# ENHANCED CYBERCRIME UNDERGROUND ECONOMY USING MACHINE LEARNING APPROACHES



Major Project submitted in partial fulfillment of the requirement for the award of the

degree of

**BACHELOR OF TECHNOLOGY IN**

**INFORMATION TECHNOLOGY**

Under the esteemed guidance of

**Mrs. K. GNANA MAYURI**

**Assistant Professor**

By

#### T. HARINI 19R11A1252

**M. SAI CHARAN 19R11A1228**

**K. GOVTHAM 19R11A1221**



**Department of Information Technology**

**Accredited by NBA**

**Geethanjali College of Engineering and Technology**

**(UGC Autonomous)**

(Affiliated to J.N.T.U.H, Approved by AICTE, New Delhi) Cheeryal(V), Keesara (M), Medchal.Dist.-501 301.

**April-2023**

Geethanjali College of Engineering & Technology

**(UGC Autonomous)**

(Affiliated to JNTUH, Approved by AICTE, New Delhi) Cheeryal (V), Keesara(M), Medchal Dist.-501 301.

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**Accredited by NAAC A+**



**CERTIFICATE**

This is to certify that the B.Tech Major Project Report Entitled “**ENHANCED CYBERCRIME UNDERGROUND ECONOMY USING MACHINE LEARNING**

**APPROACHES”** is a bonafide work done by **T. HARINI (19R11A1252), M. SAI CHARAN (19R11A1228), K. GOVTHAM (19R11A1221)** in partial fulfillment of the requirement of the award for the degree of Bachelor of Technology in “**INFORMATION TECHNOLOGY** '' from Jawaharlal Nehru Technological University, Hyderabad during the year 2022-2023.

#### Internal Guide HOD - IT

**Mrs. K. GNANA MAYURI Dr. K. Srinivas**

#### Assistant Professor Professor

**External Examiner**

Geethanjali College of Engineering & Technology

**(UGC Autonomous)**

(Affiliated to JNTUH Approved by AICTE, New Delhi) Cheeryal(V), Keesara(M), Medchal Dist.-501301

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**Accredited by NBA**



**DECLARATION BY THE CANDIDATE**

We, **T. Harini, M. Sai Charan, K. Govtham** bearing Roll Nos. **19R11A1252, 19R11A1228, 19R11A1221** hereby declare that the project report entitled **“ENHANCED CYBERCRIME UNDERGROUND ECONOMY USING MACHINE LEARNING**

**APPROACHES”** is done under the guidance of **Mrs. K. Gnana Mayuri**, **Assistant Professor**, Department of Information Technology, Geethanjali College of Engineering and Technology, is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Information Technology**.

This is a record of bonafide work carried out by us in **Geethanjali College of Engineering and Technology** and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other Universityor Institute for the award of any other degree or diploma.

#### T. HARINI - 19R11A1252

**M. SAI CHARAN - 19R11A1228**

#### K. GOVTHAM -19R11A1221

**Department of IT, Geethanjali College of Engineering and Technology.**

# ACKNOWLEDGEMENT

We are greatly indebted to the authorities of Geethanjali College of Engineering and Technology, Cheeryal, Medchal Dist., for proving us the necessary facilities to successfully carry out this major project work titled **“ENHANCED CYBERCRIME UNDERGROUND ECONOMY USING MACHINE LEARNING APPROACHES”.**

Firstly, we thank and express our solicit gratitude to **Professor, Dr. K. Srinivas, HOD, Professor, IT department**, Geethanjali College of Engineering and Technology, for his invaluablehelp and support which helped us a lot in successfully completing our major project.

Secondly, we express our gratitude to **Assistant Prof. Mrs. K. Gnana Mayuri**, internal guide, Geethanjali College of Engineering and Technology, for his suggestion and encouragement which helped us in the successful completion of our major project.

We would like to express our sincere gratitude to our **Principal Dr. S. UDAY KUMAR**

for providing the necessary infrastructure to complete our project.

We convey our gratitude to our **Chairman, Mr. G. RAVINDER REDDY,** for his invaluablesupport and encouragement for propelling our innovation forward.

Finally, we would like to express our heartfelt thanks to our parents who were very supportive both financially and mentally and for their encouragement to achieve our set goals.

#### T. HARINI - 19R11A1252.

**M. SAI CHARAN - 19R11A1228.**

**K. GOVTHAM -19R11A1221.**

# ABSTRACT

Despite the rapid escalation of cyber threats, there has still been little research into the foundations of the subject or methodologies that could serve to guide Information Systems researchers and practitioners who deal with cybersecurity. In addition, little is known about Crime-as-a-Service (CaaS), a criminal business model that underpins the cybercrime underground. This research gap and the practical cybercrime problems we face have motivated us to investigate the cybercrime underground economy by taking a data analytics approach from a design science perspective.

To achieve this goal, we propose (1) a data analysis framework for analyzing the cybercrime underground, (2) CaaS and crimeware definitions, and (3) an associated classification model. In addition, we (4) develop an example application to demonstrate how the proposed framework and classification modelcould be implemented in practice. We then use this application to investigate the cybercrimeunderground economy by analyzing a large dataset obtained from the online hacking community. By taking a design science research approach, this study contributes to the design artifacts, foundations, and methodologies in this area. Moreover, it provides useful practical insights to practitioners by suggesting guidelines as to how governments and organizations in all industries can prepare for attacks by the cybercrime underground.

**Key Words:** Crimeware-as-a-Service, crimeware, hacking community, , underground economy, machine learning.

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Fig No.** | **Title** | **Page No.** |
| 2.1 | Requirements Gathering Stage | 17 |
| 2.2 | Analysis Stage | 19 |
| 2.3 | Designing Stage | 20 |
| 2.4 | Development Stage | 21 |
| 2.5 | Integration & Test Stage | 22 |
| 2.6 | Installation & Acceptance Test | 23 |
| 2.7 | SDLC Umbrella Model | 24 |
| 3.1 | The Architectural Model | 26 |
| 3.2 | Use Case Diagram | 26 |
| 3.3 | Class Diagram | 27 |
| 3.4 | Sequence Diagram | 28 |
| 3.5 | Activity Diagram | 29 |
| 3.6 | Component Diagram | 30 |
| 3.7 | Deployment Diagram | 31 |
| 3.8 | Data Flow Diagram | 32 |
| 4.1 | Project Scheduling | 40 |

# LIST OF SCREENSHOTS

|  |  |  |
| --- | --- | --- |
| **Fig No.** | **Title** | **Page No.** |
| 5.1 | User Interface | 44 |
| 5.2 | Home Page | 53 |
| 5.3 | Uploading the Malware Dataset | 54 |
| 5.4 | Graphical Representation of Various Cybercrime Attacks | 54 |
| 5.5 | Training and Testing of Data | 55 |
| 5.6 | Prediction Accuracy | 56 |
| 5.7 | Classification Performance Graph | 56 |
| 5.8 | Uploading the Test Dataset | 57 |
| 5.9 | Prediction of Various Cybercrime Attacks | 58 |

# LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Title** | **Page No.** |
| 1.1 | Literature Forms | 11 |
| 1.2 | Literature Resources | 12 |
| 4.1 | Test Cases | 43 |

# TABLE OF CONTENTS

**Contents Page No.**

ACKNOLEDGEMENT iv

[ABSTRACT v](#_TOC_250036)

[LIST OF FIGURES vi](#_TOC_250035)

[LIST OF SCREENSHOTS vii](#_TOC_250034)

[LIST OF TABLES vii](#_TOC_250033)

[TABLE OF CONTENTS viii](#_TOC_250032)

CHAPTER 1: INTRODUCTION 1

* 1. [Introduction to the Project 1](#_TOC_250031)
  2. [Project Category 3](#_TOC_250030)
  3. [Objectives 3](#_TOC_250029)
  4. Problems Formulation 4
  5. [Identification/Reorganization of Need 4](#_TOC_250028)
  6. [Existing System 5](#_TOC_250027)
  7. [Limitations 5](#_TOC_250026)
  8. [Proposed System 5](#_TOC_250025)
  9. [Advantages of Proposed System 6](#_TOC_250024)
  10. [Unique Features of the System 6](#_TOC_250023)
  11. Reasons for Literature Survey 6
  12. [Literature Forms 11](#_TOC_250022)
  13. [Literature Resources 12](#_TOC_250021)

CHAPTER 2: REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATIONS

* 1. [Feasibility Study 13](#_TOC_250020)
  2. Software Requirements Specification 13
  3. [Validation 15](#_TOC_250019)
  4. [Expected Hurdles 16](#_TOC_250018)
  5. [SDLC Model Used 16](#_TOC_250017)

CHAPTER 3: SYSTEM DESIGN 25

* 1. [Design Approach 25](#_TOC_250016)
  2. [Detail Design 25](#_TOC_250015)
  3. Architecture Model 25
     1. [UML Diagrams 26](#_TOC_250014)
  4. [Algorithm 32](#_TOC_250013)
  5. [User Interface Design 33](#_TOC_250012)
  6. [Methodology 33](#_TOC_250011)

CHAPTER 4: IMPLEMENTATION, TESTING AND MAINTENANCE 35

* 1. [Introduction to Languages, IDE’s used for Implementation 35](#_TOC_250010)
     1. [Tools and Technologies used for Implementation 37](#_TOC_250009)
  2. Coding Standards of Languages Used 38
  3. [Project Scheduling 39](#_TOC_250008)
  4. [Testing Techniques and Test Plans 40](#_TOC_250007)
     1. [Test Cases 43](#_TOC_250006)

CHAPTER 5: RESULTS AND DISCUSSIONS 44

* 1. [User Interface Representation 44](#_TOC_250005)
     1. [Brief Description of Various Modules of the System 44](#_TOC_250004)
  2. [Coding 45](#_TOC_250003)
     1. [Sample Code 47](#_TOC_250002)
  3. [Snapshots of system with brief detail of each 53](#_TOC_250001)

CHAPTER 6: CONCLUSION AND FUTURE ENHANCEMENT 59

[REFERENCES/BIBLIOGRAPHY 60](#_TOC_250000)

# INTRODUCTION

## Introduction to the Project

As the threat posed by massive cyberattacks (e.g., ransomware and distributed denial of service attacks (DDoS)) and cybercrimes has grown, individuals, organizations, and governments have struggled to find ways to defend against them. In 2017, ransomware known as WannaCry was responsible for nearly 45,000 attacks in almost 100 countries. The explosive impact of cybercrime has put governments under pressure to increase their cybersecurity budgets. United States President Barack Obama proposed spending over $19 billion on cybersecurity as part of his fiscal year 2017 budget, an increase of more than 35% since 2016.Global cyberattacks (such as WannaCry and Petya) are executed by highly organized criminal groups, and organized or national-level crime groups have been behind many recent attacks. Typically, criminal groups buy and sell hacking tools and services on the cybercrime black market, wherein attackers share a range of hacking-related information. This online underground market is operated by groups of attackers, and it in turn supports the underground cybercrime economy. The cybercrime underground has thus emerged as a new type of organization that both operates black markets and enables cybercrime conspiracies to flourish.

Because organized cybercrime requires an online network to exist and to conduct its attacks, it is highly dependent on closed underground communities (e.g., Hack forums and Cracking zilla). The anonymity these closed groups offer means that cybercrime networks are structured differently than traditional Mafia-style hierarchies, which are vertical, concentrated, rigid, and fixed. In contrast, cybercrime networks are lateral, diffuse, fluid, and evolving. Since cyberspace is a network of networks, the threat posed by the rise of highly professional network-based cybercrime business models, such as Crimeware-as-a-Service (CaaS), remains mostly invisible to governments, organizations, and individuals.

Even though Information Systems (IS) researchers and practitioners are taking an increasing interest in cybercrime, due to the critical issues arising from the rapid increase in cyber threats, few have attempted to put this new interest on a solid foundation or develop suitable methodologies. Previous studies have not analyzed the underground economy behind cybercrime in depth. Furthermore, little is known about CaaS, one of the primary businesses

models behind the cybercrime underground. There is an overall lack of understanding, both in research and practice, of the nature of this underground and the mechanisms underlying it.

This research gap, and the practical problems faced by cybercriminals, motivates our study. We take a data analytics approach and investigate the cybercrime economy from a design science perspective. To achieve this goal, we (1) propose a data analysis framework for analyzing the cybercrime underground to guide researchers and practitioners; (2) define CaaS and crimeware to better reflect their features from both academic research and business practice perspectives; (3) use this to build a classification model for CaaS and crimeware; and (4) build an application to demonstrate how the proposed framework and classification model could be implemented in practice. We then evaluate this application by applying it in a case study, namely investigating the cybercrime economy by analyzing a large dataset from the online hacking community.

This study takes a design science research (DSR) approach. Design science “creates and evaluates information technology artifacts intended to solve identified problems”. DSR involves developing a range of IT artifacts, such as decision support systems, models, frameworks, tools, methods, and applications. Where behavioral science research seeks to develop and justify theories that explain or predict human or organizational phenomena, DSR seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts. DSR’s contribution is to add value to the literature and practice in terms of “design artifacts, design construction knowledge (e.g., foundations), and/or design evaluation knowledge (e.g., methodologies)” .

This study follows these DSR guidelines and contributes design artifacts, foundations, and methodologies. In particular, DSR must demonstrate that design artifacts are “implementable” in the business environment to solve an important problem, so we provide an implementable framework rather than a conceptual one. We also create a front-end application as a case example to demonstrate how the proposed framework and classification model could be implemented in practice. In addition, this study contributes to design theory.

As for foundations, DSR should have a creative development of constructs, models, methods, or instantiations that extend the design science knowledge base. This study therefore adds to the knowledge base by providing foundational elements such as constructs (definitions, frameworks, and applications), a model (classification model), a method

(analysis), and instantiations (applications). As for methodologies, the creative development and use of evaluation methods provide DSR contributions. Accordingly, this study uses dynamic analysis to conduct an ex-ante evaluation of the classification model. It also conducts an ex-post evaluation of a front-end application using observational methods (case examples). From a practical perspective, this study also provides practitioners with useful insights by making suggestions to guide governments and organizations in all industries in solving the problems they face when preparing for attacks from the cybercrime underground.

## Project Category

Our project enhanced cybercrime underground economy using machine learning approaches can encompass a range of project categories depending on the specific objectives of the project. It could involve research-based projects aimed at gaining a better understanding of emerging trends and patterns in cybercrime activities, network or system administration projects focused on implementing security measures to prevent cyber-attacks, internet-based projects involving the development of web-based tools for monitoring and analyzing cybercrime activities, application or system development projects aimed at creating software tools for detecting and preventing cyber-attacks, or industry automation projects focused on automating the process of analyzing cybercrime data. The choice of the project category will depend on the project's goals and the organization's needs. Regardless of the project category, a data analytics approach can help organizations better understand the underground economy of cybercrime and develop effective strategies for preventing and mitigating cyber threats.

## Objectives

Despite the rapid escalation of cyber threats, there has still been little research into the foundations of the subject or methodologies that could serve to guide Information Systems researchers and practitioners who deal with cybersecurity. In addition, little is known about Crime-as-a-Service (CaaS), a criminal business model that underpins cybercrime underground. This research gap and the practical cybercrime problems we face have motivated us to investigate the cybercrime underground economy by taking a data analytics approach from a design science perspective. To achieve this goal, we propose (1) a data analysis framework for analyzing the cybercrime underground, (2) CaaS and crimeware

definitions, and (3) an associated classification model. In addition, we (4) develop an example application to demonstrate how the proposed framework and classification model could be implemented in practice. We then use this application to investigate the cybercrime underground economy by analyzing a large dataset obtained from the online hacking community. By taking a design science research approach, this study contributes to the design artifacts, foundations, and methodologies in this area. Moreover, it provides useful practical insights to practitioners by suggesting guidelines as to how governments and organizations in all industries can prepare for attacks by cybercrime underground.

## Problem Formulation

Enhanced cybercrime underground economy using machine learning approaches involves using advanced data analytics techniques to gain insights into the underground economy of cybercrime. This approach involves analyzing large volumes of data from a variety of sources to identify emerging trends, actors involved in cybercrime activities, and vulnerabilities in the system. By using data analytics tools and techniques, organizations can better understand the tactics and techniques used by cybercriminals and develop effective strategies for detecting, preventing, and mitigating cyber threats. However, this approach also involves addressing key challenges such as the lack of reliable data, complex data, rapidly evolving threats, lack of coordination, and lack of awareness. By formulating these challenges clearly and developing targeted strategies to address them, organizations can take a proactive approach to protecting themselves from cyber threats associated with the underground economy of cybercrime. Overall, a data analytics approach can help organizations stay ahead of the latest cyber threats and minimize the risk of data breaches, financial loss, and other negative consequences associated with cybercrime.

## Identification/Reorganization of Need

The need for enhanced cybercrime underground economy using machine learning approaches is becoming increasingly apparent in today's technology-driven world. The frequency and severity of cyber threats are on the rise, posing a significant risk to organizations and individuals alike. The evolving threat landscape and lack of visibility into the underground economy of cybercrime make it challenging for organizations to understand the scope and nature of the threats they face. Furthermore, many organizations lack effective

strategies for detecting, preventing, and mitigating cyber threats, leading to financial loss, data breaches, and other negative consequences. By recognizing these factors and understanding the need for a data analytics approach to the cybercrime underground economy, organizations can take a proactive approach to cybersecurity and protect themselves from financial losses, data breaches, and other negative consequences associated with cybercrime. A data analytics approach can provide valuable insights into the nature and scope of cyber threats, enabling organizations to develop effective strategies for managing cyber risks and complying with regulatory requirements.

## Existing System

Information Systems (IS) researchers and practitioners are taking an increasing interest in cybercrime, due to the critical issues arising from the rapid increase in cyber threats, few have attempted to put this new interest on a solid foundation or develop suitable methodologies. Previous studies have not analyzed the underground economy behind cybercrime in depth. Furthermore, little is known about CaaS, one of the primary business models behind the cybercrime underground. There is an overall lack of understanding, both in research and practice, of the nature of this underground and the mechanisms underlying it.

## Limitations

The limitations of existing system are-:

* + - Stretched Resources
    - Operational Mishaps
    - High Costs
    - Low Performance

## Proposed System

We take a data analytics approach and investigate the cybercrime economy from a design science perspective. we are using machine algorithms like navie\_bayes, we (1)propose a data analysis framework for analyzing the cybercrime underground (2) define CaaS and crimeware to better reflect their features from both academic research (3) use this to build a classification model for CaaS and crimeware; and (4) build an application to demonstrate how the proposed framework and classification model could be implemented in practice.

## Advantages of Proposed System

* + - High Accuracy
    - Improved threat detection
    - Real-time incident response
    - Improved resource allocation

## Unique Features of the System

* + - We propose a data analysis framework for analyzing the cybercrime underground, CaaS, crimeware, and an associated classification model.
    - We investigate the cybercrime underground economy by analyzing a large dataset obtained from the online hacking community.
    - We take a data analytics approach and investigate the cybercrime economy from a design science perspective.
    - We first filtered the messages to select only those that carried significant risks and then divided them into categories. To determine if a given message is dangerous, our classification model checks whether it falls into which category.
    - Focuses on a real-time monitoring application that aims to monitor cybercrime- related discussions on social networks.
    - We will predict whether the data contains any cybercrime signature.
  1. **Reason for Literature Survey**
     + To gain an understanding of the existing research and debates relevant to a particular topicor area of study.
     + It helps one to build more knowledge in the field we want to do our project in i.e., CyberSecurity and Cybercrime.
     + Explains the essence of the underground economy that has grown around it.
     + Speculates on how the organization of criminal activity may evolve in cyberspace. It begins by examining organized crime in the "real world"; after defining "organized crime," it considers the advantages organization offers for group criminality in the "real world."

### Crimeware-as-a-Service—A Survey of Commoditized Crimeware in the Underground Market.

Crimeware-as-a-service (CaaS) has become a prominent component of the underground economy. CaaS provides a new dimension to cybercrime by making it more organized, automated, and accessible to criminals with limited technical skills. This paper dissects CaaS and explains the essence of the underground economy that has grown around it. The paper also describes the various crimeware services that are provided in the [underground market](https://www.sciencedirect.com/topics/computer-science/underground-market).

### Organized Cybercrime? How Cyberspace May Affect the Structure of Criminal Relationships

This article speculates on how the organization of criminal activity may evolve in cyberspace. It begins by examining organized crime in the "real world"; after defining "organized crime," it considers the advantages organization offers for group criminality in the "real world." The article then identifies the two models of organized criminal activity that have emerged in the "real world" - the "gang" model and the hierarchical American Mafia model - and explains why neither model is likely to establish itself in cyberspace. This portion of the article explains that both extant models evolved in response to empirical constraints that characterize activity in the "real world," constraints that are for the most part absent in cyberspace. The article then considers how organized criminal activity may manifest itself in the cyber world, drawing upon military analyses of netwar in so doing. It concludes that, as opposed to the fixed, hierarchical organizational models found in the "real world," criminal organization in the cyber world will be transient, lateral, and fluid, all of which can pose real challenges for law enforcement.

### Positioning and Presenting Design Science Research for Maximum Impact

Design science research (DSR) has staked its rightful ground as an important and legitimate Information Systems (IS) research paradigm. We contend that DSR has yet to attain its full potential impact on the development and use of information systems due to gaps in the understanding and application of DSR concepts and methods. This essay aims to help researchers (1) appreciate the levels of artifact abstractions that may be DSR contributions,

(2) identify appropriate ways of consuming and producing knowledge when they are preparing journal articles or other scholarly works, (3) understand and position the

knowledge contributions of their research projects, and (4) structure a DSR article so that it emphasizes significant contributions to the knowledge base. Our focal contribution is the DSR knowledge contribution framework with two dimensions based on the existing state of knowledge in both the problem and solution domains for the research opportunity under study. In addition, we propose a DSR communication schema with similarities to more conventional publication patterns, but which substitutes the description of the DSR artifact in place of a traditional results section. We evaluate the DSR contribution framework and the DSR communication schema via examinations of DSR exemplar publications.

### Design Science in Information Systems Research

Two paradigms characterize much of the research in the Information Systems discipline: behavioral science and design science. The behavioral-science paradigm seeks to develop and verify theories that explain or predict human or organizational behavior. The design- science paradigm seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts. Both paradigms are foundational to the IS discipline, positioned as it is at the confluence of people, organizations, and technology. Our objective is to describe the performance of design-science research in Information Systems via a concise conceptual framework and clear guidelines for understanding, executing, and evaluating the research. In the design-science paradigm, knowledge and understanding of a problem domain and its solution are achieved in the building and application of the designed artifact. Three recent exemplars in the research literature are used to demonstrate the application of these guidelines. We conclude with an analysis of the challenges of performing high-quality design-science research in the context of the broader IS community.

### A Design Science Research Methodology for Information Systems Research

The paper motivates, presents, demonstrates in use, and evaluates a methodology for conducting design science (DS) research in information systems (IS). DS is of importance in a discipline oriented to the creation of successful artifacts. Several researchers have pioneered DS research in IS, yet over the past 15 years, little DS research has been done within the discipline. The lack of a methodology to serve as a commonly accepted framework for DS research and of a template for its presentation may have contributed to its slow adoption. The design science research methodology (DSRM) presented here incorporates principles, practices, and procedures required to carry out such research and

meets three objectives: it is consistent with prior literature, it provides a nominal process model for doing DS research, and it provides a mental model for presenting and evaluating DS research in IS. The DS process includes six steps: problem identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation, and communication. We demonstrate and evaluate the methodology by presenting four case studies in terms of the DSRM, including cases that present the design of a database to support health assessment methods, a software reuse measure, an Internet video telephony application, and an IS planning method. The designed methodology effectively satisfies the three objectives and has the potential to help aid the acceptance of DS research in the IS discipline.

### Design Theory in Information Systems

The aim of this paper is to explore an important category of information systems knowledge that is termed “design theory”. This knowledge is distinguished as the fifth of five types of theory: (i) theory for analysing and describing, (ii) theory for understanding, (iii) theory for predicting, (iv) theory for explaining and predicting, and (v) theory for design and action. Examples of design theory in information systems are provided, with associated research methods. The limited understanding and recognition of this type of theory in information systems indicates that further debate concerning its nature and role in our discipline is needed.

### [The Anatomy of a Design Theory](https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1413&context=jais)

Design work and design knowledge in Information Systems (IS) is important for both research and practice. Yet there has been comparatively little critical attention paid to the problem of specifying design theory so that it can be communicated, justified, and developed cumulatively. In this essay we focus on the structural components or anatomy of design theories in IS as a special class of theory. In doing so, we aim to extend the work of Walls, Widemeyer and El Sawy (1992) on the specification of information systems design theories (ISDT), drawing on other streams of thought on design research and theory to provide a basis for a more systematic and useable formulation of these theories. We identify eight separate components of design theories: (1) purpose and scope, (2) constructs, (3) principles of form and function, (4) artifact mutability, (5) testable propositions, (6) justificatory knowledge (kernel theories), (7) principles of implementation, and (8) an expository

instantiation. This specification includes components missing in the Walls et al. adaptation of Dubin (1978) and Simon (1969) and also addresses explicitly problems associated with the role of instantiations and the specification of design theories for methodologies and interventions as well as for products and applications. The essay is significant as the unambiguous establishment of design knowledge as theory gives a sounder base for arguments for the rigor and legitimacy of IS as an applied discipline and for its continuing progress. A craft can proceed with the copying of one example of a design artifact by one artisan after another. A discipline cannot.

### The Novelty of ‘Cybercrime’An Assessment in Light of Routine Activity Theory

Recent discussions of ‘cybercrime’ focus upon the apparent novelty or otherwise of the phenomenon. Some authors claim that such crime is not qualitatively different from ‘terrestrial crime’, and can be analysed and explained using established theories of crime causation. One such approach, oft cited, is the ‘routine activity theory’ developed by Marcus Felson and others. This article explores the extent to which the theory’s concepts and aetiological schema can be transposed to crimes committed in a ‘virtual’ environment. Substantively, the examination concludes that, although some of the theory’s core concepts can indeed be applied to cybercrime, there remain important differences between ‘virtual’ and ‘terrestrial’ worlds that limit the theory’s usefulness. These differences, it is claimed, give qualified support to the suggestion that ‘cybercrime’ does indeed represent the emergence of a new and distinctive form of crime.

### Organised Crime Groups in Cyberspace: A Typology

Three categories of organised groups that exploit advances in information and communications technologies (ICT) to infringe legal and regulatory controls: (1) traditional organised criminal groups which make use of ICT to enhance their terrestrial criminal activities; (2) organised cybercriminal groups which operate exclusively online; and (3) organised groups of ideologically and politically motivated individuals who make use of ICT to facilitate their criminal conduct are described in this article. The need for law enforcement to have in-depth knowledge of computer forensic principles, guidelines, procedures, tools, and techniques, as well as anti-forensic tools and techniques will become more pronounced with the increased likelihood of digital content being a source of disputes or forming part of

underlying evidence to support or refute a dispute in judicial proceedings. There is also a need for new strategies of response and further research on analysing organised criminal activities in cyberspace.

### Social Change and Crime Rate Trends: A Routine Activity Approach

In this paper we present a "routine activity approach" for analyzing crime rate trends and cycles. Rather than emphasizing the characteristics of offenders, with this approach we concentrate upon the circumstances in which they carry out predatory criminal acts. Most criminal acts require convergence in space and time of likely offenders, suitable targets and the absence of capable guardians against crime. Human ecological theory facilitates an investigation into the way in which social structure produces this convergence, hence allowing illegal activities to feed upon the legal activities of everyday life. In particular, we hypothesize that the dispersion of activities away from households and families increases the opportunity for crime and thus generates higher crime rates. A variety of data is presented in support of the hypothesis, which helps explain crime rate trends in the United States 1947- 1974 as a byproduct of changes in such variables as labor force participation and single-adult households.

## Literature Forms

|  |  |
| --- | --- |
| **FORMS** | **COUNT** |
| 1. Article | 1 |
| 2. Survey Papers | 2 |
| 3. Journal | 2 |
| 4. Report | 1 |
| 5. Websites | 3 |
| 6. Research Papers | 2 |

**Table 1.1 Literature Resources**

## Literature Resources

|  |  |  |
| --- | --- | --- |
| **FORM TYPE** | **TITLE** | **INSIGHTS** |
| **1. ARTICLE** | Massive ransomware cyber-attack hits nearly 100 countries around the world. | This article is more than **5 years old** massive ransomware cyber-attack hits nearly100 countries around the world |
| **2. SURVEY PAPERS** | Crimeware-as-a- service - A survey of commoditized crimeware in the underground market. | The actors, value chains, and modes of operation in underground crimeware marketplaces are examined, and three facilitating technologies are identified that arelikely to significantly expand the reach of cybercriminals. |
| **3. JOURNAL** | The Novelty of ‘Cybercrime’: An Assessment in Light of Routine Activity Theory. | Recent discussions of ‘cybercrime’ focus uponthe apparent novelty or otherwise of the phenomenon. Some authors claim that such crime is not qualitatively different from ‘terrestrial crime’, and can be analyzed and explained using established theories of crime causation. |

**Table 1.2 Literature Resources**

# REQUIREMENT ANALYSIS AND SYSTEM

**SPECIFICATIONS**

## Feasibility Study

### Economic Feasibility

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, there is nominal expenditure and economic feasibility for certain.

### Operational Feasibility

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration.

### Technical Feasibility

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web-based user interface for audit workflow at NIC-CSD. Thus, it provides an easy access to. the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles.

## Software Requirement Specification

A Software Requirements Specification (SRS) – a requirements specification for a software system is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In

addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints). System requirements specification: A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development lifecycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* + - Business requirements describe in business terms what must be delivered or accomplished to provide value.
    - Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
    - Process requirements describe activities performed by the developing organization. For instance, process requirements could specify. Preliminary investigation examines project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

#### User Interfaces

The user interface of this system is a user-friendly python Graphical User Interface.

#### Hardware Interfaces

The interaction between the user and the console is achieved through python capabilities.

#### Software Interfaces

The required software is python. **Operating Environment** Windows XP.

**Software Requirements:**

* + - Operating System - Windows7/8
    - Programming Language - Python
    - Backend - Database

### Hardware Requirements:

* + - Processor - Pentium P–IV
    - Speed - 1.1 GHz
    - RAM - 256 MB (min)
    - Hard Disk - 20 GB

## Validation

Validation of Enhanced Cybercrime Underground Economy using machine learning approaches involves ensuring that the software product meets the specified requirements and performs as expected. In this context, validation is conducted through various testing methods, including functional testing, regression testing, performance testing, and security testing. Functional testing ensures that the software is performing the functions it was designed to do. In the case of Enhanced Cybercrime Underground Economy, this could involve testing that the machine learning algorithms are accurately identifying cybercrime activities and providing meaningful insights. Regression testing ensures that any changes or updates to the software have not introduced new errors or issues. Performance testing evaluates the software's performance under different workloads and usage scenarios, while security testing ensures that the software is secure and protected against cyber threats. In addition to testing, validation of Enhanced Cybercrime Underground Economy using machine learning approaches also involves reviewing the software documentation and user feedback to ensure that the software product meets the needs of its intended users. This may involve working with experts in the field of cybercrime to ensure that the software product accurately represents the underground economy and provides valuable insights to law enforcement agencies.

## Expected Hurdles

* + - **Data Availability:** One of the most significant challenges in data analytics is obtaining the necessary data. Cybercriminal activities are often conducted in secrecy and on platforms that are not accessible to the public or law enforcement agencies. As a result, obtaining data on cybercrime activities can be a challenge.
    - **Data Security:** Handling sensitive and confidential data requires a high level of security. Data breaches and cyber-attacks can occur, and it is essential to ensure that appropriate security measures are in place to protect the data.
    - **Complexity:** The cybercrime underground economy is complex, and understanding the relationships between different actors and their activities can be challenging. It is essential to have a deep understanding of the cybercrime landscape to identify patterns and relationships in the data.
    - **Legal and Ethical Issues:** The use of data analytics in the cybercrime underground economy raises legal and ethical questions. Privacy concerns and ethical considerations must be taken into account when using data analytics to investigate cybercrime activities.
    - **Data Quality**: Even if the data is available, it may not be of high quality. The data may be incomplete, inconsistent, or even intentionally misleading. Ensuring that the data used in the analysis is of high quality is critical to obtaining accurate and actionable insights.

## SDLC Model Used

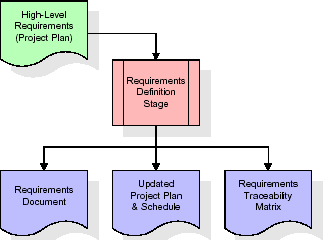
Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

### Stages in SDLC

* + - Requirement Gathering
    - Analysis
    - Designing
    - Coding
    - Testing
    - Maintenance

### Requirements Gathering Stage

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



#### Fig 2.1 Requirements Gathering Stage

These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

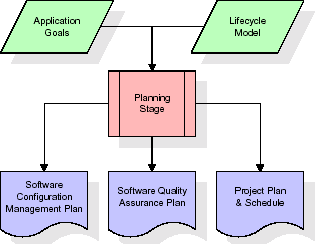
In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* + - Feasibility study is all about identification of problems in a project.
    - No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
    - Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

### Analysis Stage

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.

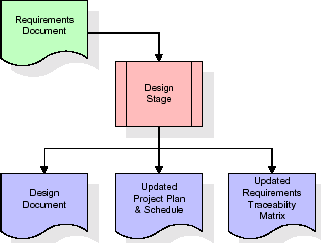


#### Fig 2.2 Analysis Stage

The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high-level estimates of effort for the out stages.

### Designing Stage

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

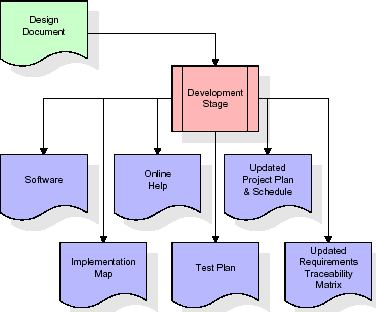


#### Fig 2.3 Designing Stage

When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

### Development (Coding) Stage

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.

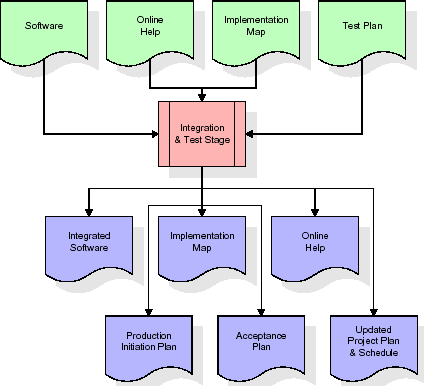


#### Fig 2.4 Development Stage

The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

### Integration & Test Stage

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

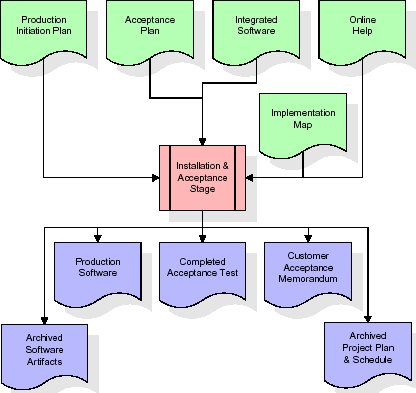


#### Fig 2.5 Integration and Test Stage

The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

### Installation & Acceptance Test

During the installation and acceptance stage, the software artefacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer. After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



#### Fig 2.6 Installation & Acceptance Test

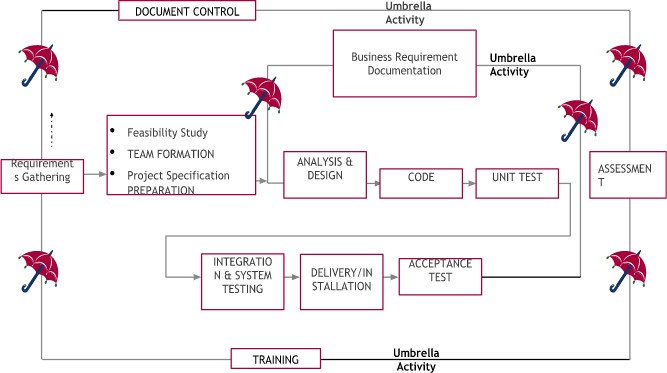
The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labour data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

### Maintenance

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

### Umbrella Model

The Umbrella Model is a software development methodology that combines the best practices of various SDLC models to create a flexible and adaptable approach to software development. The Umbrella Model consists of three main phases: Planning, Execution, and Closure. In the Planning Phase, project goals and requirements are defined, and a project plan is developed. In the Execution Phase, the development team works to create the software product, using various SDLC models as needed. The Closure Phase involves testing, documentation, and training, and concludes with the delivery of the software product to the client. The Umbrella Model emphasizes communication and collaboration throughout the software development process, allowing for flexibility and adaptability to meet the evolving needs of the project. By incorporating best practices from various SDLC models, the Umbrella Model enables organizations to choose the approach that best fits their needs, ultimately leading to successful software development projects.



**Fig 2.7 SDLC Umbrella Model**

# SYSTEM DESIGN

## Design Approach

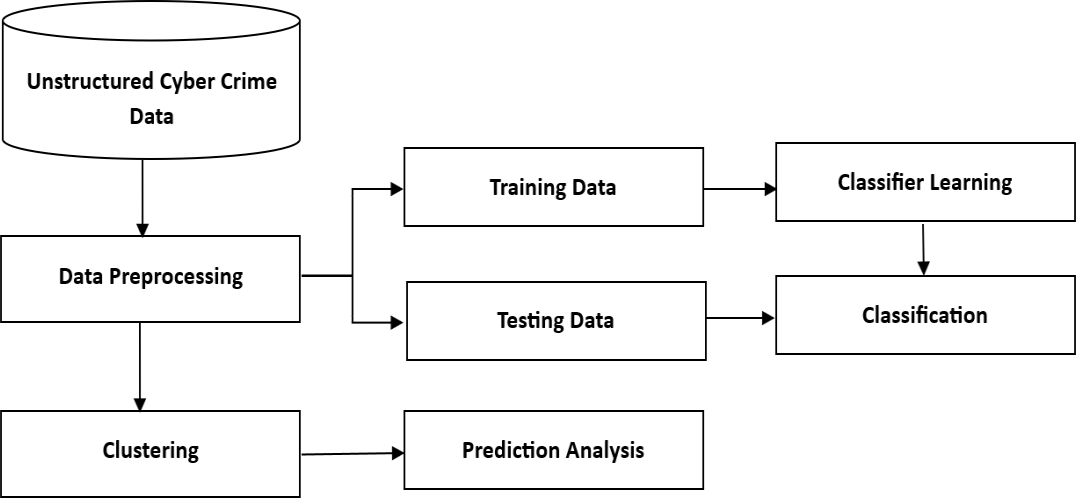
A Software Requirements Specification (SRS) – a requirements specification for a software system is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition, to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements that impose constraints on the design or implementation System requirements specification: A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of clients and stakeholders to help identify business problems and propose solutions. Within the systems development lifecycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers.

## Detail Design

This study takes a design science research (DSR) approach. Design science “creates and evaluates information technology artifacts intended to solve identified problems”. DSR involves developing a range of IT artifacts, such as decision support systems, models, frameworks, tools, methods, and applications. Where behavioral science research seeks to develop and justify theories that explain or predict human or organizational phenomena, DSR seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts. DSR’s contribution is to add value to the literature and practice in terms of “design artifacts, design construction knowledge and/or design evaluation knowledge.

## Architectural Model

Architectural diagram is a visual representation that maps out the physical implementation for components. In a software system, the term architecture refers to various functions, their implementations, and their interactions with each other.

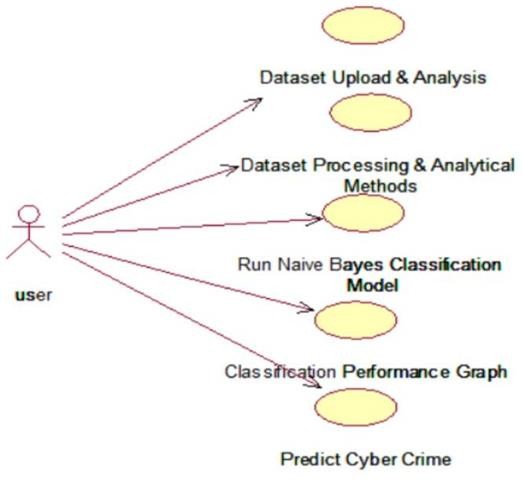


**Fig 3.1 The Architectural Model**

### UML Diagrams

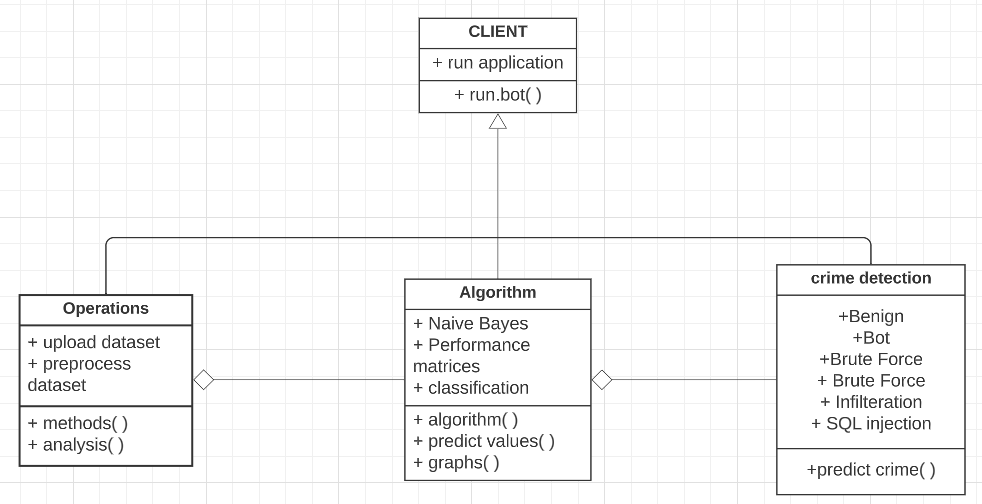
#### Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system and depicts the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.



#### Fig 3.2 Use-case Diagram

1. **Class Diagram**

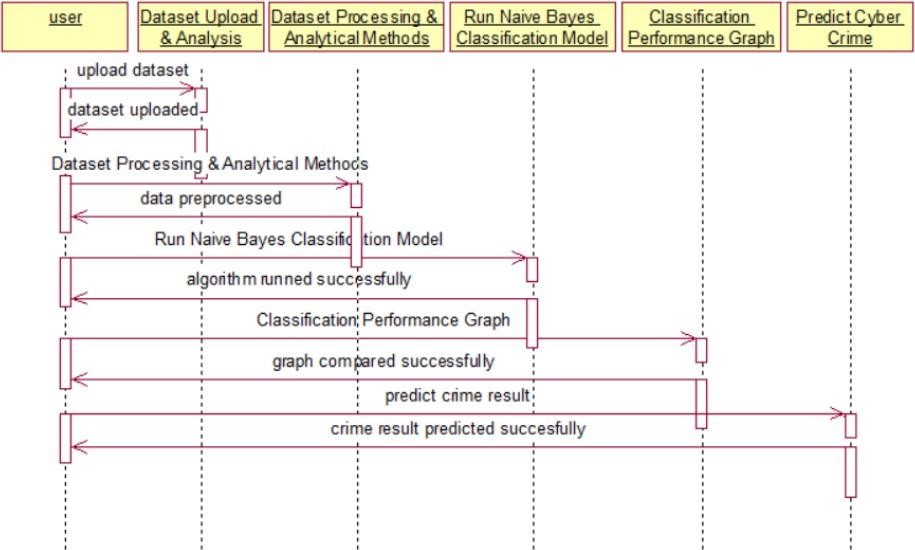
The class diagram is the main building block of object-oriented modeling. It is used both for general conceptual modeling of the system of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

#### Fig 3.3 Class Diagram

1. **Sequence Diagram**

A sequence diagram 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. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects neededto carry out the functionality of the scenario.

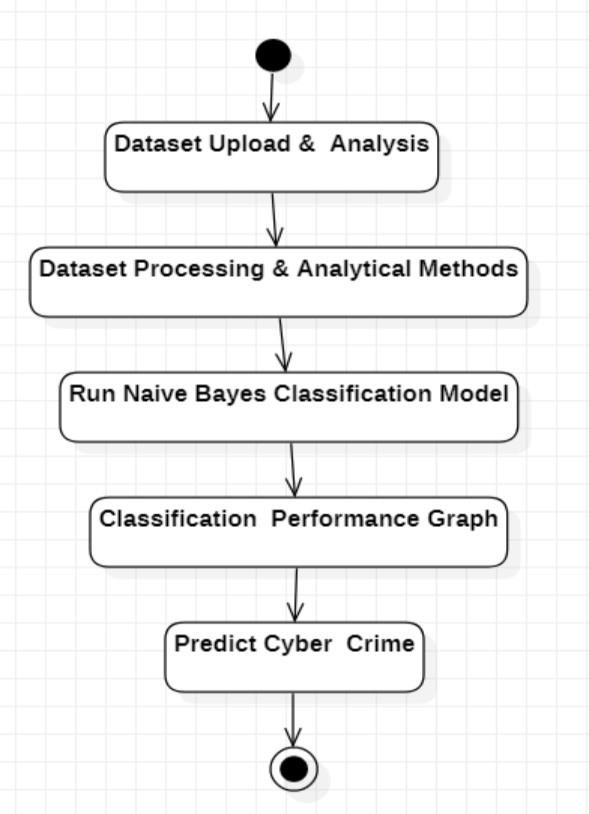
Sequence diagrams are typically associated withuse case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



#### Fig 3.4 Sequence Diagram

1. **Activity Diagram**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flowchart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So, the control flow is drawn fromone operation to another. This flow can be sequential, branched or concurrent.

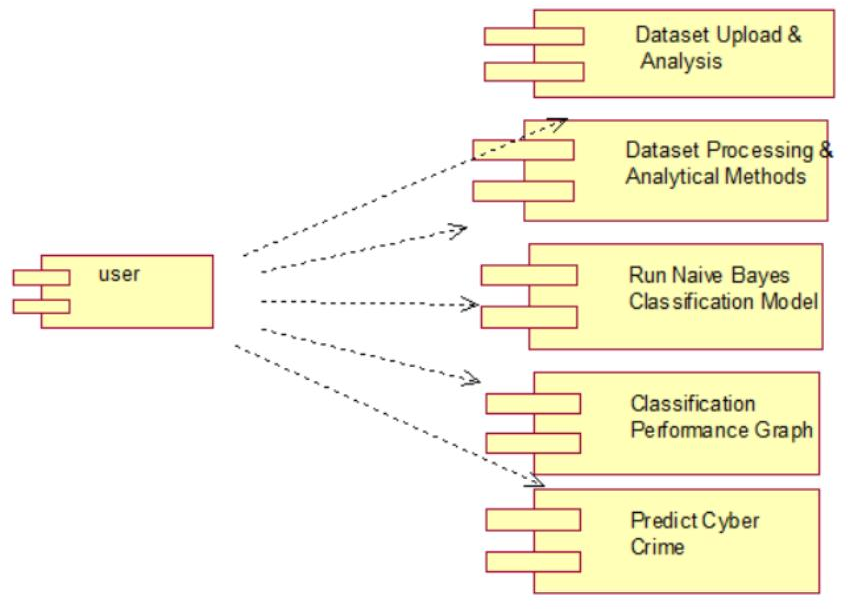


#### Fig 3.5 Activity Diagram

1. **Component Diagram**

In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.

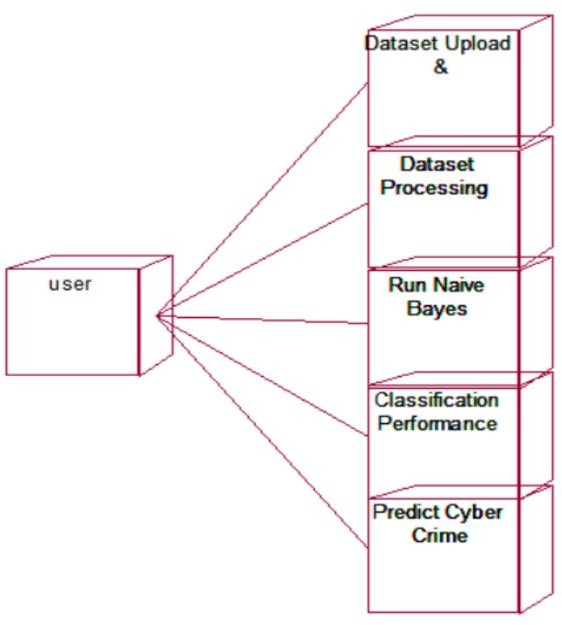


**Fig 3.6 Component Diagram**

### Deployment Diagram

A deployment diagram in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g., JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

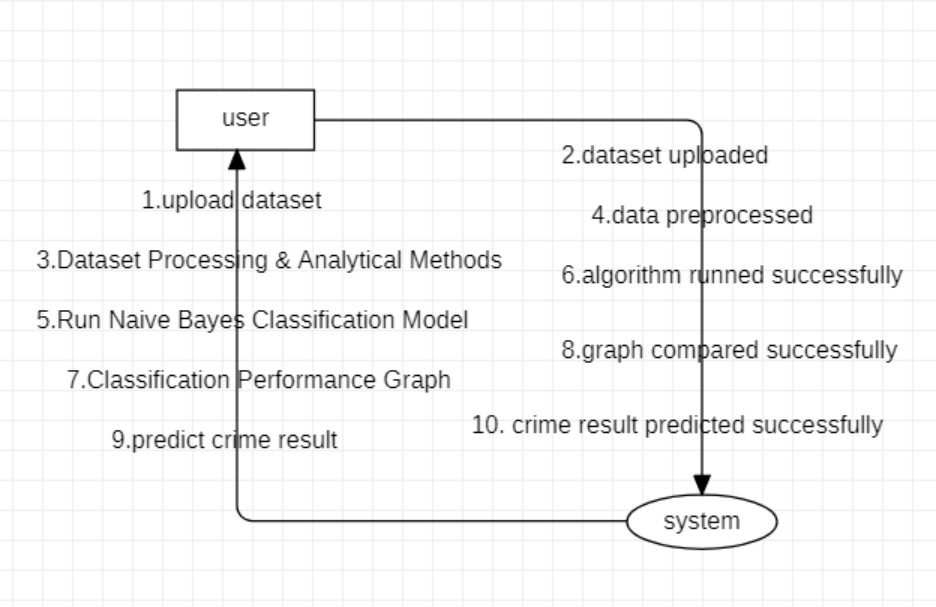


#### Fig 3.7 Deployment Diagram

1. **Data Flow Diagram**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy- to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model.



**Fig 3.8 Data Flow Diagram**

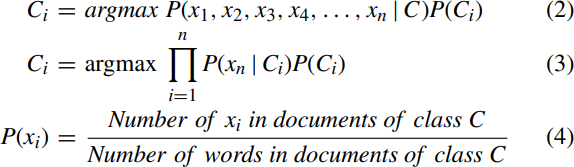
## Algorithm

#### Naive Bayes Classifier Algorithm

We employ the Naive Bayes algorithm, a probabilistic classification algorithm that addresses probabilistic reasoning under uncertainty, because it is the simplest approach for text classification. Its predictions self-correct as new information is encountered, so they become more accurate with more data. The conditional probability is given by Bayes’ theorem:



Here, P(Cj) and P(Ci |d) are the prior and posterior probabilities of class Ci , while P(d) and P(d|Ci) are the prior and posterior probabilities of the predictor d. The dependent feature vector is x = (x1, x2, . . . , xn) and Bayes’ theorem gives us the following.



Basing the probabilistic classifier on the naïve Bayes model simplifies the conditional independence assumptions for the CaaS and crimeware classes. The likelihood of the document having feature xi can then be computed by dividing ‘‘the number of features xi in documents of class C’’ by ‘‘the number of words in documents of class C’’ (Equation 4).

## User Interface Design

The user interface design of enhanced cybercrime underground economy using machine learning approaches should aim to provide a simple and user-friendly experience for analysts to interact with the system. The interface should be organized in a clear and intuitive manner, with all relevant data easily accessible. Visualizations should be incorporated to help analysts understand complex data and identify patterns, including charts, graphs, and maps. Customization features should allow analysts to adjust the layout, choose which data to display, and set up alerts for specific events. Collaboration features should also be included, such as chat, comment threads, and shared workspaces, to facilitate teamwork among analysts. Finally, security should be a top priority, with multi-factor authentication, user roles and permissions, and encryption of sensitive data. By focusing on these aspects, the user interface can provide a powerful tool for analysts to explore and understand the cybercrime underground economy.

## Methodology

#### Dataset Upload & Analysis

Using this module, we will upload the dataset and then perform analysis methods such as finding various cybercrime and their count and then clean the dataset by removing missing values.We have downloaded our dataset from Kaggle.

#### Kaggle Dataset:

Kaggle allows users to find datasets they want to use in building AI models, publish datasets, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges.

#### Dataset Processing & Analytical Methods:

Using this module, we will encode attack labels with integer ID and then split dataset into trainand test where application used 80% dataset to train classification Naïve Bayes algorithm and20% test its prediction performance.

#### Run Naive Bayes Classification Model:

Using this module, we will train classification algorithm with above 80% dataset and then build a prediction model.

#### Classification Performance Graph:

Using this module, we will plot naïve Bayes accuracy and precision graph to know its performance accuracy.

#### Predict Cyber Crime:

Using this module, we will upload a test cybercrime network dataset and then the classificationmodel will predict whether the test data contains any cybercrime signature.

# IMPLEMENTATION, TESTING AND MAINTENANCE

## Introduction to Languages, IDE’s used for Implementation

### Python

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

### History of Python

Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980s and was first released in February 1991.

#### Why Python was Created?

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to- understand syntax) that could access the Amoeba system calls. So, he decided to create a language that was extensible. This led to design of a new language which was later named Python.

#### Why the Name Python?

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

### Features of Python:

#### A Simple Language which is Easier to Learn

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun

and allows you to focus on the solution rather than syntax. If you are a newbie, it's a great choice to start your journey with Python.

#### Free and Open-Source

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software’s written in it, you can even make changes to the Python's source code. Python has a large community constantly improving it in each iteration.

#### Portability

You can move Python programs from one platform to another, and run it without any changes. It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

#### Extensible and Embeddable

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.

This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

#### A high-level, interpreted language

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on. Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

#### Large Standard Libraries to Solve Common Tasks

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQLdb.

Standard libraries in Python are well tested and used by hundreds of people. So, you can be sure that it won't break your application.

#### Object-Oriented

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively. With OOP, you are able to divide these complex problems into smaller sets by creating objects.

### Applications of Python:

#### Simple Elegant Syntax

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural.

#### Not overly strict

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

#### Expressiveness of the language

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

### Tools and Technologies used for Implementation

#### Machine Learning

Machine learning is a branch of [artificial intelligence (AI)](https://www.ibm.com/in-en/topics/artificial-intelligence) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. Machine learning is an important component of the growing field of data science. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expand and grow, the market demandfor data scientists will increase.

#### Tkinter

Tkinter is a library written in Python that is widely used to create GUI applications. It is very easy to build a GUI using Tkinter and the process is even faster. Tkinter has several widgets that can be used while developing GUI. These include buttons, radio buttons, checkboxes.

#### Pandas

Pandas is an open-source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named [Numpy,](https://www.activestate.com/products/python/python-packages/) which provides support for multi-dimensional arrays.

#### Numpy

Numpy is a library for the Python programming language, adding support for large, multi- dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. Moreover, Numpy forms the foundation of the Machine Learning stack.

#### Matplotlib

Matplotlib is one of the most popular and oldest plotting libraries in Python which is used in Machine Learning. In Machine learning, it helps to understand the huge amount of data through different visualizations.

#### 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 consistentinterface in Python.

## Coding Standards of Language Used

#### Imports, Blank Lines, and the Indentations

The import should be in a particular sequence. At first, the standard libraries, then the third party, and at the last, the local libraries should be imported. The preferred method of indentation is spaces, the 4 spaces indentation is accepted and accurate.

#### The Length of the Line and the Line Breaks

The length of the line should not be greater than 79 characters. In the case of docstrings and comments where a block of text is large, it is limited to 72 characters.

#### Whitespaces, Trailing Commas, and String Quotes

One should avoid extra white spaces, there must be a single white space around both sides of an operator, one after the comma and none inside opening or closing of parenthesis.

#### Naming Conventions

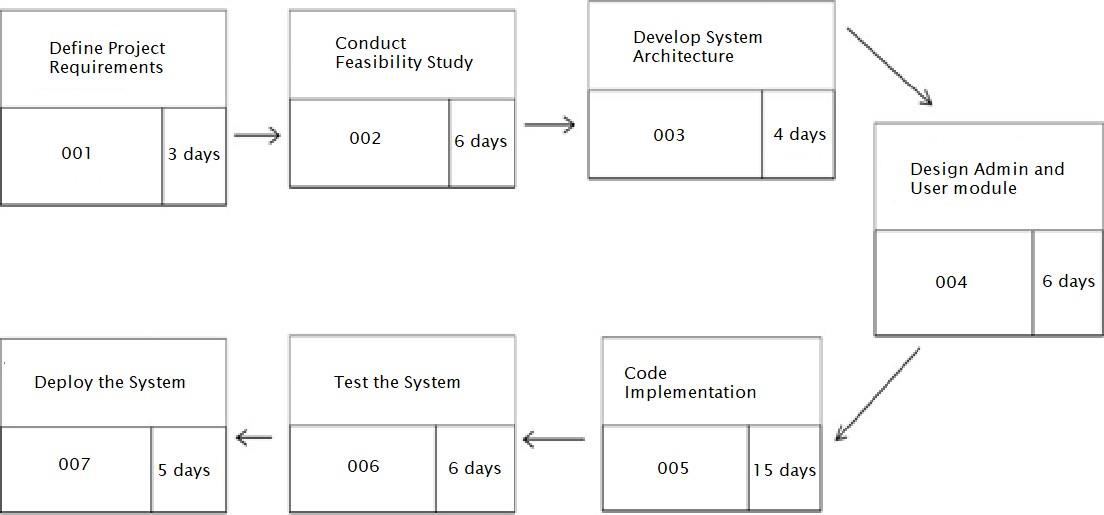
Use grammatically correct variable names, the class name should start with an uppercase and must follow camelCase convention If more than two words are to be used. In the same way, a function name should be joined with an underscore, and it must be lowercase.

## Project Scheduling

Project scheduling is a process of determining the start and end dates of project activities, defining the sequence of work, and allocating resources to complete the project within a specified timeframe. The purpose of project scheduling is to create a plan for executing and completing a project on time, within budget, and with the desired quality standards.

The project schedule serves as a roadmap for the project team, providing a clear understanding of what needs to be done, when it needs to be done, and who is responsible for doing it. It helps to track progress, identify potential delays and risks, and make necessary adjustments to keep the project on track. Effective project scheduling is crucial for the success of any project, as it ensures that the project is completed on time, within budget, and to the desired quality standards.

PERT (Program Evaluation and Review Technique) - PERT is a statistical tool used to analyze and represent the tasks and activities required to complete a project. It uses a network diagram to represent the project's critical path and determines the expected duration of the project. The PERT diagram shows the sequence of tasks, the duration of each task, and the dependencies between tasks.



**Fig 4.1 Project Scheduling**

## Testing Techniques and Test Plans

Software testing methodologies are ways to plan and carry out tests to assess software applications. We have employed the following testing methodologies in out project-:

#### Implementation and Testing

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive.

Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested tothe satisfaction of the user.

#### Implementation

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifying as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix

printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

#### Testing

Testing is the process where the test data is prepared and is used for testing the modules individuallyand later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property function as a unit. The test data should be chosen suchthat it passed through all possible condition. Actually, testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out duringthe testing period.

#### System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to use the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus, the code was exhaustively checked for all possible correct data and the outcomes were also checked.

#### Module Testing

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus, all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example, the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are

prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

#### Integration Testing

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modulesare connected and tested. The testing results are very correct. Thus, the mapping of jobs with resources is done correctly by the system.

#### Acceptance Testing

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptableand ready for the operation.

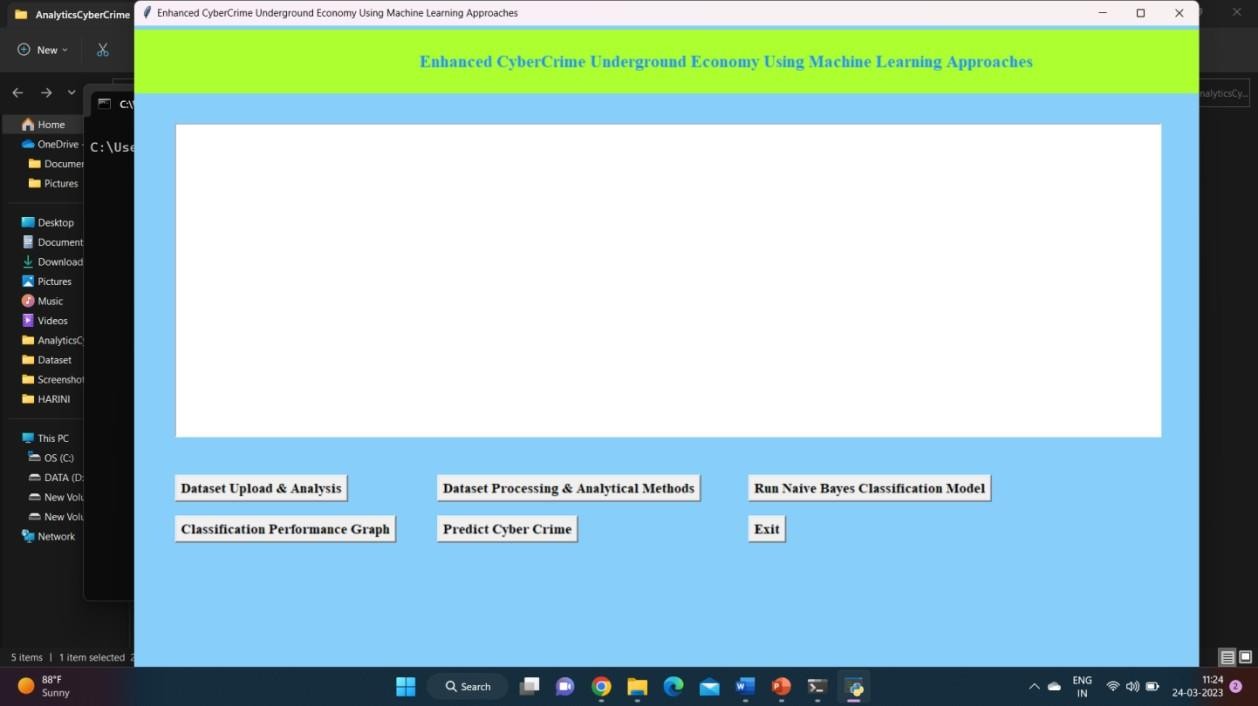
## Test Cases

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test**  **Case Id** | **Test Case Name** | **Test Case Desc.** | **Test Steps** | | | **Test**  **Case Status** | **Test Priority** |
| **Step** | **Expected** | **Actual** |
| 01 | Dataset Upload & Analysis | Test whether the Dataset is uploaded or not. | If the Dataset is not uploaded | we cannot do further operations. | If the Dataset uploaded, we will do further operations | Pass | High |
| 02 | Dataset Processing & Analytical Methods | Verify the Either Dataset is  Preprocess or not into the System. | If Dataset may not Preprocess. | We cannot do the further operations. | Dataset is Preprocessed | Pass | High |
| 03 | Run Naive Bayes Classification Model | Verify the Run Naive Bayes Classification algorithm will run or not | Without training model | We cannot run Naive Bayes Classificatio n algorithm | We can run Naive Bayes Classification algorithm | Pass | High |
| 04 | Classification Performance Graph | Test whether the Graph is displaying or not. | Without displaying graph | We cannot do further operations | We can do further operations | Pass | High |
| 05 | Predict Cyber Crime | Verify whether the data is tested or not | Without Predicting result | We cannot get accuracy results | We can get accuracy results | Pass | High |

**Table 4.1 Test Cases**

# RESULTS AND DISCUSSIONS

## User Interface Representation



**Fig 5.1: User Interface**

### Brief Description of Various Modules of the System

#### Defining Goals

The first step is to identify the conceptual scope of the analysis. Specifically, this step identifies the analysis context, namely the objectives and goals. To gain an in-depth understanding of the current CaaS research, we investigated the cybercrime underground, which operates as a closed community. Thus, the goal of the proposed framework is to “investigate the cybercrime underground economy.”

#### Identifying Sources

The second step is to identify the data sources, based on the goals defined by Step 1. This step should consider what data is needed and where it can be obtained. Since the goal of this study is to investigate the cybercrime underground, we consider data on the cybercrime

underground community. We, therefore collected such data from the community itself and obtained a malware databasefrom a leading global cybersecurity research firm.

#### Selecting Analytical Methods

A diverse range of items are sold in the cybercrime underground, with different degrees of associated risk. we focused mainly on items critical to hacking. We first filtered the messages to select only those that carried significant risks, and then divided them into categories.

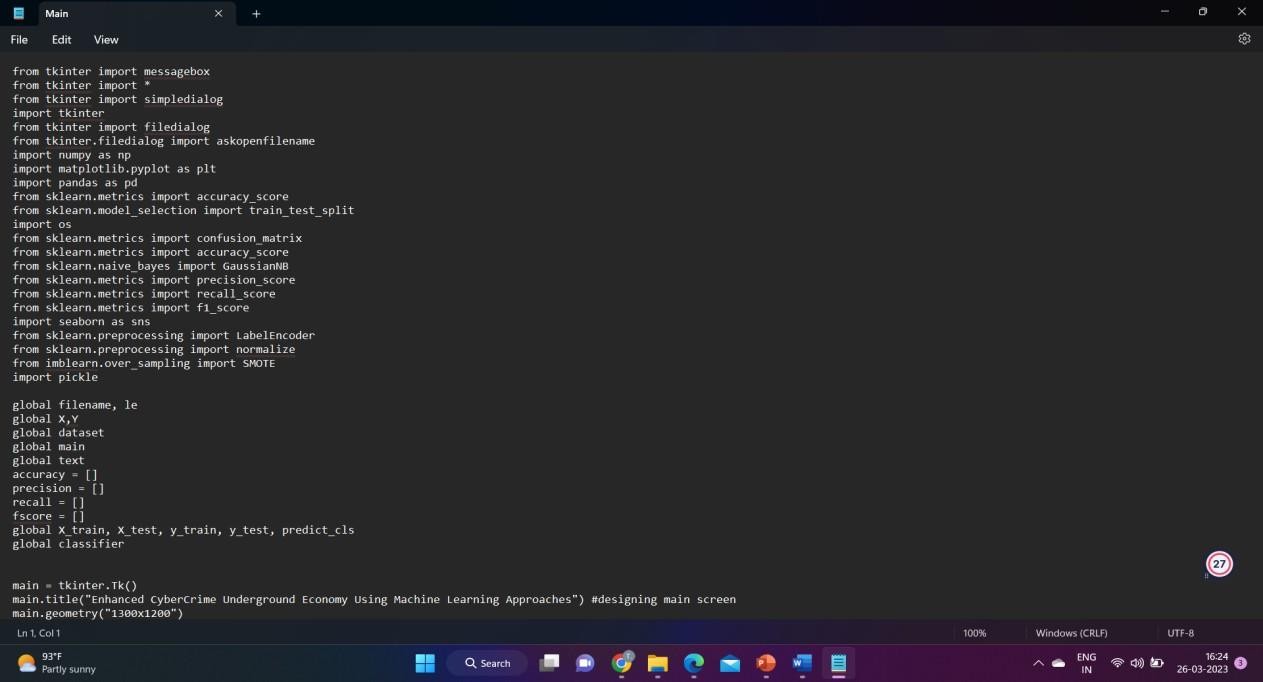
#### Implementing an Application

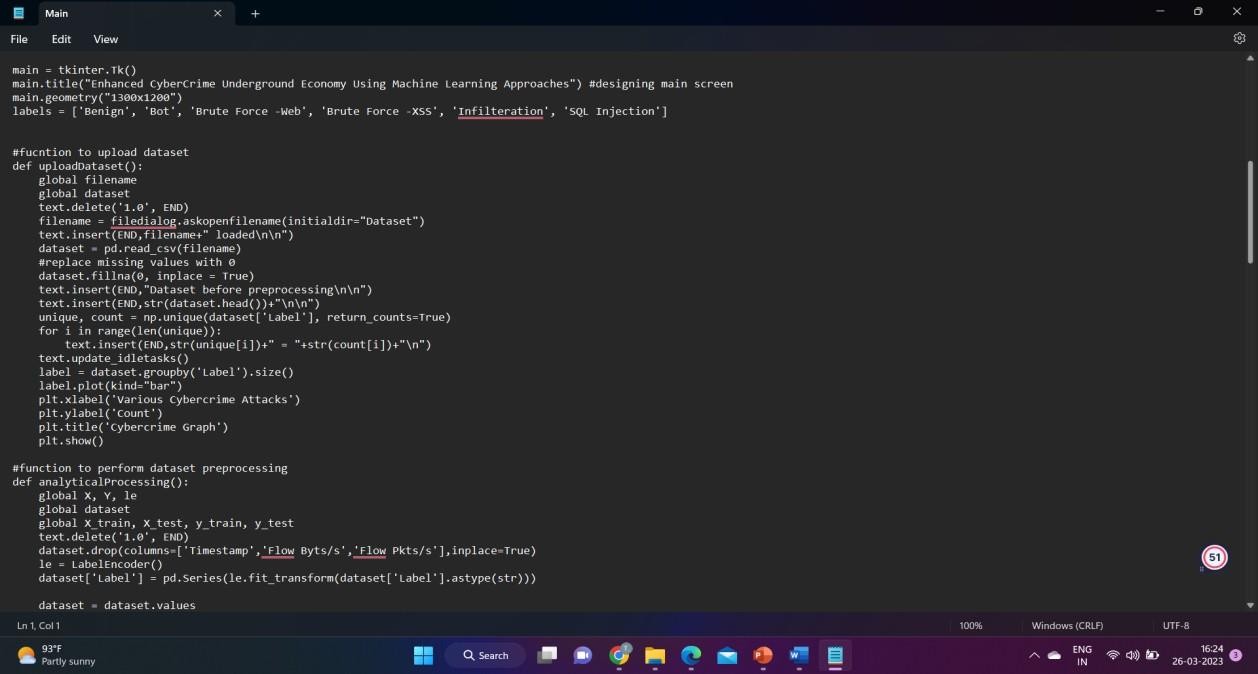
Although organizations emphasize the measures, they take to prevent cybercrime, their overalleffectiveness has yet to be empirically demonstrated in practice. In the last step of our framework, we demonstrate the use of the proposed CaaS and crimeware definitions, classification model, and analysis framework.

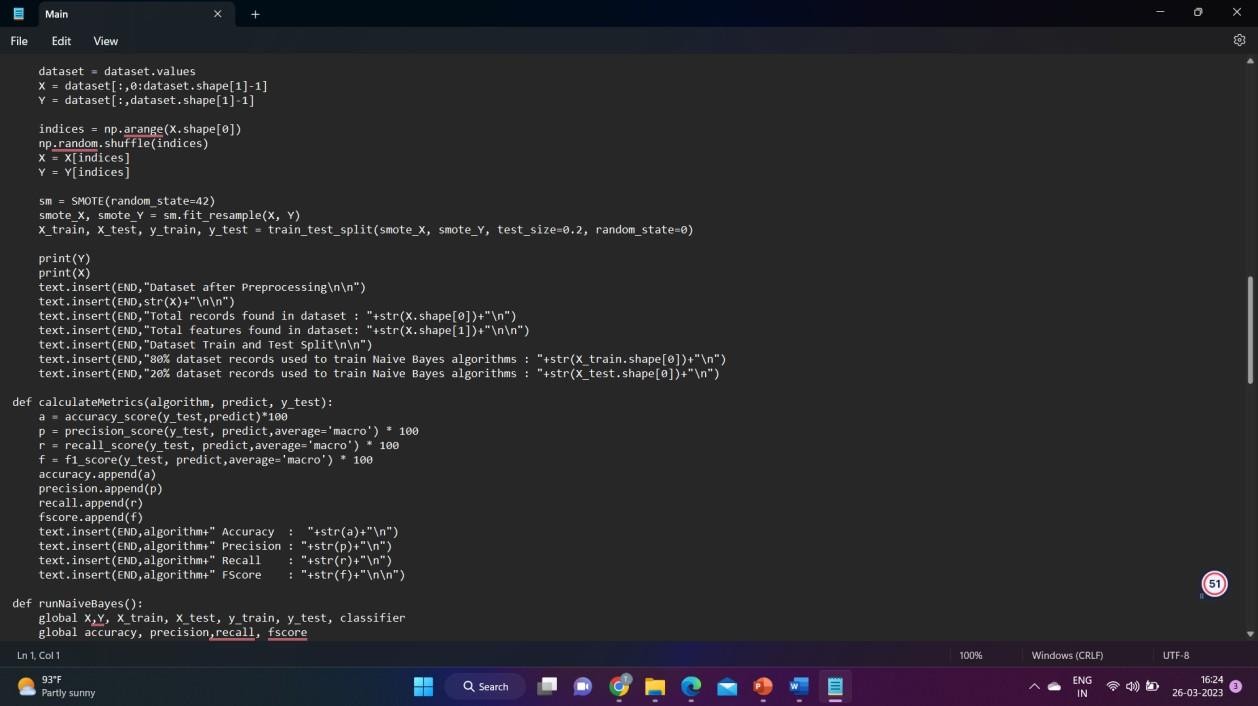
#### Analysis And Results Data

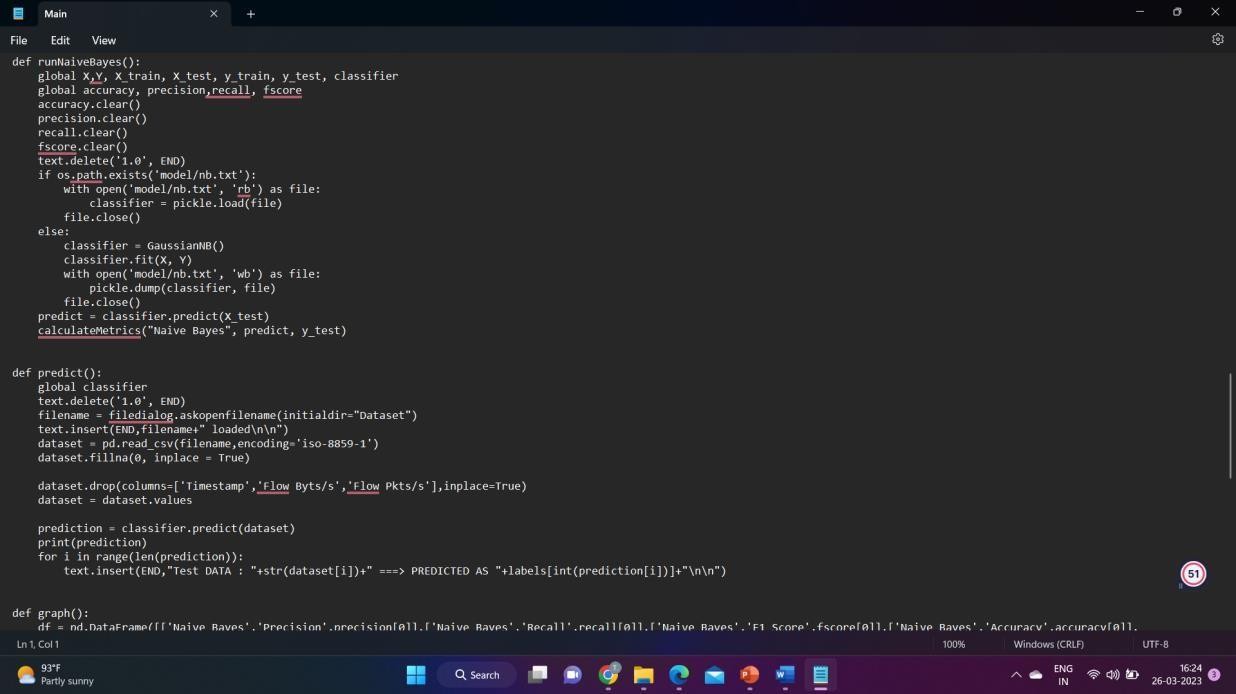
The data analysis step of the proposed framework involves four steps. Here, we report the data analysis results: CaaS and crimeware classification and market trends, cybercrime market dynamics, and potential hacking targets.

## Coding









### Sample Code

from tkinter import messagebox from tkinter import \*

from tkinter import simpledialog import tkinter

from tkinter import filedialog

from tkinter.filedialog import askopenfilename import numpy as np

import matplotlib.pyplot as plt import pandas as pd

from sklearn.metrics import accuracy\_score

from sklearn.model\_selection import train\_test\_split import os

from sklearn.metrics import confusion\_matrix from sklearn.metrics import accuracy\_score from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import precision\_score from sklearn.metrics import recall\_score from sklearn.metrics import f1\_score import seaborn as sns

from sklearn.preprocessing import LabelEncoder from sklearn.preprocessing import normalize from imblearn.over\_sampling import SMOTE import pickle

global filename, le global X,Y

global dataset global main global text accuracy = [] precision = [] recall = [] fscore = []

global X\_train, X\_test, y\_train, y\_test, predict\_cls global classifier

main = tkinter.Tk()

main.title("Enhanced CyberCrime Underground Economy Using Machine Learning Approaches") #designing main screen

main.geometry("1300x1200")

labels = ['Benign', 'Bot', 'Brute Force -Web', 'Brute Force -XSS', 'Infilteration', 'SQL Injection'] #fucntion to upload dataset

def uploadDataset(): global filename global dataset

text.delete('1.0', END)

filename = filedialog.askopenfilename(initialdir="Dataset") text.insert(END,filename+" loaded\n\n")

dataset = pd.read\_csv(filename) #replace missing values with 0

dataset.fillna(0, inplace = True) text.insert(END,"Dataset before preprocessing\n\n") text.insert(END,str(dataset.head())+"\n\n")

unique, count = np.unique(dataset['Label'], return\_counts=True) for i in range(len(unique)):

text.insert(END,str(unique[i])+" = "+str(count[i])+"\n") text.update\_idletasks()

label = dataset.groupby('Label').size() label.plot(kind="bar") plt.xlabel('Various Cybercrime Attacks') plt.ylabel('Count')

plt.title('Cybercrime Graph') plt.show()

#function to perform dataset preprocessing def analyticalProcessing():

global X, Y, le global dataset

global X\_train, X\_test, y\_train, y\_test text.delete('1.0', END)

dataset.drop(columns=['Timestamp','Flow Byts/s','Flow Pkts/s'],inplace=True) le = LabelEncoder()

dataset['Label'] = pd.Series(le.fit\_transform(dataset['Label'].astype(str))) dataset = dataset.values

X = dataset[:,0:dataset.shape[1]-1] Y = dataset[:,dataset.shape[1]-1] indices = np.arange(X.shape[0]) np.random.shuffle(indices)

X = X[indices] Y = Y[indices]

sm = SMOTE(random\_state=42)

smote\_X, smote\_Y = sm.fit\_resample(X, Y)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(smote\_X, smote\_Y, test\_size=0.2,

random\_state=0) print(Y) print(X)

text.insert(END,"Dataset after Preprocessing\n\n") text.insert(END,str(X)+"\n\n")

text.insert(END,"Total records found in dataset : "+str(X.shape[0])+"\n") text.insert(END,"Total features found in dataset: "+str(X.shape[1])+"\n\n") text.insert(END,"Dataset Train and Test Split\n\n")

text.insert(END,"80% dataset records used to train Naive Bayes algorithms : "+str(X\_train.shape[0])+"\n")

text.insert(END,"20% dataset records used to train Naive Bayes algorithms : "+str(X\_test.shape[0])+"\n")

def calculateMetrics(algorithm, predict, y\_test): a = accuracy\_score(y\_test,predict)\*100

p = precision\_score(y\_test, predict,average='macro') \* 100 r = recall\_score(y\_test, predict,average='macro') \* 100

f = f1\_score(y\_test, predict,average='macro') \* 100 accuracy.append(a)

precision.append(p) recall.append(r) fscore.append(f)

text.insert(END,algorithm+" Accuracy : "+str(a)+"\n") text.insert(END,algorithm+" Precision : "+str(p)+"\n") text.insert(END,algorithm+" Recall : "+str(r)+"\n") text.insert(END,algorithm+" FScore : "+str(f)+"\n\n")

def runNaiveBayes():

global X,Y, X\_train, X\_test, y\_train, y\_test, classifier global accuracy, precision,recall, fscore accuracy.clear()

precision.clear() recall.clear() fscore.clear() text.delete('1.0', END)

if os.path.exists('model/nb.txt'):

with open('model/nb.txt', 'rb') as file:

classifier = pickle.load(file) file.close()

else:

classifier = GaussianNB() classifier.fit(X, Y)

with open('model/nb.txt', 'wb') as file: pickle.dump(classifier, file) file.close()

predict = classifier.predict(X\_test) calculateMetrics("Naive Bayes", predict, y\_test)

def predict(): global classifier

text.delete('1.0', END)

filename = filedialog.askopenfilename(initialdir="Dataset") text.insert(END,filename+" loaded\n\n")

dataset = pd.read\_csv(filename,encoding='iso-8859-1') dataset.fillna(0, inplace = True)

dataset.drop(columns=['Timestamp','Flow Byts/s','Flow Pkts/s'],inplace=True) dataset = dataset.values

prediction = classifier.predict(dataset) print(prediction)

for i in range(len(prediction)):

text.insert(END,"Test DATA : "+str(dataset[i])+" ===> PREDICTED AS "+labels[int(prediction[i])]+"\n\n")

def graph():

df = pd.DataFrame([['Naive Bayes','Precision',precision[0]],['Naive Bayes','Recall',recall[0]],['Naive Bayes','F1 Score',fscore[0]],['Naive Bayes','Accuracy',accuracy[0]], columns=['Algorithms','Performance Output','Value'])

df.pivot("Algorithms", "Performance Output", "Value").plot(kind='bar')

plt.show() def close():

main.destroy()

font = ('times', 16, 'bold')

title = Label(main, text='Enhanced CyberCrime Underground Economy Using Machine Learning Approaches')

title.config(bg='greenyellow', fg='dodger blue') title.config(font=font)

title.config(height=3, width=120) title.place(x=0,y=5)

font1 = ('times', 12, 'bold') text=Text(main,height=20,width=150) scroll=Scrollbar(text) text.configure(yscrollcommand=scroll.set) text.place(x=50,y=120) text.config(font=font1)

font1 = ('times', 13, 'bold')

uploadButton = Button(main, text="Dataset Upload & Analysis", command=uploadDataset) uploadButton.place(x=50,y=550)

uploadButton.config(font=font1)

processButton = Button(main, text="Dataset Processing & Analytical Methods", command=analyticalProcessing)

processButton.place(x=370,y=550) processButton.config(font=font1)

nbButton = Button(main, text="Run Naive Bayes Classification Model", command=runNaiveBayes)

nbButton.place(x=750,y=550) nbButton.config(font=font1)

graphButton = Button(main, text="Classification Performance Graph", command=graph) graphButton.place(x=50,y=600)

graphButton.config(font=font1)

predictButton = Button(main, text="Predict Cyber Crime", command=predict) predictButton.place(x=370,y=600)

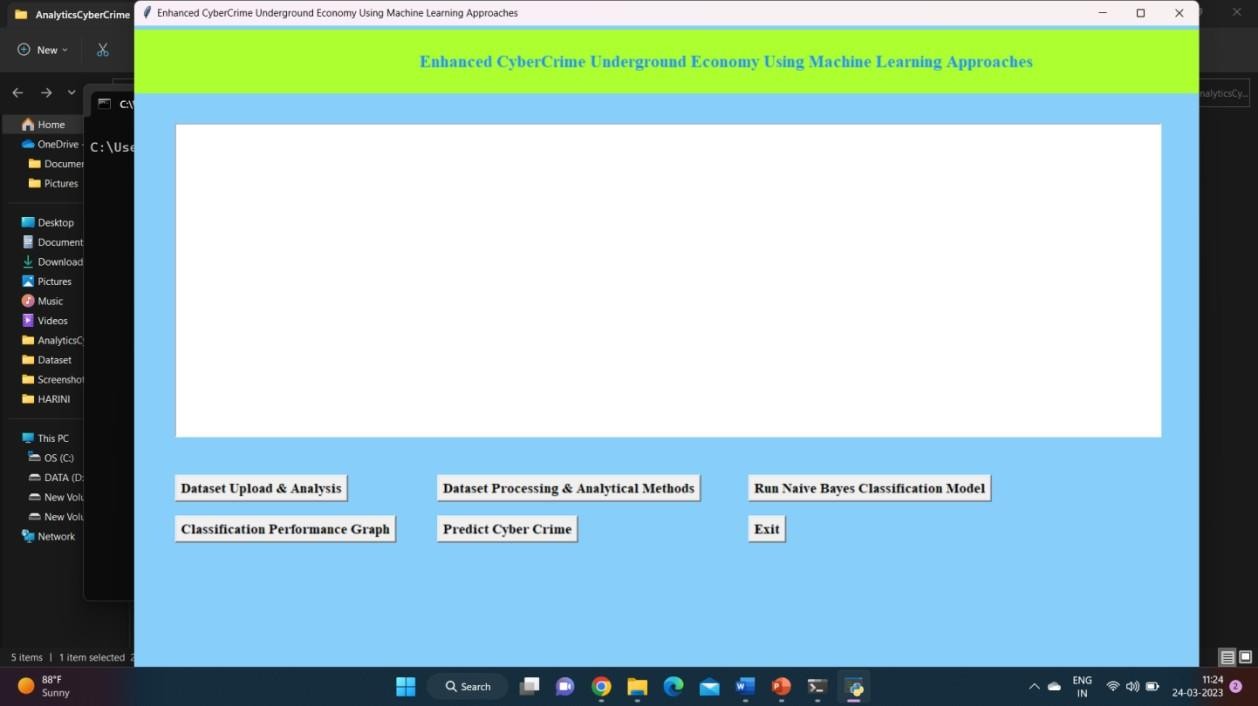
predictButton.config(font=font1)

closeButton = Button(main, text="Exit", command=close) closeButton.place(x=750,y=600) closeButton.config(font=font1) main.config(bg='LightSkyBlue')

main.mainloop()

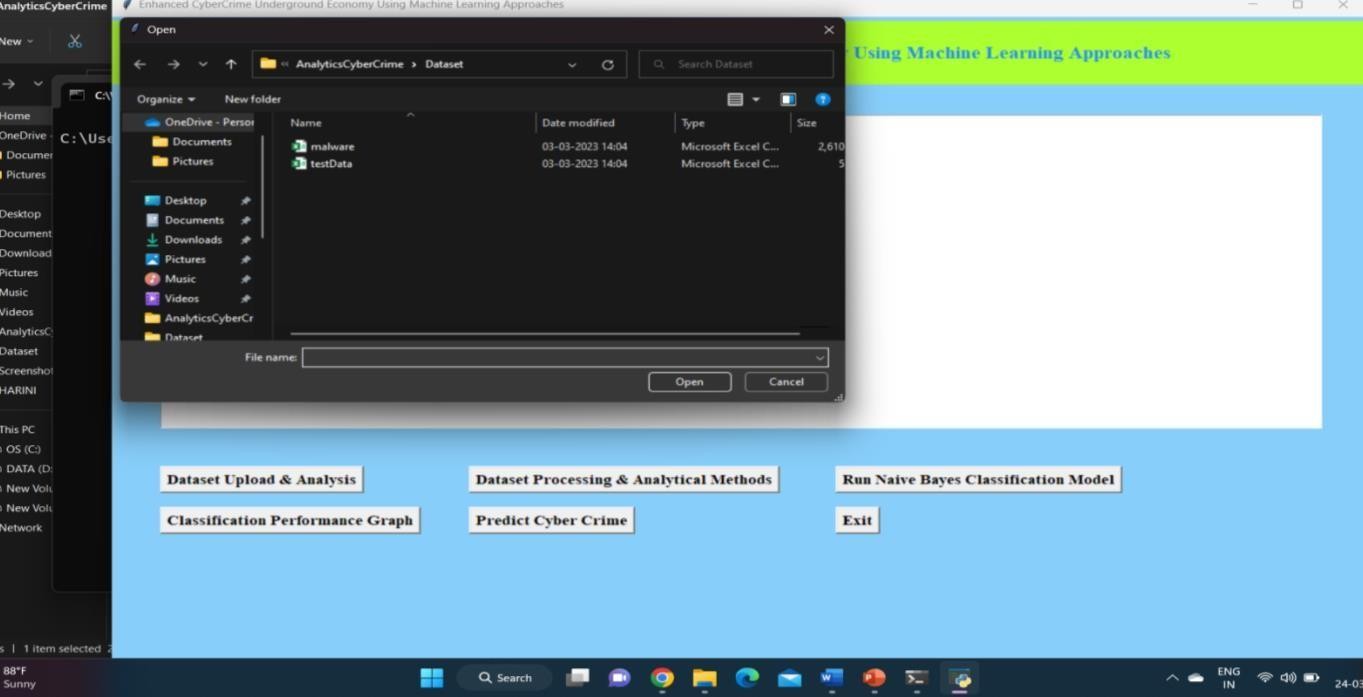
## Snapshots of System with Brief Detail of Each

To run project double, click on the ‘run.bat’ file to get below output.



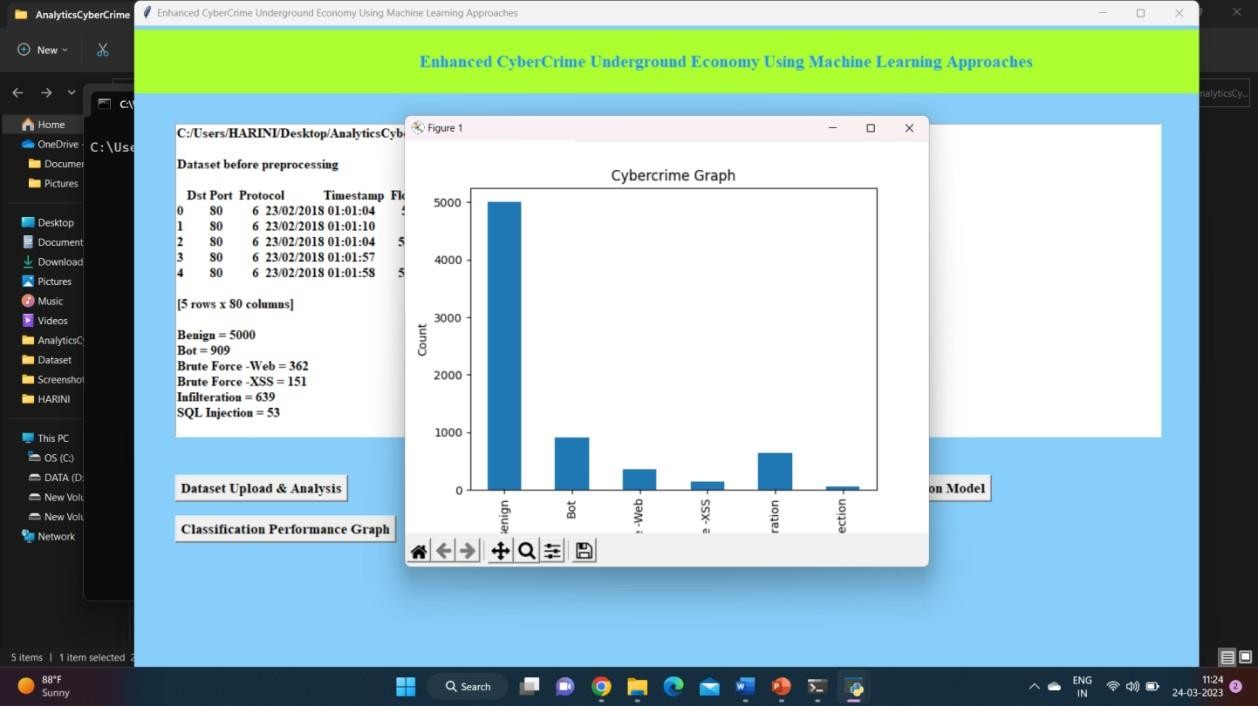
#### Fig 5.2 Home Page

In above screen click on ‘Dataset Upload & Analysis’ button to upload dataset and get below output.



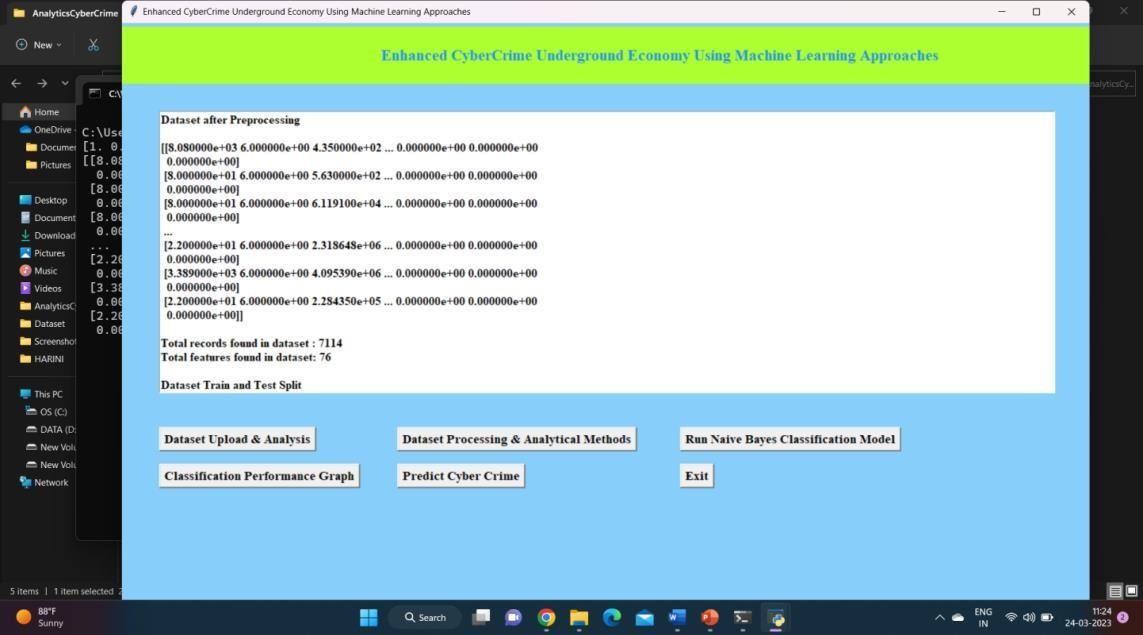
#### Fig 5.3 Uploading the Malware Dataset

In above screen selecting and uploading ‘malware.csv’ file and then click on ‘Open’ button to loaddataset and get below analysis output.



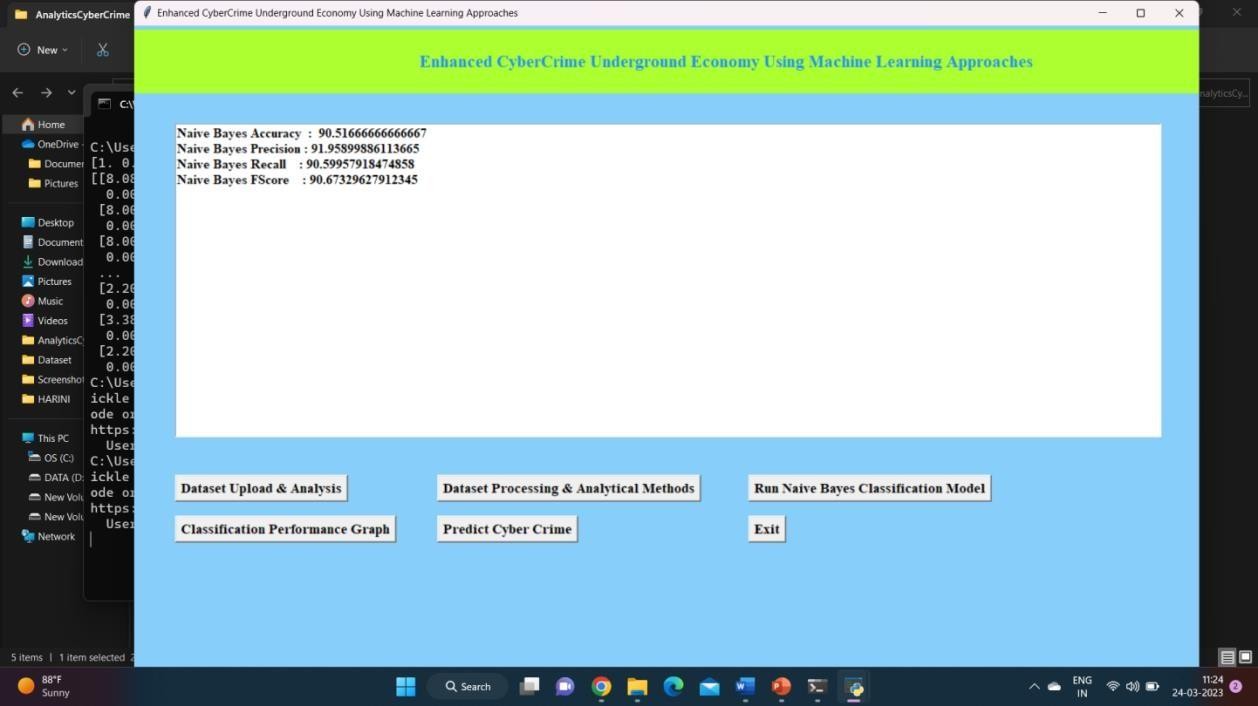
#### Fig 5.4 Graphical Representation of Various Cybercrime Attacks

In above screen we can see dataset loaded and dataset contains some non-numeric characters and algorithm will not take such non-numeric data so we need to convert it into integer ID and in abovescreen we can various cybercrime and its count so by seeing this we can analyse on which cybercrime most money is using and in above graph we can see that analysis output. Now close above graph and then click on ‘Dataset Processing & Analytical Methods’ button to clean dataset and then split data into train and test.



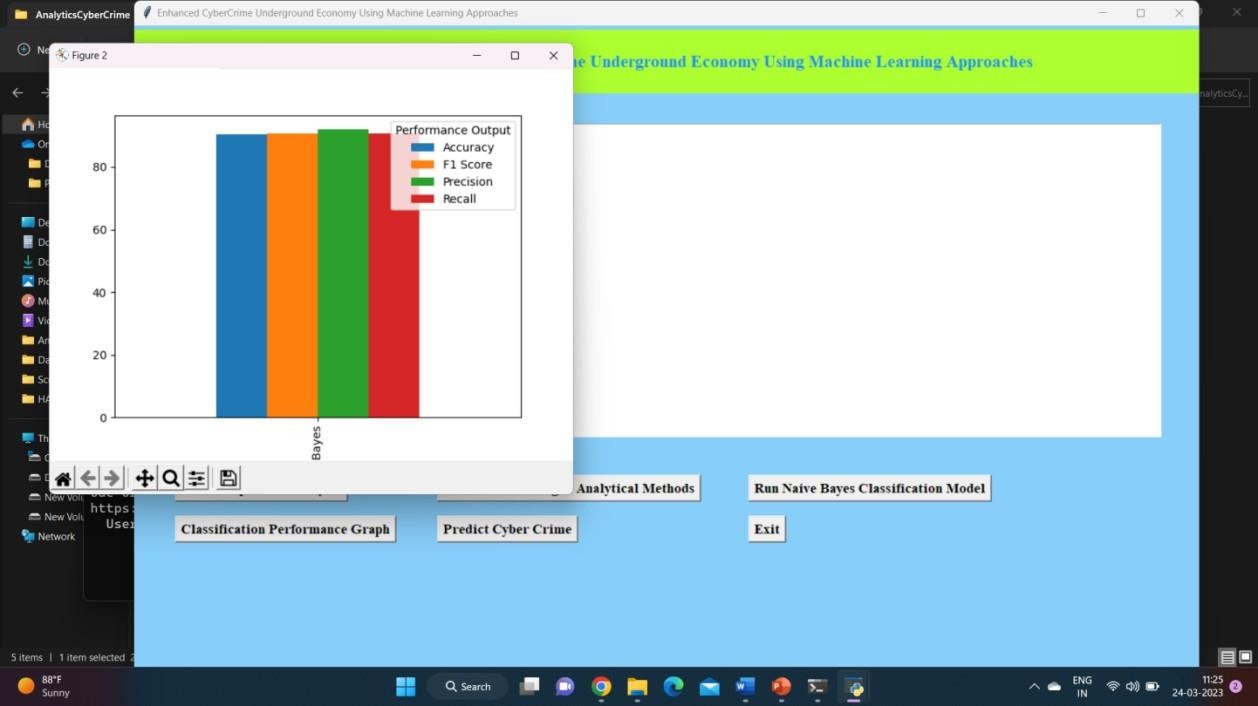
#### Fig 5.5 Training and Testing of Data

In above screen we can see all data is converted to numeric format and then we can see total datasetand then we can see 80% dataset used for training and 20 for testing and now dataset is ready and now click on ‘Run Naive Bayes Classification Model’ button to train classification model and get below output.



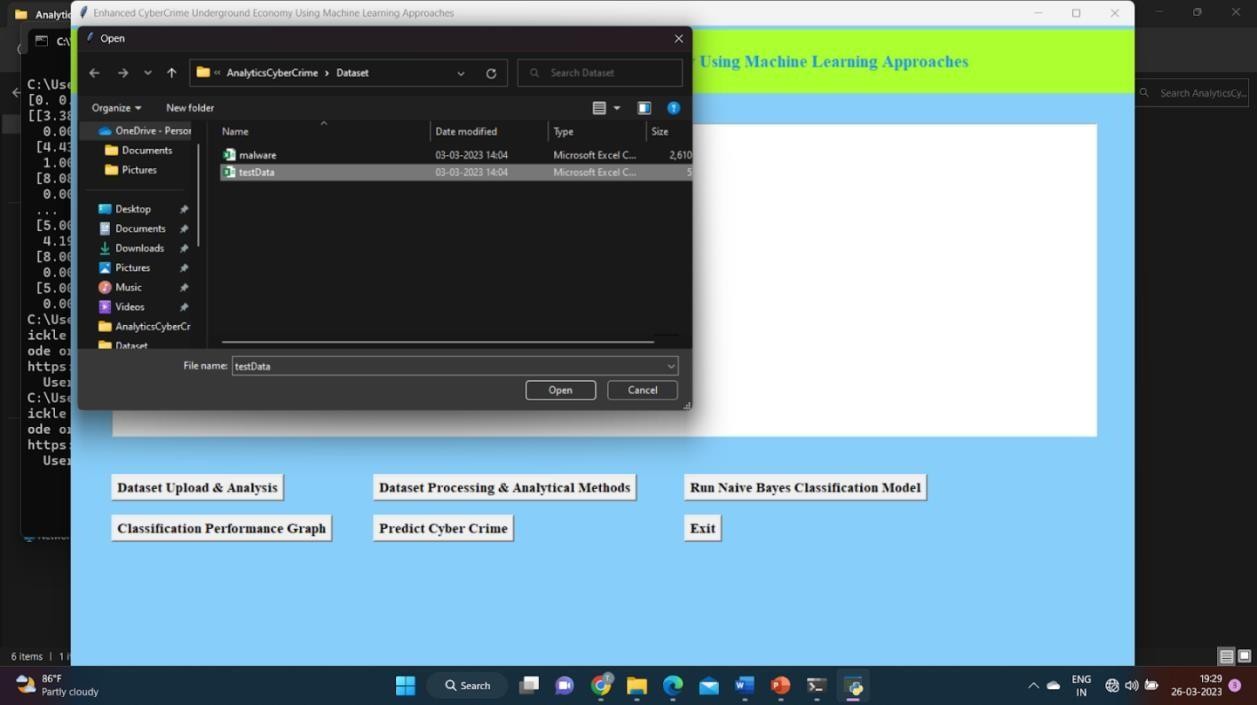
#### Fig 5.6 Prediction Accuracy

In above screen Naïve Bayes training is completed and we got its prediction accuracy as 90% and now click on ‘Classification Performance Graph’ button to get below graph.



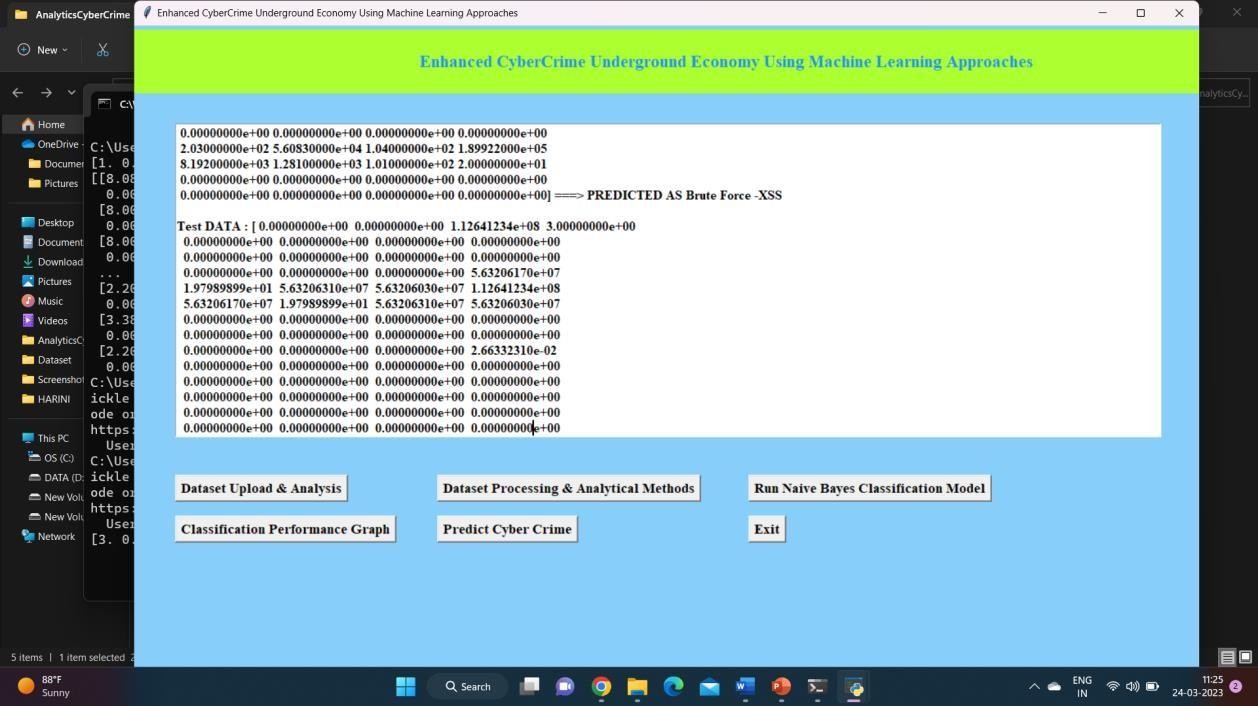
#### Fig 5.7 Classification Performance Graph

In above graph x-axis represents algorithm names and each different colour bar represents different metrics such as Accuracy, precision etc. in above graph we can see all metrics performance is above 90% and now close above graph and then click on ‘Predict Cyber Crime’ button to upload test data and get below output.



#### Fig 5.8 Uploading the Test Dataset

In above screen selecting and uploading ‘testData.csv’ file and then click on ‘Open’ button to get below output.



#### Fig 5.9 Prediction of Various Cybercrime Attacks

In above screen in square bracket, we can see network traffic data and after arrow symbol we can see the type of cybercrime attack prediction. Scroll down above screen to view all cybercrime prediction.

# CONCLUSION AND FUTURE SCOPE

This study also has important implications for society. Over the last few years, the world has been facing cyberterrorism and cyberwar threats from nation-sponsored attackers. Pollitt defined cyberterrorism as “the premeditated, politically motivated attack against information, computer systems, computer programs and data which results in violence against non-combatant targets by subnational groups or clandestine agents.” Unlike most cybercrime, which is primarily motivated by monetary gain, cyberterrorists are politically motivated. As a result, governments should, for example, strengthen their ability to protect their citizens in online virtual environments by enhancingtheir immediate responses to threats such as cyberespionage and cyberterrorism. This issue therefore has profound implications in terms of the need for a global cyber defense to maintain a cyber-safe environment.

# REFERENCES/BIBLIOGRAPHY

1. J. C. Wong and O. Solon. (2017, May 12). Massive ransomware cyber-attack hits early 100 countries around the world. [Online]. https://[www.theguardian.com/technology/2017/may/12/global-cyber-attack-ransomware-nsa-](http://www.theguardian.com/technology/2017/may/12/global-cyber-attack-ransomware-nsa-uk-) [uk-](http://www.theguardian.com/technology/2017/may/12/global-cyber-attack-ransomware-nsa-uk-) nhs
2. “FACT SHEET: Cybersecurity National Action Plan,” ed: The White House, 2016.
3. A. K. Sood and R. J. Enbody, “Crimeware-as-a-service—A survey of commoditized crimewarein the underground market,” Int. J. Crit. Infr. Prot., vol. 6, no. 1, pp. 28–38, 2013.
4. S. W. Brenner, “Organized Cybercrime? How Cyberspace May Affect the Structure of CriminalRelationships,” N. C. J. Law & Technol., vol. 4, no. 1, pp. 1-50, 2002.
5. K. Hughes, “Entering the world-wide web,” ACM SIGWEB Newsl., vol. 3, no. 1, pp. 4– 8, 1994.
6. S. Gregor and A. R. Hevner, “Positioning and Presenting Design Science Research for MaximumImpact,” MIS Quart., vol. 37, no. 2, pp. 337-356, 2013.
7. A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design Science in Information SystemsResearch,” MIS Quart., vol. 28, no. 4, pp. 75-105, 2004.
8. K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A Design Science Research Methodology for Information Systems Research,” J. Manag. Inf. Syst., vol. 24, no. 3, pp. 45–77, 2007.
9. S. Gregor, “Design theory in information systems,” Aust. J. Inf. Syst., vol. 10, no. 1, pp. 14–22,2002.
10. S. Gregor and D. Jones, “The Anatomy of a Design Theory,” J. the Assoc. Inf. Syst., vol. 8, no. 5, pp. 313–335, 2007.
11. M. Yar, “The Novelty of 'Cybercrime': An Assessment in Light of Routine Activity Theory,”Eur. J. Criminol., vol. 2, no. 4, pp. 407–427, 2005.
12. K.-K. R. Choo, “Organised Crime Groups in Cyberspace: a Typology,” Trends in OrganizedCrime, vol. 11, no. 3, pp. 270–295, 2008.
13. L. E. Cohen and M. Felson, “Social Change and Crime Rate Trends: A Routine ActivityApproach,” Am. Sociol. Rev., vol. 44, pp. 588–608, 1979.
14. M. Felson, “Routine Activities and Crime Prevention in the Developing Metropolis,” Criminol., vol. 25, no. 4, pp. 911–932, 1987.