within the cyber domain demands a knowledge base of threat information and an expressive way to represent this knowl-edge. This purpose is served by the use of taxonomies, sharing standards, and on tologies. This Paper introduces the Cyber Threat Intelligence (CTI) model, which enables cyber defenders to explore their threat intelligence capabilities and under- stand their position against the ever-changing cyber threat landscape. In addition, we use our model to analyze and evaluate several existing taxonomies, sharing standards, and ontologies relevant to cyber threat intelligence. Our results show that the cyber security community lacks an ontology covering the complete spectrum ofthreat intelligence. To conclude, we argue the importance of developing a multi- layered cyber threat intelligence ontology based on the CTI model and the steps should be taken under consideration, which are the foundation of our future work.

2. **Paper Name:** :- Probabilistic Threat Detection for Risk Management in Cyber-physical Medical Systems

   Author: Aakarsh Rao, Nadir Carreón, Roman Lysecky, and Jerzy Rozenblit,

   Abstract: Medical devices are complex cyber-physical systems exposed to numerous security risks and vulnerabilities. This article presents a dynamic risk management and automated threat mitigation approach based on a probabilistic threat estimation framework. A smart-connected pacemaker case study illustrates the approach

3. **Paper Name:** Managing cyber threat intelligence in a graph database

   Author name: Seulgi Lee, Hyeisun Cho, Nakhyun Kim, Byungik Kim

   Abstract: Efforts to cope jointly with the ever-increasing number of breach incidents have resulted in the establishment of the standard format and protocol and given birth to many consultative groups. In addition, various channels that distributeCyber Threat Intelligence information free of charge have emerged, and studies on utilizing such channels have spread. As the market for sharing information professionally is expanding, the need to manage the shared information in various ways inorder to achieve better result has arisen. This Paper proposes a standardized management structure and method based on the standardized format and a meaning andstandard of Cyber Threat Intelligence that can be shared outside when loading OS- INT information collected from various channels into the graph database. This pa- per also proposes a method of supporting the detection provided by existing security equipment with the information saved in the graph database and an effective method of analysis. Lastly, the Paper discusses the advantages that can be expected from saving cyber threat information information in the graph database developed using information collected from the outside.

4. **Paper Name:** Optimized Reactive Power Flow of DFIG Power Converters for Better Reliability Performance Considering Grid Codes

   Author: Dao Zhou

   Abstract: If there is no reactive power exchange between a doubly fed induction generator (DFIG) and a grid, the various characteristics of the power converters in a DFIG wind turbine system cause the lifetime expectancy of a rotor-side converter (RSC) to be significantly less than that of a grid-side converter (GSC). In order to fulfill modern grid this Paper, the additional stress of a power semiconductordue to the reactive power injection is first evaluated in terms of a modulation indexand the current loading. Then, an optimized reactive power flow is proposed in the case where an over-excited reactive power support is applied with the joint compensation from both the RSC and the GSC. Finally, some experimental validations are performed at a downscale DFIG prototype. It is concluded that, among the different combined reactive power support strategies, the best scheme will tradeoff the lifetime between the GSC and the RSC

5. **Paper Name:** A Design of IL-CyTIS for Automated Cyber Threat Detection

   Author: Joseph Yoo

   Abstract: As cyber squabbling has been intensified, the necessity of sharing cyber threat information has increased. Therefore, attempts to develop a technology to upgrade and Machine the related system will continue. In particular, it is anticipated that automated response and analysis using machine learning will be actively conducted. In this Paper, we design and propose IL-CyTIS (a unified and lightened information structure) by customizing STIX (a cyber threat information expression standard) for the input and analysis via machine learning before conducting a study using machine learning to derive new information from existing cyber threat information. Then, we discuss its actual application in machine learning.

6. **Paper Name:** Cyber Threat Detection and Application Analysis

   Author: Shuangmao Yang, Ji Wang,Jing Zhang, Hao Li

   Abstract: With the security situation in Cyberspace constantly becoming worse, Cyber threat detection has attracted a lot of researching attentions. In this Paper, existing detection technologies are firstly reviewed. Secondly, a framework of capturingthe abnormal traffic of botnets is proposed. Major modules and key detection techniques are presented at the same time. The hidden threat detection in physically isolated network is also discussed, and a detection system capable of detecting and locating hidden malicious programs is proposed and validated by experiments. Conclusion and future researching suggestions are given finally.

7. **Paper Name:**A Study on a Cyber Threat Intelligence Analysis (CTI) Platformfor the Proactive Detection of Cyber Attacks Based on Automated Analysis \

   Author name: Byung Ik Kim, Nakhyun Kim, Seulgi Lee, Hyeisun Cho, Junhyung Park

   Abstract: This Paper proposes an automated cyberattack analysis platform thatis designed to analyze and respond to cyberattacks, which are becoming ever more intelligent and advanced. The ICT information generated during previous cyberattacks will be collected to analyze cyberattacks automatically, and the relationshipbetween the collected information, level of re-exploitation, and similar ICT infor-mation among cyberattacks will be automatically analyzed. If the values that arecurrently being monitored are entered into the developed platform, the most similarcyberattacks in the past and the current phase of attacks will be provided to the analyst. In addition, a system capable of blocking attacks in advance before damagesare caused could be developed by providing response/ analysis guideline informationon the potential future attack inflow.

8. **Paper Name:**Data-Mining a Mechanism against Cyber Intrusion: A Review

Author: Shipra Ravi Kumar Prof. J.S.Jassi Suman Avdhesh Yadav Ravi Sharma

Abstract:- Data mining is the process in that analyzing of data from different perspective and summarizes that data into some useful information which can be used to enhance the revenue generation, cost cutting etc.. In data mining, cluster formation plays a vital role which is data can be divided into different groups. Clustering is the technique in which grouping is based on similar type of data relevant to different attributes. WEKA is the most important tool of data mining which is used to allocate and clustering of data with use of various machine learning algorithms. The purpose of this Paper is to compare different algorithms of machine learning on the subject of types of data set, their size, number of clusters and cyber privacy platform. We also discuss different types of cyber Intrusion in computing world.

9. **Paper name:** Optimal Machine Learning Algorithms for Cyber Threat Detection

Author: Hafiz M. Farooq

Abstract:—With the exponential hike in cyber Intrusion, organizations are now striving for better data mining techniques in order to analyze security logs received from their IT infrastructures to ensure effective and automated cyber threat detection. Machine Learning (ML) based analytics for security machine data is the next emerging trend in cyber security, aimed at mining security data to uncover advanced targeted cyber Intrusion actors and minimizing the operational overheads of maintaining static correlation rules. However, selection of optimal machine learning algorithm for security log analytics still remains an impeding factor against the success of data science in cyber security due to the risk of large number of false-positive detections, especially in the case of large-scale or global Security Operations Center(SOC) environments. This fact brings a dire need for an efficient machine learning based cyber threat detection model, capable of minimizing the false detection rates.In this Paper, we are proposing optimal machine learning algorithms with their implementation framework based on analytical and empirical evaluations of gathered results, while using various prediction, classification and forecasting algorithms.

10. **Paper Name:** Intrusion network Threat Detection Advances, Challenges and Fu-ture Directions

    Author: Nickson M. Karie

    Abstract: It is predicted that, the number of connected INetwork (Intrusion network) devices will rise to 38.6 billion by 2025 and an estimated 50 billion by 2030. The increased deployment of Intrusion network devices into diverse areas of our life has provided us with significant benefits such as improved quality of life and task automation. However, each time a new Intrusion network device is deployed, new and unique security Intrusion emerge or are introduced into the environment under which the device must operate. Instantaneous detection and mitigation of every se- curity threat introduced by different Intrusion network devices deployed can be very challenging. This is because many of the Intrusion network devices are manufactured with no consideration of their security implications. In this Paper therefore, we review existing literature and present Intrusion network threat detection research advances with a focus on the various Intrusion network security challenges as well as the current developments towards combating cyber security Intrusion in Intrusion network networks. However, this Paper also highlights several future research directions in the Intrusion network domain

CHAPTER 3
PROBLEM STATEMENT

3.1 Problem Statement:

To Threat detection is the practice of analyzing the entirety of a security ecosystemto identify any malicious activity that could compromise the network. If a threatis detected, then mitigation efforts must be enacted to properly neutralize the threatbefore it can exploit any present vulnerabilities.

CHAPTER 4

PROJECT REQUIREMENT

4.1 External Interface Requirement:

4.1.1 User Interface:

Application Based Threat Detection

4.1.2 Hardware Interfaces:

RAM : 8 GB

As we are using Machine Learning Algorithm and Various High Level LibrariesLaptop RAM minimum required is 8 GB.Hard Disk :
40 GB
Data Set of CT Scan images is to be used hence minimum 40 GB Hard Disk memoryis required.
Processor : Intel i5 Processor

Pycharm IDE that Integrated Development Environment is to be used and data loading should be fast hence Fast Processor is required
IDE : Pycharm

Best Integrated Development Environment as it gives possible suggestions at thetime of typing code snippets that makes typing feasible and fast.
Coding Language : Python Version 3.5

Highly specified Programming Language for Machine Learning because of avail-ability of High Performance Libraries.
Operating System : Windows 10

Latest Operating System that supports all type of installation and development Environment.

4.1.3 Software Interfaces:

Operating System: Windows 10

IDE: Pycharm ,Spyder
Programming Language : Python

4.2 Non Functional Requirement:

4.2.1 Performance Requirements:

The performance of the functions and every module must be well. The overall performance of the software will enable the users to work efficiently. Performance of encryption of data should be fast. Performance of the providing virtual environmentshould be fast Safety Requirement. The application is designed in modules where errors can be detected and xedeasily. This makes it easier to install and update new functionality if required.

4.2.2 Safety Requirement:

The application is designed in modules where errors can be detected and fixed easily.This makes it easier to install and update new functionality if required.

4.2.3 Software Quality Attributes:

Our software has many quality attribute that are given below:-

1. Adaptability and Availability: This software is adaptable by all users. This software is freely available to all users. The availability of the software is easy for everyone.
2. Maintainability: After the deployment of the project if any error occurs then it can be easily maintained by the software developer.
3. Reliability: The performance of the software is better which will increase the reliability of the Software.
4. User Friendliness: Since, the software is a GUI application; the output generated is much user friendly in its behavior.
5. Integrity: Integrity refers to the extent to which access to software or data by unauthorized persons can be controlled.
6. Security: Users are authenticated using many security phases so reliable security is provided.
7. Testability: The software will be tested considering all the aspects.

CHAPTER 5

SYSTEM ANALYSIS

5.1 System Architecture:

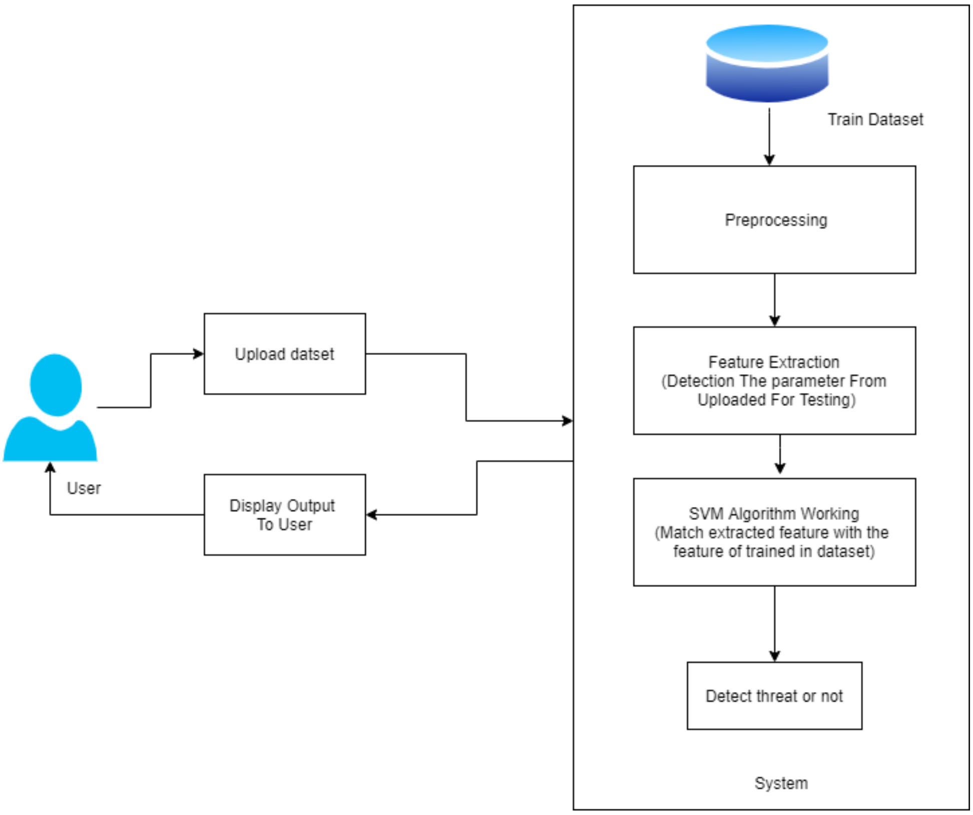


FIGURE 5.1: SYSTEM ARCHITECTURE 1

5.1.1 Module:

1. Admin

2. In this module, the Admin has to log in by using valid user name and password.After login successful he can do some operations such as View All Users andAuthorize, View All E-Commerce Website and Authorize, View All Productsand Reviews, View All Products Early Reviews, View All Keyword Search Details, View All Products Search Ratio, View All Keyword Search Results, View All Product Review Rank Results.

3. View and Authorize Users

In this module, the admin can view the list of users who all registered. In this,the admin can view the user’s details such as, user name, email, address and admin authorizes the users.

4. View All Products Search Ratio, View All Keyword Search Results, View All Product Review Rank Results.

5. Ecommerce User

6. In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password Once Login is successful user will do some operations like Add Products, View All Products with reviews, View All Early Product’s reviews, View All Purchased Transactions.

7. End User

5.1.2 Data Flow Diagram:

In Data Flow Diagram, we Show that flow of data in our system in DFD0 we show that base DFD in which rectangle present input as well as output and circle show our system,In DFD1 we show actual input and actual output of system input of our system is text or image and output is rumor detected likewise in DFD 2 we present
operation of user as well as admin.

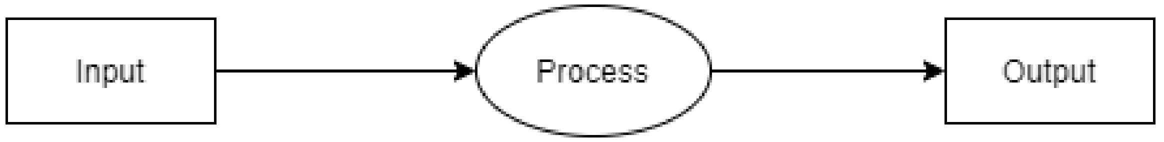


FIGURE 5.2: DATA FLOW(0) DIAGRAM

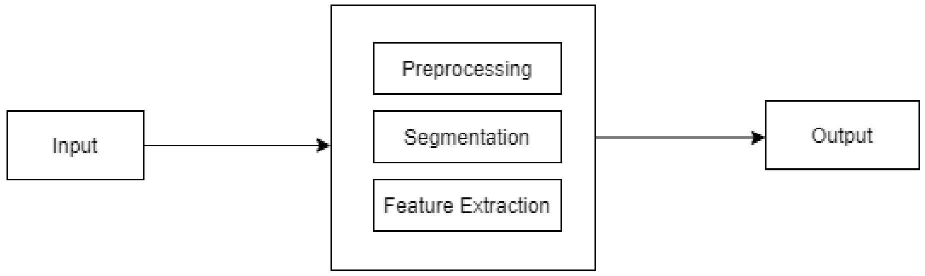


FIGURE 5.3: DATA FLOW(1) DIAGRAM

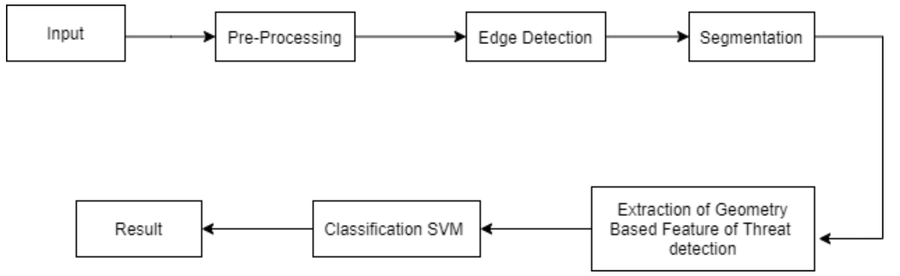


FIGURE 5.4: DATA FLOW(2) DIAGRAM

5.2 UML Diagrams:

Unified Modeling Language is a standard language for writing software blueprints. TheUML may be used to visualize, specify, construct and document the artifacts of a soft- ware intensive system. UML is process independent, although optimally it should be used in process that is use case driven, architecture-centric, iterative, and incremental. The Number of UML Diagram is available.

1. Class Diagram:

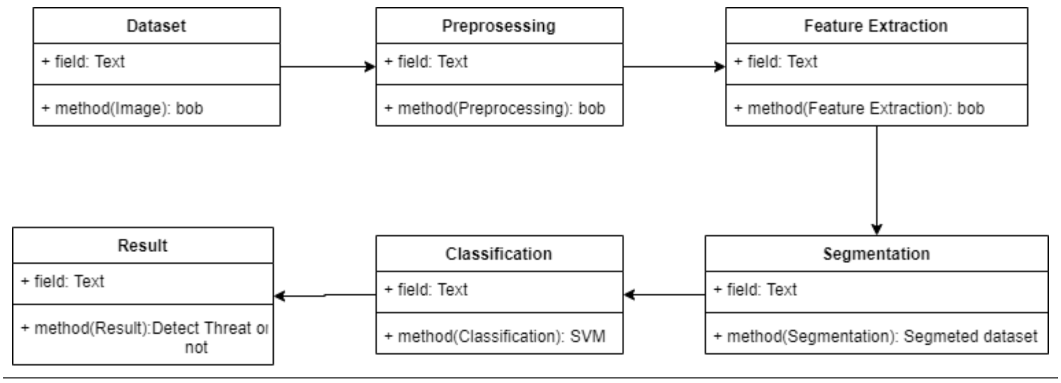


FIGURE 5.5: CLASS DIAGRAM

2. Use Case Diagram:

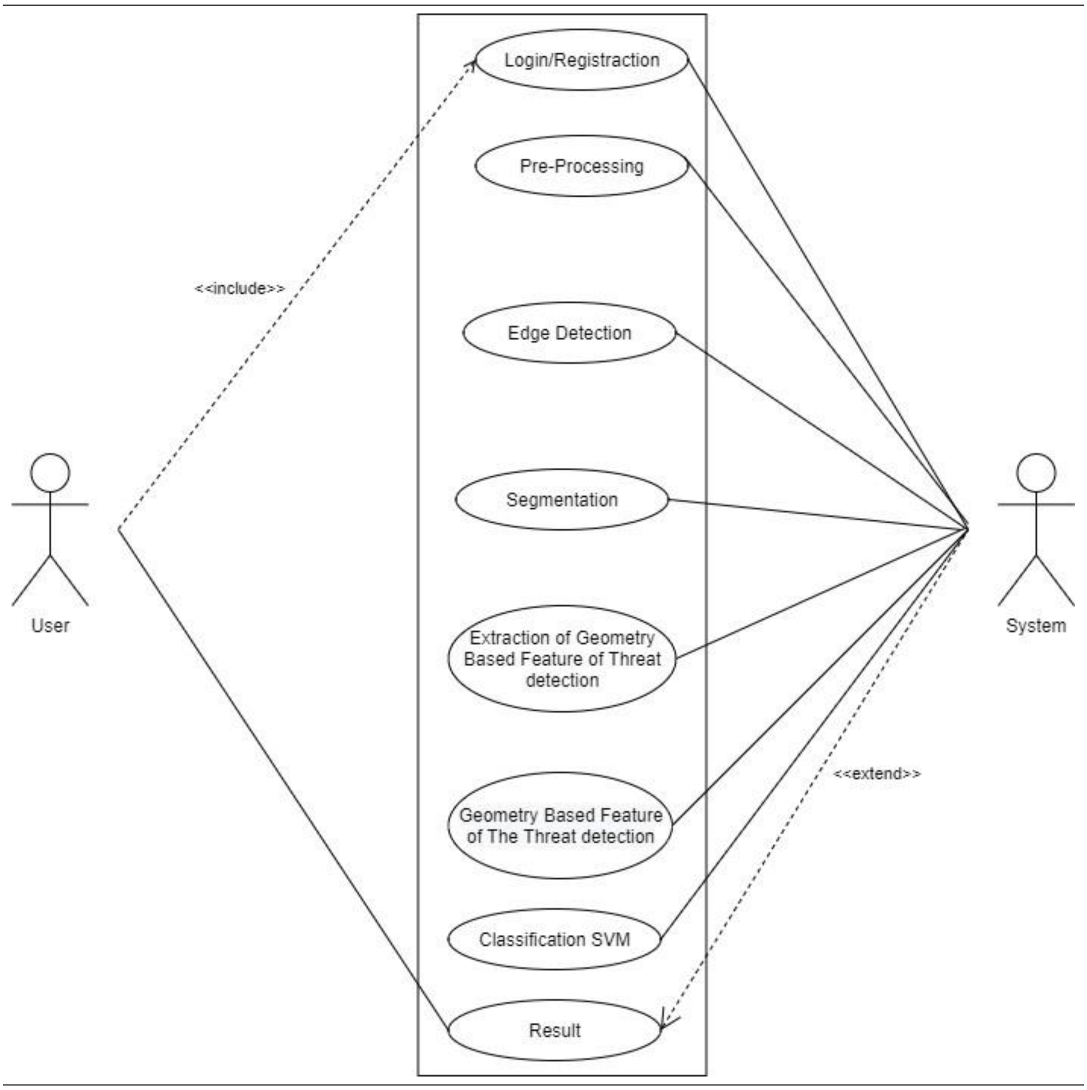


FIGURE 5.6: USE CASE DIAGRAM

3. Activity Diagram:

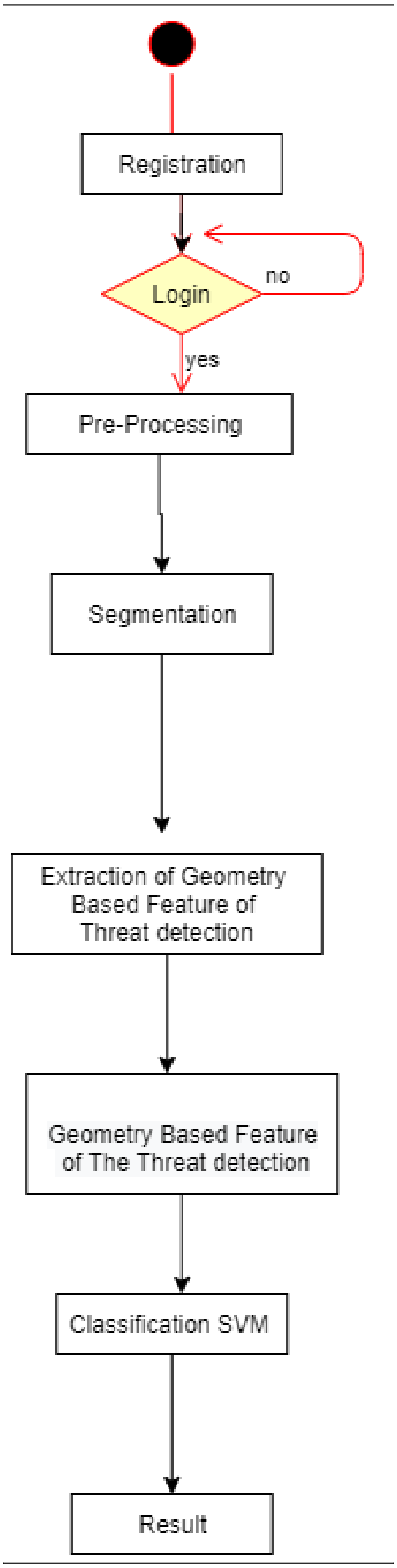

Registration
Login
no
yes
Pre-Processing
Segmentation
Extraction of Geometry Based Feature of Threat detection
Geometry Based Feature of The Threat detection
Classification SVM
Result

FIGURE 5.7: ACTIVITY DIAGRAM

4. Sequence Diagram:

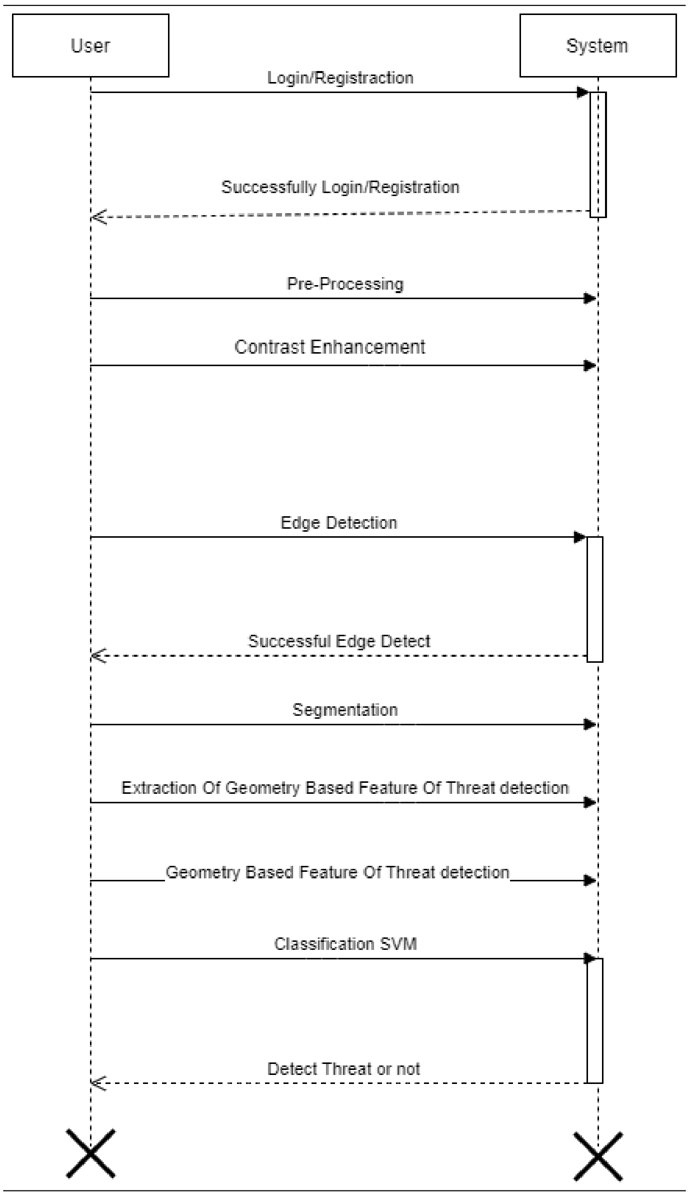
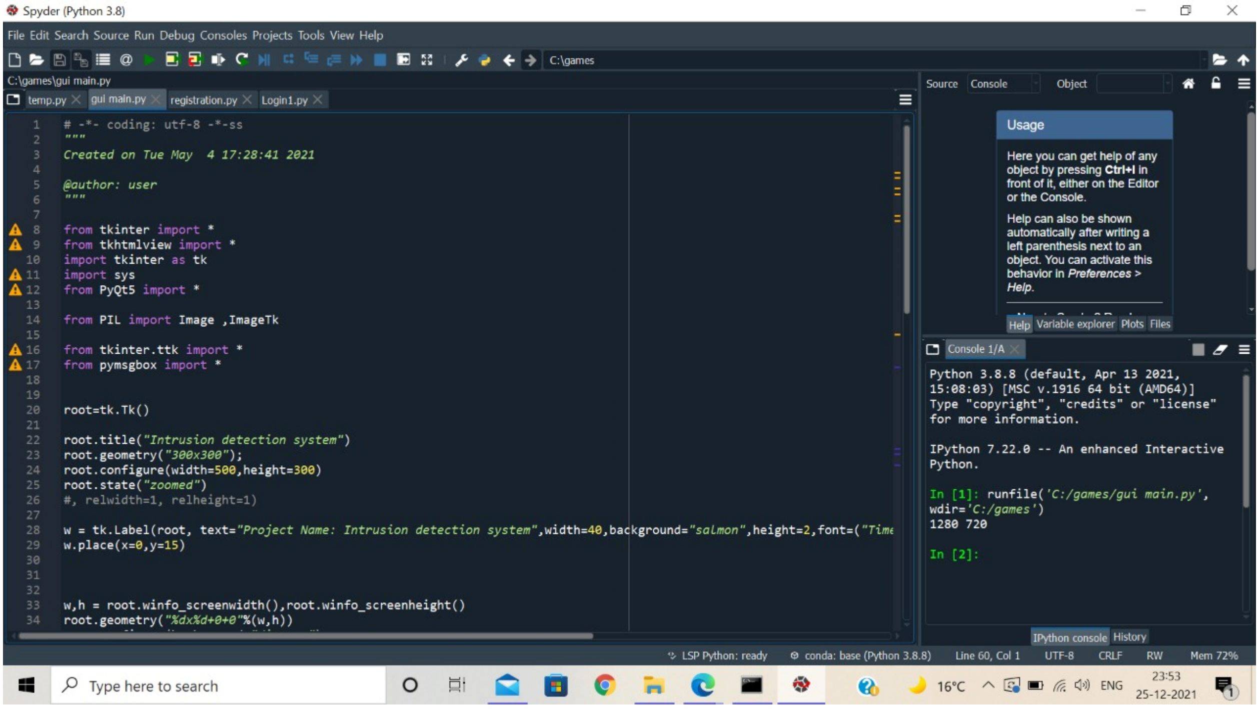


FIGURE 5.8: SEQUENCE DIAGRAM

CHAPTER 6

CODES AND OUTPUT

6.1 Simulation Results



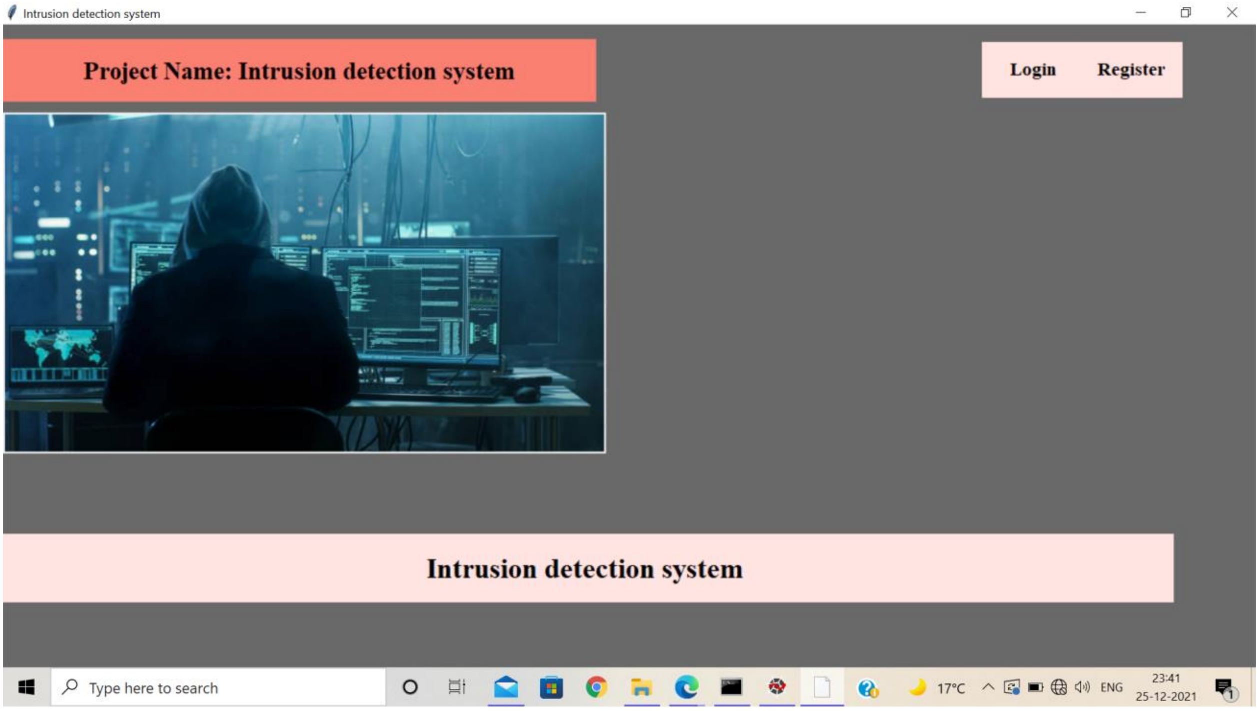
```
# -*- coding: utf-8 -*-ss
"""
Created on Tue May  4 17:28:41 2021

@author: user
"""

from tkinter import *
from tkhtmlview import *
import tkinter as tk
import sys
from PyQt5 import *

from PIL import Image ,ImageTk

from tkinter.ttk import *
from pymsgbox import *


root=tk.Tk()

root.title("Intrusion detection system")
root.geometry("300x300");
root.configure(width=500,height=300)
root.state("zoomed")
#, relwidth=1, relheight=1)

w = tk.Label(root, text="Project Name: Intrusion detection system",width=40,background="salmon",height=2,font=("Time
w.place(x=0,y=15)



w,h = root.winfo_screenwidth(),root.winfo_screenheight()
root.geometry("%dx%d+0+0"%(w,h))
```

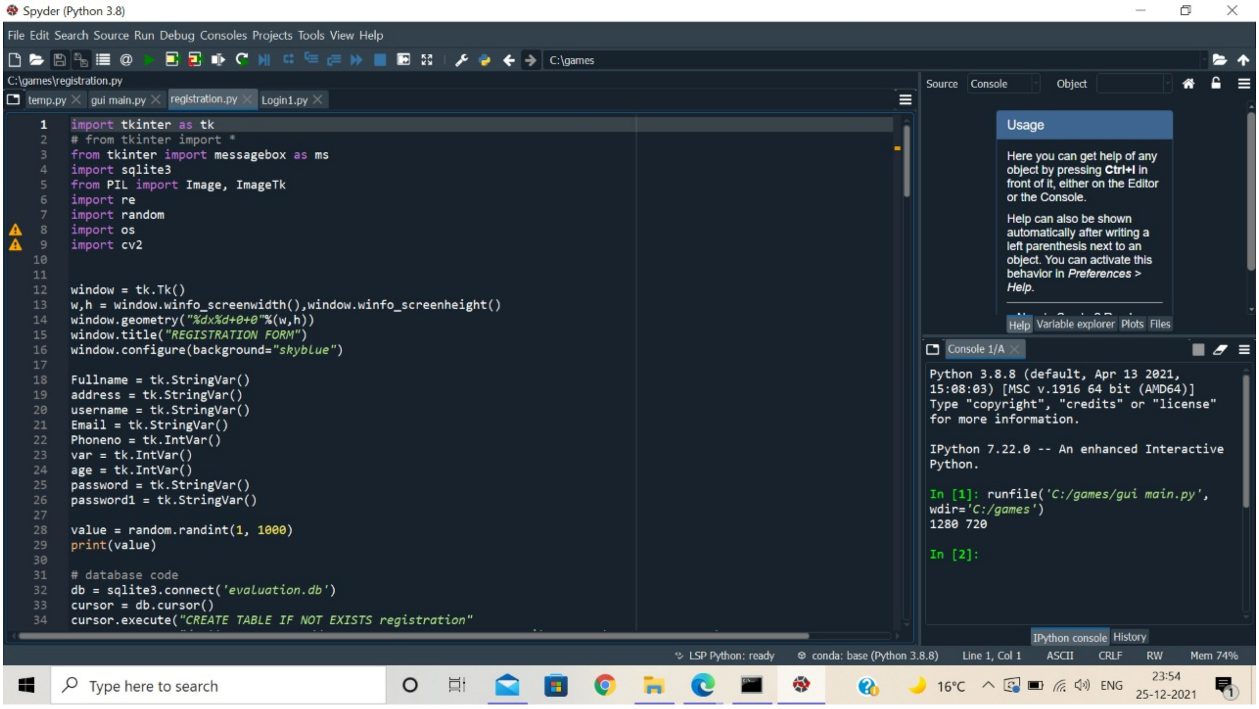
Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

```
Python 3.8.8 (default, Apr 13 2021,
15:08:03) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license"
for more information.

IPython 7.22.0 -- An enhanced Interactive
Python.

In [1]: runfile('C:/games/gui main.py',
wdir='C:/games')
1280 720

In [2]:
```


Intrusion detection system
Project Name: Intrusion detection system
Login
Register
Intrusion detection system
Type here to search
17°C
ENG
23:41
25-12-2021

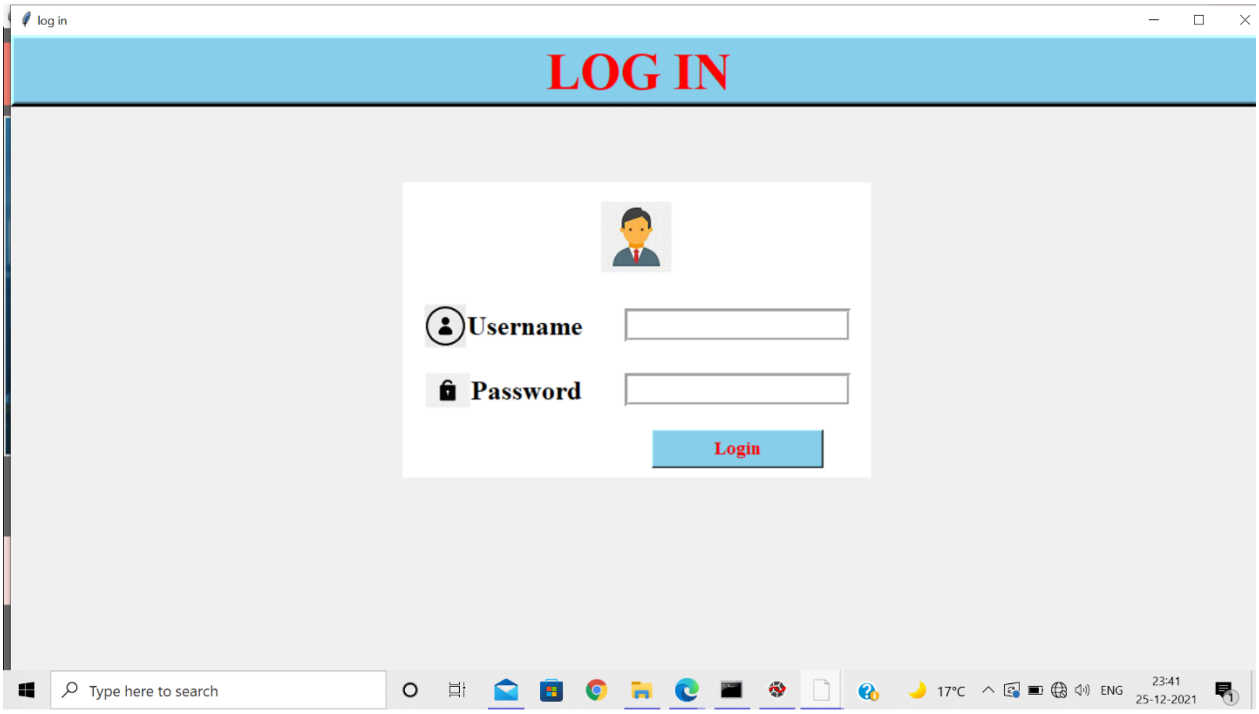


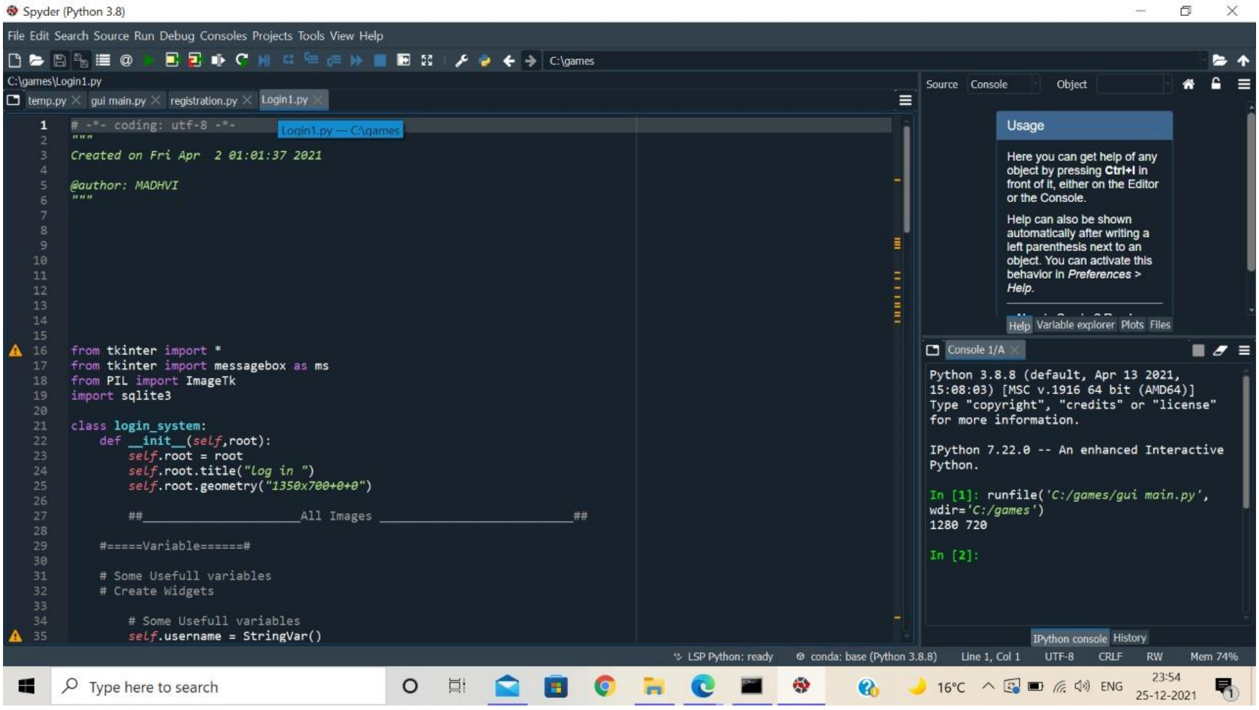
```python
import tkinter as tk
# from tkinter import *
from tkinter import messagebox as ms
import sqlite3
from PIL import Image, ImageTk
import re
import random
import os
import cv2


window = tk.Tk()
w,h = window.winfo_screenwidth(),window.winfo_screenheight()
window.geometry("%dx%d+0+0"%(w,h))
window.title("REGISTRATION FORM")
window.configure(background="skyblue")

Fullname = tk.StringVar()
address = tk.StringVar()
username = tk.StringVar()
Email = tk.StringVar()
Phoneno = tk.IntVar()
var = tk.IntVar()
age = tk.IntVar()
password = tk.StringVar()
password1 = tk.StringVar()

value = random.randint(1, 1000)
print(value)

# database code
db = sqlite3.connect('evaluation.db')
cursor = db.cursor()
cursor.execute("CREATE TABLE IF NOT EXISTS registration"
```

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

```
Python 3.8.8 (default, Apr 13 2021,
15:08:03) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license"
for more information.

IPython 7.22.0 -- An enhanced Interactive
Python.

In [1]: runfile('C:/games/gui main.py',
wdir='C:/games')
1280 720

In [2]:
```





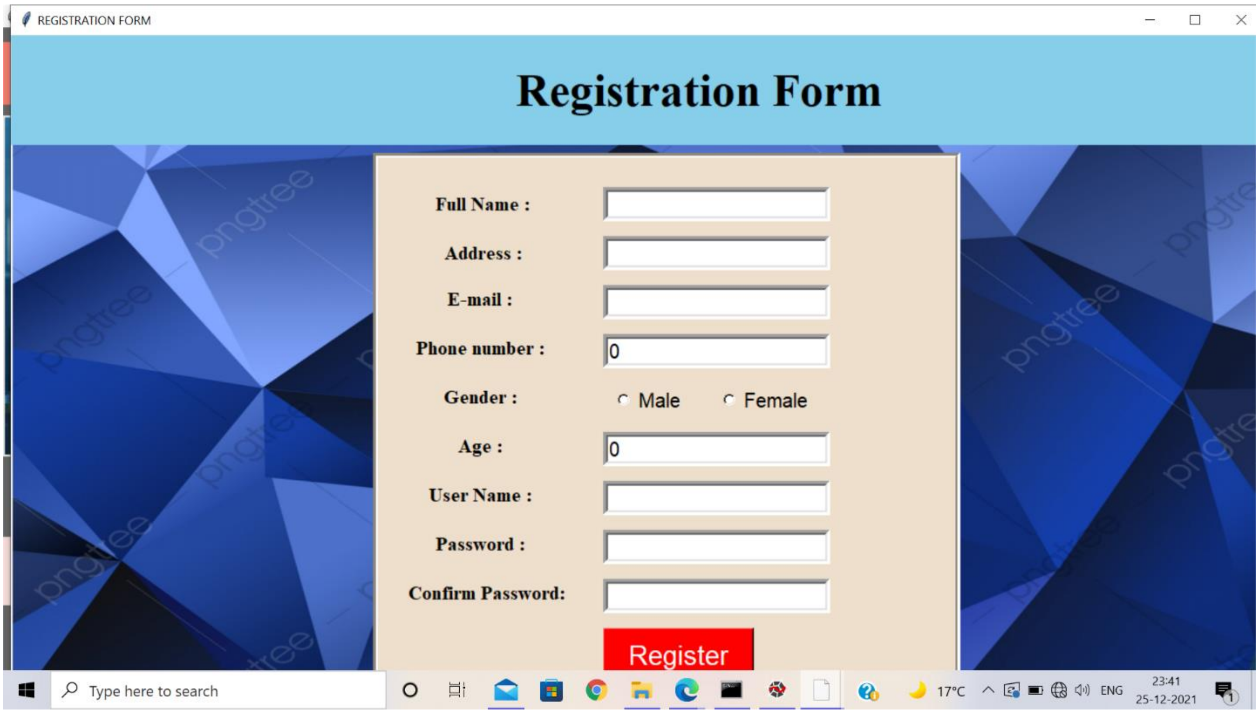
```
# -*- coding: utf-8 -*-
"""
Created on Fri Apr  2 01:01:37 2021

@author: MADHVI
"""









from tkinter import *
from tkinter import messagebox as ms
from PIL import ImageTk
import sqlite3

class login_system:
    def __init__(self,root):
        self.root = root
        self.root.title("Log in ")
        self.root.geometry("1350x700+0+0")

        ##______________________All Images ___________________________##

    #=====Variable======#

    # Some Usefull variables
    # Create Widgets

        # Some Usefull variables
        self.username = StringVar()
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

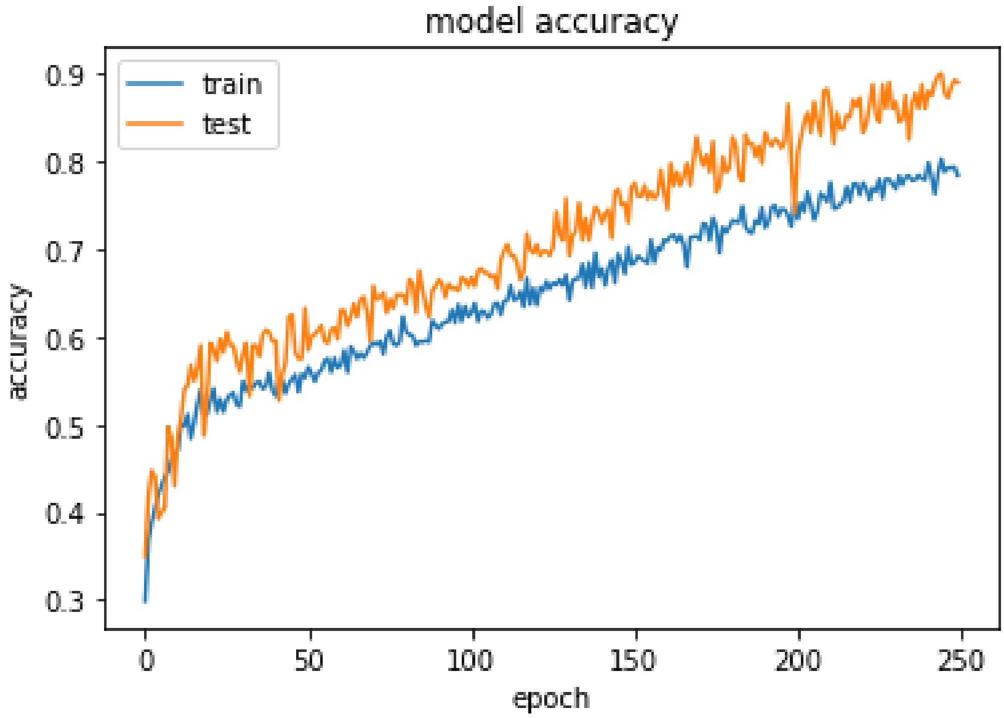
```
Python 3.8.8 (default, Apr 13 2021,
15:08:03) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license"
for more information.

IPython 7.22.0 -- An enhanced Interactive
Python.

In [1]: runfile('C:/games/gui main.py',
wdir='C:/games')
1280 720

In [2]:
```

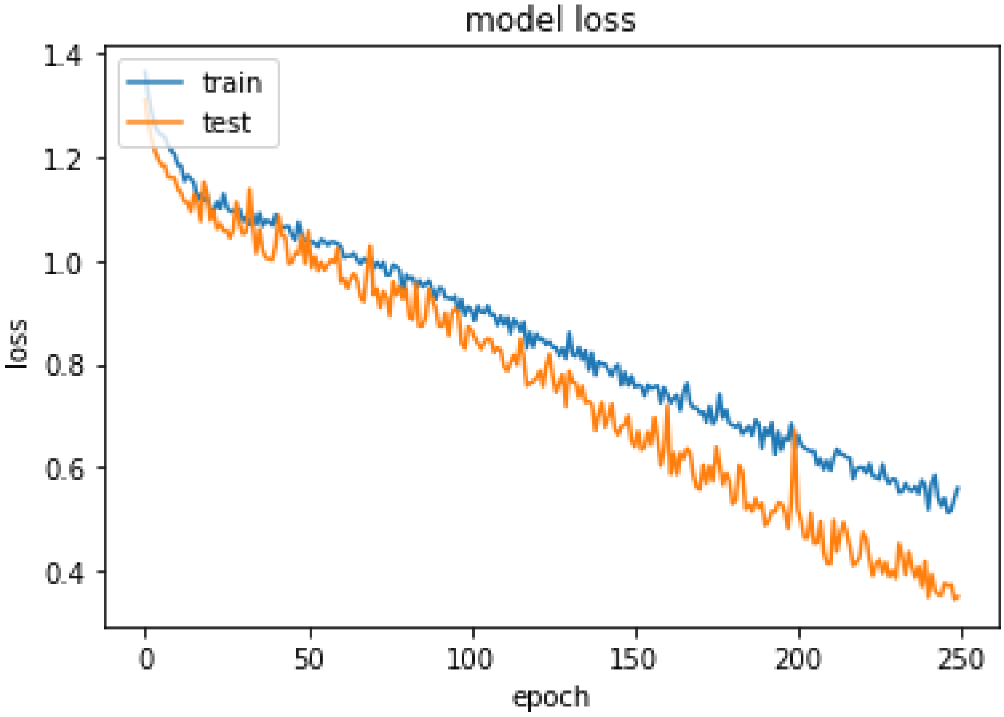


Registration Form

Full Name :	
Address :	
E-mail :	
Phone number :	0
Gender :	○ Male ○ Female
Age :	0
User Name :	
Password :	
Confirm Password:	

6.2 Graphs


model accuracy
accuracy
epoch
train
test
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
50
100
150
200
250


model loss
loss
epoch
train
test
1.4
1.2
1.0
0.8
0.6
0.4
0
50
100
150
200
250

CHAPTER 7

SOFTWARE INFORMATION

7.1 Software Information: Python

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philos- ophy emphasizes code readability with its notable use of significant whitespace. Itslanguage constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple pro- gramming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a ”batteries included” lan- guage due to its comprehensive standard library.

Python was created in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting.

Python 3.0, released in 2008, was a major revision of the language that is notcompletely backward-compatible, and much Python 2 code does not run unmodifiedon Python 3.

The Python 2 language was officially discontinued in 2020 (first planned for2015), and ”Python 2.7.18 is the last Python 2.7 release and therefore the last Python2 release.”[30] No more security patches or other improvements will be released forit. With Python 2's end-of-life, only Python 3.6.x and later are supported.

Python interpreters are available for many operating systems. A global com- munity of programmers develops and maintains CPython, a free and open-source reference implementation. A non-profit organization, the Python Software Founda-tion, manages and directs resources for Python and CPython development.

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde

Informatica (CWI) in the Netherlands as a successor to the ABC lan- guage (itself inspired by SETL), capable of exception handling and interfacing withthe Amoeba operating system. Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12July 2018, when he announced his ”permanent vacation” from his responsibilities as Python’s Benevolent Dictator For Life, a title the Python community bestowed upon him to reflect his long-term commitment as the project’s chief decision-maker.He now shares his leadership as a member of a five-person steering council. In Jan-uary 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry Warsaw, Carol Willing and Van Rossum to a five-member ”Steering Council” to lead the project.

Anaconda:

Anaconda: Anaconda is a free and open-source distribution of the Python andR programming languages for scientific computing (data science, machine learningapplications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintainedby Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012.As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.

Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes onlyconda, Python, the packages they depend on, and a small number of other packages.Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI).

The big difference between conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python datascience and the reason conda exists.

When pip installs a package, it automatically installs any dependent Python packages without checking if these conflict with previously installed package [citationneeded]. It will install a package and any of its dependencies regardless of the state of the existing installation [citation needed]. Because of this, a user with a working installation of, for example, Google Tensorflow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by Tensorflow. In some cases, the package may appear to work but produce different results in detail.

In contrast, conda analyses the current environment including everything currently installed, and, together with any version limitations specified (e.g. the user

may wish to have Tensorflow version 2,0 or higher), works out how to install a compatible set of dependencies, and shows a warning if this cannot be done.

Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or the user's own private repository or mirror, using the conda install command. Anaconda, Inc. compiles and builds the packages available in the Anaconda repository itself, and provides binaries for Windows 32/64bit, Linux 64 bit and MacOS 64-bit. Anything available on PyPI may be installed into a conda environment using pip, and conda will keep track of what it has installeditself and what pip has installed.

Custom packages can be made using the conda build command, and can beshared with others by uploading them to Anaconda Cloud, PyPI or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 in- cludes Python 3.7. However, it is possible to create new environments that include

any version of Python packaged with conda.

CHAPTER 8
APPLICATION

8.1 Applications:

- Selective logging. The IPS only records network activity when it takes action, maintaining the privacy of network users.

- Privacy protection. The IPS compares network traffic against a list of known malicious traffic and does not store or view content.

- Reputation-managed protection. The IPS subscribes to a reputation-based list of known malicious sites and domains, which it uses to proactively protect the university.

- Multiple threat protection. The IPS offers zero-day threat protection, mitigates brute force password attempts, and provides protection against availability threats, such as DDoS and DoS attempts.

CHAPTER 9

FUTURE SCOPE

9.1 Future Scope:

1. Automated monitoring systems: Alongside manual processes, companies can enhance their IT posture by integrating automated threat detection systems.

2. Penetration testing: Penetration testing allows companies to identify vulnerabilities in their systems, networks, and web applications.

CHAPTER 10

CONCLUSION

10.1 CONCLUSION

The industrial Intrusion network based network is rapidly growing in the coming future. The detection of software piracy and malware Intrusion are the main challenges in the field of cybersecurity using Intrusion network-based big data. We proposed a combined Machine learning-based approach for the identification of piratedand malware files. First, the Tensor Flow neural network is proposed to detect the pirated features of original software using software plagiarism. We collected 100 programmers' source codes files from KC99 to investigate the proposed approach. The source code is preprocessed to clean from noise and to capture further the high-quality features which include useful tokens. Then, TFIDF and Log TF weighting techniques are used to zoom the contribution of each feature in terms of source codesimilarity. The weighting values are then used as input to the designed Machine learning approach. Secondly, we proposed a novel methodology based on convolution neural network and color image visualization to detect malware using Intrusionnetwork.

CHAPTER 11

REFERENCES

[1] C. R. Srinivasan, B. Rajesh, P. Saikalyan, K. Premsagar, and E. S. Yadav, “Areview on the different types of INetwork (Intrusion network),” J. Adv. Res. Dyn. Control Syst., vol. 11, no. 1, pp. 154–158, 2019.

[2] G. J. Joyia, R. M. Liaqat, A. Farooq, and S. Rehman, “Internet of Medical Things (IOMT): Applications, benefits and future challenges in healthcare do-main,” J. Commun., vol. 12, no. 4, pp. 240–247, 2017.

[3] Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “INetwork forsmart cities,” IEEE Internet Things J., vol. 1, no. 1, pp. 22–32, Feb. 2014.

[4] E. B. Karbab, M. Debbabi, A. Derhab, and D. Mouheb, “Android malware detection using Machine learning on API method sequences,” Dec. 2017, arXiv:1712.08996. [Online]. Available: https://arxiv.org/abs/1712.08996

[5] S. Jabbar, K. R. Malik, M. Ahmad, O. Aldabbas, M. Asif, S. Khalid, K. Han,and S. H. Ahmed, “A methodology of real-time data fusion for localized big data analytics,” IEEE Access, vol. 6, pp. 24510–24520, 2018.

[6] F. Ullah, J. Wang, M. Farhan, M. Habib, and S. Khalid, “Software plagiarism detection in multiprogramming languages using machine learning approach,” Concurrency Comput., Pract. Exper., to be published.

[7] D.-K. Chae, J. Ha, S.-W. Kim, B. Kang, and E. G. Im, “Software plagiarism detection: A graph-based approach,” in Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage., Nov. 2013, pp. 1577–1580.

[8] Y. Akbulut and O. Dönmez, “Predictors of digital piracy among Turkish un-dergraduate students,” Telematics Inform., vol. 35, no. 5, pp. 1324–1334,

[9] M. ShanmughaSundaram and S. Subramani, “A measurement of similarityto identify identical code clones,” Int. Arab J. Inf. Technol., vol. 12, pp. 735–740,Dec. 2015.

[10] C. Ragkhitwetsagul, “Measuring code similarity in large-scaled code Cor- pora,” in Proc. IEEE Int. Conf. Softw. Maintenance Evol. (ICSME), Oct. 2016, pp. 626–630.