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). Mapping these threats enables advanced reflection of governance strategies, policies, and activities that can be developed or improved to minimize risks and avoid harmful consequences. Enhanced collaboration among governments, industries, and civil society actors is vital to increase preparedness and resilience against malicious use and abuse of AI.

Machine learning is an important component of the growing field of data science. Through the use of statistical methods, different type of algorithms is trained to make classifications or predictions, and to uncover key insights in this project. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics.

Machine learning algorithms build a model based on this project data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of datasets, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

CONTEXTS

LIST OF THE FIGURES

LIST OF SCREENSHOTS

ABBREVATIONS

AI	ARTIFICIAL INTELLIGENCE
ML	MACHINE LEARNING
HW	HARDWARE
SVGA	SUPER VIDEO GRAPHICS ARRAY
UML	UNIFIED MODELING LANGUAGE
JDBC	JAVA DATABASE CONNECTIVITY
ODBC	OPEN DATABASE CONNECTIVITY

CHAPTER 1
INTRODUCTION

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 cybersecurity 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 cybercrime1 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 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 cybersecurity 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.

CHAPTER 2
LITERATURE SURVEY

1.Author:Brundage,

Title: *The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation* Description: This paper explores the various malicious applications of AI, particularly in areas such as cyberattacks, disinformation, and surveillance. The authors present a framework for forecasting the potential malicious uses of AI and discuss strategies to prevent and mitigate its harmful effects. The paper emphasizes the importance of collaboration between researchers, policymakers, and industry leaders to develop preventive measures and safeguards against AI misuse.

2.Author:Horvitz

Title: *Artificial Intelligence as a Tool for Harmful Use in Cybersecurity* Description: This paper examines the use of AI in enhancing cybersecurity threats, including automated cyberattacks, phishing, and the generation of malware. The authors focus on the dual-use nature of AI, where advancements that improve cybersecurity can also be leveraged by cybercriminals to exploit vulnerabilities. The paper discusses potential countermeasures to safeguard digital systems against AI-driven cybercrime.

3.Author:Goodfellow

Title: *Explaining and Harnessing Adversarial Examples* Description: In this influential work, the authors explore how AI systems, particularly deep learning models, are vulnerable to adversarial attacks. These attacks involve feeding AI models with malicious inputs designed to mislead or confuse them. The study highlights the potential risks of adversarial examples in AI-driven systems and emphasizes the need for better defences to protect against these kinds of malicious manipulations in security-critical applications.

4.Author:Zhang

Title: *Deepfake's and Beyond: A Survey of Fake Media Detection* Description: This paper provides an overview of the technology behind deepfakes, AI-generated media that can be used for disinformation campaigns, fraud, and reputation damage. The authors review various techniques for detecting deepfakes and other forms of AI-generated fake media, highlighting the challenges of distinguishing between real and synthetic content. The paper also discusses the societal impact of deepfakes and the ethical implications of AI's role in misinformation.

5.Author:Dastin,J.

Title: AI used in the creation of autonomous weapons Description: This article discusses the growing concerns regarding the use of AI in autonomous weapons systems. The paper focuses on the potential dangers of AI-controlled military technologies, such as drones and robots, that can be deployed in warfare or conflict zones. The author outlines the ethical dilemmas associated with autonomous weapons, including accountability for their actions and the risk of their malicious use by state or non-state actors.

CHAPTER 3

EXISTING 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 cybersecurity 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.

CHAPTER 4

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.

CHAPTER 5
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.

CHAPTER 6

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.

CHAPTER 7

SYSTEM ARCHITECTURE

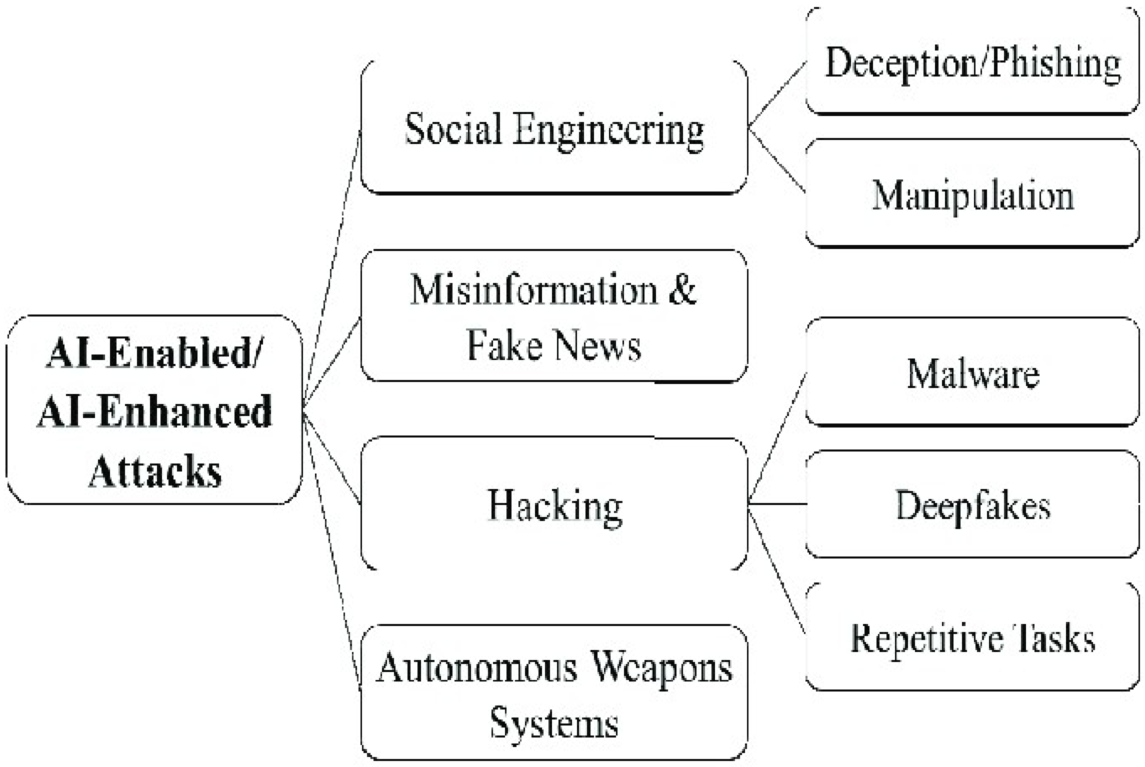


FIG_1: System Architecture

CHAPTER 8

SYSTEM REQUIREMENTS

➢ **H/W System Configuration**: - The Hardware System Configuration for the Artificial Intelligence-based project analysing the malicious use and abuse of AI should be designed to support both the computation-heavy requirements of AI and the storage needs of handling large datasets. Below is a typical hardware configuration that can be used for such projects:

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

CHAPTER 9
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).

CHAPTER 10

SYSTEM STUDY AND 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

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

Economic 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.

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.

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.

CHAPTER 11

PRELIMINARY INVESTIGATION

The first and foremost strategy for development of a project starts from the thought of designing a mail enabled platform for a small firm in which it is easy and convenient of sending and receiving messages, there is a search engine ,address book and also including some entertaining games. When it is approved by the organization and our project guide the first activity, i.e.. preliminary investigation begins. The activity has three parts:

- Request Clarification
- Feasibility Study
- Request Approval

11.1 Request Clarification

After the approval of the request to the organization and project guide, with an investigation being considered, the project request must be examined to determine precisely what the system requires.

Here our project is basically meant for users within the company whose systems can be interconnected by the Local Area Network (LAN). In today's busy schedule man need everything should be provided in a readymade manner. So taking into consideration of the vastly use of the net in day to day life, the corresponding development of the portal came into existence.

11.2 Feasibility Analysis

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibilities that have to be analyzed are

- Operational Feasibility
- Economic Feasibility
- Technical Feasibility

Operational Feasibility

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the Admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

Economic Feasibility

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer-based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. Since the system is a network based, any number of employees connected to the LAN within that organization can use this tool from at any time. The Virtual Private Network is to be developed using the existing resources of the organization. So the project is economically feasible.

Technical Feasibility

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The organization needs IBM compatible machines with a graphical web browser connected to the Internet and Intranet. The system is developed for platform independent environment. Java Server Pages, JavaScript, HTML, SQL server and WebLogic Server are used to develop the system. The technical feasibility has been carried out. The system is technically feasible for development and can be developed with the existing facility.

11.3 Request Approval

Not all request projects are desirable or feasible. Some organization receives so many project requests from client users that only few of them are pursued. However, those projects that are both feasible and desirable should be put into schedule. After a project request is approved, it cost, priority, completion time and personnel requirement is estimated and used to determine where to add it to any project list. Truly speaking, the approval of those above factors, development works can be launched.

CHAPTER 12

SYSTEM DESIGN AND DEVELOPMENT

Input Design

Input Design plays a vital role in the life cycle of software development, it requires very careful attention of developers. The input design is to feed data to the application as accurate as possible. So inputs are supposed to be designed effectively so that the errors occurring while feeding are minimized. According to Software Engineering Concepts, the input forms or screens are designed to provide to have a validation control over the input limit, range and other related validations. This system has input screens in almost all the modules. Error messages are developed to alert the user whenever he commits some mistakes and guides him in the right way so that invalid entries are not made. Let us see deeply about this under module design.

Input design is the process of converting the user created input into a computer-based format. The goal of the input design is to make the data entry logical and free from errors. The error is in the input are controlled by the input design. The application has been developed in user-friendly manner. The forms have been designed in such a way during the processing the cursor is placed in the position where must be entered. The user is also provided with in an option to select an appropriate input from various alternatives related to the field in certain cases. Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page.

Output Design

The Output from the computer is required to mainly create an efficient method of communication within the company primarily among the project leader and his team members, in other words, the administrator and the clients. The output of VPN is the system which allows the project leader to manage his clients in terms of creating new clients and assigning new projects to them, maintaining a record of the project validity and providing folder level access to each client on the user side depending on the projects allotted to him. After completion of a project, a new project may be assigned to the client. User authentication procedures are maintained at the initial stages itself. A new user may be created by the administrator himself or a user can himself register as a new user but the task of assigning projects and validating a new user reset with the administrator only.

The application starts running when it is executed for the first time. The server has to be started and then the internet explorer in used as the browser. The project will run on the local area network so the server machine will serve as the administrator while the other connected systems can act as the clients. The developed system is highly user friendly and can be easily understood by anyone using it even for the first time.

System Design uml Diagrams

System Design along with UML Diagrams for your project on Artificial Intelligence and Malicious Use Detection.

System Design Overview

The AI Malicious Use Detection System is designed to monitor and identify the potential misuse of AI technologies such as deepfakes, AI-enabled cyberattacks, disinformation, and more. It comprises several modules to process user data, detect misuse patterns, generate alerts, and store/report data for analysis.

Key Components of the System:

1. User Interface (UI): Provides interaction with the users.
2. AI Model for Malicious Use Detection: Uses machine learning algorithms to detect suspicious AI use.
3. Database Management System: Stores user, threat, and system-related data.
4. External APIs: Integrates with external data sources like weather, news, and market data.
5. Reporting and Analytics Module: Allows admins to generate reports and analyse trends.
6. Admin Panel: Allows administrators to configure the system, review reports, and analyse user data.

High-Level System Architecture

The system can be divided into several layers:

1. Frontend (Client Layer): This is the user interface where users can interact with the system.
2. Backend (Server Layer): This layer handles processing, detection algorithms, and data handling.
3. Database Layer: Stores structured data, logs, and results from the AI detection engine.
4. External Data Layer: Fetches data from APIs like weather and market data

UML Diagrams

a. Use Case Diagram

for the Artificial Intelligence and Malicious Use Detection System:

Use Case Diagram Overview

The Use Case Diagram represents the interactions between users (actors) and the system's functions (use cases). In this system, we have two main actors:

- **User**: Can view recommendations and report malicious use.
- **Admin**: Can analyse data and configure the system.

Key Use Cases:

1. **View Recommendations:** The user can view AI-related recommendations based on the input data.
2. **Report Malicious Use:** The user can report any suspicious AI use.
3. **Analyse Data:** The admin can analyse data to identify malicious use patterns.
4. **Configure System:** The admin can configure system settings and parameters.

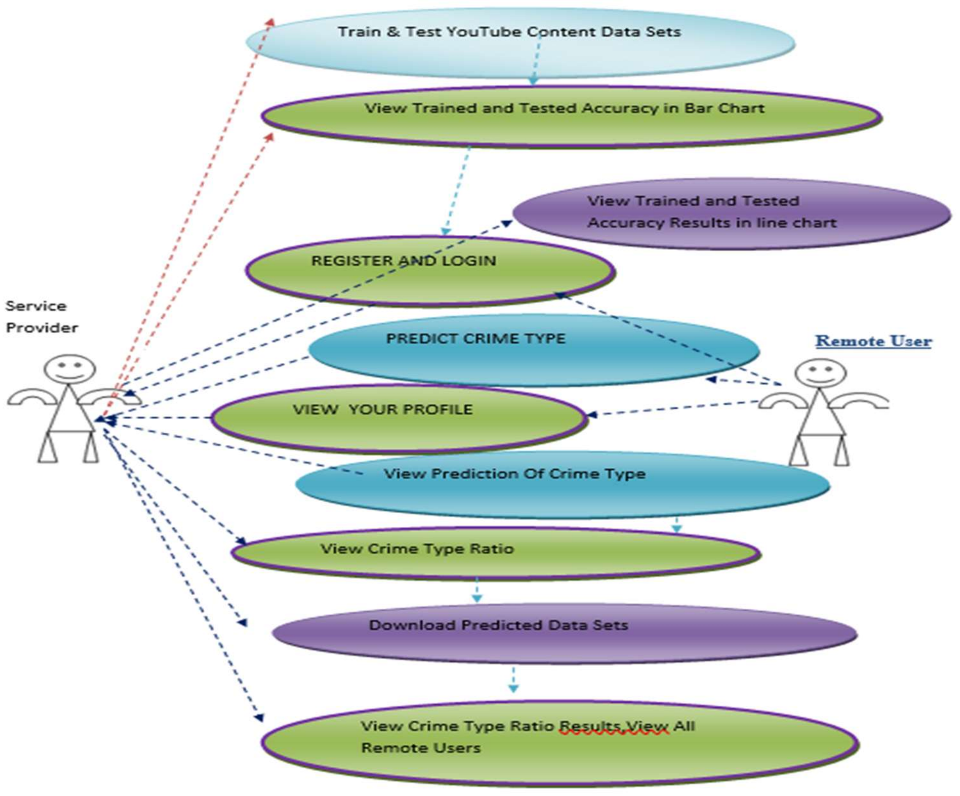

Train & Test YouTube Content Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results in line chart
REGISTER AND LOGIN
Service Provider
PREDICT CRIME TYPE
Remote User
VIEW YOUR PROFILE
View Prediction Of Crime Type
View Crime Type Ratio
Download Predicted Data Sets
View Crime Type Ratio Results,View All Remote Users

Fig_2: UML diagram

b. Class Diagram

for the Artificial Intelligence and Malicious Use Detection System. This diagram represents the core classes, their attributes, methods, and relationships within the system.

Class Diagram Overview

The Class Diagram includes:

1. **User:** Represents a user of the system (e.g., a regular user or an administrator).
2. **Malicious Use Detection Engine:** This class represents the detection engine responsible for analysing data and identifying malicious AI usage.

3. **Report:** This class represents the reports generated based on the analysis of AI misuse.
4. **Database:** This class handles the interactions with the database (storing and retrieving user data, reports, etc.).

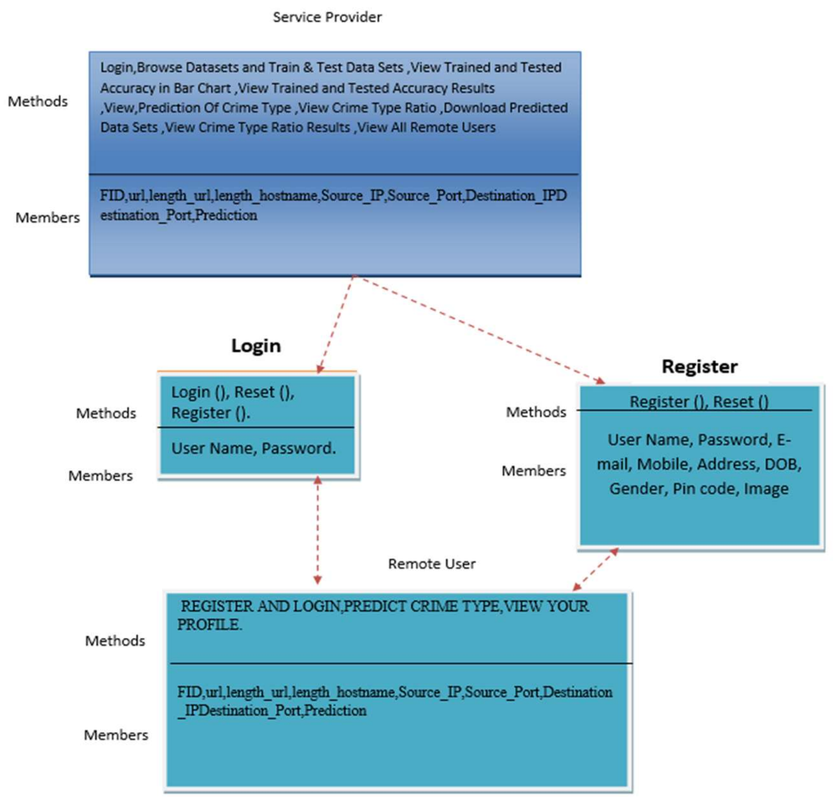

Service Provider
Methods
Login,Browse Datasets and Train & Test Data Sets ,View Trained and Tested Accuracy in Bar Chart ,View Trained and Tested Accuracy Results ,View,Prediction Of Crime Type ,View Crime Type Ratio ,Download Predicted Data Sets ,View Crime Type Ratio Results ,View All Remote Users
Members
FID,url,length_url,length_hostname,Source_IP,Source_Port,Destination_IPDestination_Port,Prediction
Login
Methods
Login (), Reset (), Register ().
Members
User Name, Password.
Register
Methods
Register (), Reset ()
Members
User Name, Password, E-mail, Mobile, Address, DOB, Gender, Pin code, Image
Remote User
Methods
REGISTER AND LOGIN,PREDICT CRIME TYPE,VIEW YOUR PROFILE.
Members
FID,url,length_url,length_hostname,Source_IP,Source_Port,Destination_IPDestination_Port,Prediction

Fig_3: Service Provider

c. Sequence Diagram

for the Artificial Intelligence and Malicious Use Detection System. This diagram represents the flow of interaction between different components of the system when a user inputs data, the detection engine processes the data, and results are returned.

Sequence Diagram Overview

The Sequence Diagram demonstrates the flow of messages and interactions between:

- **User:** A person interacting with the system.
- **UI:** The user interface that receives and sends data.
- **MaliciousUseDetectionEngine:** The AI engine responsible for analysing the input data.
- **Database:** Where data (user, report, and detection results) is stored.
- **External API:** The service that provides additional data for analysis, such as news or weather.

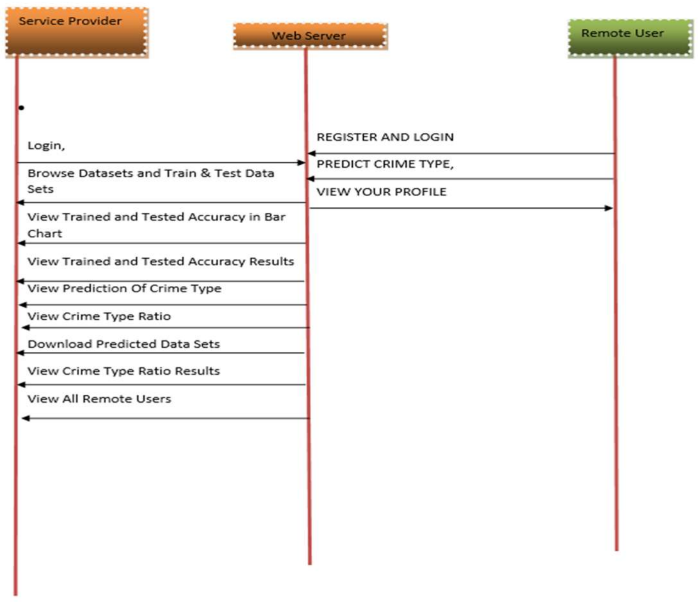

Service Provider
Web Server
Remote User
Login,
Browse Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Crime Type
View Crime Type Ratio
Download Predicted Data Sets
View Crime Type Ratio Results
View All Remote Users
REGISTER AND LOGIN
PREDICT CRIME TYPE,
VIEW YOUR PROFILE

Fig_4: Sequence Diagram

d. Activity Diagram

for the Artificial Intelligence and Malicious Use Detection System. This diagram represents the workflow or sequence of actions that occur in the system, from the user's input to the analysis and result display.

Activity Diagram Overview

The Activity Diagram demonstrates the sequence of activities that happen in the system:

1. The user enters input data.
2. The system processes the data.
3. The detection engine analyses the data for potential malicious AI use.
4. The results are displayed to the user.
5. If the data is invalid, the user is prompted to re-enter the correct information.

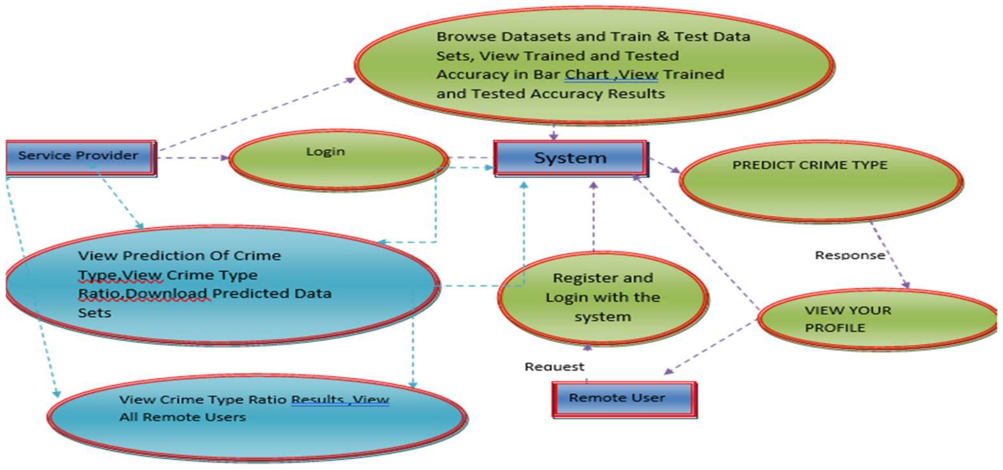

Browse Datasets and Train & Test Data Sets, View Trained and Tested Accuracy in Bar Chart ,View Trained and Tested Accuracy Results
Service Provider
Login
System
PREDICT CRIME TYPE
View Prediction Of Crime Type,View Crime Type Ratio,Download Predicted Data Sets
Register and Login with the system
Response
VIEW YOUR PROFILE
Request
Remote User
View Crime Type Ratio Results ,View All Remote Users

Fig_5: Activity Diagram

Flowchart Diagram

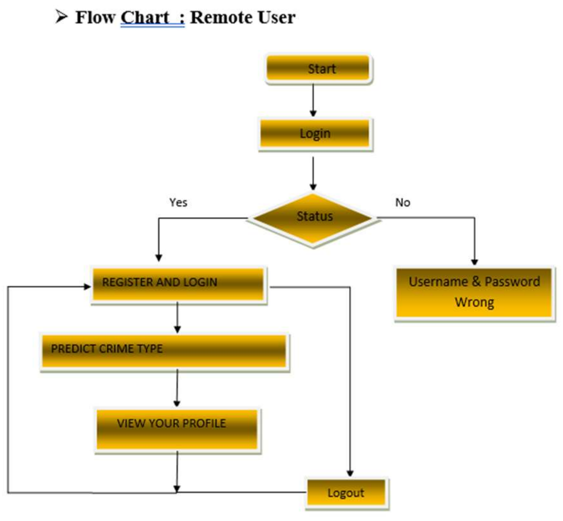
For the Artificial Intelligence and Malicious Use Detection System. This diagram represents the sequential steps in the process of data entry, analysis, detection, and results display.

Flowchart Diagram Overview

The Flowchart Diagram outlines the following main steps:

1. The user inputs data into the system.

2. The system validates the input.
3. If the input is valid, the system processes the data, fetches external data (if needed), analyses it for malicious use, and then displays the results.
4. If the input is invalid, the user is prompted to re-enter the correct data.



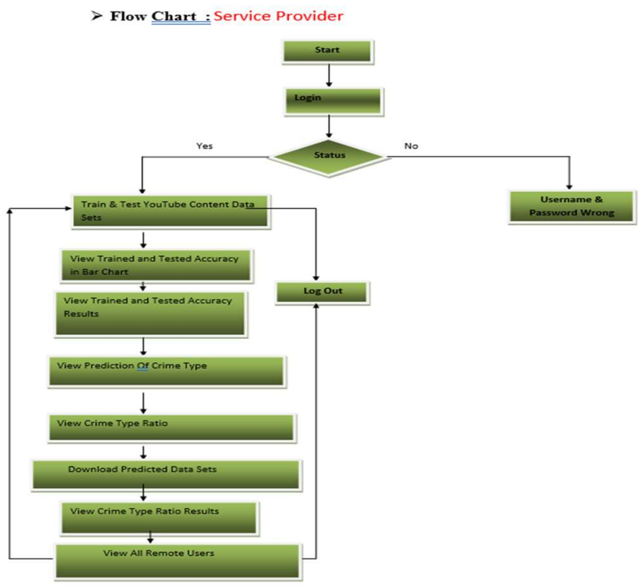

Flow Chart : Service Provider
Start
Login
Status
Yes
No
Train & Test YouTube Content Data Sets
Username & Password Wrong
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
Log Out
View Prediction Of Crime Type
View Crime Type Ratio
Download Predicted Data Sets
View Crime Type Ratio Results
View All Remote Users

Fig_6: Flowchart Diagram

CHAPTER 13

SOFTWARE ENVIRONMENT

13.1 What is 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

Advantages of Python :-

Let's see how Python dominates over other languages.

1. Extensive Libraries

Python downloads with an extensive library and it *contain 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.

2. 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.

3. 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.

4. 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.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

6. 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.

7. 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.

8. 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.

9. 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.

10. 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.

11. Interpreted

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

Advantages of Python Over Other Languages

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2.Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 GitHub annual survey showed us that Python has overtaken Java in the most popular programming language category.

3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and machine learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language

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.

1. 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.

2. 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.

3. 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.

4. 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.

5. Simple

No, we're not kidding. Python's simplicity can indeed be a problem. Take my example. I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

13.2 History of Python :

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wickenden Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So, I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

13.3 What is Machine Learning:

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of *building models of data*.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models *tunable parameters* that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Categories Of Machine Leaning :

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into *classification* tasks and *regression* tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as *clustering* and *dimensionality reduction.* Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

Challenges in Machines Learning :

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.

Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

Applications of Machines Learning:

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection
- Fraud prevention
- Recommendation of products to customer in online shopping

How to Start Learning Machine Learning?

Arthur Samuel coined the term “Machine Learning” in 1959 and defined it as a “Field of study that gives computers the capability to learn without being explicitly programmed”.

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer Is The Best Job of 2019 with a *344%* growth and an average base salary of $146,085 per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

(a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

(b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!! Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

(c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Kera's, TensorFlow, Scikit-learn, etc.

So if you want to learn ML, it's best if you learn Python! You can do that using various online resources and courses such as Fork Python available Free on GeeksforGeeks.

Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

(a) Terminologies of Machine Learning

- **Model** – A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
- **Feature** – A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.

- **Target (Label)** – A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
- **Training** – The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
- **Prediction** – Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

(b) Types of Machine Learning

- **Supervised Learning** – This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
- **Unsupervised Learning** – This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
- **Semi-supervised Learning** – This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
- **Reinforcement Learning** – This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

Advantages of Machine learning :-

1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed (automation)

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement

As ML algorithms gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Disadvantages of Machine Learning :-

1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

4. High error-susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

CHAPTER 14

SYSTEM TESTING

14.1 Testing Methodology

The following are the Testing Methodologies:

- Unit Testing.
- Integration Testing.
- User Acceptance Testing.
- Output Testing.
- Validation Testing.

Unit Testing

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All-important processing path are tested for the expected results. All error handling paths are also tested.

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

The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

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.

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.

Validation Checking

Validation checks are performed on the following fields.

Text Field:

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

Numeric Field:

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

Preparation of Test Data

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

Using Live Test Data:

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

Using Artificial Test Data:

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

The package "Virtual Private Network" has satisfied all the requirements specified as per software requirement specification and was accepted.

User Training

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

Maintenance

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user's requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

14.2 Testing Strategy:

A strategy for system testing integrates system test cases and design techniques into a well-planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation. A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

CHAPTER 15

IMPLEMANTATION

The implementation of the Artificial Intelligence and Malicious Use Detection System involves integrating multiple components, including machine learning models, a web interface, and backend services. The system begins by setting up the development environment, including Python, Django, and required libraries like TensorFlow and scikit-learn for machine learning. The first step in the implementation process is collecting and preprocessing data, followed by integrating external data sources like weather and news APIs, which are fetched using Python's requests library. The core of the system relies on machine learning models that are trained using labeled datasets, where patterns of malicious AI use are identified. Models like Random Forest or Neural Networks are used to predict whether an input contains potentially harmful AI behavior. Once trained, these models are integrated into the Django backend, where user inputs are processed and analyzed in real time. The frontend, built using HTML, CSS, and JavaScript, allows users to submit data for analysis and receive results on a web page. The system is designed to be scalable, with Docker used for containerization and cloud platforms like AWS or Heroku for deployment. Finally, thorough testing is performed using unit tests for backend logic and Selenium for frontend automation, ensuring the robustness of the system before it is deployed. This comprehensive implementation ensures that the system can accurately detect and classify potentially malicious use of AI in real-world scenarios.

CHAPTER 16
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

CHAPTER 17

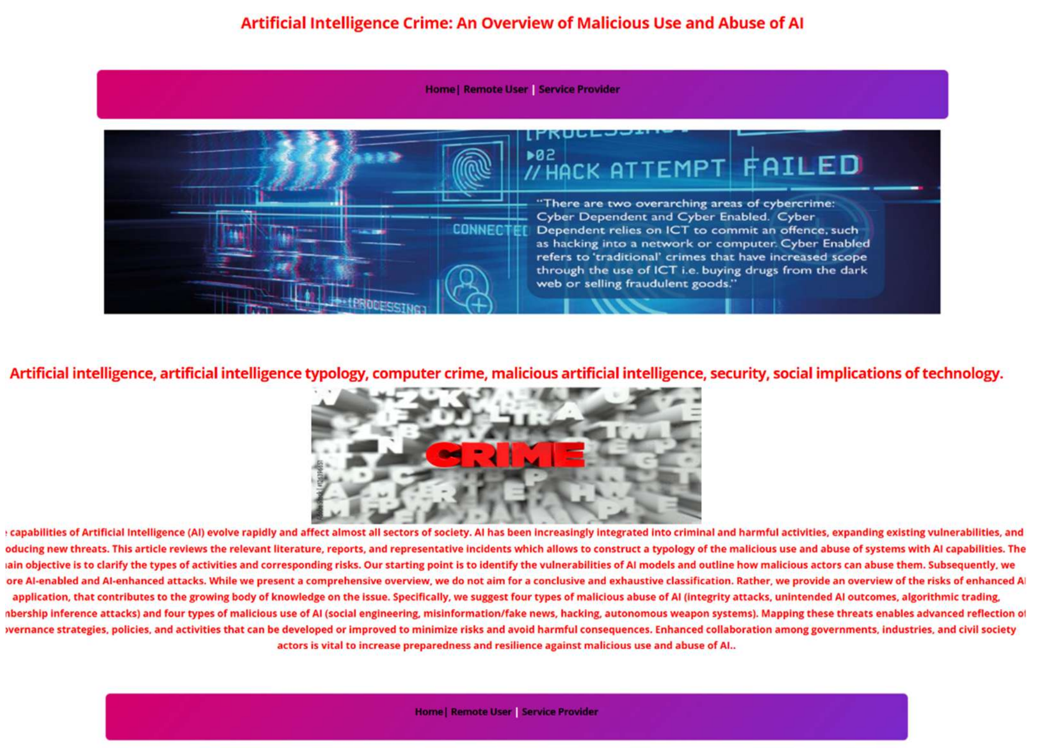
SAMPLE CODE

```
import builtins
import sys

### Registry and builtin stateless codec functions

try:
    from _codecs import *
except ImportError as why:
    raise SystemError('Failed to load the builtin codecs: %s' % why)

__all__ = ["register", "lookup", "open", "EncodedFile", "BOM", "BOM_BE",
           "BOM_LE", "BOM32_BE", "BOM32_LE", "BOM64_BE", "BOM64_LE",
           "BOM_UTF8", "BOM_UTF16", "BOM_UTF16_LE", "BOM_UTF16_BE",
           "BOM_UTF32", "BOM_UTF32_LE", "BOM_UTF32_BE",
           "CodecInfo", "Codec", "IncrementalEncoder", "IncrementalDecoder",
           "StreamReader", "StreamWriter",
           "StreamReaderWriter", "StreamRecoder",
           "getencoder", "getdecoder", "getincrementalencoder",
           "getincrementaldecoder", "getreader", "getwriter",
           "encode", "decode", "iterencode", "iterdecode",
           "strict_errors", "ignore_errors", "replace_errors",
           "xmlcharrefreplace_errors",
           "backslashreplace_errors", "namereplace_errors",
           "register_error", "lookup_error"]

### Constants

#
# Byte Order Mark (BOM = ZERO WIDTH NO-BREAK SPACE = U+FEFF)
# and its possible byte string values
# for UTF8/UTF16/UTF32 output and little/big endian machines
#

# UTF-8
BOM_UTF8 = b'\xef\xbb\xbf'

# UTF-16, little endian
BOM_LE = BOM_UTF16_LE = b'\xff\xfe'

# UTF-16, big endian
BOM_BE = BOM_UTF16_BE = b'\xfe\xff'

# UTF-32, little endian
BOM_UTF32_LE = b'\xff\xfe\x00\x00'
```

```
# UTF-32, big endian
BOM_UTF32_BE = b'\x00\x00\xfe\xff'

if sys.byteorder == 'little':

    # UTF-16, native endianness
    BOM = BOM_UTF16 = BOM_UTF16_LE

    # UTF-32, native endianness
    BOM_UTF32 = BOM_UTF32_LE

else:

    # UTF-16, native endianness
    BOM = BOM_UTF16 = BOM_UTF16_BE

    # UTF-32, native endianness
    BOM_UTF32 = BOM_UTF32_BE

# Old broken names (don't use in new code)
BOM32_LE = BOM_UTF16_LE
BOM32_BE = BOM_UTF16_BE
BOM64_LE = BOM_UTF32_LE
BOM64_BE = BOM_UTF32_BE
class IncrementalEncoder(object):
    """
    An IncrementalEncoder encodes an input in multiple steps. The input can
    be passed piece by piece to the encode() method. The IncrementalEncoder
    remembers the state of the encoding process between calls to encode().
    """
    def __init__(self, errors='strict'):
        """
        Creates an IncrementalEncoder instance.

        The IncrementalEncoder may use different error handling schemes by
        providing the errors keyword argument. See the module docstring
        for a list of possible values.
        """
        self.errors = errors
        self.buffer = ""

    def encode(self, input, final=False):
        """
        Encodes input and returns the resulting object.
        """
        raise NotImplementedError
    def reset(self):
        """
        Resets the encoder to the initial state.
        """

    def getstate(self):
        """
        Return the current state of the encoder.
        """
        return 0

    def setstate(self, state):
        """
        Set the current state of the encoder. state must have been
        returned by getstate().
        """
# Tell modulefinder that using codecs probably needs the encodings
# package
_false = 0
if _false:
    import encodings

### Tests

if __name__ == '__main__':

    # Make stdout translate Latin-1 output into UTF-8 output
    sys.stdout = EncodedFile(sys.stdout, 'latin-1', 'utf-8')


    # Have stdin translate Latin-1 input into UTF-8 input

    sys.stdin = EncodedFile(sys.stdin, 'utf-8', 'latin-1')
```

CHAPTER 18

SCREENSHOTS


Artificial Intelligence Crime: An Overview of Malicious Use and Abuse of AI

Home| Remote User | Service Provider

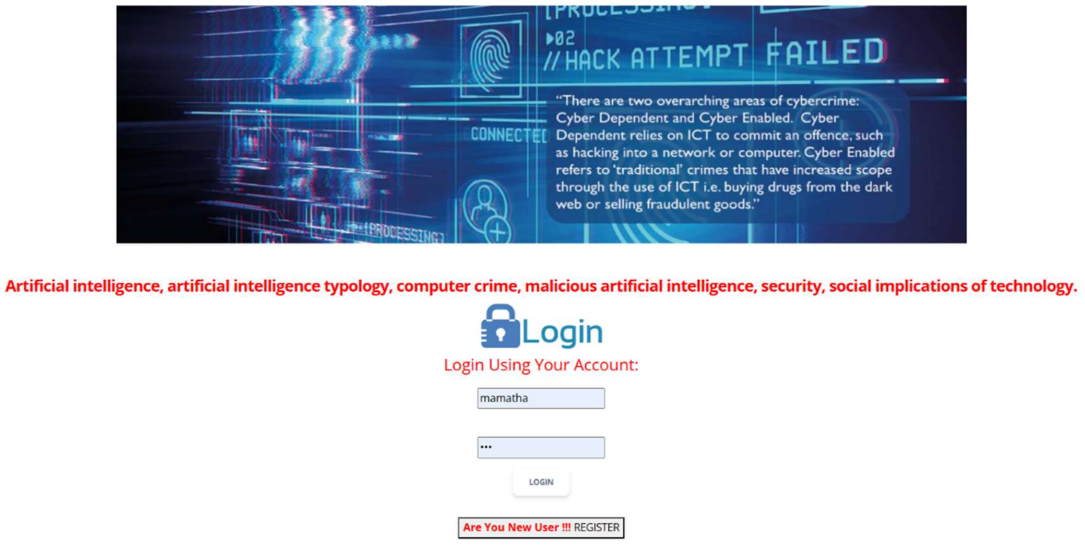
"There are two overarching areas of cybercrime: Cyber Dependent and Cyber Enabled. Cyber Dependent relies on ICT to commit an offence, such as hacking into a network or computer. Cyber Enabled refers to 'traditional' crimes that have increased scope through the use of ICT i.e. buying drugs from the dark web or selling fraudulent goods."

Artificial intelligence, artificial intelligence typology, computer crime, malicious artificial intelligence, security, social implications of technology.

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 oducing 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 ain 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 ore 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, nbership inference attacks) and four types of malicious use of AI (social engineering, misinformation/fake news, hacking, autonomous weapon systems). Mapping these threats enables advanced reflection of overnance strategies, policies, and activities that can be developed or improved to minimize risks and avoid harmful consequences. Enhanced collaboration among governments, industries, and civil society actors is vital to increase preparedness and resilience against malicious use and abuse of AI..

Home| Remote User | Service Provider

Screenshot 1: Home page


"There are two overarching areas of cybercrime: Cyber Dependent and Cyber Enabled. Cyber Dependent relies on ICT to commit an offence, such as hacking into a network or computer. Cyber Enabled refers to 'traditional' crimes that have increased scope through the use of ICT i.e. buying drugs from the dark web or selling fraudulent goods."

Artificial intelligence, artificial intelligence typology, computer crime, malicious artificial intelligence, security, social implications of technology.

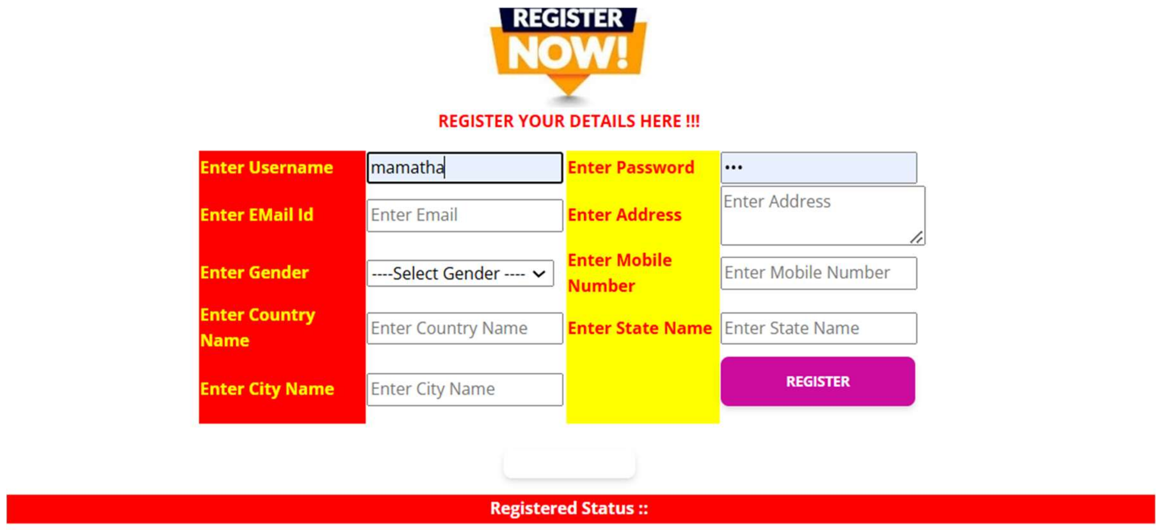
Login

Login Using Your Account:

mamatha

•••

Screenshot 2: login page

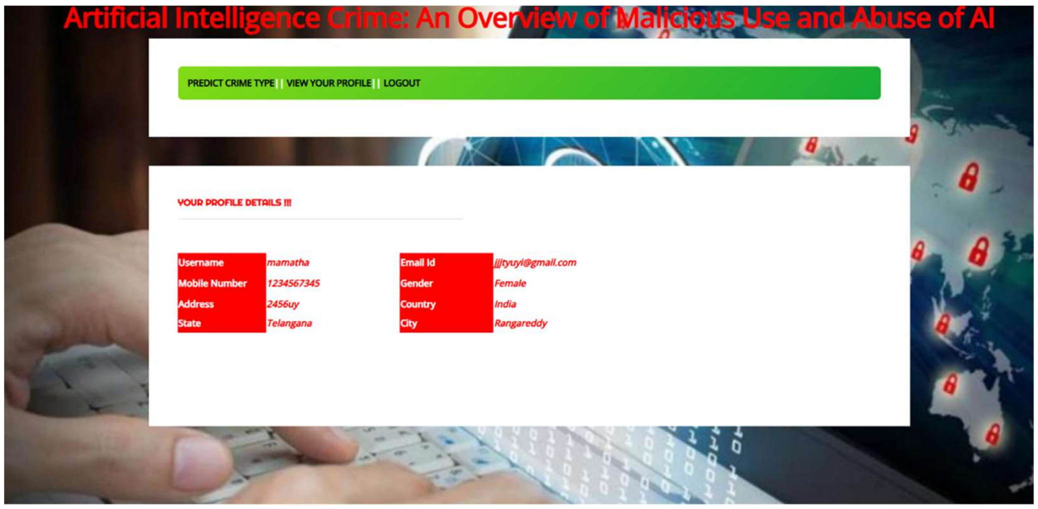


REGISTER YOUR DETAILS HERE !!!

Enter Username	mamatha	Enter 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		

Registered Status ::

Screenshot 3: Registering process

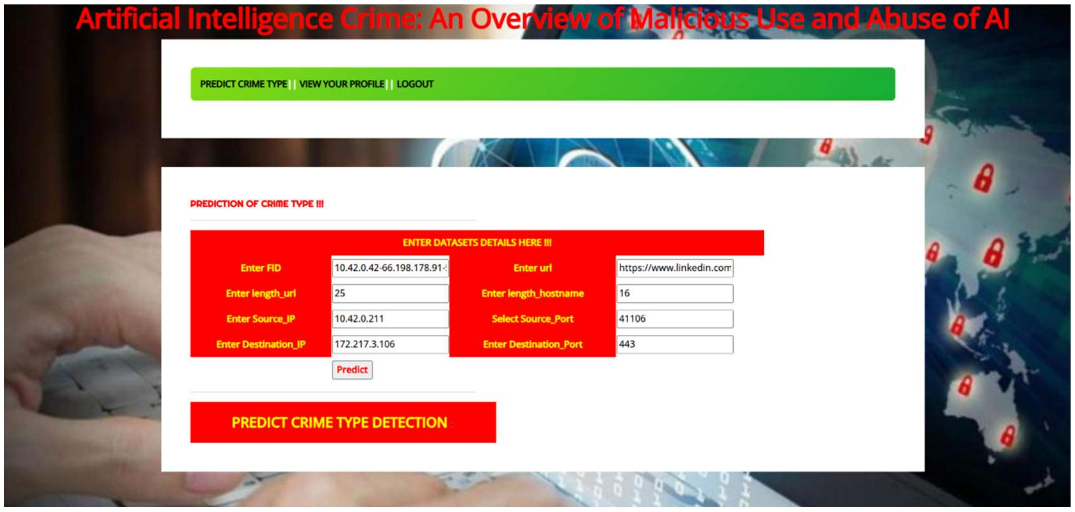


Artificial Intelligence Crime: An Overview of Malicious Use and Abuse of AI

YOUR PROFILE DETAILS !!!

Username	mamatha	Email Id	jjjtyuyi@gmail.com
Mobile Number	1234567345	Gender	Female
Address	2456uy	Country	India
State	Telangana	City	Rangareddy

Screenshot 4: User profile page


Artificial Intelligence Crime: An Overview of Malicious Use and Abuse of AI

PREDICT CRIME TYPE || VIEW YOUR PROFILE || LOGOUT

PREDICTION OF CRIME TYPE !!!

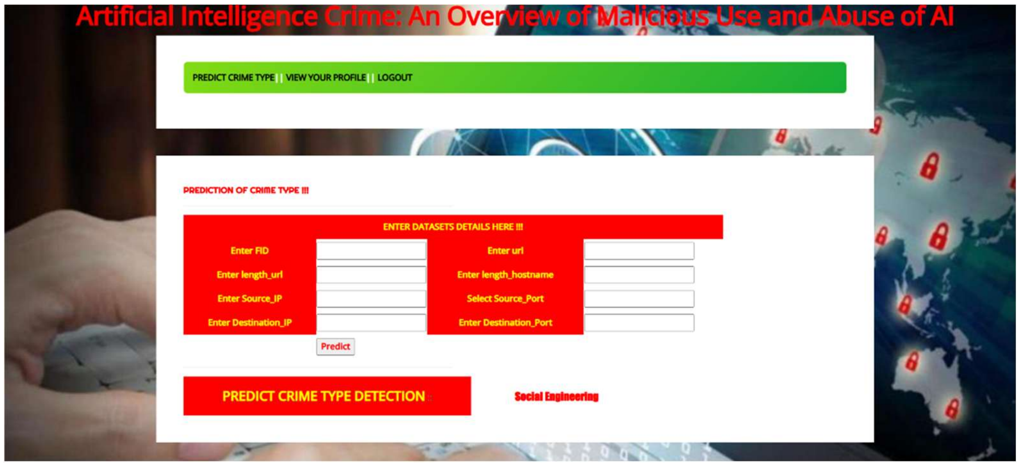
ENTER DATASETS DETAILS HERE !!!

Enter FID	10.42.0.42-66.198.178.91-	Enter url	https://www.linkedin.com
Enter length_url	25	Enter length_hostname	16
Enter Source_IP	10.42.0.211	Select Source_Port	41106
Enter Destination_IP	172.217.3.106	Enter Destination_Port	443

Predict

PREDICT CRIME TYPE DETECTION

Screenshot 5: Entering Dataset Page


Artificial Intelligence Crime: An Overview of Malicious Use and Abuse of AI

PREDICT CRIME TYPE || VIEW YOUR PROFILE || LOGOUT

PREDICTION OF CRIME TYPE !!!

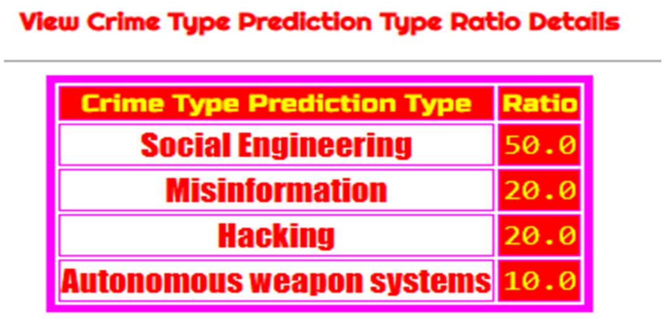
ENTER DATASETS DETAILS HERE !!!

Enter FID		Enter url	
Enter length_url		Enter length_hostname	
Enter Source_IP		Select Source_Port	
Enter Destination_IP		Enter Destination_Port	

Predict

PREDICT CRIME TYPE DETECTION Social Engineering

Screenshot 6: Predicted Crime


View Crime Type Prediction Type Ratio Details

Crime Type Prediction Type	Ratio
Social Engineering	50.0
Misinformation	20.0
Hacking	20.0
Autonomous weapon systems	10.0

Screenshot 7: Crime Ratio

CHAPTER 19
FUTURE SCOPE OF THE PROJECT

The future scope of the Artificial Intelligence and Malicious Use Detection System is vast and holds immense potential for growth and refinement. As AI technologies continue to evolve, so do the methods by which malicious actors exploit them. The project could be expanded to incorporate more advanced machine learning techniques, such as deep learning models, to improve the accuracy and detection capabilities for new types of AI-enabled attacks. Additionally, the system could be adapted to monitor emerging AI applications across various sectors, including healthcare, finance, and autonomous systems, where the risk of malicious misuse is particularly high. The incorporation of real-time data streams from multiple external sources, like social media, IoT devices, and global security databases, would further enhance the system's ability to detect and mitigate potential threats. Moreover, continuous updates to the classification system, using empirical data and feedback loops, could make the tool more adaptable to the rapidly changing landscape of AI threats. Finally, expanding the project to include international collaboration could allow for a more global perspective on AI risks, ensuring that the system remains robust and effective in addressing the full spectrum of potential malicious uses. Ultimately, the future scope of this project lies in developing a proactive, adaptive, and globally integrated approach to combating AI abuse and ensuring the safe deployment of AI technologies across all domains.

CHAPTER 20
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.

[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.

[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 andW.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.

[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.

[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.

[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.
[9] D. R. Coats, *Worldwide Threat Assessment of the U.S. Intelligence Community*. New York, NY, USA, 2017, p. 32.
[10] L. Floridi, ``Soft ethics: Its application to the general data protection regulation and its dual advantage,'' *Philosophy Technol.*, vol. 31, no. 2, pp. 163_167, Jun. 2018, doi: 10.1007/s13347-018-0315-5.
[11] P. S. Chauhan and N. Kshetri, ``2021 state of the practice in data privacy and security,'' *Computer*, vol. 54, no. 8, pp. 125_132, Aug. 2021, doi: 10.1109/MC.2021.3083916.
[12] S. Gordon and R. Ford, ``On the de_nition and classi_cation of cybercrime,'' *J. Comput. Virol.*, vol. 2, no. 1, pp. 13_20, Aug. 2006, doi: 10.1007/s11416-006-0015-z.
[13] *Cybercrime*. United Nations: Of_ce Drugs. Accessed: May 19, 2021. http://www.unodc.org/unodc/en/cybercrime/index.html
[14] M. Brundage, S. Avin, J. Clark, and H. Toner, ``The malicious use of arti_cial intelligence: Forecasting, prevention, and mitigation,'' 2018, *arXiv:1802.07228*.
[15] T. C. King, N. Aggarwal, M. Taddeo, and L. Floridi, ``Arti_cial intelligence crime: An interdisciplinary analysis of foreseeable threats and solutions,'' *Sci. Eng. Ethics*, vol. 26, no. 1, pp. 89_120, Feb. 2020, doi: 10.1007/s11948-018-00081-0.
[16] V. Ciancaglini, ``Malicious uses and abuses of arti_cial intelligence,'' in *Trend Micro Research; United Nations Interregional Crime and Justice Research Institute (UNICRI); Europol's European Cybercrime Centre (EC3)*, Nov. 2020. [Online]. Available: https://www.europol.europa.eu/ publications-documents/malicious-uses-and-abuses-of-arti_cialintelligence
[17] K. D. Fiedler, V. Grover, and J. T. C. Teng, ``An empirically derived taxonomy of information technology structure and its relationship to organizational structure,'' *J Manage. Inf. Syst.*, vol. 13, pp. 9_34, Jun. 1996, doi: 10.1080/07421222.1996.11518110.

[18] N. Bostrom, ``Information hazards: A typology of potential harms from knowledge,'' *Rev. Contemp. Philosophy*, vol. 10, pp. 44_79, May 2011.
[19] W. B. Carper and W. E. Snizek, ``The nature and types of organizational taxonomies: An overview,'' *Acad. Manage. Rev.*, vol. 5, no. 1, pp. 65_75, Jan. 1980.
[20] (Apr. 21, 2021). *Proposal for a Regulation Laying Down Harmonised Rules on Arti_cial Intelligence_Arti_cial Intelligence Act*. European Commission. Accessed: May 19, 2021. [Online]. Available: https://digital-strategy.ec.europa.eu/en/library/proposal-regulationlaying-down-harmonised-rules-arti_cial-intelligence-arti_cialintelligence