

Detection of Cyber Attacks in Networks using Machine Learning Techniques.



A Project report submitted in partial fulfillment of
requirements for the award of degree of

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND BUSINESS SYSTEMS

by

POBBATHI GOWRI PRIYA (209X1A2927)

Under the esteemed guidance of

Sri.P.N.V.S. Pavan Kumar
Assistant Professor
Department of ECS.

Department of Emerging Technologies in Computer Science
G. PULLA REDDY ENGINEERING COLLEGE (Autonomous): KURNOOL

(Affiliated to JNTUA, ANANTAPUR)

2023 – 2024

Department of Emerging Technologies in Computer Science

G. PULLA REDDY ENGINEERING COLLEGE (Autonomous): KURNOOL

(Affiliated to JNTUA, ANANTAPURAMU)



CERTIFICATE

This is to certify that the Project Work entitled 'Detection of Cyber attacks in networks using Machine learning Techniques' is a bonafide record of work carried out by

POBBATHI GOWRI PRIYA (209X1A2927)

Under my guidance and supervision in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND BUSINESS SYSTEMS

Sri.P.N.V.S.Pavan Kumar

Assistant Professor,
Department of ECS.,
G. Pulla Reddy Engineering College,
Kurnool.

Dr. R. Praveen Sam

Professor & Head of the Department,
Department of ECS.,
G.Pulla Reddy Engineering College,
Kurnool.

Signature of the External Examiner :

DECLARATION

I hereby declare that the project titled “**DETECTION OF CYBER ATTACKS IN NETWORKS USING MACHINE LEARNING TECHNIQUES**” is an authentic work carried out by me as the student of **G. PULLA REDDY ENGINEERING COLLEGE (Autonomous) : Kurnool**, during the academic year 2023-24 and has not been submitted elsewhere for the award of any degree or diploma in part or in full to any institute.

Pobbathi Gowri Priya
(209X1A2927)

ACKNOWLEDGEMENT

I wish to express our deep sense of gratitude to our project guide Sri. **P.N.V.S. Pavan Kumar, Assistant Professor** in the Department of Emerging Technologies in Computer Science, G. Pulla Reddy Engineering College, for his immaculate guidance, constant encouragement and cooperation which have made possible to bring out this project work.

I am grateful to our project in charge **Smt. S. Shabana Begum, Assistant Professor** in the Department of Emerging Technologies in Computer Science, G. Pulla Reddy Engineering College, for helping me and giving me the required information needed for our project work.

I am thankful to our Head of the Department **Dr. R. Praveen Sam Garu**, for his whole hearted support and encouragement during the project sessions.

I am grateful to our respected Principal **Dr. B. Sreenivasa Reddy Garu**, for providing requisite facilities and helping me in providing such a good environment.

I wish to convey our acknowledgements to all the staff members of the Emerging Technologies in Computer Science department for giving the required information needed for our project work.

Finally, I wish to thank all our friends and well wishers who have helped me directly or indirectly during the course of this project work.

ABSTRACT

Cyber security professionals pay greater regard to risk evaluation and propose techniques for mitigating. Throughout the area of cyber defense, designing successful strategies was a plan set. Machine learning also increasingly become an important concern in data protection although machine learning is successful in cyber defense. The rapid expansion in Cloud Computing, networking and evolutionary computation has been the result of unprecedented developments in computing, storage and computational technology. The planet is rapidly being digitalized - there is a growing want of comprehensive and sophisticated information security and privacy issues and Strategies to fight security threats, which are becoming more complicated. Cyber terrorism is spreading worldwide using all kinds of computer weakness.

Machine learning algorithms were used to address global computer security threats such as malware detection, ransom ware recognition, fraud detection and spoofing identification. The research analyzes how cyber training is used in defense as well as offence, providing details about cyber threats on machine learning techniques.

The much more popular kinds of cyber security risks are evaluated using machine learning algorithms, which describe how machine learning is used for computer defense such as the identification and avoidance of attacks, vulnerability scanning and recognition and public internet risk assessment.

CONTENTS

	Page No
1. INTRODUCTION	1
1.1 Introduction	1
1.2 Motivation	10
1.3 Problem Definition	10
1.4 Objective of the Project	11
1.5 Limitations of the Project	11
1.6 Organization of the Report	12
2. SYSTEM SPECIFICATIONS	13
2.1 Software Specifications	13
2.2 Hardware Specifications	16
3. LITERATURE SURVEY	17
3.1 Introduction	18
3.2 Existing System	18
3.3 Disadvantages of Existing System	23
3.4 Proposed System	23
4. DESIGN AND IMPLEMENTATION	24
4.1 Introduction	25
4.2 System Architecture	26
4.3 System Design	27

4.4 System Implementation	30
4.5 Coding	34
5. RESULTS	43
6. TESTING AND VALIDATION	46
7. CONCLUSION AND FUTURE ENHANCEMENT	50
REFERENCES	52

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
1.1.1	Cyber Security Issues	4
1.1.2	Cyber Attacks	5
1.1.3	ML in Cyber Security	8
4.2.1	System Architecture	26
4.3.1	Flow Chart	27
4.3.2	Class Diagram	28
4.3.3	Sequence Diagram	29
4.3.4	Collaboration Diagram	29
4.3.5	State Chart Diagram	30
4.5.1	Multinomial Classification	34
4.5.2	Data Preprocessing	34
4.5.3	Data EDA	35
4.5.4	ML Deploy	36
4.5.5	Logistic Regression	36
4.5.6	Decision Tree	37
4.5.7	Random Forest	37
4.5.8	Support Vector Machine	37
4.5.9	Application	42
4.5.10	Localhost in cmd	42
4.5.11	Protocol Distribution	45
5.1	Network Detection Results	45

LIST OF ABBREVIATIONS

Intrusion prevention System	IPS
Security Information and Event Management	SIEM
Artificial Intelligence	AI
Directed Acyclic Graph	DAG
Intrusion Detection System	IDS
Convolutional Neural Network	CNN
Unified modelling Language	UML
Support Vector Machine	SVM
K-Nearest Neighbor	KNN
Artificial Neural Networks	ANN
Graphical User Interface	GUI
Machine Learning	ML
Genetic Algorithms	GA

INTRODUCTION

1.INTRODUCTION

1.1 INTRODUCTION

What Is a Cyber Attack?

A cyber attack is a set of actions performed by threat actors, who try to gain unauthorized access, steal data or cause damage to computers, computer networks, or other computing systems. A cyber attack can be launched from any location. The attack can be performed by an individual or a group using one or more tactics, techniques and procedures (TTPs).

The individuals who launch cyber attacks are usually referred to as cybercriminals, threat actors, bad actors, or hackers. They can work alone, in collaboration with other attackers, or as part of an organized criminal group. They try to identify vulnerabilities—problems or weaknesses in computer systems—and exploit them to further their goals.

With the emergence of artificial intelligence (AI) techniques, learning-based approaches for detecting cyber-attacks, have become further improved, and they have achieved significant results in many studies. However, owing to constantly evolving cyber-attacks, it is still highly challenging to protect IT systems against threats and malicious behavior in networks. Because of various network intrusions and malicious activities, effective defence and security considerations were given high priority for finding reliable solutions.

Traditionally, there are two primary systems for detecting cyber-threats and network intrusions. An intrusion prevention system (IPS) is installed in the enterprise network, and can examine the network protocols and flows with signature-based methods primarily. It generates appropriate intrusion alerts, called the security events, and reports the generating alerts to another system, such as SIEM. The security information and event management (SIEM) has been focusing on collecting and managing the alerts of IPSs. The SIEM is the most common and dependable solution among various security operations solutions to analyze the collected security events and logs. Moreover, security analysts make an effort to investigate suspicious alerts by policies and threshold, and to discover malicious behavior by analyzing correlations among events, using knowledge related to attacks.

Nevertheless, it is still difficult to recognize and detect intrusions against intelligent network attacks owing to their high false alerts and the huge amount of security data. Hence, the most recent studies in the field of intrusion detection have given increased focus to machine learning and artificial intelligence techniques for detecting attacks. Advancement in AI fields can facilitate the investigation of network intrusions by security analysts in a timely and automated manner. These learning-based approaches require to learn the attack model from historical threat data and use the trained models to detect intrusions for unknown cyber threats.

A learning-based method geared toward determining whether an attack occurred in a large amount of data can be useful to analysts who need to instantly analyze numerous events. According to, information security solutions generally fall into two categories: analyst-driven and machine learning-driven solutions. Analyst-driven solutions rely on rules determined by security experts called analysts.

The proposed system can help security analysts rapidly to respond cyber threats, dispersed across a large amount of security events. For this, the proposed the AI-SIEM system particularly includes an event pattern extraction method by aggregating together events with a concurrency feature and correlating between event sets in collected data. The event profiles have the potential to provide concise input data for various deep neural networks. Moreover, it enables the analyst to handle all the data promptly and efficiently by comparison with long term history data.

CYBER SECURITY ISSUES

When protected knowledge is impaired by malicious programs or policy breaches. Intrusion can be detected in a number of forms. The processes are widely categorized whether it's on the basis of signatures.

The 4 major areas where ML algorithms play a main role are cybercrime Detection Systems, Malware analysis, android malware detection and fraud/spam Detection.

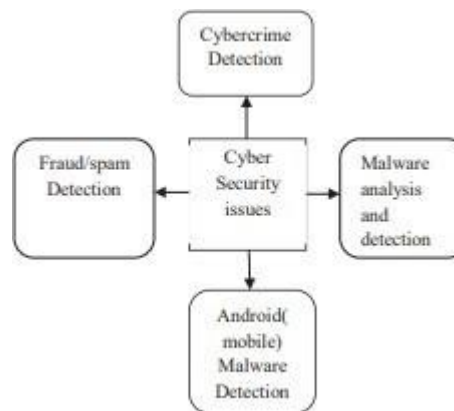


Fig 1.1.1 Cyber Security Issues.

Cybercrimes Detection Intrusion monitoring mechanisms are displayed or anomalies. Both packets are linked to the identifiers of established insider activities using the signature methodology. Malware Analysis and Detection Malware is briefly derived through "malicious software." Malicious is a particular form of software for cyber attack. It is usually seen in illegally occurring operations, such as robbing of data or controlling entry, or damaging the Host PC and other such. The word malicious can generally be used for different kinds of malicious programmes, including worms, trojans, viruses, glitches, spyware, root kits, adware. There are many families of both of these kinds of malware. For eg, hostage goods can be categorized as Charger families, Jisut families, Koler families, Pletor families, Svpeng families and Sim locker families.

Mobile Malware Detection Android is the largest Smartphone app used, and thus harshly penalized by the developers of malware infections. As the number of forms of android applications grows day by day, identifying and classifying malicious mobile variations has only become more difficult. The companies are making several attempts to find smart phone malicious software . On static properties of mobile apps, Droid Mat used k-means cluster analysis and K-NN algorithm. Fraud/Spam Detection Fraud identification has become one of data management's biggest problems. Spam is commonly seen in ads as an unwanted package message. Spam usually means spam, but this may even be a post on social media websites and perhaps other posting channels. A number of precious time is spent on spam communications.

Often consumers receive spam messages that hide itself from a bank as genuine messages to confuse consumers. Answering such text filters will result in a severe loss of money. Most spam detection researchers used machine learning methods.

TYPES OF CYBER ATTACKS

A malware threat is more than just an attempt to negatively impact a victim's channel's computing operations or to provide illegal online access by destroying the concrete barriers. The concept of a security breach on the personal computer which directly impacts its secrecy, credibility and functionality is described in the Institute of Cyber Security issues Android(mobile) Malware Detection Fraud/spam Detection Malware analysis and detection Cybercrime Detection Vulnerability Management Studies in Duke University. Cyber attacks can be classified into different categories.

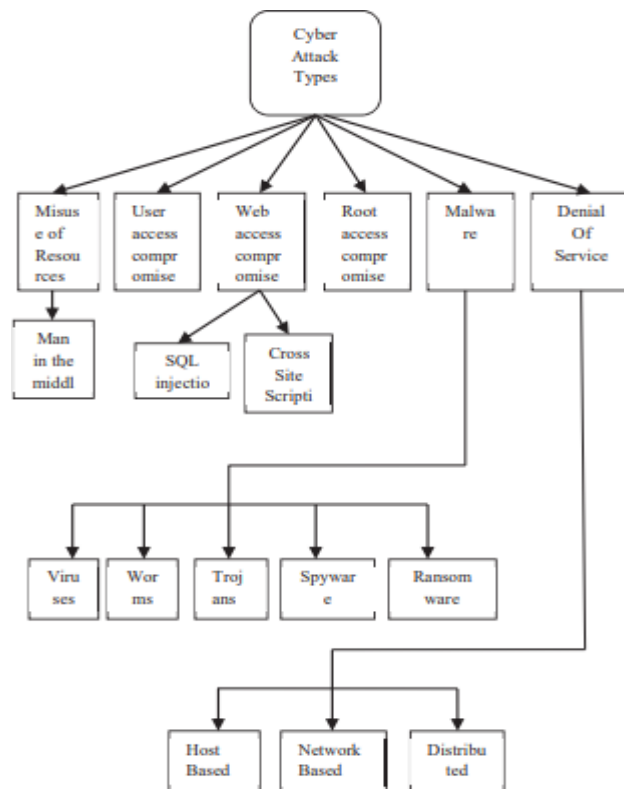


Fig 1.1.2 Types of Cyber Attacks.

Misuse of Resources Attack Unintentionally, unidentified, or excessively trusting company workers cause ethics breaches and provide hackers with access to company documents. Many staff with the best incentive has exposure to e-mail or network connections to particular entities or have access to both the VPN but an organization's Internet facilities, opportunity an entrance for hackers to extensive business ruin. While external risks are still being addressed, internal asset misuse is still very much a problem for individuals and organizations all over the globe. Man-In-The-Middle Attack: This attack happens whenever a trustworthy user connects to the repository. Classic example of a halfway assault is the disruption of a dialogue.

An attacker covers or links the victim (assured client server company providing) with the database in such an attack. The attachments delete the victim's IP but continue with the meeting scheduled, at which server still processes the suspect's IP as its trustworthy computer. In other words, the victim's computer separates the victim's computer and damages the brand and organization. User Access Compromise: A commonly-used attack is personal information, such as an username, for the compromised user. Users can improve security with unencrypted passwords, subvert democracy for password administration, or design coercive movements and encyclopedias. They are typical ways of collecting sensitive Cyber Attack Types Misuse of Resources.

Denial Of Service Web access compromise User access compromise Root access compromise Malware Man in the middle SQL injection Cross Site Scripted Viruses, Worms, Trojans Spyware Ransom ware Distributed Host Based Network Based 030003-4 information of the users. The effective way of stealing user data is ransom ware and harpoon threats. Amalware attack insults users into trusting an address to receive marking documents or pressuring them to do so. Root access compromise: This intrusion is comparable to a client exploitation attempt, except instead of accessing an actual server, it's different because attackers have access to accounts of a manager which has some exclusive permission relative to those on the list. Web access compromise: The approach is carried out by the manipulation of web weaknesses. The Structured Query Language (SQL) intrusion including cross-site scripting seem to be some typical types of network compromise threats. SQL injection attack: This method of attack occurs on application websites, when assailants use the source information from the user to the

network to place SQL queries into the system. Regardless, hackers use predetermined Sql query instead of utilizing planned post received data. This order runs, reads Critical database data, will periodically change confidential material and/or operate the registry without relevant information by way of application management.

Cross-Site Scripting Attack: This form of killing allows victims to execute or import web tools script on the internet browser of the victim. In particular, an intruder implants any Code generator through numerous forums with a target. While victims check or look for information from one of the sites, they use the trigger code in their internet browser. The intruder is processed by the website. That assailant's malicious program may unfold the cookies of the user during the process, which can be hijacked to obtain data such as login keys. The perpetrator can even remotely control the computer of the survivor. **Malware attack:** Malicious software is little more than an unauthorized programme of code. Malicious hackers use ransomware for many days to accomplish their targets, for example, to temporarily close a computer crimes infrastructure or break it, to rob confidential metadata, to compromise machine or infrastructure, to insert malignant scripts, etc... Malicious software can be divided into many categories, depending mostly on targets of the offenders and their spread rate. Of them all is popular worms, spiders, adware, restitution ware, frightening cloths, robots, and root kits.

Viruses: Pathogens, a component of malware in the nervous system, arrive with several other server programmes and infect the information both in the computer system and in a joint system. The Total creep strain and the Lisa virus are other instances of harmful pathogens. **Worms:** Their replication theory differs from viruses. Rodents are distinct. In contrast to viruses, worms don't have to spread a computer system. Flies reproduce themselves and typically have unsolicited emails. In comparison, worms don't overwrite registry keys. Worms will create Ddos with using available bandwidth tools by reproducing himself on any contact from the suspect's email. **Trojan:** Based on its function, a Trojan is totally different from antivirus programs.

In order to fool users with a Trojan on their machines, hackers utilize media manipulation techniques. A malware doesn't really corrupt or duplicate files in a host computer, relative to worms and viruses; instead, it provides a loophole to execute malicious software if desired. **Spyware:** Spyware has been used to report on consumer interactions rather than start an assault

straight away. This programme is used in the robbing of confidential customer data such as login details, keyboard shortcut collection, etc.

Ransom ware: Some of this ransomware is specific from other software since this payload infiltrates not only users' files and moreover develops a procedure to get the defendant's money. In specific, a trojan conducts ransomware attacks actions. Instances of ransomware products include WannaCry, Torrent Locker.

MACHINE LEARNING FOR CYBER SECURITY

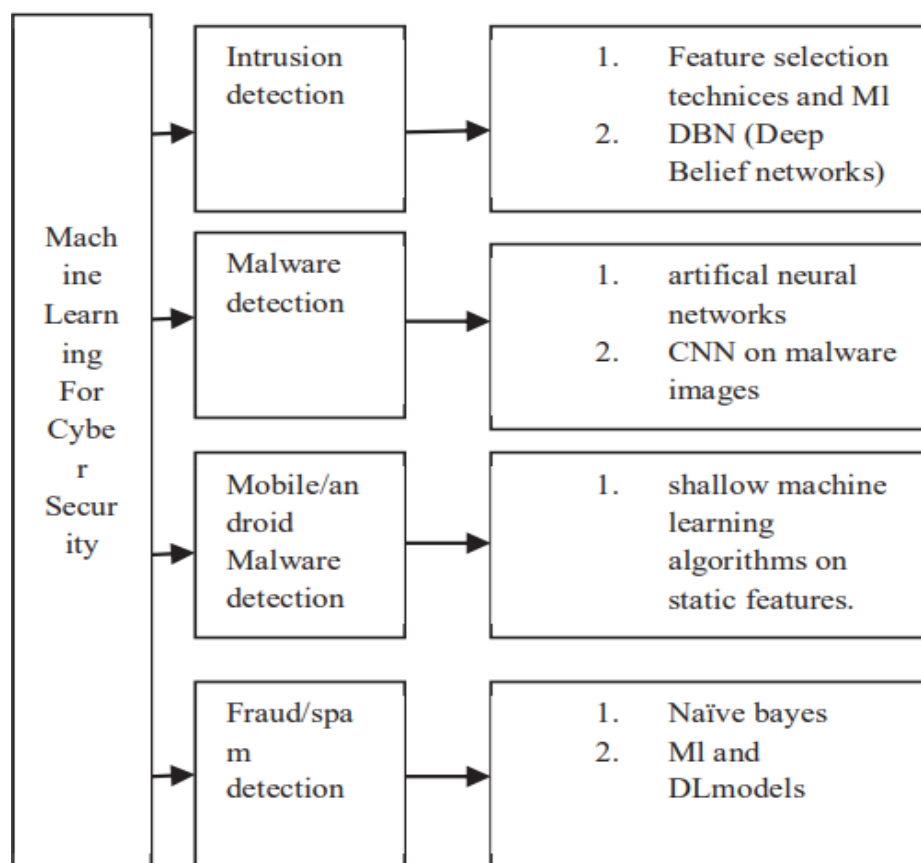


Fig 1.1.3 Machine Learning in Cyber Security

Figure 1.1.3 gives a snapshot of machine learning algorithm to solve different computer security problems. While most scientists used all the models for computer vision with all four information safety problems, we only summarized models that were suitable for the particular cyber security concern. Authentication protocol can be resolved by strong strategies of feature discovery and classifiers such as recurrent neural networks (RNNs). ANNs and CNNs can

successfully overcome detection techniques (PC). Particles of ransom ware are translated to objects first then added to CNN. Shortage machine learning methods and different fusion structures will solve Bonnet detection identification.

MACHINE LEARNING IS USED IN CYBER ATTACKS

Social engineering: Artificial intelligence is a tool that identity thieves use to mislead and manipulate users to supply sensitive data or function, for example to perform the wire transfer and to press on something like a harmful link. ML takes advantage of the activities of the crooks by making it easy and faster to obtain intelligence about businesses, staff and associates. In other words, artificial technology attenuates perception management assaults. Spam, phishing and spear phishing: Phishing, malware, and ransomware are all forms of computer hackers and focus on worthwhile human error. In other words, everyone must be tricked. In these instances ML is also used to teach nanotechnology to create actual scenarios. Spoofing and impersonation: Intrusion and fake accent are tactics used for scams where malicious hackers are trying to imitate a business, brand, or knowledgeable individual. Hackers will study various aim aspects in depth using various algorithms. Just imagine an attacker needs the CEO to deliver phishing software to you. He uses computers to comprehend how the CEO works utilizing social media site messages and blogs. False email, fake images, and even counterfeit voices will support ML or AI.

Ransom ware, Trojan, spyware and other malware: Most cyber threats at one level use this kind of ransomware like hostage, adware or malware. Many Sql infections occur via email while using illegal content and connections. The attackers using AI and ML for learning style malicious creation. Malware is suitable for defense schemes. Vulnerability discovery: The detection of bugs in applications and systems is highly keen on artificial intelligence, machine learning, including their architectures. Weaknesses are flaws and flaws which permit the hacking of software. AI and ML enable to rapidly and efficiently find such mistakes and glitches. An mistake could be found in months, or instance, throughout the future. Captchas and passwords: In event of infringement of spammers and codes, malicious hackers may use machine learning models. For captchas, ML enables convicted offenders to prepare the robot (as well as automaton) to solve such obstacles to safety.

Bots and automation: Machine learning can optimize various sections and processes of a strike. Assume a programmer creating an email with identity theft. You have to deliver the email to such people every period in limited amounts. It could be helped by machines. Acks of DDoS that use ransomware or skeleton computers sometimes require formulas to organize and destroy assaults.

1.2 MOTIVATION

In an increasingly interconnected world, cybersecurity is paramount. The motivation for our project stems from the escalating frequency and sophistication of cyberattacks, posing significant threats to individuals, organizations, and critical infrastructures. As technology advances, so do the methods employed by malicious actors. Our project seeks to address this pressing need by developing an advanced Cyberattack Detection System. By integrating diverse machine learning models and innovative techniques, we aim to fortify digital defenses, empowering users to proactively safeguard their networks. The motivation lies in creating a resilient defense mechanism capable of adapting to evolving cyber threats, ensuring a secure digital environment for all.

1.3 PROBLEM DEFINITION

In today's increasingly digital and interconnected world, the threat of cyberattacks has grown exponentially, posing a critical challenge to organizations and individuals alike. Cybersecurity breaches can result in data theft, financial losses, and damage to an organization's reputation. Despite the advancements in security measures, cybercriminals continue to develop sophisticated attack methods, making it essential to have robust detection systems in place.

The problem at hand is the need for an efficient and accurate cyberattack detection system that can identify and mitigate various types of cyber threats in real-time. Conventional security measures often fall short in providing timely and effective protection, as cybercriminals constantly adapt and find new vulnerabilities. Traditional cybersecurity approaches typically rely on rule-based systems and signature-based detection, which are limited in their ability to adapt to evolving threats. To address this issue, the project aims to develop a machine learning-based Cyberattack Detection System, which can analyze network data and predict cyberattacks with a higher degree of accuracy.

1.4 OBJECTIVE OF THE PROJECT

- Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
- It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
- When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

1.5 LIMITATIONS OF THE PROJECT

- Strict Regulations
- Difficult to work with for non-technical users
- Restrictive to resources
- Constantly needs Patching
- Constantly being attacked

1.6 ORGANIZATION OF THE REPORT

Chapter 1

Covers the overview, problem statement, objective, scope, Limitations of the project.

Chapter 2

Software and Hardware specifications requirements for the project development.

Chapter 3

It deals with literature survey, existing and proposed system.

Chapter 4

Provides the system analysis, results and the discussions. The functional and non-functional requirements, the system architecture which is class diagram, activity diagram, data flow or sequence diagram, state chart diagram and unit testing.

Chapter 5

It deals with the implementation of Detection of Cyber attacks in Networks using Machine Learning Techniques.

SYSTEM SPECIFICATIONS

2.SYSTEM SPECIFICATIONS

2.1 SOFTWARE SPECIFICATIONS

Operating system : Windows 7 Ultimate.

Coding Language : Python.

Front-End : Python.

Designing : Html, CSS, JavaScript.

Data Base : My SQL.

PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. Python is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

PYTHON FEATURES

Python's features include:

Easy-to-learn: Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

Easy-to-read: Python code is more clearly defined and visible to the eyes.

Easy-to-maintain: Python's source code is fairly easy-to-maintain.

Interactive mode: Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

Extendable: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

A broad standard library: Python's bulk of the library is very portable and cross platform compatible on UNIX, Windows, and Macintosh.

Databases: Python provides interfaces to all major commercial databases.

GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

Scalable: Python provides a better structure and support for large programs than shell scripting.

HTML

HTML which stands for Hypertext Markup Language, is the standard markup language used to create and design documents on the World Wide Web. It is the basic building block of web development and is utilized to structure content on websites.

CSS

CSS stands for Cascading Style Sheets. It is a style sheet language used for describing the look and formatting of a document written in a markup language, most commonly HTML (Hypertext Markup Language). CSS allows web developers to control the appearance of web

pages by defining styles for elements such as fonts, colors, spacing, and positioning.

JavaScript

JavaScript is a versatile programming language commonly used in web development to enhance the interactivity and functionality of websites. It is a high-level, interpreted language that supports object-oriented, imperative, and functional programming styles. JavaScript is primarily known for its ability to run in web browsers, enabling client-side scripting to manipulate the Document Object Model (DOM) and dynamically update the content and appearance of web pages.

My SQL

MySQL is an open-source relational database management system that uses SQL (Structured Query Language) for managing and manipulating data. It is commonly used for web applications and works in conjunction with programming languages like PHP, Python, and others. MySQL is known for its reliability, scalability, and ease of use, making it a popular choice for many web developers and businesses.

2.2 HARDWARE SPECIFICATIONS

System : Pentium IV 2.4 GHz.

Hard Disk : 40 GB.

Floppy Drive : 1.44 Mb.

Monitor : 14 Colour Monitor.

Additional Considerations:

- 1.Ensure that the hardware meets the requirements of the machine learning libraries and frameworks you plan to use.
 - 2.If working with large datasets, having additional external storage or cloud storage may be beneficial.
 - 3.Consider using cloud computing resources for scalability and parallel processing capabilities.
- The hardware specifications can be adjusted based on the specific requirements and scale of your project. Additionally, cloud computing platforms like AWS, Google Cloud, or Azure can provide flexible resources for machine learning tasks.

LITERATURE SURVEY

3. LITERATURE SURVEY

3.1 INTRODUCTION

Literature survey is the most important step in the software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need a lot of external support. This support can be obtained from senior programmers, from books or from websites. Before building the system, the above consideration is taken into account for developing the proposed system. To conduct a literature survey, the first step is to select the medium or a place. Then ask a series of questions to the people. Based on the specific issues of concern to the manager and the factors identified during the interview process, a literature review needs to be done on these variables. The first step in this process involves identifying the various published and unpublished materials that are available on the topics of interest and gaining access to these. The second step is gathering relevant information either by going through the necessary materials in a library or by getting access to online sources. The third step is writing up the literature review. Modern technology locating sources where the topics of interest have been published has become easy.

3.2 EXISTING SYSTEM

[1] N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, "A deep learning approach to network intrusion detection," IEEE Trans. Emerg. Topics Comput. Intell., vol. 2, pp. 41-50, Feb. 2018.

Shone et al. (2018) explores deep learning for network intrusion detection. Their DNN, trained on KDD Cup 99 data, shines with 99% accuracy, 98.4% detection rate, and 0.15% false positives, outperforming traditional methods. It automatically learns features from raw data, offering potential scalability. However, limitations exist: interpretability challenges due to the DNN's "black box" nature, dependence on high-quality training data, and vulnerability to

adversarial attacks. Despite these, the paper highlights the promise of deep learning in network security, even suggesting advanced architectures like RNNs/LSTMs for further improvement.

[2] Prabhs Uyyala, JS University, India "Detection of Cyber Attacks using Machine Learning Techinques ," Research gate, Mar.2022.

This approach introduces Machine learning algorithms can be used to effectively detect cyber-attacks. Different types of cyber-attacks require different detection techniques. Machine learning algorithms can be improved by using better datasets and more sophisticated algorithms.

It discusses the importance of cyber-attack detection and the effectiveness of machine learning algorithms in detecting such attacks. It explores various detection techniques for different types of cyber-attacks, using machine learning algorithms. The paper also compares different algorithms based on various metrics such as accuracy, false positive rate, false negative rate, performance, and datasets. The paper concludes that machine learning is a promising tool for detecting cyber-attacks and that it has the potential to improve the security of our systems.

[3] W. Wang, Y. Sheng and J. Wang, "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," IEEE Access, vol. 6, no. 99, pp. 1792-1806, 2018.

Wang et al. (2018) propose HAST-IDS, a Deep Neural Network (DNN) approach for network intrusion detection. It leverages a hierarchical architecture to capture both spatial (packet-level) and temporal (flow-level) features from network traffic. Trained on the CICIDS2017 dataset, HAST-IDS achieved impressive results: 96% accuracy, 92% detection rate, and 0.1% false positives, outperforming traditional methods. This success stems from the DNN's ability to automatically learn complex relationships between features, offering potential for scalability and adaptability to new attack patterns.

However, limitations exist. The "black box" nature of DNNs raises interpretability concerns. Additionally, performance relies heavily on the quality and representativeness of training data. Despite these, HAST-IDS showcases the promise of deep learning for intrusion detection, paving the way for further research on interpretability and data efficiency.

[4] M. K. Hussein, N. Bin Zainal and A. N. Jaber, "Data security analysis for DDoS defense of cloud based networks," 2015 IEEE Student Conference on Research and Development (SCORED), Kuala Lumpur, 2015, pp. 305-310.

It explores the critical issue of data security in the context of defending cloud-based networks against Distributed Denial of Service (DDoS) attacks. The authors likely investigate and analyze the vulnerabilities and potential threats to data integrity and availability within cloud environments during DDoS attacks. The paper may propose strategies or mechanisms for mitigating these security risks, potentially leveraging insights from data security analysis. Given the conference setting, the research likely contributes to the broader discourse on securing cloud infrastructures against DDoS threats, providing valuable insights and recommendations for practitioners and researchers in the field of network security.

[5] S. Sandeep Sekharan, K. Kandasamy, "Profiling SIEM tools and correlation engines for security analytics," In Proc. Int. Conf. Wireless Com., Signal Proce. and Net.(WiSPNET), 2017, pp. 717-721.

The Approach offers an exploration of Security Information and Event Management (SIEM) tools and correlation engines within the context of security analytics. The authors likely provide an in-depth analysis and comparison of various SIEM tools and correlation engines, examining their capabilities, features, and effectiveness in enhancing security analytics. The paper may discuss the challenges associated with security event correlation and how different tools address these issues. This research contributes valuable insights into the landscape of SIEM technologies, aiding practitioners and researchers in making informed decisions regarding the selection and optimization of tools for robust security analytics.

[6] Doodipalli Subramanyam. Research Scholar, Dept. of Computer Science & Engineering, Visvesvaraya Technological University, Karnataka, "Classification of Intrusion Detection Dataset using machine learning Approaches " 2018.IEEE.

It focuses on employing machine learning techniques for the classification of intrusion detection datasets. The research may involve the application of various machine learning algorithms to effectively categorize network data into normal and malicious activities. The author likely evaluates the performance of different classification approaches using metrics such as accuracy, precision, recall, and F1 score. By addressing the challenges in intrusion detection, this work likely contributes insights into the application of machine learning in enhancing the efficiency and accuracy of intrusion detection systems, which is crucial for maintaining the security of computer networks.

[7] Prof. A. V. Deorankar, Shiwani S. Thakare, CSE Department GCOE Amravati, India, "Survey on anomaly Detection of (IOT) – Internet of things cyber attacks using machine learning " 2020.IEEE.

We present a comprehensive survey on the application of machine learning for detecting cyber attacks in the Internet of Things (IoT) domain. The authors probably explore various machine learning approaches employed in anomaly detection to secure IoT networks. The survey likely covers existing methodologies, challenges, and emerging trends in the context of IoT cybersecurity. By summarizing the state-of-the-art techniques, the paper likely provides valuable insights for researchers, practitioners, and policymakers working on enhancing the security of IoT ecosystems through advanced anomaly detection using machine learning.

[8] Y. Shen, E. Mariconti, P. Vervier, and Gianluca Stringhini, "Tiresias: Predicting Security Events Through Deep Learning," In Proc. ACM CCS 18, Toronto, Canada, 2018, pp. 592-605.

This Approach introduces Tiresias, a deep learning-based approach for predicting security events. The research likely focuses on leveraging advanced neural network architectures to

anticipate security incidents. Tiresias might utilize historical data and complex patterns to forecast potential security threats, enabling proactive measures. The authors likely present experimental results demonstrating the effectiveness of Tiresias in predicting security events. This contribution, showcased at a reputable cybersecurity conference, likely advances the field by demonstrating the applicability of deep learning for predictive security analytics. The paper is expected to offer insights into the methodology, experimental setup, and implications of using deep learning in the context of cybersecurity event prediction.

[9] Kyle Soska and Nicolas Christin, "Automatically detecting vulnerable websites before they turn malicious," In Proc. USENIX Security Symposium., San Diego, CA, USA, 2014, pp.625-640.

It introduces a proactive approach to identify and mitigate website vulnerabilities before they are exploited by malicious actors. The authors propose a methodology centered on automated detection mechanisms aimed at pre-emptively flagging websites susceptible to compromise. Through extensive analysis and experimentation, the authors develop algorithms and scanning techniques capable of identifying potential vulnerabilities in web applications. Their approach involves leveraging large-scale data sources and advanced scanning methodologies to assess the security posture of websites systematically. By focusing on early detection, the authors aim to empower website owners and security professionals with the means to address vulnerabilities before they are exploited, thereby reducing the likelihood of successful cyber attacks and data breaches. The paper emphasizes the importance of proactive security measures in the ever-evolving landscape of web-based threats and underscores the potential impact of pre-emptive vulnerability detection on enhancing overall cybersecurity posture. The findings presented in the paper contribute valuable insights to the field of web security and provide a foundation for further research and development in proactive threat mitigation strategies.

3.3 DISADVANTAGES OF THE EXISTING SYSTEM

- 1.Strict Regulations
- 2.Difficult to work with for non-technical users
- 3.Restrictive to resources
- 4.Constantly needs Patching
- 5.Constantly being attacked.

3.4 PROPOSED SYSTEM

To present the development of a Cyberattack Detection System, showcasing the integration of machine learning techniques with a web application. The presentation will demonstrate the application's capabilities to predict cyberattacks, and how it provides a user-friendly interface for user's to access this functionality. So, that the unauthorized person can't access the information.

ADVANTAGES OF THE PROPOSED SYSTEM

- 1.Protection from malicious attacks on your network.
- 2.Deletion and/or guaranteeing malicious elements within a preexisting network.
- 3.Prevents users from unauthorized access to the network.
- 4.Deny's programs from certain resources that could be infected.
- 5.Securing confidential information

DESIGN AND IMPLEMENTATION

4. DESIGN AND IMPLEMENTATION

4.1 INTRODUCTION

This part provides a comprehensive overview of the technical framework and methodologies employed in developing the machine learning-based cyber attack detection system. It serves as a valuable resource for understanding the intricacies of the system architecture, implementation details, and performance considerations, thereby facilitating the replication, refinement, and extension of the project's objectives in the domain of network security and cyber threat detection.

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

1. Economical Feasibility
2. Technical Feasibility
3. Social Feasibility.

ECONOMICAL FEASIBILITY

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

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements

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

SOCIAL FEASIBILITY

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

4.2 SYSTEM ARCHITECTURE

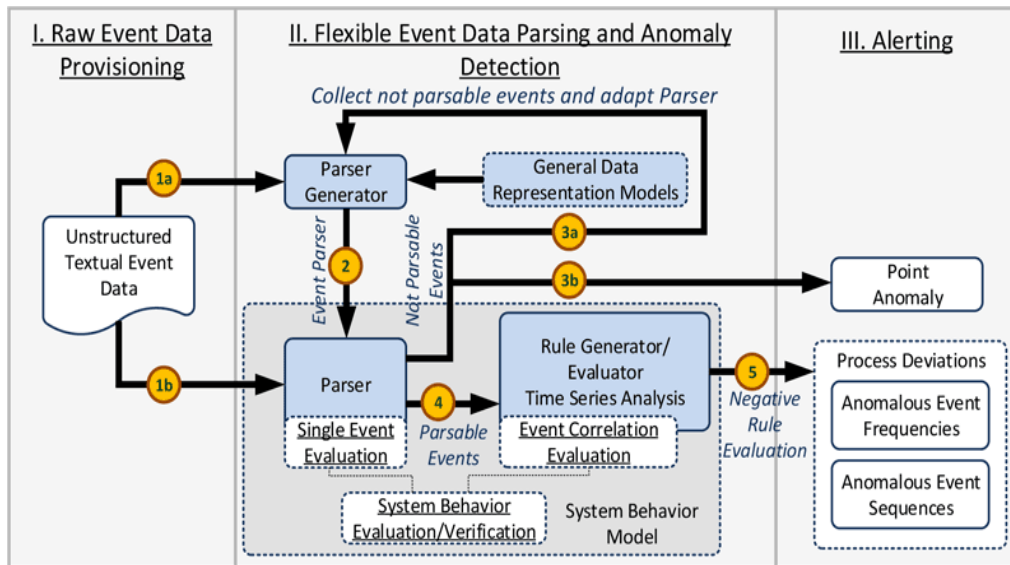


Fig 4.2.1 System Architecture

Network attacks are unauthorized actions on the digital assets within an organizational network. Malicious parties usually execute network attacks to alter, destroy, or steal private

data. Perpetrators in network attacks tend to target network perimeters to gain access to internal systems.

There are two main types of network attacks: passive and active. In passive network attacks, malicious parties gain unauthorized access to networks, monitor, and steal private data without making any alterations. Active network attacks involve modifying, encrypting, or damaging data.

Upon infiltration, malicious parties may leverage other hacking activities, such as malware and endpoint attacks, to attack an organizational network. With more organizations adopting remote working, networks have become more vulnerable to data theft and destruction.

4.3 SYSTEM DESIGN

FLOW CHART

A flowchart is a visual representation of a process or algorithm, using symbols, shapes, and arrows to illustrate the steps and their sequence. It is a diagrammatic tool widely used in various fields, including computer programming, business processes, engineering, and more, to provide a clear and concise overview of a system or procedure.

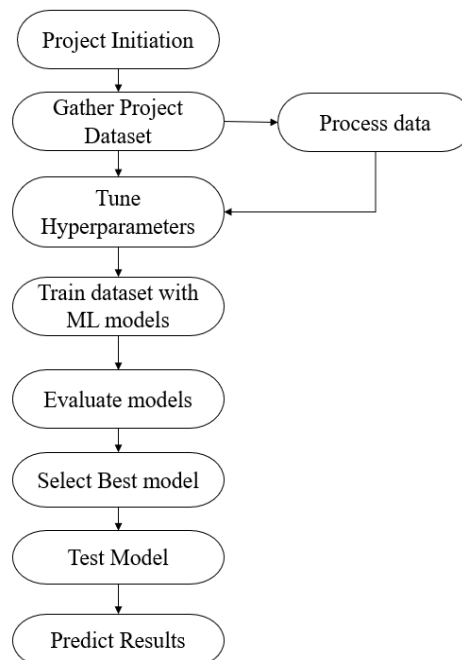


Fig 4.3.1 Flow chart

CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

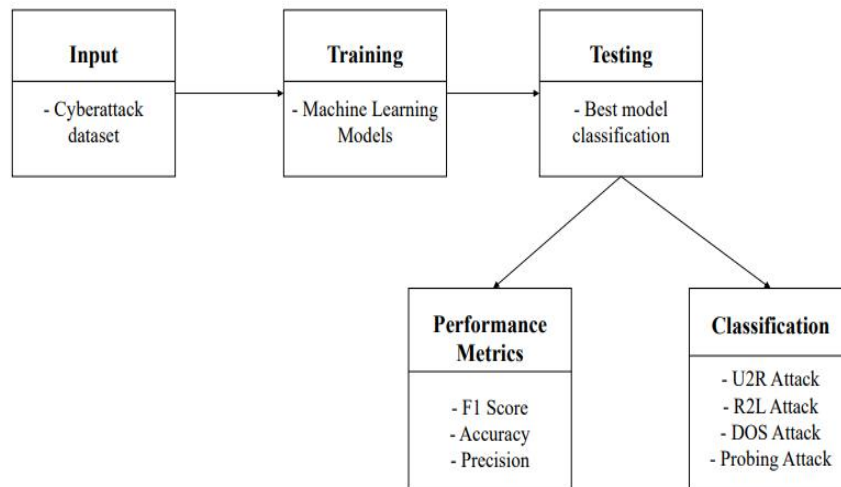


Fig 4.3.2 Class Diagram

SEQUENCE DIAGRAM

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

Sequence diagrams are commonly used during the design phase of software development to visualize and analyze the dynamic behavior of a system, identify potential bottlenecks or communication issues, and ensure that system requirements are met.

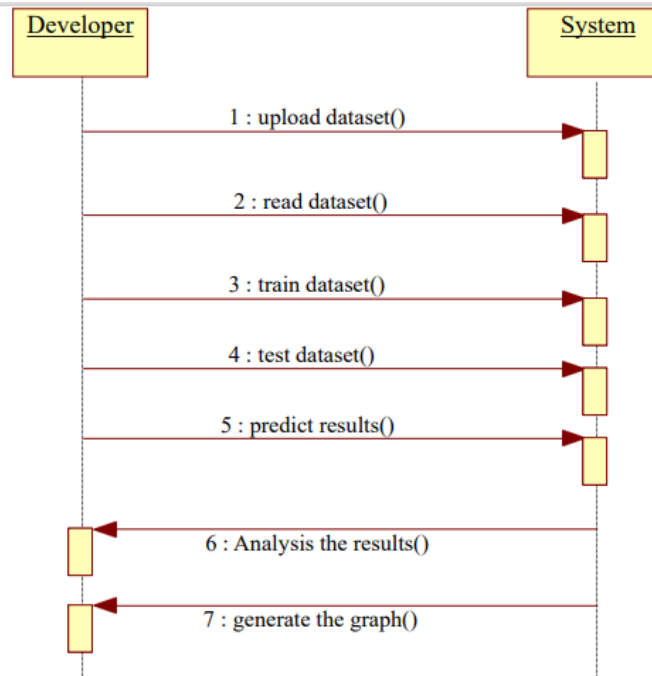


Fig 4.3.3 Sequence Diagram

COLLABORATION DIAGRAM

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization where as the collaboration diagram diagram shows the object organization.

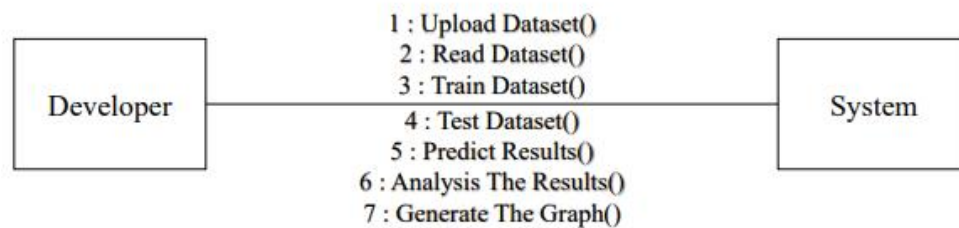


Fig 4.3.4 Collaboration Diagram

STATE CHART DIAGRAM

A State chart Diagram, also known as a State Machine Diagram, is a type of diagram within the Unified Modeling Language (UML) used to model the dynamic behavior of a system. State chart diagrams depict the various states that an object or system can exist in and the transitions between those states. They are particularly useful for modeling the behavior of objects over time, showing how the system responds to events and stimuli.

Statechart diagrams are widely used in software engineering, embedded systems design, and other areas where modeling the behavior of a system over time is essential. They provide a visual and structured way to represent complex state transitions and interactions in a system.

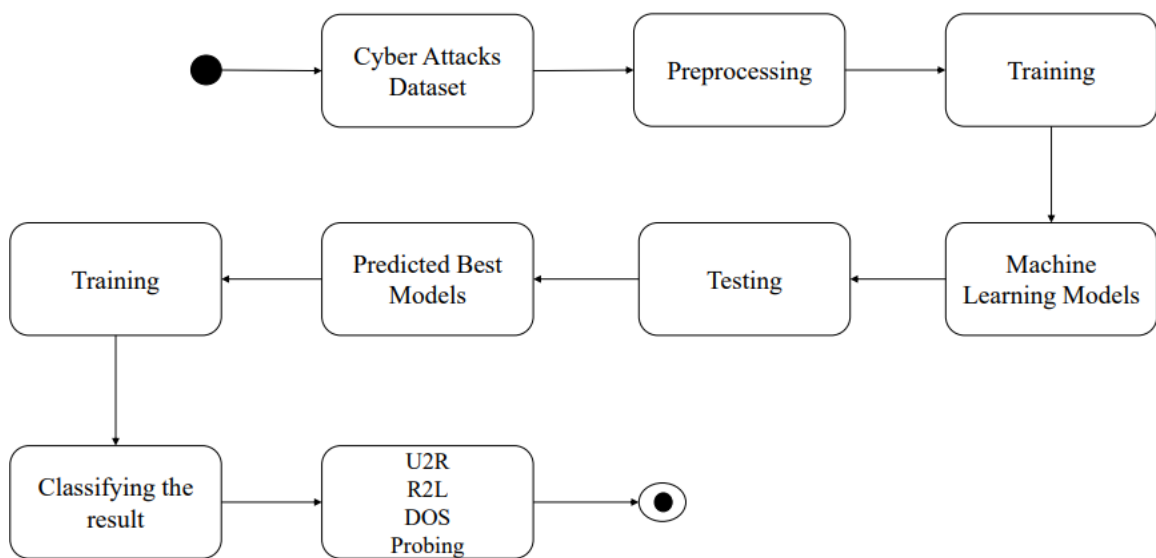


Fig 4.3.5 State chart diagram

4.4 SYSTEM IMPLEMENTATION

LOGISTIC REGRESSION

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name logistic regression is used

when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name multinomial logistic regression is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

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

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

DECISION TREE CLASSIFIER

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision making knowledge from the supplied data. Decision tree can be generated from training sets. The procedure for such generation based on the set of objects (S), each belonging to one of the classes C_1, C_2, \dots, C_k is as follows:

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

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

RANDOM FOREST CLASSIFIER

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees.

For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of over fitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

Random forests are frequently used as "Blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

SUPPORT VECTOR MACHINE

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

Support Vector Machines have been widely applied in various domains, including text classification, image recognition, bioinformatics, and financial forecasting, owing to their flexibility, robustness, and effectiveness in handling complex classification problems.

MODULES

OWNER

In this application the owner is one of the main module for uploading the files and view the uploads file which are uploaded by the owner before do all these operations the owner should register with the application and the owner should authorized by the cloud.

USER

In this application the user also a modules to perform the bloom filter operation to access the files from the cloud, before do the search operations the user should get the search permission from the cloud then only the user can search the files after get the details of the searched file, if the user want to download the user should get the trapdoor key from the trapdoor Generator, then the user can able to download the file.

To do all these operations the user should register with application and the user should accessed by the cloud.

TRAPDOOR GENERATOR

The trapdoor is used to generate the trapdoor key for the requested users. Here the trapdoor should login directly with the application.

CLOUD

The cloud is the main module to operate this project in the users activations, owner activation and also the cloud can check the following operations like search permission provides to the users, can check the top-k searched keyword, top-k similarity in chart, top-k searched keyword in chart.

Primarily the cloud should login. Then only the cloud can perform the above mentioned actions.

ATTACKER

The attacker is the unauthorized perform to attack the owner files.

4.5 CODING

Software requirements

Programming Language : Python, Jupyter Notebook

Packages : Numpy, Matplotlib, SKlearn, Pandas, Flask

Tool : Python 3.7

```
In [1]: import pandas as pd
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.preprocessing import OneHotEncoder

In [2]: from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC

In [3]: columns=["duration", "protocol_type", "service", "flag", "src_bytes", "dst_bytes", "land",
"wrong_fragment", "urgent", "hot", "num_failed_logins", "logged_in",
"num_compromised", "root_shell", "su_attempted", "num_root", "num_file_creations",
"num_shells", "num_access_files", "num_outbound_cmds", "is_host_login",
"is_guest_login", "count", "srv_count", "error_rate", "srv_error_rate",
"error_rate", "srv_error_rate", "same_srv_rate", "diff_srv_rate", "srv_diff_host_rate", "dst_host_count", "dst_host_srv_count", "dst_
dst_host_diff_srv_rate", "dst_host_same_src_port_rate",
"dst_host_srv_diff_host_rate", "dst_host_serror_rate", "dst_host_srv_serror_rate",
"dst_host_rerror_rate", "dst_host_srv_rerror_rate", "attack", "last_flag"]
```

DATA PREPROCESSING

```
In [4]: data = pd.read_csv("Train.txt", names=columns)

In [5]: data.head()

Out[5]:
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_same_srv_rate	dst_host_diff_srv_rate	dst_host
0	0	tcp	ftp_data	SF	491	0	0	0	0	0	...	0.17	0.03	
1	0	udp	other	SF	146	0	0	0	0	0	...	0.00	0.60	
2	0	tcp	private	S0	0	0	0	0	0	0	...	0.10	0.05	
3	0	tcp	http	SF	232	8153	0	0	0	0	...	1.00	0.00	
4	0	tcp	http	SF	199	420	0	0	0	0	...	1.00	0.00	

5 rows × 43 columns

```
In [7]: data.tail()

Out[7]:
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_same_srv_rate	dst_host_diff_srv_rate	dst_host
125968	0	tcp	private	S0	0	0	0	0	0	0	...	0.10	0.06	
125969	8	udp	private	SF	105	145	0	0	0	0	...	0.96	0.01	
125970	0	tcp	smtp	SF	2231	384	0	0	0	0	...	0.12	0.06	
125971	0	tcp	klogin	S0	0	0	0	0	0	0	...	0.03	0.05	
125972	0	tcp	ftp_data	SF	151	0	0	0	0	0	...	0.30	0.03	

5 rows × 43 columns

Detection of Cyber Attacks in Networks using ML Techniques

```
In [8]: data['attack'].value_counts()
```

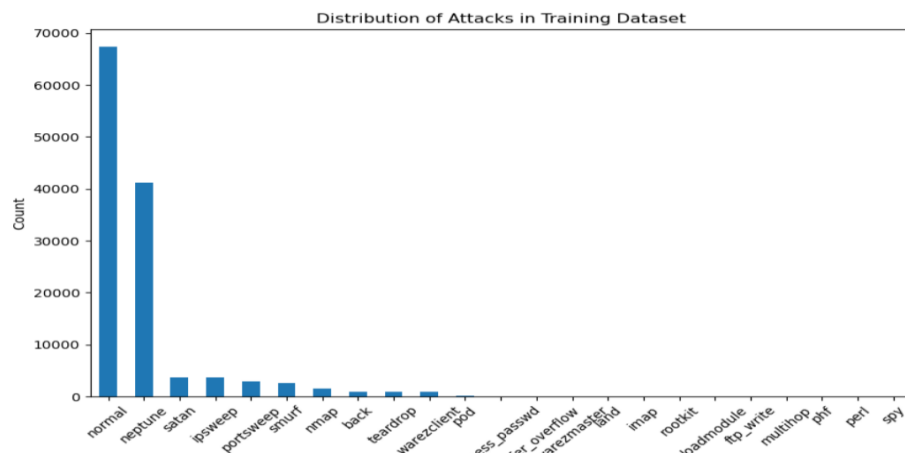
```
Out[8]: attack
normal      67343
neptune     41214
satan       3633
ipsweep     3599
portsweep   2931
smurf       2646
nmap        1493
back        956
teardrop    892
warezclient 890
pod         201
guess_passwd 53
buffer_overflow 30
warezmaster 20
land        18
imap        11
rootkit     10
loadmodule   9
ftp_write    8
multihop     7
phf          4
perl         3
spy          2
Name: count, dtype: int64
```

DATA EDA

```
In [9]: import pandas as pd
import matplotlib.pyplot as plt
```

```
In [10]: attack_counts = data['attack'].value_counts()

# Create a bar plot
plt.figure(figsize=(10, 6))
attack_counts.plot(kind='bar')
plt.title('Distribution of Attacks in Training Dataset')
plt.xlabel('Attack Type')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



Detection of Cyber Attacks in Networks using ML Techniques

```
In [14]: encoder = LabelEncoder()
categorical_columns = ['protocol_type', 'service', 'flag', 'attack']
for column in categorical_columns:
    data[column] = encoder.fit_transform(data[column])
```

```
In [15]: data
```

```
Out[15]:
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_same_srv_rate	dst_host_diff_srv_rate	dst
0	0	1	20	9	491	0	0	0	0	0	...	0.17	0.03	
1	0	2	44	9	146	0	0	0	0	0	...	0.00	0.60	
2	0	1	49	5	0	0	0	0	0	0	...	0.10	0.05	
3	0	1	24	9	232	8153	0	0	0	0	...	1.00	0.00	
4	0	1	24	9	199	420	0	0	0	0	...	1.00	0.00	
...
125968	0	1	49	5	0	0	0	0	0	0	...	0.10	0.06	
125969	8	2	49	9	105	145	0	0	0	0	...	0.96	0.01	
125970	0	1	54	9	2231	384	0	0	0	0	...	0.12	0.06	
125971	0	1	30	5	0	0	0	0	0	0	...	0.03	0.05	
125972	0	1	20	9	151	0	0	0	0	0	...	0.30	0.03	

125973 rows x 43 columns

```
In [18]: X = data.drop(['attack', 'last_flag'], axis=1)
y = data['attack']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [19]: models = {
    'Logistic Regression': LogisticRegression(max_iter=1000),
    'Naive Bayes': MultinomialNB(),
    'Decision Trees': DecisionTreeClassifier(),
    'Random Forest': RandomForestClassifier(),
    'Support Vector Machine': SVC()
}
```

```
In [20]: scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

ML DEPLOY

LOGISTIC REGRESSION

```
In [21]: logistic_model = LogisticRegression(max_iter=1000, solver='lbfgs') # Increase max_iter as needed
logistic_model.fit(X_train, y_train)

C:\Users\harik\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
n_iter_i = _check_optimize_result(
```

```
Out[21]: LogisticRegression
LogisticRegression(max_iter=1000)
```

```
In [22]: y_pred = logistic_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
confusion = confusion_matrix(y_test, y_pred)
```

```
In [23]: print("Logistic Regression:")
print(f"Accuracy: {accuracy}")
```

Logistic Regression:
Accuracy: 0.9862671164913673

DECISION TREE CLASSIFIER

```
In [25]: decision_tree_model = DecisionTreeClassifier(random_state=42)
decision_tree_model.fit(X_train, y_train)
```

```
Out[25]: DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)
```

```
In [26]: y_pred = decision_tree_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
confusion = confusion_matrix(y_test, y_pred)
```

```
In [27]: print("Decision Tree Model:")
print(f"Accuracy: {accuracy}")
```

```
Decision Tree Model:
Accuracy: 0.9967453859892836
```

RANDOM FOREST CLASSIFIER

```
In [29]: random_forest_model = RandomForestClassifier(random_state=42)
random_forest_model.fit(X_train, y_train)
```

```
Out[29]: RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
In [30]: y_pred = random_forest_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
confusion = confusion_matrix(y_test, y_pred)
```

```
In [31]: print(f"Accuracy: {accuracy}")
```

```
Accuracy: 0.9980154792617583
```

SUPPORT VECTOR MACHINE

```
In [33]: svm_model = SVC(kernel='linear', C=1.0, random_state=42)
svm_model.fit(X_train, y_train)
```

```
Out[33]: SVC
SVC(kernel='linear', random_state=42)
```

```
In [34]: y_pred = svm_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
confusion = confusion_matrix(y_test, y_pred)
```

```
In [35]: print(f"Accuracy: {accuracy}")
```

```
Accuracy: 0.9894820400873189
```

APPLICATION

BACKEND SOURCE CODE

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import joblib

app = Flask(__name__)
model = joblib.load('model.pkl')

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():

    int_features = [float(x) for x in request.form.values()]

    if int_features[0]==0:
        f_features=[0,0,0]+int_features[1:]
    elif int_features[0]==1:
        f_features=[1,0,0]+int_features[1:]
    elif int_features[0]==2:
        f_features=[0,1,0]+int_features[1:]
    else:
        f_features=[0,0,1]+int_features[1:]

    if f_features[6]==0:
        fn_features=f_features[:6]+[0,0]+f_features[7:]
    elif f_features[6]==1:
```

```
        fn_features=f_features[:6]+[1,0]+f_features[7:]
    else:
        fn_features=f_features[:6]+[0,1]+f_features[7:]

    final_features = [np.array(fn_features)]
    predict = model.predict(final_features)

    if predict==0:
        output='Normal'
    elif predict==1:
        output='DOS'
    elif predict==2:
        output='PROBE'
    elif predict==3:
        output='R2L'
    else:
        output='U2R'

    return render_template('index.html', output=output)

@app.route('/results',methods=['POST'])
def results():

    data = request.get_json(force=True)
    predict = model.predict([np.array(list(data.values()))])

    if predict==0:
        output='Normal'
    elif predict==1:
        output='DOS'
    elif predict==2:
```



```
        output='PROBE'
    elif predict==3:
        output='R2L'
    else:
        output='U2R'

    return jsonify(output)

if __name__ == "__main__":
    app.run()
```

HTML CODE

```
/* Style inputs with type="text", select elements */
input[type=text], select {
    width: 100%; /* Full width */
    padding: 12px; /* Some padding */
    border: 1px solid #ccc; /* Gray border */
    border-radius: 4px; /* Rounded borders */
    box-sizing: border-box; /* Make sure that padding and width stays in place */
    margin-top: 6px; /* Add a top margin */
    margin-bottom: 16px; /* Bottom margin */
    resize: vertical /* Allow the user to vertically resize the textarea (not horizontally) */
}

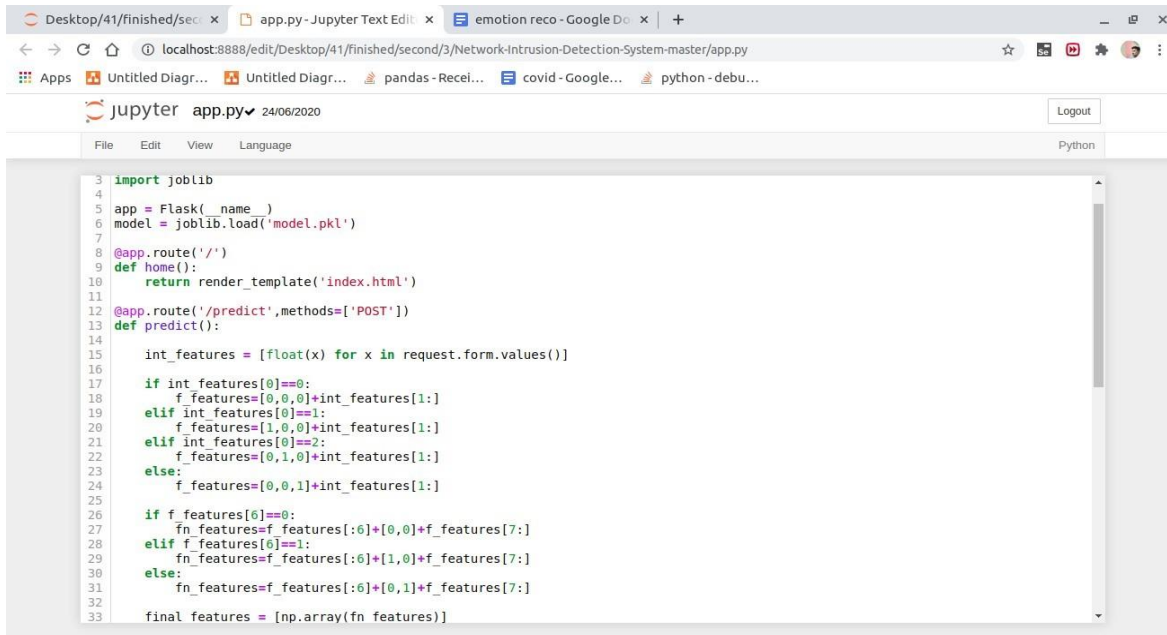
/* Style the submit button with a specific background color etc */
input[type=submit] {
    background-color: #4CAF50;
    color: white;
    padding: 12px 20px;
    border: none;
```

```
border-radius: 4px;
cursor: pointer;
}

/* When moving the mouse over the submit button, add a darker green color */
input[type=submit]:hover {
background-color: #45a049;
}

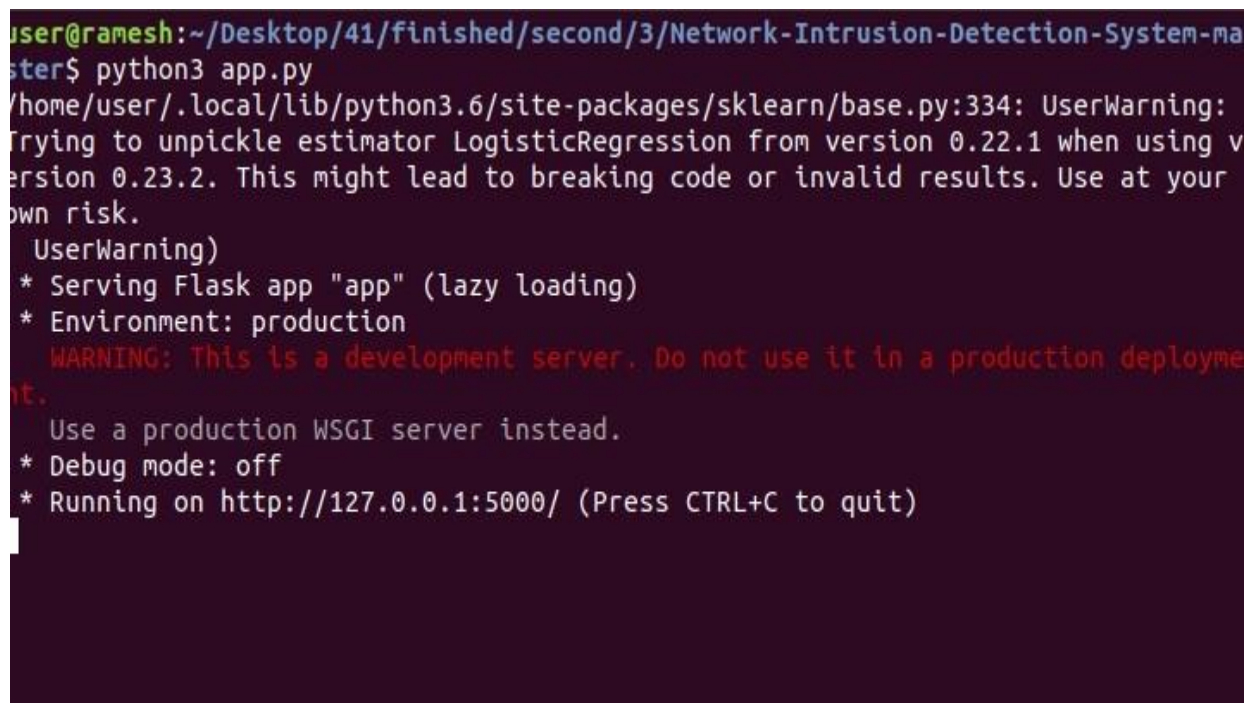
/* Add a background color and some padding around the form */
.login {
border-radius: 5px;
background-color: #f2f2f2;
padding: 20px;
}
```

APPLICATION



```
3 import joblib
4
5 app = Flask(__name__)
6 model = joblib.load('model.pkl')
7
8 @app.route('/')
9 def home():
10     return render_template('index.html')
11
12 @app.route('/predict', methods=['POST'])
13 def predict():
14
15     int_features = [float(x) for x in request.form.values()]
16
17     if int_features[0]==0:
18         f_features=[0,0,0]+int_features[1:]
19     elif int_features[0]==1:
20         f_features=[1,0,0]+int_features[1:]
21     elif int_features[0]==2:
22         f_features=[0,1,0]+int_features[1:]
23     else:
24         f_features=[0,0,1]+int_features[1:]
25
26     if f_features[6]==0:
27         fn_features=f_features[:6]+[0,0]+f_features[7:]
28     elif f_features[6]==1:
29         fn_features=f_features[:6]+[1,0]+f_features[7:]
30     else:
31         fn_features=f_features[:6]+[0,1]+f_features[7:]
32
33     final_features = [np.array(fn_features)]
```

LOCAL HOST IN CMD



```
user@ramesh:~/Desktop/41/finished/second/3/Network-Intrusion-Detection-System-master$ python3 app.py
/home/user/.local/lib/python3.6/site-packages/sklearn/base.py:334: UserWarning:
Trying to unpickle estimator LogisticRegression from version 0.22.1 when using version 0.23.2. This might lead to breaking code or invalid results. Use at your own risk.
  UserWarning)
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

RESULTS

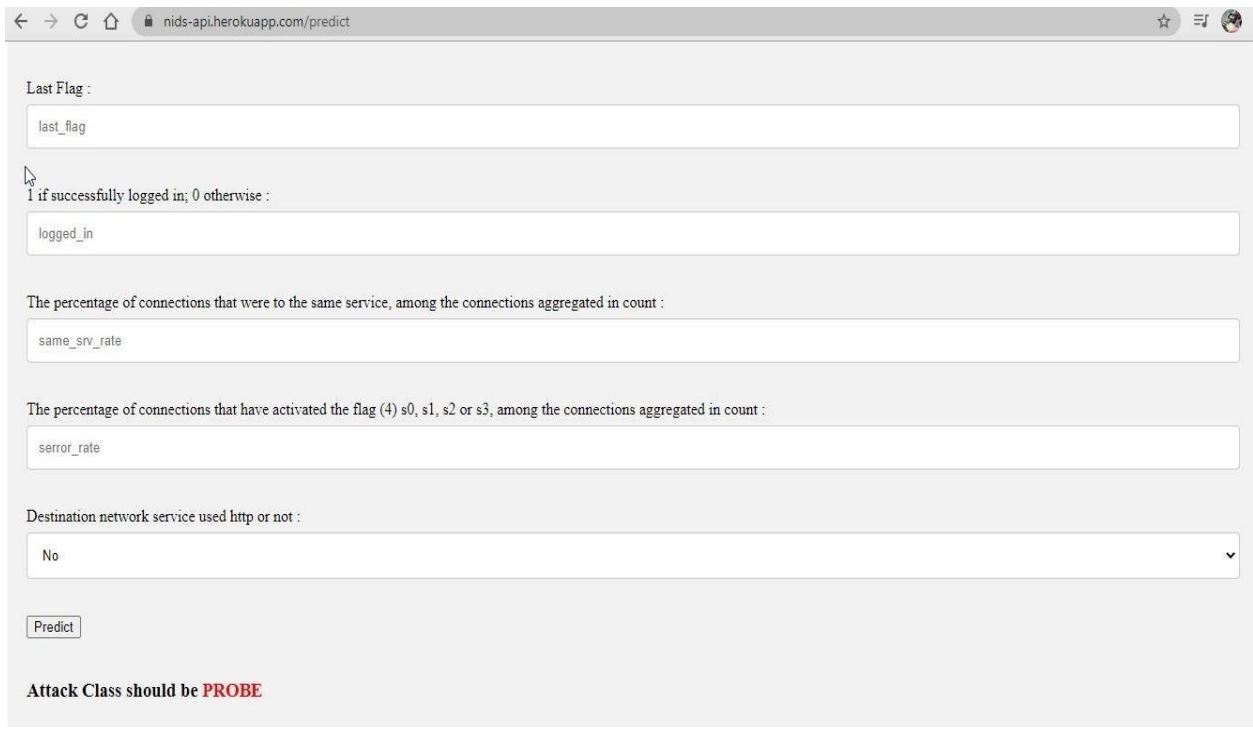
5.RESULTS

The experiments were conducted in Machine learning libraries like numpy, pandas, scikitlearn. Python language is used to develop the application with jupyter notebook IDE. Predictions can be done by four algorithms like SVM, ANN, RF, CNN this paper helps to identify which algorithm predicts the best accuracy rates which helps to predict best results to identify the cyber attacks happened or not.

Network Intrusion Detection System

Attack:
satan
Number of connections to the same destination host as the current connection in the past two seconds :
175
The percentage of connections that were to different services, among the connections aggregated in dst_host_count :
0.84
The percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_count :
0.00
The percentage of connections that were to the same service, among the connections aggregated in dst_host_count :
0.00
Number of connections having the same port number :
1

Status of the connection –Normal or Error :
Other
Last Flag :
18
1 if successfully logged in; 0 otherwise :
0
The percentage of connections that were to the same service, among the connections aggregated in count :
0.01
The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count :
0.10
Destination network service used http or not :
No
Predict



The screenshot shows a web browser window with the URL `nids-api.herokuapp.com/predict`. The page contains several input fields for features used in a machine learning model:

- Last Flag :** A text input field containing the value `last_flag`.
- 1 if successfully logged in; 0 otherwise :** A text input field containing the value `logged_in`.
- The percentage of connections that were to the same service, among the connections aggregated in count :** A text input field containing the value `same_srv_rate`.
- The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count :** A text input field containing the value `error_rate`.
- Destination network service used http or not :** A dropdown menu with the value `No` selected.

Below the input fields is a **Predict** button. At the bottom of the form, the output is displayed: **Attack Class should be PROBE**.

THE ATTACKS THAT ARE PREDICTED USING THIS APPLICATION ARE

Denial-of-Service-Attack (DoS): Intrusion where a for every child means to make a host out of reach to its genuine reason by momentarily or in some cases for all time disturbing administrations by flooding the objective machine with gigantic measures of solicitations and henceforth over-burdening the host.

User-to-Root-Attack (U2R): A classification of usually utilized move by the culprit start by attempting to access a client's previous access and misusing the openings to acquire root control.

Remote-to-Local-Attack (R2L): The interruption in which the aggressor can send information bundles to the objective however has no client account on that machine itself, attempts to abuse one weakness to acquire nearby access shrouding themselves as the current client of the objective machine.

Probing-Attack: The sort in which the culprit attempts to accumulate data about the PCs of the organization and a definitive target doing so is to move beyond the firewall and acquiring root access.

TESTING AND VALIDATION

6.TESTING AND VALIDATION

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

UNIT TESTING

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

INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Organization and preparation of functional tests is focused on requirements, key functions, or

special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

WHITE BOX TESTING

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

TESTING STRATEGY AND APPROACH

Field testing will be performed manually and functional tests will be written in detail.

TESTING OBJECTIVES

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

FEATURES TO BE TESTED

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

TEST RESULTS: All the test cases mentioned above passed successfully. No defects encountered.

ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

TEST RESULTS: All the test cases mentioned above passed successfully. No defects encountered.

CONCLUSION AND FUTURE ENHANCEMENTS

7.CONCLUSION AND FUTURE ENHANCEMENTS

In this approach, we have proposed the AI-SIEM system using event profiles and artificial neural networks. The novelty of our work lies in condensing very large-scale data into event profiles and using the deep learning-based detection methods for enhanced cyber- threat detection ability. The AI-SIEM system enables the security analysts to deal with significant security alerts promptly and efficiently by comparing long term security data. By reducing false positive alerts, it can also help the security analysts to rapidly respond to cyber threats dispersed across a large number of security events.

For the evaluation of performance, we performed a performance comparison using two benchmark datasets (NSLKDD, CICIDS2017) and two datasets collected in the real world. First, based on the comparison experiment with other methods, using widely known benchmark datasets, we showed that our mechanisms can be applied as one of the learning-based models for network intrusion detection. Second, through the evaluation using two real datasets, we presented promising results that our technology also outperformed conventional machine learning methods in terms of accurate classifications.

At the present time, assessments of support vector machine, ANN, CNN, Random Forest and significant learning estimations reliant upon current CICIDS2017 data set were presented moderately. Results show that the significant learning estimation performed generally best results over SVM, ANN, RF and CNN. We will use port scope attempts just as other attack types with AI and significant learning computations, Apache Hadoop and shimmer advancements together with on this data set later on. Every one of these estimation assists us with recognizing the digital assault in network. It occurs in the manner that when we think about long back a long time there might be such countless assaults occurred so when these assaults are perceived then the highlights at which these assaults are going on will be put away in some datasets. So by utilizing these datasets we will anticipate if digital assault is finished. These forecasts should be possible by four calculations like SVM, ANN, RF, CNN this paper assists with distinguishing which calculation predicts the best precision rates which assists with foreseeing best outcomes to recognize the digital assaults occurred or not.

REFERENCES

REFERENCES

- [1] N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, "A deep learning approach to network intrusion detection," IEEE Trans. Emerg. Topics Comput. Intell., vol. 2, pp. 41-50, Feb. 2018.
- [2] Prabhs Uyyala, JS University, India "Detection of Cyber Attacks using Machine Learning Techniqes ," Research gate, Mar.2022.
- [3] W. Wang, Y. Sheng and J. Wang, "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," IEEE Access, vol. 6, no. 99, pp. 1792-1806, 2018.
- [4] M. K. Hussein, N. Bin Zainal and A. N. Jaber, "Data security analysis for DDoS defense of cloud based networks," 2015 IEEE Student Conference on Research and Development (SCORED), Kuala Lumpur, 2015, pp. 305-310.
- [5] S. Sandeep Sekharan, K. Kandasamy, "Profiling SIEM tools and correlation engines for security analytics," In Proc. Int. Conf. Wireless Com., Signal Proce. and Net.(WiSPNET), 2017, pp. 717-721.
- [6] Doodipalli Subramanyam. Research Scholar, Dept. of Computer Science & Engineering, Visvesvaraya Technological University, Karnataka, "Classification of Intrusion Detection Dataset using machine learning Approaches " 2018.IEEE.
- [7] Prof. A. V. Deorankar, Shiwani S. Thakare, CSE Department GCOE Amravati, India, "Survey on anomaly Detection of (IOT) – Internet of things cyber attacks using machine learning " 2020.IEEE.
- [8] Y. Shen, E. Mariconti, P. Vervier, and Gianluca Stringhini, "Tiresias: Predicting Security Events Through Deep Learning," In Proc. ACM CCS 18, Toronto, Canada, 2018, pp. 592-605.

[9] Kyle Soska and Nicolas Christin, "Automatically detecting vulnerable websites before they turn malicious,", In Proc. USENIX Security Symposium., San Diego, CA, USA, 2014, pp.625-640.

[10] Kehe Wu, Zuge Chen, Wei Li, "A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks", *Access IEEE*, vol. 6, pp. 50850-50859, 2018