

A
MINOR PROJECT REPORT
ON
ARTIFICIAL INTELLIGENCE CRIME: AN OVERVIEW OF
MALICIOUS USE AND ABUSE OF AI

Submitted in partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

by

GOPU MANISH KUMAR 20P81A6606

Under the guidance of

Mrs. D.SRILATHA

Assistant Professor



VIGNANA BHARATHI ENGINEERING COLLEGE

(Approved by AICTE, Accredited by NAAC & Affiliated to JNTUH & TS SBTET)

Koheda Road, Chinthapalliguda(V), Ibrahimpatnam(M), R.R Dist-501510

2023-2024



VIGNANA BHARATHI ENGINEERING COLLEGE

(Approved by AICTE, Accredited by NAAC & Affiliated to JNTUH & TS SBTET)

Koheda Road, Chinthapalliguda(V), Ibrahimpatnam(M), R.R Dist-501510

2023-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report titled “**ARTIFICIAL INTELLIGENCE CRIME AN OVERVIEW OF MALICIOUS USE AND ABUSE OF AI**” being submitted by **Mr.GOPU MANISH KUMAR (20P81A6606)** in Bachelor’s of Technology IV Year- I Semester **CSE-Artificial Intelligence & Machine Learning** is a record bonafide work carried out by us. The results embodied in this report have not been submitted to any other University for the award of any degree. The results of the investigations enclosed in this report have been verified and found satisfactory.

Mrs. D. SRILATHA
Assistant Professor,
CSE.
VBEC

Mr.P.MANINDER
Head of the Department,
CSE.
VBEC

External Examiner

Dr.C.RAMA SHESHAGIRI RAO
Principal,
VBEC



VIGNANA BHARATHI ENGINEERING COLLEGE

(Approved by AICTE, Accredited by NAAC & Affiliated to JNTUH & TS SBTET)

Koheda Road, Chinthapalliguda(V), Ibrahimpatnam(M), R.R Dist-501510

2023-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

I here by declare that the project report entitled “**ARTIFICIAL INTELLIGENCE CRIME AN OVERVIEW OF MALICIOUS USE AND ABUSE OF AI**” is an original work done and submitted in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science and Engineering-ARTIFICIAL INTELLIGENCE & MACHINE LEARNING** and it is a record of bonafide project work carried out by us under the guidance of **Mrs.D.SRILATHA, Assistant Professor, Department of CSE.**

I further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other Institute or University.

GOPU MANISH KUMAR 20P81A6606

ACKNOWLEDGEMENT

The satisfaction of completing this project would be incomplete without mentioning our gratitude towards all the people who have supported us. Constant guidance and encouragement have been instrumental in the completion of this project.

First and Foremost, we thank the Chairman, Principal, Vice Principal for availing infrastructural facilities to complete the mini project in time.

I offer our sincere gratitude to our guide **Mrs.D.SRILATHA**, Assistant Professor, CSE Department, Vignana Bharathi Engineering College, for her immense support, timely co-operation and valuable advice throughout the course of our project work.

I would like to thank the Head of Department, **Mr. P. MANINDAR**, for his meticulous care and cooperation throughout the project work.

I also owe the gratitude towards **Dr. C. RAMA SESHAGIRI RAO**, Principal of VBEC, for his suggestions and encouragement given to us in completing the project.

I also thank the **Project Review Committee Members, Faculty members and Management of VBEC** for their valuable suggestions.

GOPU MANISH KUMAR 20P81A6606

ABSTRACT

The capabilities of Artificial Intelligence (AI) evolve rapidly and affect almost all sectors of society. AI has been increasingly integrated into criminal and harmful activities, expanding existing vulnerabilities, and introducing new threats. This article reviews the relevant literature, reports, and representative incidents which allows to construct a typology of the malicious use and abuse of systems with AI capabilities.

The main objective is to clarify the **types of activities and corresponding risks**. Our starting point is to identify the vulnerabilities of AI models and outline how malicious actors can abuse them. Subsequently, we explore AI-enabled and AI-enhanced attacks. While we present a comprehensive overview, we do not aim for a conclusive and exhaustive classification. Rather, we provide an overview of the risks of enhanced AI application, that contributes to the growing body of knowledge on the issue. Specifically, we suggest four types of malicious abuse of AI (integrity attacks, unintended AI outcomes, algorithmic trading, membership inference attacks) and four types of malicious use of AI (social engineering, misinformation/fake news, hacking, autonomous weapon systems).

TABLE OF CONTENTS

Chapter No				Page no
1			INTRODUCTION	1-2
	1.1		Motivation	1
	1.2		Problem Statement	2
2			LITERATURE SURVEY	3
3			ANALYSIS	4-7
	3.1		Existing System	4
	3.2		Proposed System	5-6

	3.3		System Requirements	7
		3.3.1	Hardware Requirements	7
		3.3.2	Software Requirements	7
4			SYSTEM DESIGN	7-13
	4.1		UML Description	8
	4.2		UML Diagrams	9-14
5			IMPLEMENTATION	15-29
	5.1		Module Description	15
	5.2		Sample Code	16-24
	5.3		Algorithms Used	25-29

6			SYSTEM STUDY	30-31
7			SOFTWARE ENVIRONMENT	32-38
	7.1		PYTHON	32
	7.2		Advantages of PYTHON	33-34
	7.3		Disadvantages of PYTHON	35
	7.4		Machine Learning	36-37
	7.5		Advantages and Disadvantages of ML	38
8			PACKAGES	39-40
9			TESTING	41-42
10			Output Screens	43-48

11			CONCLUSION	49
12			FUTURE SCOPE	50
			REFERENCES	51

List of Figures		
Figure no	Name of Figure	Page no
1	Use Case Diagram	9
2	Sequence Diagram	10
3	Class Diagram	11
4	Architecture diagram	12
5	Flow charts	13-14
6	Outputs	43-48

1. INTRODUCTION

1.1 MOTIVATION

The impact of systems using Artificial Intelligence (AI) is at the center of numerous academic studies, political debates, and reports of civil society organizations. The development of AI has become the subject of praise due to unprecedented technological capabilities, such as enhanced possibilities for automated image recognition (e.g., detection of cancer in the field of medicine). However, it has also been criticized - even feared - due to aspects such as the uncertain consequences of automation for the labor market (e.g., concerns of mass unemployment). This duality of positive vs negative aspects of the technology can also be identified in the context of cyber-security and cybercrime. Governments use AI to enhance their capabilities, whereas the same technology can be used for attacks against them .

While the recent surge in AI development has been fueled by the private sector and applications in customer-oriented applications, sectors such as defense might use similar capabilities in their operations . At the same time, it is increasingly difficult to distinguish between the actions of state and non-state actors. This has recently been demonstrated by a wave of ransomware attacks targeting public infrastructure in many countries, such as the Colonial Pipeline in the United States in May 2021. Additionally, programs and applications developed for non-malicious purposes can also be implemented or modified for malicious intent and potentially cause harm. The dual-use aspect of technology is not an entirely new problem when it comes to cyber-crime or cyber-security.

Nevertheless, how AI can be leveraged for malicious use and abuse constitutes novel vulnerabilities. Permanent assessment of the threat landscape is crucial to create and adapt governance mechanisms, develop proactive measures, and enhance (cyber)resilience. To build on previous work [14]_[16] and expand the understanding of how AI broadens the potential for malicious activities online, this article evaluates the main categories of use and abuse of AI in a criminal context. We provide several salient examples that allow us to illustrate the challenges at hand. Based on these examples, we present a typology that catalogs the main harmful AI-based activities. Developing knowledge and understanding about the potential malicious use and abuse of AI enables cyber-security organizations and governmental agencies to anticipate such incidents and increase their preparedness against attacks.

1.2 Problem Statement

The malicious use and abuse of Artificial Intelligence (AI) present a pressing challenge in contemporary society. As AI technologies advance, so does the potential for their exploitation in criminal activities. The problem stems from the dual nature of AI, which can be harnessed for both beneficial and malevolent purposes. In recent years, instances of AI-driven cyberattacks, deep fake manipulations, and autonomous systems employed in criminal operations have surged, raising concerns about the integrity of digital ecosystems.

The core issue lies in the ability of malicious actors to leverage AI algorithms for unauthorized access, data breaches, and manipulation of information at an unprecedented scale. This poses a significant threat to individual privacy, national security, and the trustworthiness of AI applications across various sectors. As AI systems become more sophisticated, the challenge of detecting and mitigating malicious activities becomes increasingly complex.

Addressing this problem requires a multifaceted approach involving technological advancements in AI security, policy frameworks, and international collaboration. Striking a balance between fostering AI innovation and safeguarding against its malicious use is imperative to ensure a secure and ethically aligned AI landscape. Efforts to establish robust regulations, ethical guidelines, and proactive security measures are crucial in mitigating the risks associated with the malevolent application of AI technologies

2.LITERATURE SURVEY

The exploration of Artificial Intelligence (AI) in the context of crime has unveiled a spectrum of potential malicious applications. A comprehensive literature survey reveals diverse instances of AI misuse, ranging from cyberattacks powered by machine learning algorithms to deepfake technology enabling identity fraud and misinformation. AI's adaptability poses challenges for law enforcement, as criminals leverage autonomous systems to optimize evasion strategies.

Ethical concerns emerge as AI algorithms exhibit biases, leading to discriminatory outcomes in predictive policing and judicial decision-making. Furthermore, the proliferation of AI-driven tools for surveillance raises privacy apprehensions. Researchers emphasize the need for robust regulations and ethical frameworks to mitigate these risks.

Despite the ominous possibilities, the literature also acknowledges AI's potential in crime prevention, such as using predictive analytics for proactive law enforcement. Striking a balance between harnessing AI for societal benefit and safeguarding against malicious use remains a critical area of investigation. As AI continues to evolve, interdisciplinary efforts are essential to stay ahead of emerging threats and ensure responsible AI deployment in the realm of law and security.

3.ANALYSIS

3.1 EXISITING SYSTEM

To build on previous work and expand the understanding of how AI broadens the potential for malicious activities online, this article evaluates the main categories of use and abuse of AI in a criminal context. We provide several salient examples that allow us to illustrate the challenges at hand. Based on these examples, we present a typology that catalogs the main harmful AI-based activities. Developing knowledge and understanding about the potential malicious use and abuse of AI enables cyber-security organizations and governmental agencies to anticipate such incidents and increase their preparedness against attacks. Furthermore, a typology is greatly useful in structuring research efforts and identifying gaps in knowledge in areas where more research is warranted.

Disadvantages

- An existing methodology not proposed the term "AI-Crime" to describe the situation in which AI technologies are re-oriented to facilitate criminal activity.
- An existing system doesn't implement for MALICIOUS ABUSE OF AI and VULNERABILITIES OF AI MODELS.

3.2 PROPOSED SYSTEM

With the typology presented in this paper, we hope to make the following contributions:

a. Add to the emerging body of knowledge that maps types of malicious use and abuse of AI systems. To understand the main concepts, threat scenarios, and possibilities is necessary to develop much-needed preventive measures and proactive responses to such attacks.

b. Help in establishing a shared language among and across different disciplines, especially between STEM disciplines and legal practitioners, as well as policymakers. Interdisciplinary research on the topic can reduce confusion caused by excessively technical or monodisciplinary language and aid in bridging existing gaps.

c. Propose mitigation strategies, as well as demonstrating that a collective effort among government, academia, and industry is needed.

The methodology is based on an analysis of the available literature on cybercrime and the potential malicious use and abuse of AI systems. A literature review informs this study and findings using the following databases: IEEE Xplore, Science Direct, Wiley Online Library, and Google Scholar. We used keywords, titles, and screened abstracts. The search terms included are (Artificial Intelligence OR AI OR Machine Learning OR ML) AND (malicious OR crime OR harmful OR cyber attack). Additionally, we examined lists of references obtained from reviewed papers and reports, as well as news sources describing past AI incidents. We only reviewed papers/reports/web pages available in English and Portuguese. After analyzing these sources, we were able to identify the different types of malicious use and abuse of AI

systems.

Machine learning (ML) has become more prevalent in recent years. This has created incentives for attackers to manipulate models (e.g., the software itself) or the underlying data, making ML models prone to integrity attacks. In integrity attacks, hackers attempt to inject false information into a system to corrupt the data, undermining their trustworthiness.

Advantages

- The system aims to propose a typology of the malicious use and abuse of AI based on empirical evidence and contemporary discourse, analyzing how AI systems are used to compromise confidentiality, integrity, and data availability.
- Objectives are limited to identifying essential elements of the malicious use and abuse of AI, and to collect evidence of their use in practice. The compiled data enable further analysis of the possible ways in which AI systems can be exploited for criminal activities.

3.3 SYSTEM REQUIREMENTS

3.3.1. HARDWARE REQUIREMENTS:

➤	Processor	:	Pentium - IV
➤	RAM	:	4 GB(min)
➤	Hard Disk	:	20 GB
➤	Key Board	:	Standard Windows Keyboard
➤	Mouse	:	Two or Three Button Mouse
➤	Monitor	:	SVGA

3.3.2 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).

4.SYSTEM DESIGN

4.1 UML DESCRIPTION

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

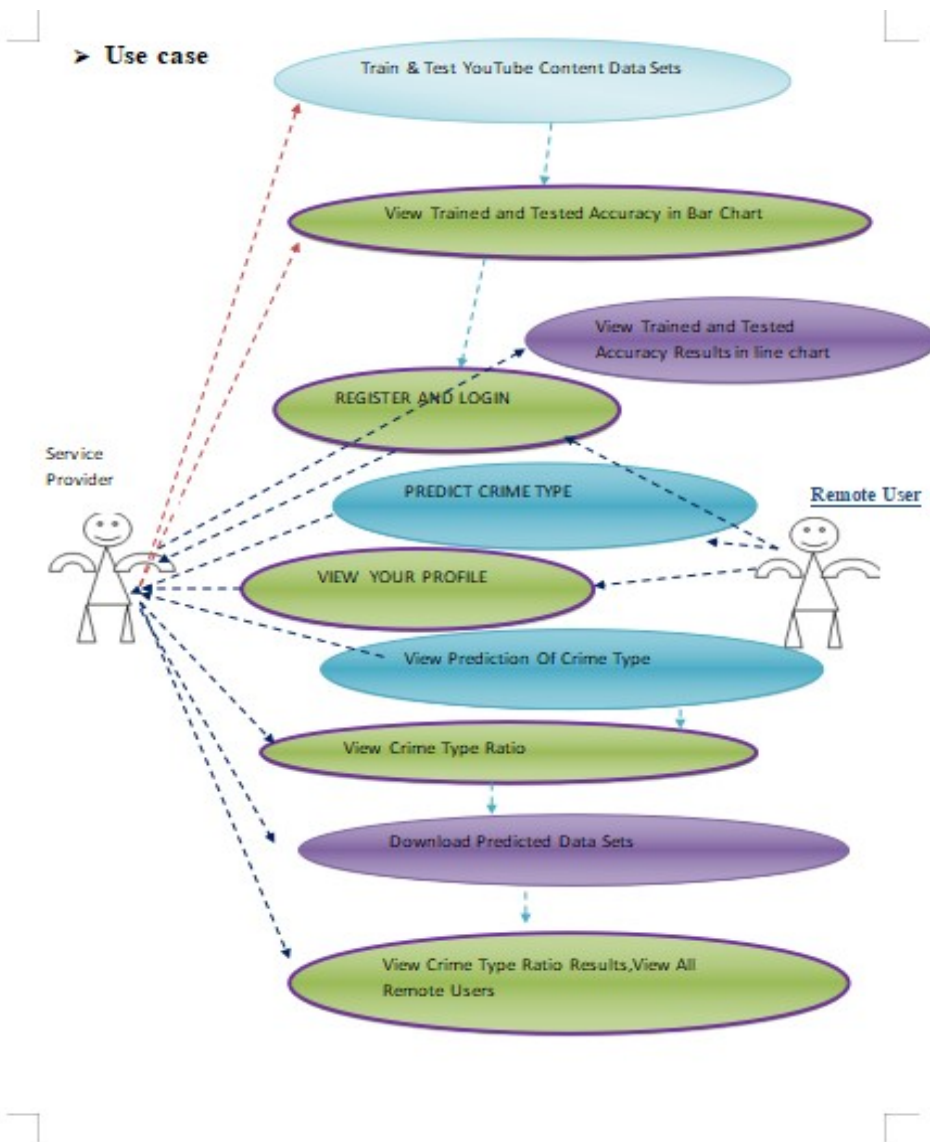
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.

4.2 UML DIAGRAMS

4.2.1.USE CASE DIAGRAM

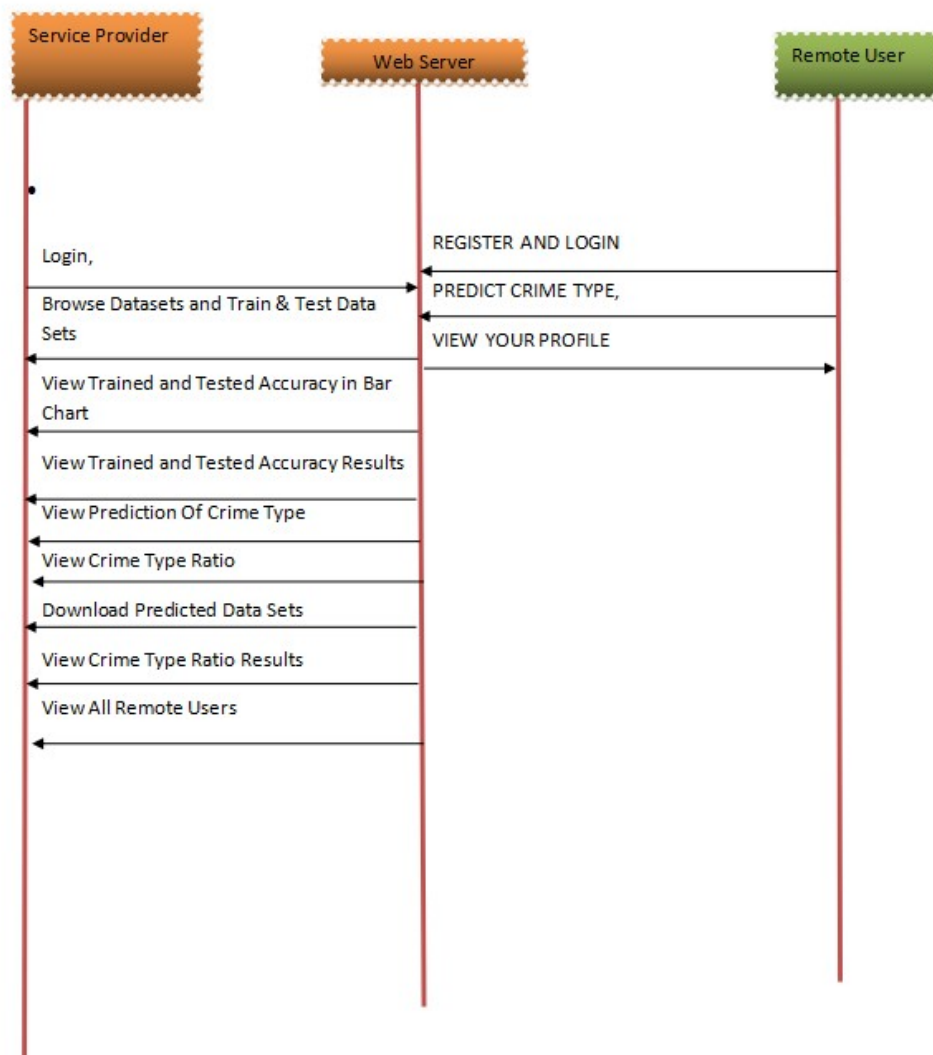
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies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



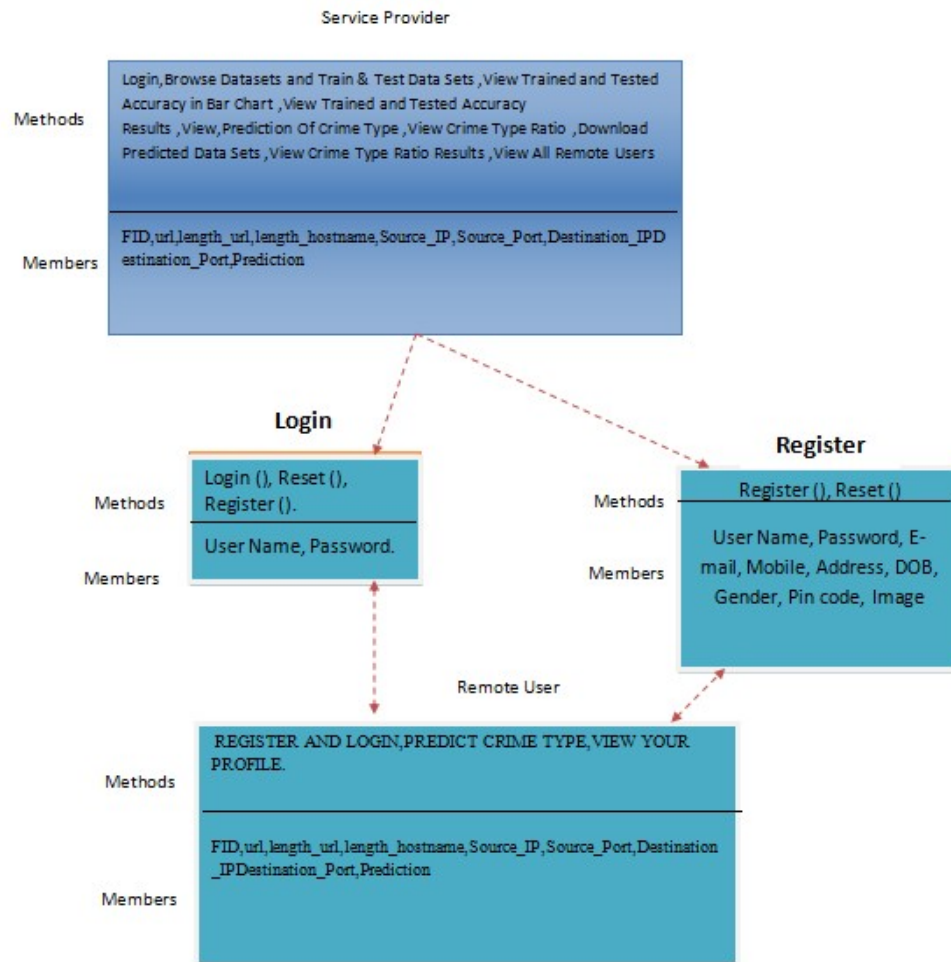
4.2.1. SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

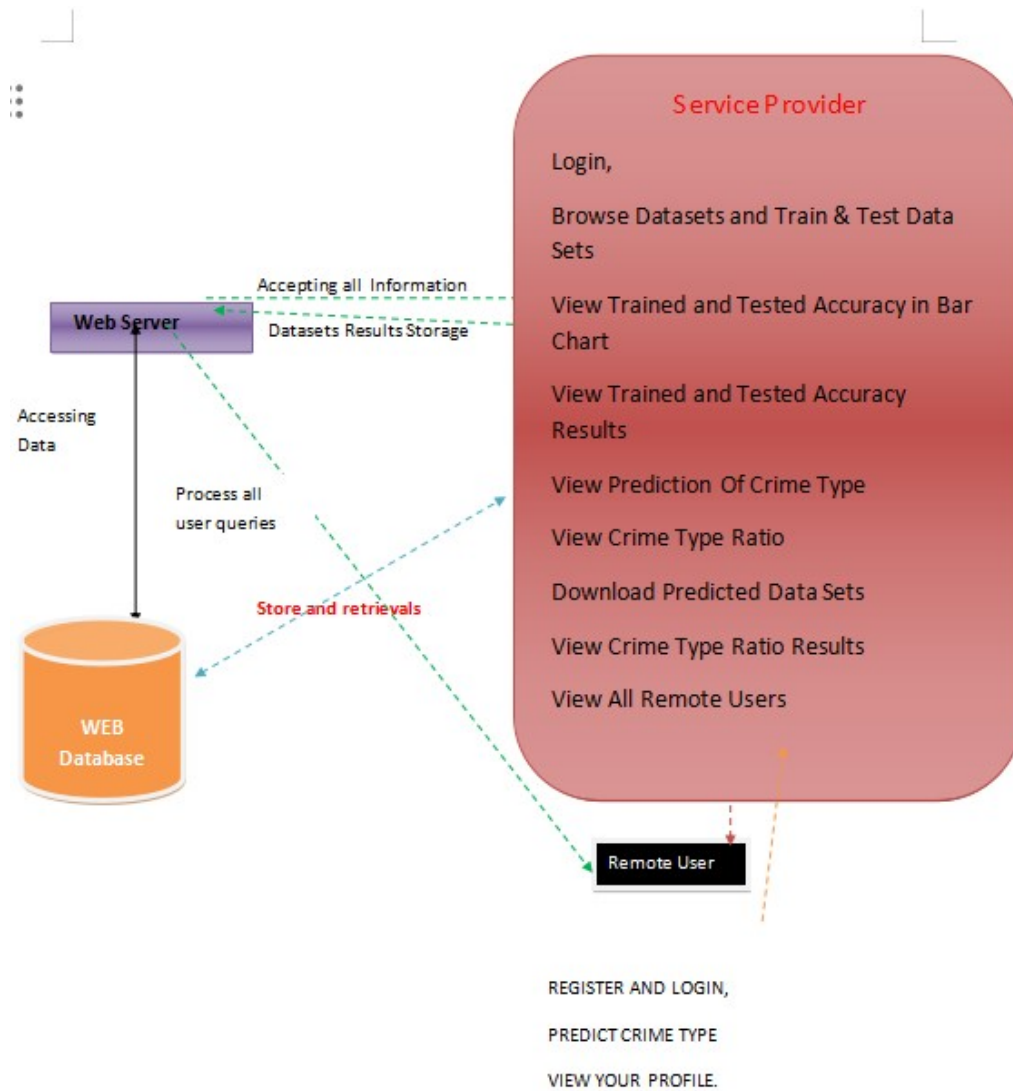
➤ Sequence Diagram



4.2.3 CLASS DIAGRAM

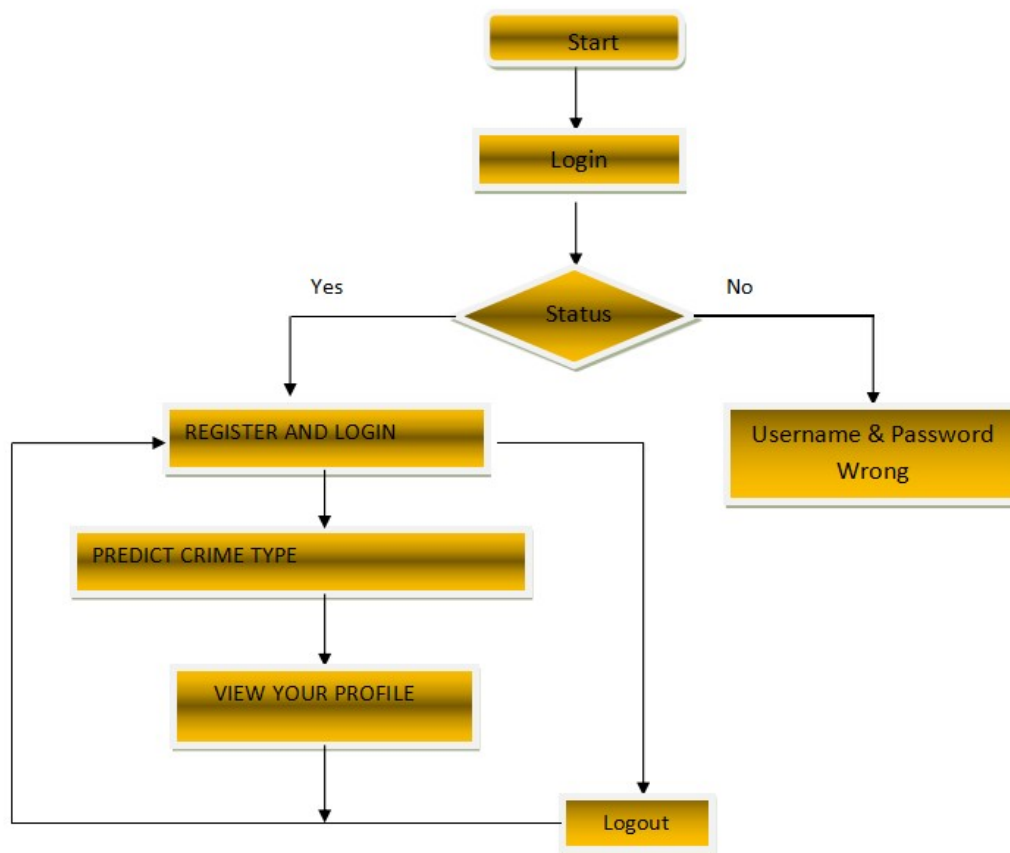


4.2.4 ARCHITECTURE DIAGRAM

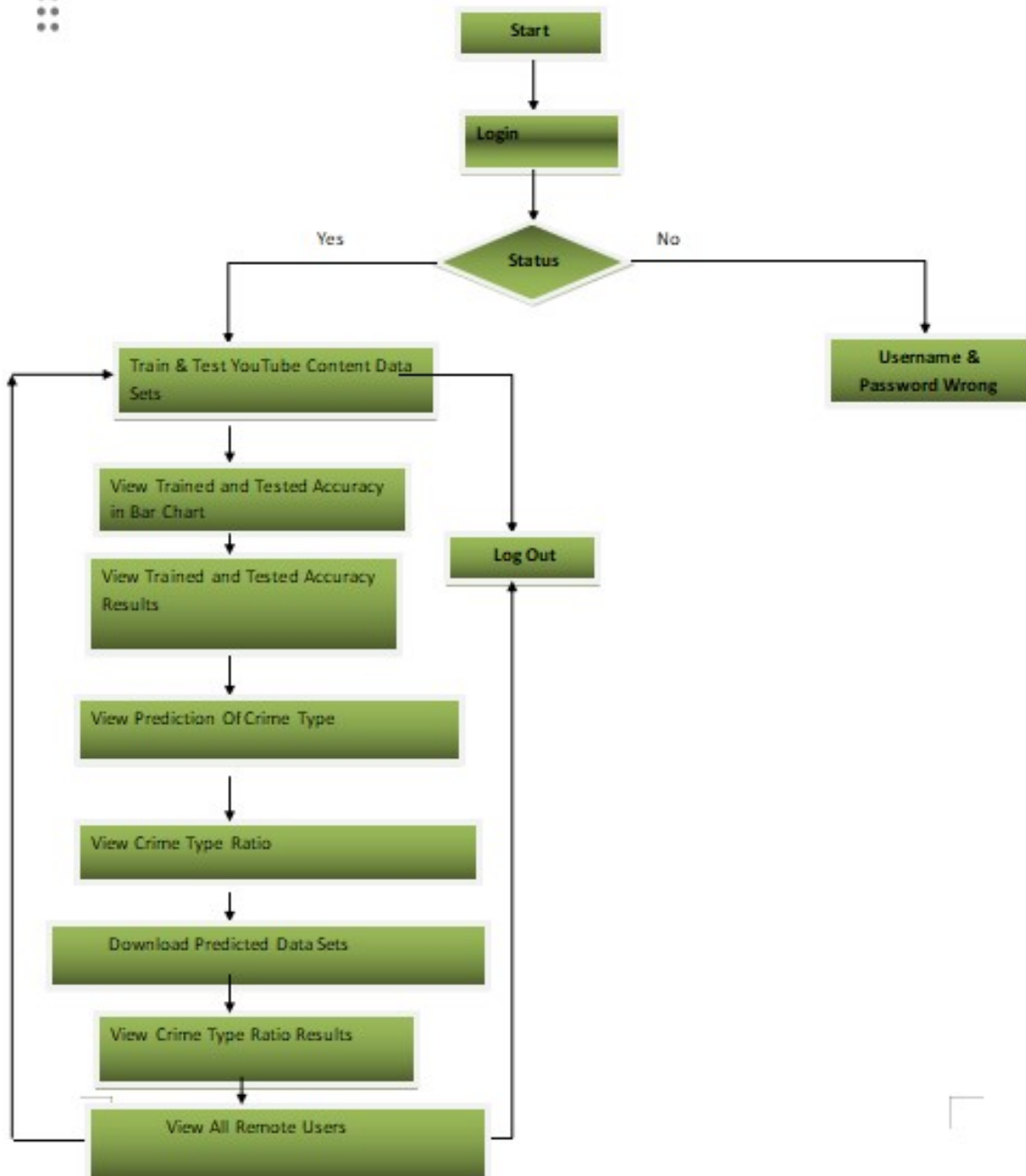


4.2.5 FLOWCHART DIAGRAM

⋮ ➤ Remote User



➤ Service Provider



5.IMPLEMENTATION

5.1 MODULE DESCRIPTION

5.1.1 Service Provider

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as Login,Browse Datasets and Train & Test Data Sets ,View Trained and Tested Accuracy in Bar Chart ,View Trained and Tested Accuracy Results,View,Prediction Of Crime Type ,View Crime Type Ratio ,Download Predicted Data Sets ,View Crime Type Ratio Results ,View All Remote Users

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.

5.1.2 Remote User

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 REGISTER AND LOGIN, PREDICT CRIME TYPE, VIEW YOUR PROFILE.

5.2 SAMPLE CODE

```
from django.db.models import Count

from django.db.models import Q

from django.shortcuts import render, redirect, get_object_or_404


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

from sklearn.ensemble import VotingClassifier

# Create your views here.

from Remote_User.models import ClientRegister_Model, prediction_Of_crime_type, detection_ratio, detection_accuracy

def login(request):

    if request.method == "POST" and 'submit1' in request.POST:

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

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

        try:

            enter = ClientRegister_Model.objects.get(username=username,password=password)
```

```

        request.session["userid"] = enter.id

        return redirect('ViewYourProfile')

    except:

        pass

    return render(request, 'RUser/login.html')

def index(request):

    return render(request, 'RUser/index.html')

def Add_DataSet_Details(request):

    return render(request, 'RUser/Add_DataSet_Details.html', {"excel_data": ""})

def Register1(request):

    if request.method == "POST":

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

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

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

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

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

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

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

```

```

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

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

        ClientRegister_Model.objects.create(username=username,email=email,
password=password,phoneno=phoneno,country=country,state=state,
city=city,address=address,gender=gender)

        obj = "Registered Successfully"

        return render(request, 'RUser/Register1.html',{'object':obj})

    else:

        return render(request,'RUser/Register1.html')

def ViewYourProfile(request):

    userid = request.session['userid']

    obj = ClientRegister_Model.objects.get(id= userid)

    return render(request,'RUser/ViewYourProfile.html',{'object':obj})

def Predict_Crime_Type(request):

    if request.method == "POST":

        if request.method == "POST":

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

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

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

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

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

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

```

```

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

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

df = pd.read_csv('Datasets.csv')

def apply_response(Label):

    if (Label == 0):

        return 0

    elif (Label == 1):

        return 1

    elif (Label == 2):

        return 2

    elif (Label == 3):

        return 3

df['results'] = df['Label'].apply(apply_response)

cv = CountVectorizer()

X = df['url']

y = df['results']

print("Url")

print(X)

print("Results")

print(y)

cv = CountVectorizer()

```

```

X = cv.fit_transform(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("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("ACCURACY")

print(naivebayes)

print("CLASSIFICATION REPORT")

print(classification_report(y_test, predict_nb))

print("CONFUSION MATRIX")

print(confusion_matrix(y_test, predict_nb))

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


# 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("ACCURACY")

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))


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))


print("Decision Tree Classifier")

dtc = DecisionTreeClassifier()

dtc.fit(X_train, y_train)

dtcpredict = dtc.predict(X_test)

print("ACCURACY")

print(accuracy_score(y_test, dtcpredict) * 100)

print("CLASSIFICATION REPORT")

print(classification_report(y_test, dtcpredict))

print("CONFUSION MATRIX")

print(confusion_matrix(y_test, dtcpredict))

models.append(('DecisionTreeClassifier', dtc))


classifier = VotingClassifier(models)

classifier.fit(X_train, y_train)
```

```
y_pred = classifier.predict(X_test)

url1 = [url]

vector1 = cv.transform(url1).toarray()

predict_text = classifier.predict(vector1)

pred = str(predict_text).replace("[", "")

pred1 = pred.replace("]", "")

prediction = int(pred1)

if (prediction == 0):

    val = 'Social Engineering'

elif (prediction == 1):

    val = 'Misinformation'

elif (prediction == 2):

    val = 'Hacking'

elif (prediction == 3):

    val = 'Autonomous weapon systems'
```



```
print(val)

print(pred1)


prediction_Of_crime_type.objects.create(
    FID=FID,
    url=url,
    length_url=length_url,
    length_hostname=length_hostname,
    Source_IP=Source_IP,
    Source_Port=Source_Port,
    Destination_IP=Destination_IP,
    Destination_Port=Destination_Port,
    Prediction=val)


return render(request, 'RUser/Predict_Crime_Type.html',{'objs': val})

return render(request, 'RUser/Predict_Crime_Type.html')
```

5.3 ALGORITHMS USED

5.3.1 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).

While the Naive Bayes classifier is widely used in the research world, it is not widespread among practitioners which want to obtain usable results. On the one hand, the researchers found especially it is very easy to program and implement it, its parameters are easy to estimate, learning is very fast even on very large databases, its accuracy is reasonably good in comparison to the other approaches. On the other hand, the final users do not obtain a model easy to interpret and deploy, they does not understand the interest of such a technique.

Thus, we introduce in a new presentation of the results of the learning process. The classifier is easier to understand, and its deployment is also made easier. In the first part of this tutorial, we present some theoretical aspects of the naive bayes classifier. Then, we implement the approach on a dataset with Tanagra. We compare the obtained results (the parameters of the model) to those obtained with other linear approaches such as the logistic regression, the linear discriminant analysis and the linear SVM. We note that the results are highly consistent. This largely explains the good performance of the method in comparison to others. In the second part, we use various tools on the same dataset (**Weka 3.6.0**, **R 2.9.2**, **Knime 2.1.1**, **Orange 2.0b** and **RapidMiner 4.6.0**). We try above all to understand the obtained results.

5.3.2 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 C_1, C_2, \dots, C_k is as follows:

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

Step 2. Otherwise, let T be some test with possible outcomes O_1, O_2, \dots, O_n . Each object in S has one outcome for T so the test partitions S into subsets S_1, S_2, \dots, S_n where each object in S_i has outcome O_i for T . T becomes the root of the decision tree and for each outcome O_i we build a subsidiary decision tree by invoking the same procedure recursively on the set S_i .

5.3.3 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.

SVM is a discriminant technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter—in contrast to *genetic algorithms (GAs)* or *perceptrons*, both of which are widely used for classification in machine learning. For perceptrons, solutions are highly dependent on the initialization and termination criteria. For a specific kernel that transforms the data from the input space to the feature space, training returns uniquely defined SVM model parameters for a given training set, whereas the perceptron and GA classifier models are different each time training is initialized. The aim of GAs and perceptrons is only to minimize error during training, which will translate into several hyperplanes' meeting this requirement.

5.3.4 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.

Logistic regression competes with discriminant analysis as a method for analyzing categorical-response variables. Many statisticians feel that logistic regression is more versatile and better suited for modeling most situations than is discriminant analysis. This is because logistic regression does not assume that the independent variables are normally distributed, as discriminant analysis does.

This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots. It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables. It provides confidence intervals on predicted values and provides ROC curves to help determine the best cutoff point for classification. It allows you to validate your results by automatically classifying rows that are not used during the analysis

5.3.5 Extra tree classifier

Extra Trees Classifier is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a “forest” to output its classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest. Each Decision Tree in the Extra Trees Forest is constructed from the original training sample. Then, at each test node, Each tree is provided with a random sample of k features from the feature-set from which each decision tree must select the best feature to split the data based on some mathematical criteria (typically the Gini Index). This random sample of features leads to the creation of multiple de-correlated decision trees. To perform feature selection using the above forest structure, during the construction of the forest, for each feature, the normalized total reduction in the mathematical criteria used in the decision of feature of split (Gini Index if the Gini Index is used in the construction of the forest) is computed. This value is called the Gini Importance of the feature. To perform feature selection, each feature is ordered in descending order according to the Gini Importance of each feature and the user selects the top k features according to his/her choice.

Extra Trees Forest for the above data with five decision trees and the value of k which decides the number of features in a random sample of features be two. Here the decision criteria used will be Information Gain. First, we calculate the entropy of the data. Note the formula for calculating the entropy is

$Entropy(S) = \sum_{i=1}^c -p_i \log_2(p_i)$ where c is the number of unique class labels and p_i is the proportion of rows with output label is i. Therefore for the given data, the entropy is:-

$Entropy(S) = \sum_{i=1}^c -p_i \log_2(p_i)$ where c is the number of unique class labels and

p_i is the proportion of rows with output label is i. Therefore for the given data, the entropy is:-

$Entropy(S) = -\frac{9}{14} \log(\frac{9}{14}) - \frac{5}{14} \log(\frac{5}{14})$ [Tex]\rightarrow
 $Entropy(S) = 0.940$ [/Tex] Let the decision trees be constructed such that:-

1st Decision Tree gets data with the features Outlook and Temperature: Note that the formula for Information Gain is:-

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum \{v \in \text{Values}(A)\} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

The Extra Trees Classifier for feature selection offers several advantages:

- **Robust to noise and irrelevant features:** Extra Trees Classifier utilizes multiple decision trees and selects features based on their importance scores, making it less sensitive to noise and irrelevant features. It can effectively handle datasets with a large number of features and noisy data.
- **Computational efficiency:** Extra Trees Classifier constructs decision trees in parallel, which can significantly speed up the training process compared to other feature selection techniques. It is particularly useful for high-dimensional datasets where efficiency is crucial.
- **Bias reduction:** The random selection of subsets and random splitting points in Extra Trees Classifier helps to reduce the bias that can arise from using a single decision tree. By considering multiple decision trees, it provides a more comprehensive evaluation of feature importance.
- **Feature ranking:** Extra Trees Classifier assigns importance scores to each feature, allowing you to rank them based on their relative importance. This ranking can provide insights into the relevance and contribution of each feature to the target variable.
- **Handling multicollinearity:** The Extra Trees Classifier can handle correlated features effectively. By randomly selecting subsets of features and utilizing random splits, it reduces the impact of multicollinearity, unlike methods that rely on explicit feature correlations.
- **Generalization and interpretability:** By selecting a subset of relevant features, Extra Trees Classifier can improve model generalization by reducing overfitting. Additionally, the selected features can provide interpretable insights into the factors that drive predictions and influence the target variable.

6. SYSTEM STUDY

6.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and 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. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY

6.2 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

6.3 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.

6.4 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.

7. SOFTWARE ENVIRONMENT

7.1 PYTHON

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like Google, Amazon, Facebook, Instagram, Dropbox, Uber... etc.

The biggest strength of Python is huge collection of standard library which can be used for the following:

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks.
- Multimedia.

7.2 ADVANTAGES OF PYTHON

Extensive Libraries: Python downloads with an extensive library and it contains code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

Extensible: As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

Embeddable: Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

Improved Productivity: The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

Simple and Easy: When working with Java, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

Readable: Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.

Object-Oriented: This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

Free and Open-Source: Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

Portable: When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA). However, you need to be careful enough not to include any system-dependent features.

Interpreted: Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled language.

7.3 DISADVANTAGES OF PYTHON

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

Speed Limitations: We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

Weak in Mobile Computing and Browsers: While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle. The reason it is not so famous despite the existence of Brython is that it isn't that secure.

Design Restrictions: As you know, Python is dynamically-typed. This means that you don't need to declare the type of variable while writing the code. It uses duck-typing. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

Underdeveloped Database Access Layers: Compared to more widely used technologies like JDBC (Java DataBase Connectivity) and ODBC (Open DataBase Connectivity), Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

7.4 MACHINE LEARNING

Machine learning is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and models that enable computers to learn from and make predictions or decisions based on data. The primary goal of machine learning is to allow computers to automatically learn patterns and improve their performance on a specific task without being explicitly programmed for that task. There are several key concepts and techniques within the field of machine learning:

1. Types of Machine Learning:

Supervised Learning: Involves training a model on a labeled dataset, where the algorithm learns to map input data to corresponding output labels.

Unsupervised Learning: Involves finding patterns and relationships in data without labeled outputs. Clustering and dimensionality reduction are common unsupervised learning tasks.

Reinforcement Learning: Involves training an agent to make decisions in an environment to achieve a goal. The agent receives feedback in the form of rewards or penalties.

2. Algorithms:

Linear Regression: Predicts a continuous output based on input features by fitting a linear equation to the observed data.

Decision Trees: Hierarchical structures that make decisions based on input features.

Random Forest: Ensemble learning method that builds multiple decision trees to improve accuracy and control overfitting.

Support Vector Machines (SVM): Classifies data points by finding the hyperplane that best separates different classes.

Neural Networks: Deep learning models inspired by the structure of the human brain, consisting of layers of interconnected nodes (neurons).

3. Deep Learning:

Deep learning is a subset of machine learning that involves neural networks with multiple layers (deep neural networks). It has shown remarkable success in various domains such as image and speech recognition, natural language processing, and game playing.

4. Data Preprocessing:

Proper preprocessing of data, including cleaning, normalization, and handling missing values, is crucial for the success of machine learning models.

5. Applications:

Machine learning finds applications in various domains, including but not limited to:

- Image and speech recognition
- Natural language processing
- Recommend systems
- Autonomous vehicles
- Health care (diagnosis and personalized medicine)
- Finance (fraud detection and risk assessment)

Machine learning is a dynamic and rapidly evolving field, with ongoing research and advancements contributing to its continued growth and application in diverse industries.

7.5 ADVANTAGES AND DISADVANTAGES OF MACHINE LEARNING

Advantages of Machine Learning:

1. Automation and Efficiency:

ML automates repetitive tasks, saving time and resources. It excels at processing and analyzing large datasets faster and more accurately than human counterparts.

2. Improved Decision Making:

ML models can analyze vast amounts of data to make data-driven predictions and decisions. This can lead to better insights and more informed choices in various domains.

3. Adaptability and Continuous Improvement:

ML models can adapt to new data, allowing them to improve and evolve over time. This adaptability enables systems to stay relevant and effective in dynamic environments.

Disadvantages of Machine Learning:

1. Data Dependency:

ML models heavily rely on quality and quantity of training data. Biased or insufficient data can lead to inaccurate predictions and reinforce existing biases.

2. Lack of Transparency and Interpretability:

Some complex machine learning models, especially in deep learning, are often considered "black boxes" where the decision-making process is not easily interpretable. This lack of transparency raises ethical and trust issues.

3. Overfitting and Generalization Challenges:

Models may overfit the training data, performing well on it but poorly on new, unseen data. Striking the right balance between fitting the training data and generalizing to new data is a common challenge.

8. PACKAGES

1) SkLearn:

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

2)Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

3)Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

4)Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

9. TESTING

9.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

9.2 Integration testing

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

1.Top Down Integration

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2. Bottom-up Integration

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure

9.3 User Acceptance Testing

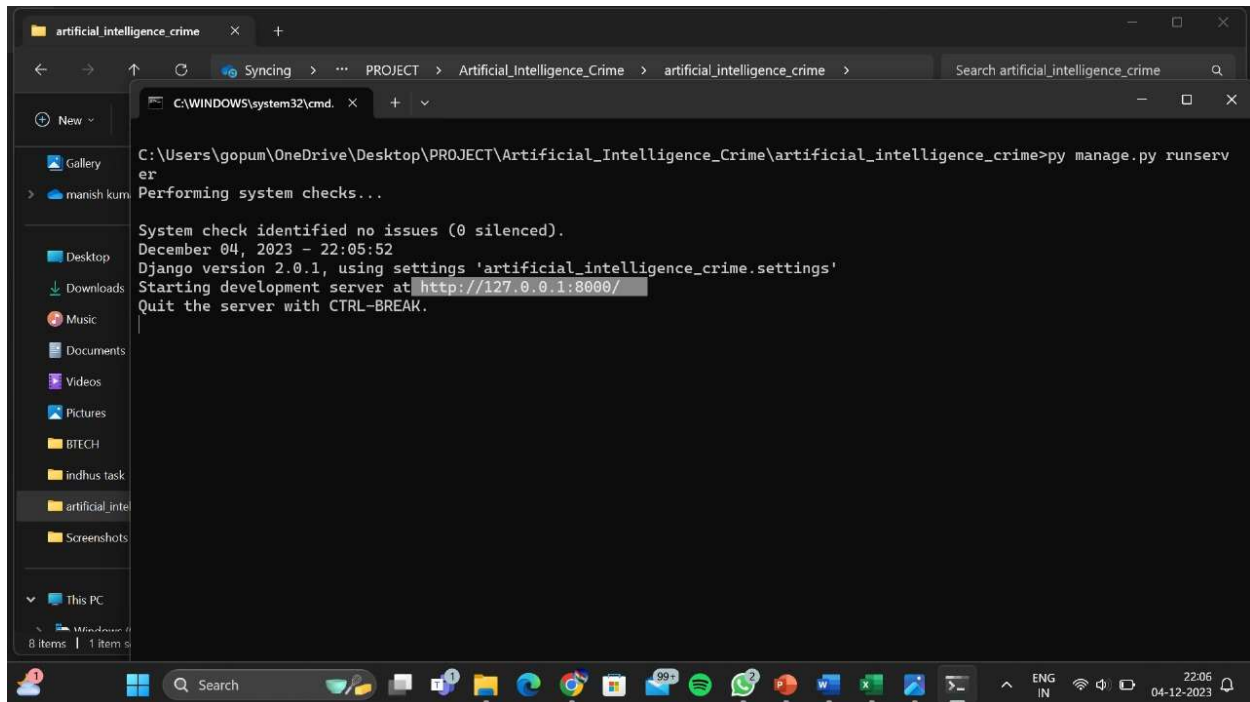
User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

9.4 Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

10. OUTPUT SCREENS

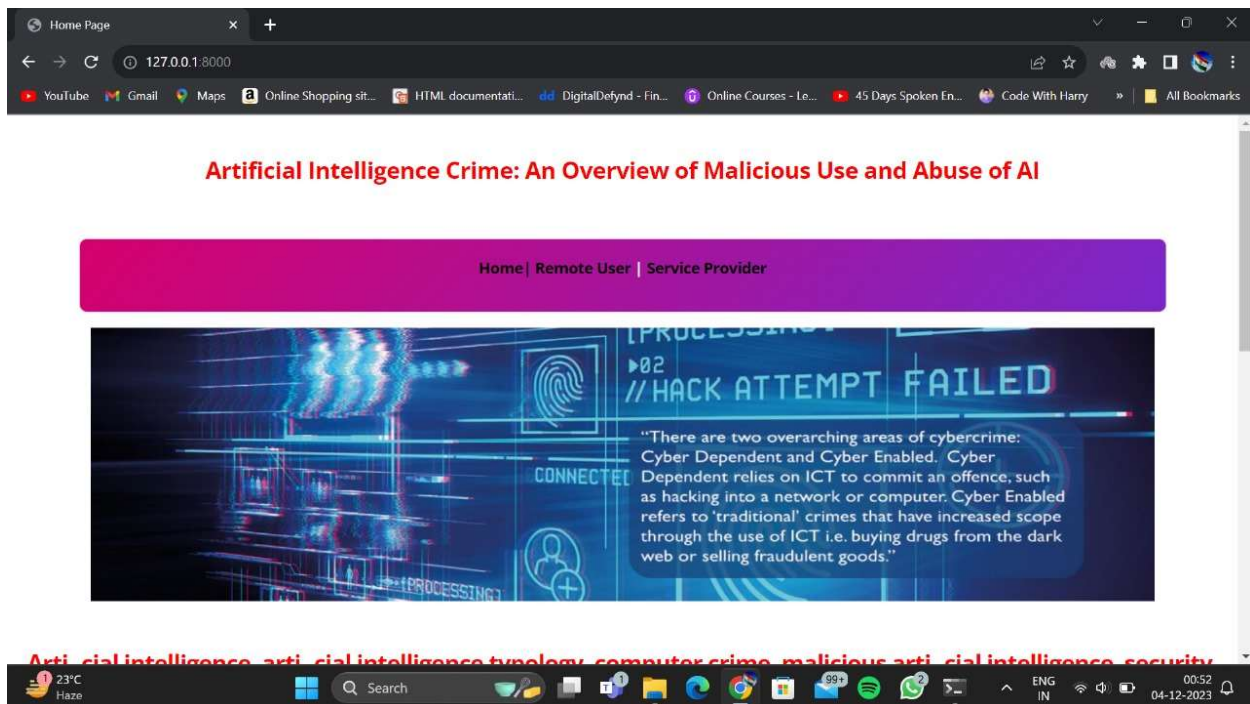
10.1 Clicking on run.bat the Command prompt shows a screen in this way.



```
C:\WINDOWS\system32\cmd. x + v
C:\Users\gopum\OneDrive\Desktop\PROJECT\Artificial_Intelligence_Crime\artificial_intelligence_crime>py manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
December 04, 2023 - 22:05:52
Django version 2.0.1, using settings 'artificial_intelligence_crime.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

10.2 HOME PAGE



10.3 USER REGISTRATION

ChatGPT x Register x +

127.0.0.1:8000/Register1/

YouTube Gmail Maps Online Shopping sit... HTML documentati... DigitalDefynd - Fin... Online Courses - Le... 45 Days Spoken En... Code With Harry All Bookmarks

Arti cial intelligence, arti cial intelligence typology, computer crime, malicious arti cial intelligence, security, social implications of technology.

REGISTER NOW!

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

Windows Search 22:59 04-12-2023

10.4 USER LOGIN

ChatGPT x Login x +

127.0.0.1:8000/login/

YouTube Gmail Maps Online Shopping sit... HTML documentati... DigitalDefynd - Fin... Online Courses - Le... 45 Days Spoken En... Code With Harry All Bookmarks

Arti cial intelligence, arti cial intelligence typology, computer crime, malicious arti cial intelligence, security, social implications of technology.

Login

Login Using Your Account:

User Name

Password

LOGIN

Are You New User !!! REGISTER

Home | Remote User | Service Provider

Windows Search 22:59 04-12-2023

10.5 USER DETAILS

PREDICT CRIME TYPE || VIEW YOUR PROFILE || LOGOUT

YOUR PROFILE DETAILS !!!

Username	manish	Email Id	manish@gmail.com
Mobile Number	1234567890	Gender	Male
Address	hyd	Country	india
State	telangana	City	hyderabad

10.6 PREDICTIVE CRIME TYPE

PREDICTION OF CRIME TYPE !!!

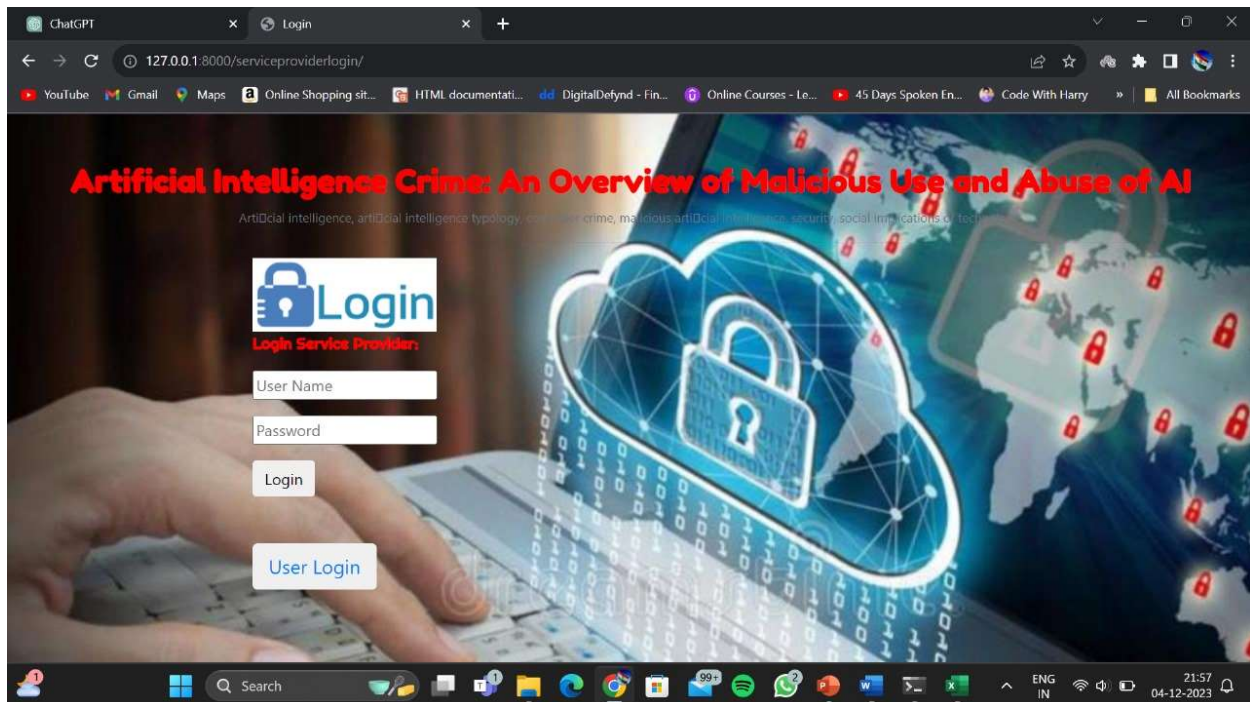
ENTER DATASETS DETAILS HERE !!!

Enter FID	<input type="text"/>	Enter url	<input type="text"/>
Enter length_url	<input type="text"/>	Enter length_hostname	<input type="text"/>
Enter Source_IP	<input type="text"/>	Select Source_Port	<input type="text"/>
Enter Destination_IP	<input type="text"/>	Enter Destination_Port	<input type="text"/>

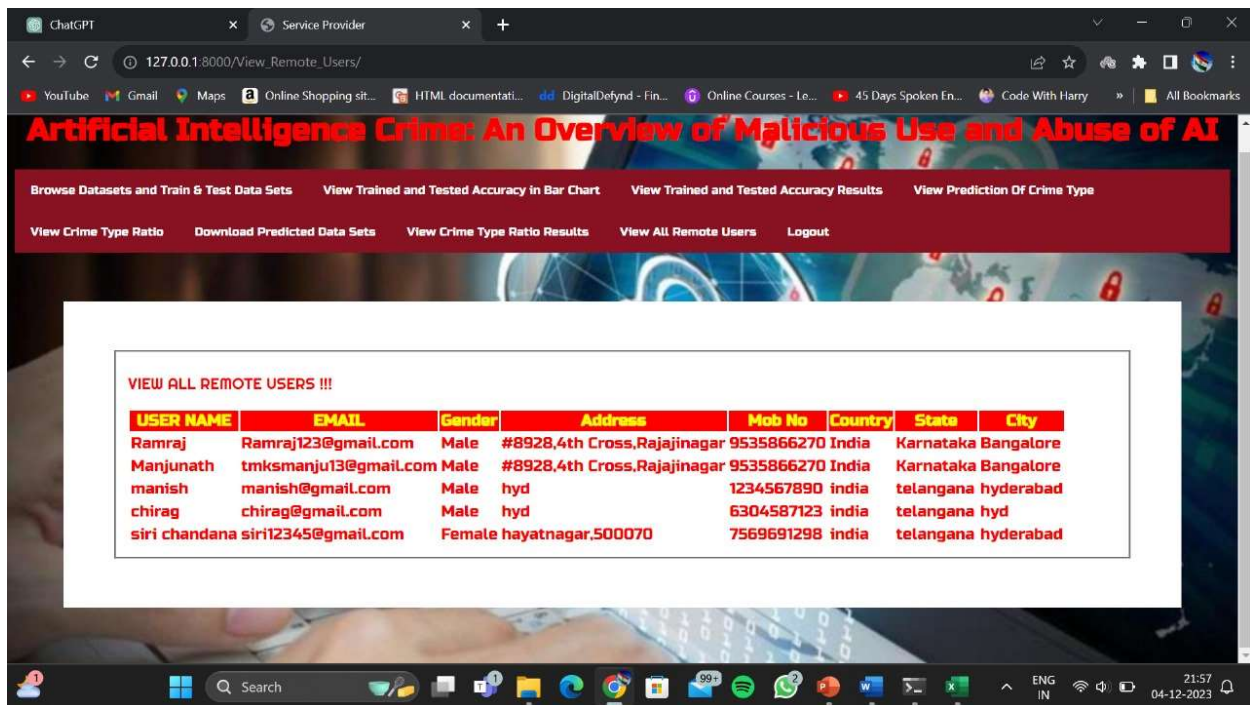
Predict

PREDICT CRIME TYPE DETECTION :

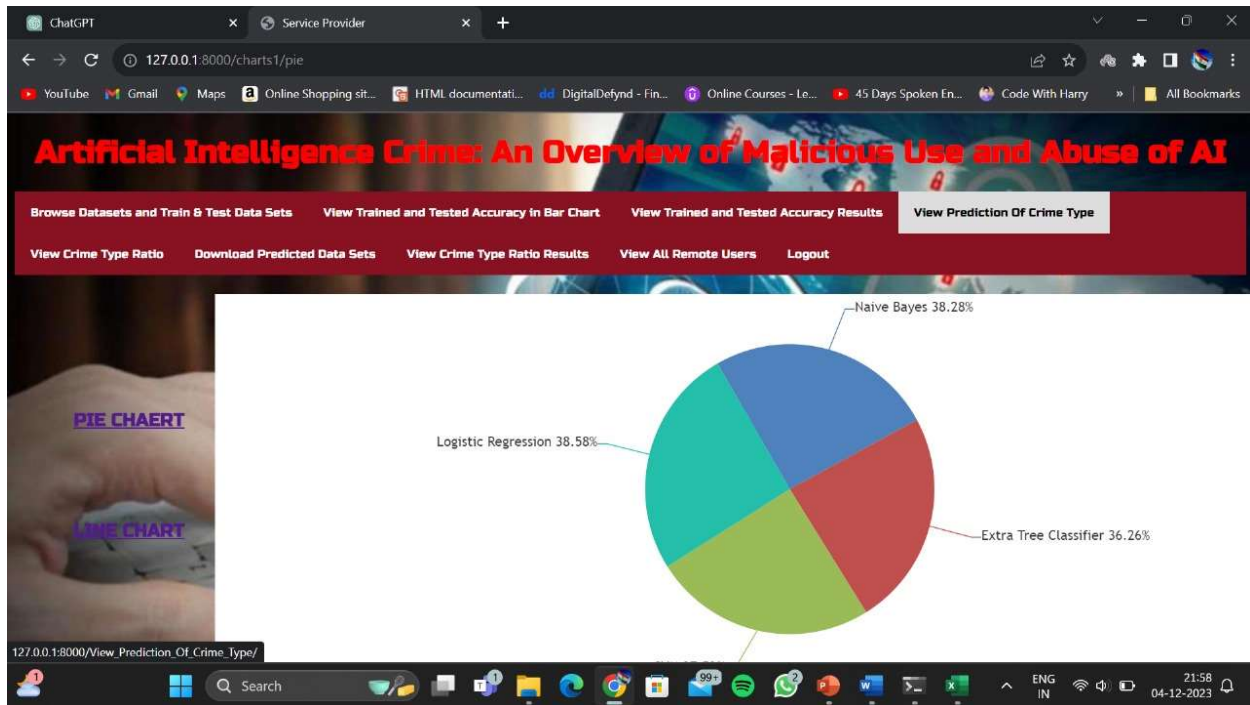
10.7ADMIN_LOGIN



10.8 REMOTE USERS



10.9 Piechart representation of Algorithms



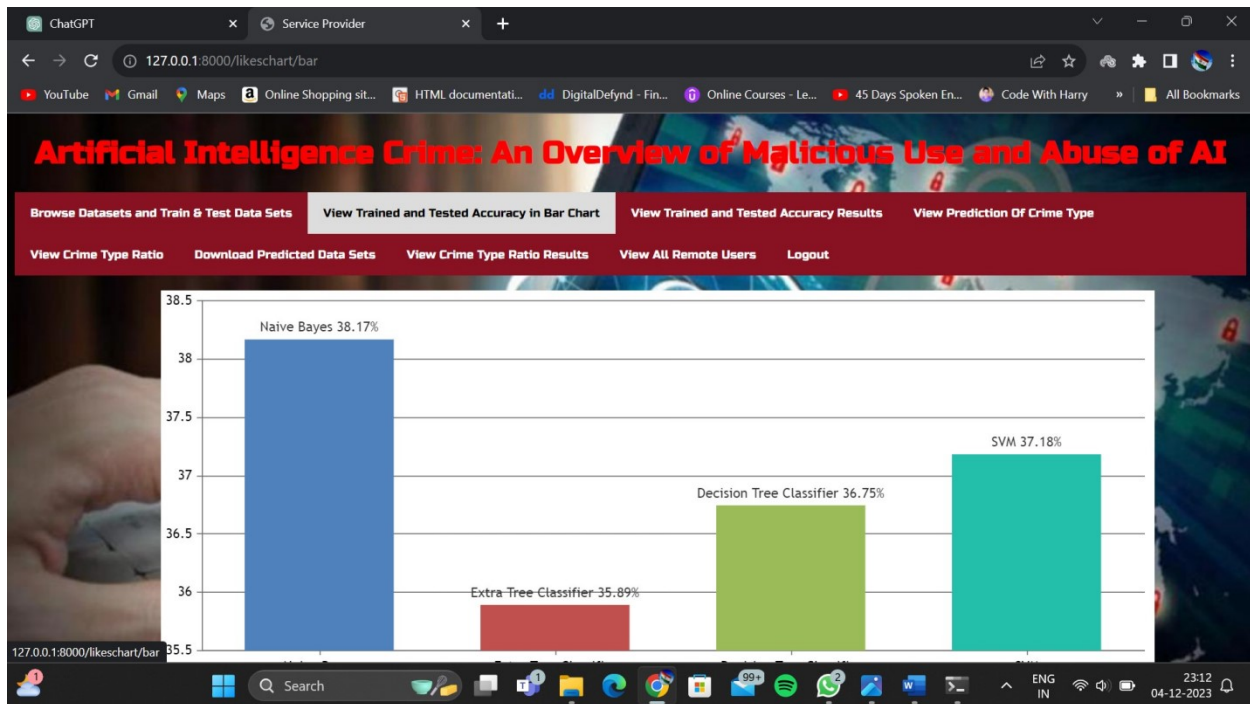
10.10 PREDICTION TYPE DETAILS

View Crime Prediction Type Details !!!

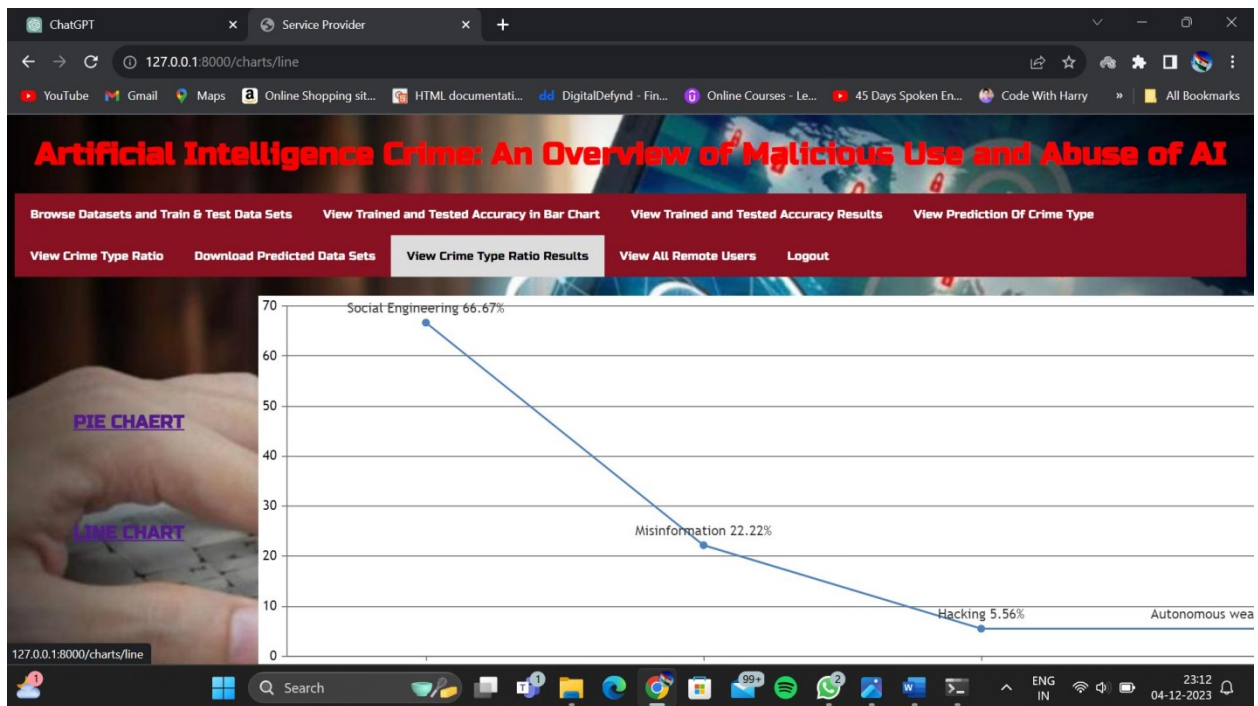
52.84.143.74-51243-80-6	http://sophie-world.com/games/port-and-starboard
172.217.10.78-10.42.0.151-443-57693-6	http://www.ktplasmachinery.com/cs/
172.217.10.74-10.42.0.211-443-33776-6	http://www.game.co.uk/en/games/nintendo-switch/nintendo-switch/
172.217.3.97-10.42.0.42-443-40598-6	http://press-preview.weebly.com

127.0.0.1:8000/View_Prediction_Of_Crime_Type/

10.11. TRAINED_ACCURACY



10.12 CRIME TYPE RESULT RATIOS



11. CONCLUSION

The threats posed by the use and abuse of AI systems must be well understood to create mechanisms that protect society and critical infrastructures from attacks. Based on the available literature, reports, and previous incidents, we focused on creating a classification of how AI systems can be used or abused by malicious actors. This includes, but is not limited to, physical, psychological, political, and economic harm. We explored the vulnerabilities of AI models, such as unintended outcomes, and AI-enabled and AI-enhanced attacks, such as forgery. This article also describes past incidents, such as the 2010 _ash crash and the Cambridge Analytica scandal, manifesting the challenges at hand. We also outlined attacks that, to the best of our knowledge, have only been demonstrated through "proof of concept", such as IBM's DeepLocker. In response to the risks presented in this paper, we have also explored some possible mitigation strategies. Industries, governments, civil society, and individuals should cooperate in developing knowledge and raising awareness while developing technical and operational systems and procedures to address the challenges.

Although this type of classification is a useful starting point, it does not come without drawbacks. Some AI-enabled or AI-enhanced attacks might not fit the categories established. Further work could use empirical methods to assess whether the classification scheme presented is generalizable and representative. When sufficient data is available, methods such as statistical analysis could be helpful to reach a more complete overview of the threat scenario. Continuously mapping the risks associated with malicious use and abuse of AI helps to enhance preparedness and increases the potential to prevent and adequately respond to attacks.

12. FUTURE SCOPE

The malicious use of artificial intelligence (AI) in crime is a growing concern. Currently, AI is exploited for various illicit activities, including cyber attacks, deepfake creation, and automated phishing. Future challenges may involve more sophisticated AI-driven attacks, necessitating advanced cybersecurity measures.

The scope of AI-related crime could expand to include autonomous weapon systems, AI-driven financial fraud, and even manipulation of AI algorithms for illegal purposes. Addressing these challenges requires interdisciplinary collaboration between technologists, policymakers, and law enforcement to develop ethical frameworks, regulations, and countermeasures.

The ongoing development of AI and its integration into various domains underscores the need for proactive efforts to anticipate and mitigate potential risks associated with its malicious use. Public awareness, ethical guidelines, and international cooperation are crucial for managing the evolving landscape of AI-related crime.

REFERENCES

- [1] K. Crawford, *Atlas of AI: Power, Politics, and the Planetary Costs of Arti_cial Intelligence*. London, U.K.: Yale Univ. Press, 2021.
- [2] D. Garcia, "Lethal arti_cial intelligence and change: The future of international peace and security," *Int. Stud. Rev.*, vol. 20, no. 2, pp. 334_341, Jun. 2018, doi: [10.1093/isr/viy029](https://doi.org/10.1093/isr/viy029).
- [3] T. Yigitcanlar, K. Desouza, L. Butler, and F. Roozkhosh, "Contributions and risks of arti_cial intelligence (AI) in building smarter cities: Insights from a systematic review of the literature," *Energies*, vol. 13, no. 6, p. 1473, Mar. 2020, doi: [10.3390/en13061473](https://doi.org/10.3390/en13061473).
- [4] I. van Engelshoven. (Oct. 18, 2019). *Speech by Minister Van Engelshoven on Arti_cial Intelligence at UNESCO, on October the 18th in Paris*. Government of The Netherlands. Accessed: Apr. 15, 2021. [Online]. Available: https://www.government.nl/documents/speeches/2019/10/18/speech-by-minister-van-engelshoven-on-arti_cial-intelligence-atunesco
- [5] O. Osoba and W. Welser IV, *The Risks of Arti_cial Intelligence to Security and the Future of Work*. Santa Monica, CA, USA: RAND Corporation, 2017, doi: [10.7249/PE237](https://doi.org/10.7249/PE237).
- [6] D. Patel, Y. Shah, N. Thakkar, K. Shah, and M. Shah, "Implementation of arti_cial intelligence techniques for cancer detection," *Augmented Hum. Res.*, vol. 5, no. 1, Dec. 2020, doi: [10.1007/s41133-019-0024-3](https://doi.org/10.1007/s41133-019-0024-3).
- [7] A. Rodríguez-Ruiz, E. Krupinski, J.-J. Mordang, K. Schilling, S. H. Heywang-Köbrunner, I. Sechopoulos, and R. M. Mann, "Detection of breast cancer with mammography: Effect of an arti_cial intelligence support system," *Radiology*, vol. 290, no. 2, pp. 305_314, Feb. 2019, doi: [10.1148/radiol.2018181371](https://doi.org/10.1148/radiol.2018181371).
- [8] J. Furman and R. Seamans, "AI and the economy," Nat. Bur. Econ. Res., NBER, Cambridge, MA, USA, Work. Paper, 2018, doi: [10.3386/w24689](https://doi.org/10.3386/w24689).