

CYBER SECURITY OF MOBILE APPLICATIONS USING ARTIFICIAL INTELLIGENCE

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

IMMAREDDY ROHINI (40110461)

TUMMALA ANUPAMA (40111357)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Category-1 University by UGC

Accredited with Grade “A++” by NAAC | 12B Status by UGC | Approved by AICTE

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119**

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **IMMAREDDY ROHINI (40110461), TUMMALA ANUPAMA (40111357)** who carried out the Project Report entitled "**CYBER SECURITY OF MOBILE APPLICATIONS USING ARTIFICIAL INTELLIGENCE**" under my supervision from Nov 2023 to April 2024.


Internal Guide

R. Anishwarya M.E., (Ph.D.)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.,

Submitted for Viva voce Examination held on 13/04/2024


Internal Examiner


External Examiner

DECLARATION

I, IMMAREDDY ROHINI (Reg. No 40110461), hereby declare that the Project Report entitled "CYBER SECURITY OF MOBILE APPLICATIONS USING ARTIFICIAL INTELLIGENCE" done by me under the guidance of R. Aishwarya M.E., (Ph.D.,) is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE: 13/04/2024

PLACE: Chennai



SIGNATURE OF CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **R.Aishwarya M.E., (Ph.D.,)** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Nowadays, with the evolution in wireless communication, there are many security threats over the internet. The intrusion detection system (IDS) helps to find the attacks on the system and the intruders are detected. Previously various machine learning (ML) techniques are applied on the IDS and tried to improve the results on the detection of intruders and to increase the accuracy of the IDS. This paper has proposed an approach to develop efficient IDS by using the principal component analysis (PCA) and the random forest classification algorithm. Where the PCA will help to organize the dataset by reducing the dimensionality of the dataset and the random forest will help in classification. Results obtained states that the proposed approach works more efficiently in terms of accuracy as compared to other techniques like SVM, Naïve Bayes, and Decision Tree. The results obtained by proposed method are having the values for performance time (min) is 3.24 minutes, Accuracy rate (%) is 96.78 %, and the Error rate (%) is 0.21 %. In the ever-evolving landscape of wireless communication, the security of internet transactions and data transfers has become a paramount concern. The proliferation of sophisticated cyber threats necessitates robust mechanisms to safeguard against unauthorized access and potential breaches. Intrusion Detection Systems (IDS) play a pivotal role in this defensive strategy by monitoring network traffic for suspicious activities and signs of malicious intent. The effectiveness of an IDS in identifying and mitigating threats hinges on its ability to discern between benign and potentially harmful behavior accurately. Historically, various machine learning (ML) techniques have been harnessed to enhance the predictive capabilities of IDS. These techniques include Support Vector Machines (SVM), Naïve Bayes classifiers, and Decision Trees, each with its strengths and limitations. However, despite their contributions to improving detection rates, these methods have encountered challenges related to high dimensionality of data and computational complexity, which can impede their efficiency and accuracy. The research presented in this paper introduces a novel approach to IDS development, which leverages the synergy between Principal Component Analysis (PCA) and the Random Forest classification algorithm. PCA serves as a powerful tool for pre-processing and transforming the data by reducing its dimensionality.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	vii
	LIST OF ABBREVIATION	ix
1	INTRODUCTION	1
2	LITERATURE SURVEY	4
	2.1 Inferences from Literature Survey	5
	2.2 Issues in Existing System	6
3	REQUIREMENT ANALYSIS	7
	3.1 Aim	7
	3.2 Scope	7
	3.3 Requirement Analysis	8
	3.4 Selecting Methodology or Process Model	10
	3.5 System Requirement & Software Environment	10
4	DESCRIPTION OF PROPOSED SYSTEM	11
	4.1 System Architecture	11
	4.2 Dataflow Diagram	12
	4.3 UML Diagrams	13
5	IMPLEMENTATION DETAILS	17
	5.1 Development and Deployment Setup	17
	5.2 Proposed Algorithm	21
	5.3 System Study	23
	5.4 System Testing	24
6	RESULTS AND DISCUSSIONS	28
	6.1 Login Page	28

6.2	Intrusion Detection System	29
6.3	Performance Analysis	30
6.4	Comparison between Proposed System and Existing System	32
6.5	Cyber Security of Mobile Applications Future	33
7	CONCLUSION	34
	REFERENCES	35
	APPENDIX	38
	A. Source Code	38
	B. Screenshots	62
	C. Research Paper	64
	D. Certificate	75

LIST OF FIGURES

FIGURE NO.	FIGURE NAMES	PAGE NO.
4.1	System Architecture	11
4.2	Data Flow Diagram	12
4.3	UML Diagrams	13
6.1	Login Page	28
6.2	Intrusion Detection System	29
6.3	Confusion Matrix	30
6.4	Comparison between Proposed System and Existing System	32
6.5	Cyber Security of Mobile Applications Future	33
B.1	KDD_Train	64

CHAPTER 1

INTRODUCTION

Intrusion of a computing system is an attempt to break into or misuse it. An intrusion is any kind of action that compromises the integrity, confidentiality and availability of some information or computer resource. Using the weakness or flaws in the system architecture, the intruder intrudes to circumvent the authentication or authorization process. With the tremendous growth of network based services and secured information on networks, network security is becoming more and more important than ever before. One solution to this is the use of Network Intrusion Detection System (NIDS) that detect attacks by observing various network activities. So it is more important that such systems should be more accurate in identifying attacks, quick to train and to generate as few false positives as possible. Therefore, it is very essential for such systems to be exceedingly correct in detecting attacks, learn quickly and create as hardly any misleading up-sides as practical. Intrusion discovery structures (IDS) help keep networks relaxed through detecting malicious intrusions. Therefore, IDS has emerged as a critical element of laptop networks. The necessities for an IDS are agility and agility. Safety is paramount in all projects to save you any loss. An essential characteristic of IDS is to offer statistics approximately unusual activity and to notify community administrators to echo/monitor and/or block suspicious connectivity. Additionally, an IDS can distinguish among internally generated assaults (coming from one's personal personnel, clients, or others) and externally generated (assaults executed with the aid of hackers). Common styles of intrusion detection structures (IDS) are community (network IDS) and host-primarily based (HIDS). Network IDS alone is based on detecting unlawful, illicit, and conflicting behavior in network visitors. Cyber threats targeting mobile applications are evolving at an alarming rate, ranging from data breaches and identity theft to malware and ransomware attacks. Traditional security measures such as firewalls and encryption techniques are no longer sufficient to effectively combat these sophisticated threats. This requires an innovative approach that can adapt and respond to the dynamic nature of cyber attacks.

Artificial Intelligence (AI) emerges as a promising solution to strengthen the security of mobile applications. Using the power of machine learning algorithms, AI can analyze vast amounts of data in real-time, detect anomalies and predict potential security breaches with a high degree of accuracy. Additionally, AI-powered solutions have the ability to continuously learn and adapt to new attack vectors, making them indispensable in the fight against cyber threats. In this paper, we study the role of AI in enhancing the security of mobile applications. We explore how AI technologies, including machine learning, natural language processing and anomaly detection, can be used to proactively identify and mitigate security vulnerabilities. Additionally, we discuss the challenges and ethical considerations associated with integrating AI into mobile application security frameworks. By integrating AI-powered security mechanisms into mobile apps, developers can strengthen their defenses against numerous cyber threats, protect user data, and increase overall trust in the digital ecosystem. The goal of this paper is to strengthen the security posture of mobile applications in the transformative potential of AI and promote a safer digital environment for all users.

Rapid Evolution of Mobile Applications: Start by elaborating on the significance of mobile applications in contemporary society. Mobile applications have fundamentally transformed various aspects of daily life, including communication, commerce, and entertainment. The ubiquity of smartphones has led to a proliferation of mobile apps, catering to diverse user needs and preferences.

Challenges in Mobile Application Security: Discuss the escalating challenges associated with securing mobile applications. The increasing prevalence of cyber threats, such as data breaches, malware attacks, and phishing scams, poses significant risks to user privacy and data integrity. Traditional security measures, while essential, often struggle to keep pace with the sophistication and agility of modern cyber attacks.

The Promise of Artificial Intelligence: Introduce the concept of Artificial Intelligence (AI) as a disruptive force in cybersecurity. AI encompasses a range of technologies, including machine learning, natural language processing, and pattern recognition, that enable systems to learn from data, identify patterns, and make autonomous decisions. Unlike static security approaches, AI offers the potential to adapt and evolve in response to emerging threats, making it a valuable asset in the battle against cyber adversaries.

AI-Powered Security Solutions: Explore the specific applications of AI in bolstering the security of mobile applications. Highlight how AI algorithms can analyze vast datasets to identify anomalies and detect potential security breaches in real-time. For example, machine learning models can learn normal user behavior patterns and flag deviations that may indicate unauthorized access or malicious activity. Similarly, AI-driven anomaly detection techniques can uncover previously unknown attack vectors and proactively thwart cyber threats before they escalate.

Continuous Learning and Adaptation: Emphasize the dynamic nature of AI-powered security solutions. Unlike traditional rule-based approaches, AI systems have the ability to continuously learn from new data and adapt their defense mechanisms accordingly. This adaptive capacity is particularly crucial in the context of mobile applications, where the threat landscape is constantly evolving. By leveraging AI, developers can stay ahead of emerging threats and minimize the risk of security breaches.

Challenges and Ethical Considerations: Acknowledge the potential challenges and ethical implications associated with integrating AI into mobile application security frameworks. Discuss concerns related to data privacy, algorithmic bias, and transparency in AI-driven decision-making. Highlight the importance of implementing robust governance frameworks and ethical guidelines to ensure responsible deployment of AI technologies in cybersecurity.

Conclusion and Future Outlook: Conclude by summarizing the transformative potential of AI in enhancing the security posture of mobile applications. Emphasize the need for collaboration between industry stakeholders, policymakers, and cybersecurity experts to harness the full benefits of AI while addressing its associated challenges. Provide a glimpse into future trends and advancements in AI-driven mobile application security, underscoring the ongoing importance of innovation and vigilance in safeguarding digital assets and user privacy.

CHAPTER 2

LITERATURE SURVEY

Literature survey is the most important step in 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 lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

2.1 INFERENCES FROM LITERATURE SURVEY

2.1.1 A Wireless Intrusion Detection System for 802.11 WPA3 Networks/2021-Conference

Our experimental results show that the AP is vulnerable to eight out of the nine attacks and the IDS is unable to detect any of them. We propose a design for a signature-based IDS, which incorporates techniques to detect all the above attacks. The main disadvantages of this system is IDS is unable to detect any of them.

2.1.2 Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset/2020

In our experiments, we considered both the binary and multiclass classification configurations. The results demonstrated that the XG Boost-based feature selection method allows for methods such as the DT to increase its test accuracy from 88.13 to 90.85% for the binary classification scheme.

2.1.3 A Hybrid Framework for Intrusion Detection in Healthcare Systems Using Deep Learning/2021-RESEARCH article

We have created a new Artificial Neural Network architecture that surpassed performance compared to the other algorithms obtaining ~99.2 % accuracy. We conducted our experiments on a Linux machine running Ubuntu 21.04 OS with 32 GB RAM and 8 GB NVidia GTX 1080 GPU to use distributed training of our algorithms to obtain results faster.

2.1.4 A Proposed Wireless Intrusion Detect ion Prevent ion and Attack System/2019-Conference

Now let's start by talking about the structure of this system. The traditional intrusion detection system automatically monitors and analyzes attacks that are representative of the presence of the human element, Research work in the development of defense systems focuses on the monitoring and analysis of attacks without citing real solutions to these attacks

2.1.5 Intrusion detection system for detecting wireless attacks in IEEE 802.11 networks/ 2019-Journal

The proposed KDE-HMM technique/method combines the advantages of both statistical and probabilistic properties to yield better results. The performance of the proposed KDE-HMM technique has been experimentally validated and it is found that the proposed KDE-HMM detects the aforementioned attacks with 98% accuracy.

2.2 ISSUES IN EXISTING SYSTEM

- Iftikhar Ahmad et. al, studied various machine learning algorithms for the intrusion detection system.
- They compared some of the techniques like SVM, Extreme learning machine and the random forest.
- The authors have stated the results as the Extreme machine learning method performs a way better as compared to other algorithms.
- B. Riyaz et. al., here worked to improve the quality of the dataset to provide it to the intrusion detection system.
- They have used a fuzzy rule-based feature selection technique for the improvement of the dataset.
- They used the KDD dataset and resulted shown dynamic growth in the result of the IDS.
- The systems which work over the internet suffer from various malicious activities.
- The major problem seen in this field is the intrusion in the system for violating the information.
- Existing results state that there may be some improvements to be done on terms of accuracy and the detection rates and the false alarm rate.
- Some other techniques can replace previously applied techniques such as SVM and Naïve Bayes.
- Also, the study states that the dataset can be improved by using some methods over it. To improve the quality of the input to the proposed system.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 AIM

The primary aim of this project is to develop an efficient Intrusion Detection System (IDS) using a combination of Principal Component Analysis (PCA) and the Random Forest classification algorithm. The focus is to enhance the accuracy of intrusion detection while minimizing the computational complexity often associated with high-dimensional datasets. By integrating PCA for dimensionality reduction and Random Forest for classification, the project seeks to achieve a more effective and time-efficient detection of intruders, ultimately improving the security of wireless communication networks.

3.2 SCOPE

1. Dimensionality Reduction: Utilizing PCA to reduce the number of features in the dataset, thereby simplifying the dataset without losing significant information. This is crucial for managing big data scenarios and high feature sets common in network traffic data.
2. Classification Improvement: Employing Random Forest, an ensemble learning method that can handle the reduced dataset to classify and identify potential intrusions with higher accuracy than other commonly used techniques such as Support Vector Machines (SVM), Naïve Bayes, and Decision Trees.
3. Performance Evaluation: Comparing the effectiveness of the proposed method with other algorithms by assessing key performance indicators such as accuracy rate, error rate, and the time required to train and test the model.
4. Application in Real-world Scenarios: Testing the IDS in diverse and dynamic environments to ensure its robustness and reliability in detecting a wide range of attacks, from simple to complex, in wireless communication networks.
5. Scalability: Ensuring that the proposed IDS can scale with growing data and evolving attack patterns, maintaining high performance without the need for excessive computational resources.
6. Usability: Providing a solution that can be potentially integrated into existing security systems and frameworks, making it accessible to organizations and individuals who require robust security measures against cyber threats.
7. Research Contribution: Contributing to the field of cybersecurity by presenting a novel approach that can be further explored and refined by researchers and practitioners to enhance network security measures.

3.3 REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analyzing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

3.3.1 Functional Requirements

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even capable of working around any failure.

Compatibility

It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

Flexibility

The flexibility of the project is provided in such a way that it has the ability to run on different environments being executed by different users.

Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information.

3.3.2 Non-Functional Requirements

Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

Maintainability

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyze, change and test the application.

3.4 SELECTING METHODOLOGY OR PROCESS MODEL

- Data Collection
- Dataset
- Data Preparation
- Model Selection
- Analyze and Prediction
- Accuracy on test set
- Saving the Trained Model

3.5 SYSTEM REQUIREMENT & SOFTWARE ENVIRONMENT

3.5.1 *Hardware Requirements*

System	:	Pentium i3 Processor.
Hard Disk	:	500 GB.
Monitor	:	15" LED
Input Devices	:	Keyboard, Mouse
Ram	:	4 GB

3.5.2 *Software Requirements*

Operating system	:	Windows 11.
Coding Language	:	Python
Database	:	MYSQL

3.5.3 *Risk and Analysis of Project*

The main objective of the system is intrusion detection using PCA (Principle component analysis) and random forest algorithm for classification.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

- The intrusion detection system works for the improvement of the system, which is affected by the intruders.
- This system can do the detection of the intruders.
- The proposed system tries to eliminate the existing problems related to the previous work.
- The proposed system consists of the two methods that are principal component analysis, and the other one is the random forest.
- The principal component analysis is used for the reduction of the dimension of the dataset; by this method, the dataset quality will be improved as the dataset may contain the correct attributes.
- After this, the random forest algorithm will be applied for the detection of the intruders

ADVANTAGES

- The error rate found in our proposed approach is very low as of .21%.
- As well, the accuracy obtained is much higher than previous algorithms.
- Also, the time taken for the performance is less than other algorithms.

4.1 SYSTEM ARCHITECTURE

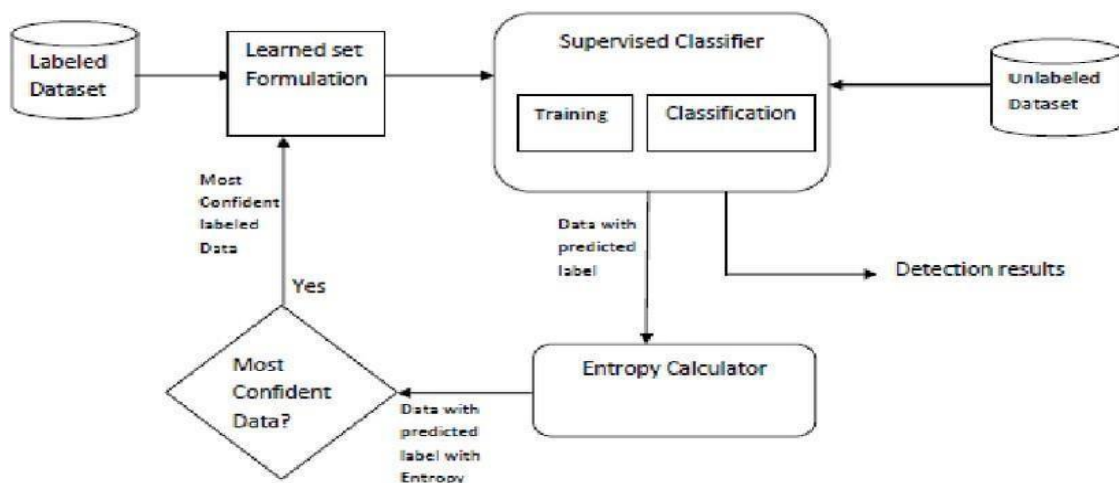


Fig 4.1 System Architecture

4.2 DATA FLOW DIAGRAM

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

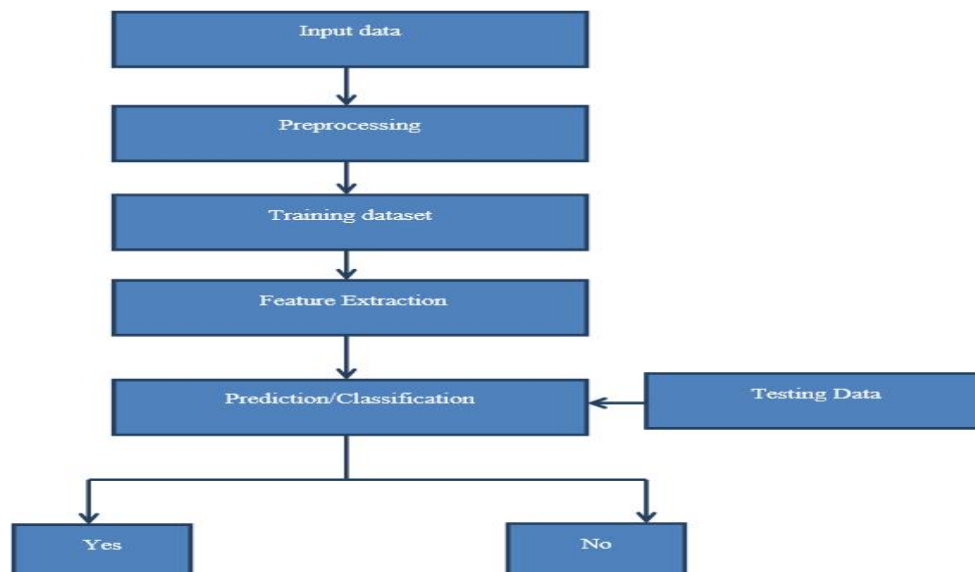


Fig 4.2 Dataflow Diagram

4.3 UML DIAGRAMS

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

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

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

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

GOALS

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

- 4.3.1 Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- 4.3.2 Provide extendibility and specialization mechanisms to extend the core concepts.
- 4.3.3 Be independent of particular programming languages and development process.
- 4.3.4 Provide a formal basis for understanding the modeling language.
- 4.3.5 Encourage the growth of OO tools market.
- 4.3.6 Support higher level development concepts such as collaborations, frameworks, patterns and components.
- 4.3.7 Integrate best practices.

4.3.1 Use Case Diagram

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

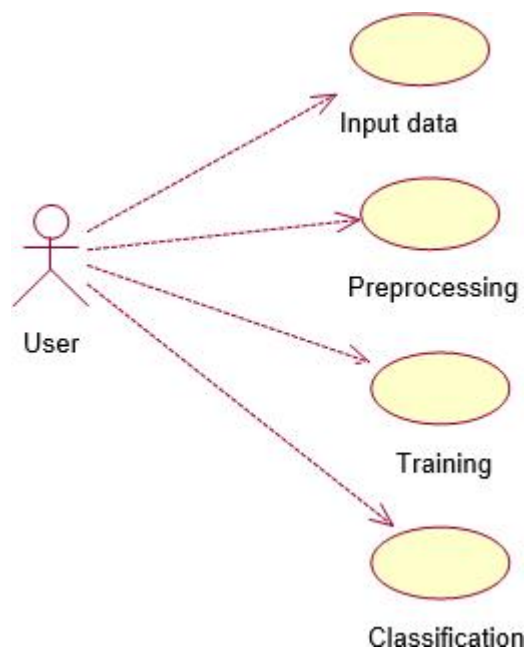


Fig 4.3.1 Use Case Diagram

4.3.2 Sequence Diagram

A sequence diagram in Unified Modelling 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.

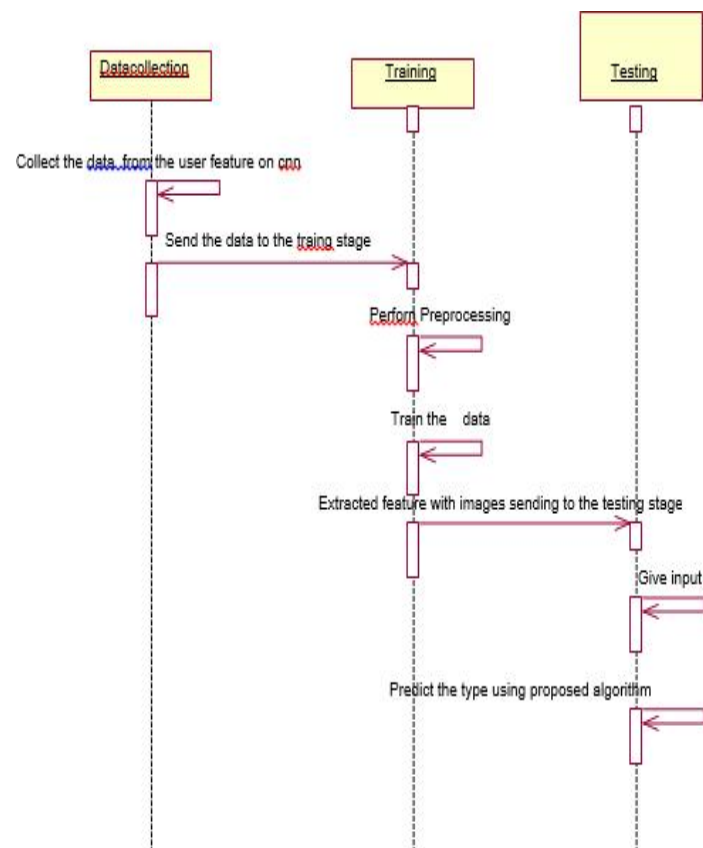


Fig 4.3.2 Sequence Diagram

4.3.3 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

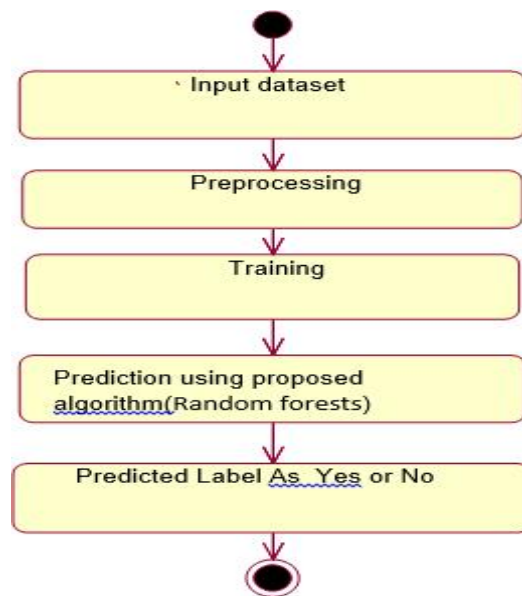


Fig 4.3.3 Activity Diagram

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

Intrusion detection systems paintings to protect structures from being focused by using attackers. This machine can locate incoming calls. The proposed device tries to conquer previous issues in the present paintings. The proposed machine includes techniques: foremost factor analysis, and the opposite is random wooded area. Principal thing analysis is used to reduce the dimensionality of a information set; With this method, the first-class of the dataset will improve because the dataset can comprise valid functions. Next, a random leap algorithm is used to come across intruders, which provides both better detection charges and fake alarms than SVM. This fig.1 explains the steps for monitoring Network Intrusion Detection System (NIDS) deployment in this situation. NIDS is a type of security software used to monitor computer networks for malicious activity. The system starts with a list of data written by individual elements. For NIDS, this information may include network traffic patterns marked as normal or disruptive. The system then uses this registration information to train the classification manager. A classifier is a machine learning algorithm that learns to recognize patterns in data associated with each category. Once the classifier is trained, it can be used to classify new, unlabeled items.

This is the first real step towards truly constructing a system studying version for statistics series. This is an critical step that depends on a good version: the better realities we get, the higher our model will complete. There are numerous techniques of records series like textual content scraping, manual intervention and so forth. The dataset used on this intrusion detection machine dataset is taken from kdd hyperlink:<http://kdd.lcs.Uci.Edu/databases/kddcup99/kddcup99.Html>.

The dataset contains 125974 character factors.

The statistics set has 42 columns, defined below. Let's alternate the records. By casting off missing information and doing away with a few sections. First permit's make a posting of section names that we need to store or store. Then, at that point, we erase or dispose of the multitude of segments other than the ones we need to hold.

At last, we dispose of or discard lines with missing qualities from the dataset. A distinction is made among education and evaluation. Head part investigation is a way frequently used to decrease the dimensionality of a data set. Head part investigation is one of the most extreme green and exact systems to diminish the dimensionality of records and favored results. This approach reduces the traits of the statistics to a favored variety of attributes, referred to as principals. This technique accepts every one of the information records as a dataset with an immense amount of capabilities and hence a very massive dataset size. This approach reduces the scale of the data by means of placing the records points at the same axis. The information factors are converted to an axis and come to be principal components. PCA can be achieved the use of the accompanying advances:

1. Take the given estimations along with the aspects in general.
2. Add the propose vector for each estimation d.
3. Add the covariance grid for the whole records set.
4. Compute eigenvectors ($e_1, e_2, e_3, \dots, e_d$) and eigenvalues ($v_1, v_2, v_3, \dots, v_d$).
5. Organize the eigenvalues in diving request and the n eigenvectors arranged by greatest eigenvalues to get $ad \times n = M$ lattice.
6. Utilizing this M, make another variant space.
7. The following lengths are top.

Irregular Backwoods is one of the most fundamental techniques in device getting to know to solve class troubles. Random forests fall underneath the class of supervised algorithms. This algorithm is achieved in awesome steps: the primary is to generate a forest of a given dataset, and the second one is to generate class predictions.

Analysis and Forecasting

In the real records set, we decided on only 10 features

1. Duration (number of seconds) of the connection
2. Protocol_type protocol type, eg tcp, udp etc.

3. Src_bytes number of data bytes from source to destination
4. Dst_bytes number of data bytes from destination to source
5. Is_hot_login 1 if the login belongs to the "hot" list; About otherwise
6. Is_guest_login 1 if login as "guest"; About otherwise
7. Diff_srv_rate % connections to different services
8. Srv_diff_host_rate % connections to different hosts
9. Indicate the normal or error status of the connection
10. Designation Normal or attackers

Indicates how long the connection took, measured in seconds.

Protocol_type: Indicates the type of communication protocol, such as TCP or UDP.

Src_bytes: This is the number of data bytes sent from the source (sender) to the destination (receiver) during the connection. Dst_bytes: Similarly, it represents the number of data bytes sent from the destination back to the source during the connection. Is_hot_login: This function indicates whether the login belongs to the "hot" list. Is_guest_login: Indicates whether the login is a guest login. Guest logins may have limited access compared to regular user logins. Diff_srv_rate: Represents the percentage of connections to different services. Srv_diff_host_rate: Similar to Diff_srv_rate, this function represents the percentage of connections to different hosts. Flag: Indicates the status of the connection, whether it is normal or an error has occurred. Labels: This is the connection classification that is categorized as "Normal" or "Attack". This is the target variable we want to predict. If a connection is classified as "Attack", it means that it is identified as potentially malicious, while "Normal" means that it is a harmless or expected behavior. Essentially, these features provide insight into various aspects of network activity, and by analyzing them, we can create models classifying connections as normal or potentially malicious based on historical patterns and known attack signatures.

Be Careful with the Check Set

We done ninety nine.1% accuracy at the take a look at set.

Save Learned Version

Step one is to transform it to .H5 or .H5. PKL library using Alex. Make positive ALEX is mounted on your environment. Next, import the module and unload the model into a .Pkl file.

Data collection

```
download_dataset("http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html") preprocess_data()
```

Data maintenance and preparation

```
select_columns(["relevant_feature_1", "relevant_feature_2", ...])  
remove_missing_values()
```

Principal Component Analysis (PCA)

```
apply_pca()
```

Random Forest Model

```
train_random_forest()
```

Analysis and Estimation

```
select_analysis_features(["selected_feature_1",  
"selected_feature_2", ...])evaluate_accuracy()
```

Save Learning Model

```
save_model_to_file("learned_model.pkl")
```

5.2 PROPOSED ALGORITHMS

Principle Component Analysis

Principal component analysis (PCA) is a statistical technique that uses orthogonal improvements to convert strongly and rapidly correlated variables into a set of uncorrelated variables. PCA is a widely used device in exploratory evaluation and gadget studying for predictive models. PCA could be very beneficial in identification process wherein there is heterogeneity amongst capabilities/variables. In the context of AI-powered mobile app cybersecurity, PCA can be used to identify and understand patterns in the behavior of mobile app data.

Simply, it works like this:

Data Collection:

First, you collect data about mobile app behavior. This may include information such as features accessed by the application, how it communicates with servers, user interactions, etc.

Feature Extraction:

PCA then takes this data and looks for patterns or correlations between different features (or characteristics) of applications. For example, it may find that certain types of applications tend to access specific parts of the device or have similar communication patterns. Dimensionality reduction: PCA helps reduce the complexity of this data by transforming it into a small set of features called principal components. These components are new variables that are combinations of the original features, but chosen to capture as much variability as possible in the data.

Anomaly detection:

Once the data is transformed, anomalies or unusual behavior can be more easily identified. For example, if an application suddenly starts accessing sensitive data that it does not normally have access to, this deviation from the norm can be flagged as a potential security threat.

Security Improvements:

With PCA, security systems can be better equipped to detect unusual activity or potential threats in mobile applications. It helps understand typical application behavior and detect deviations that could indicate malicious activity or vulnerabilities.

In short, PCA is a technique that helps to simplify and understand complex data patterns, making it useful for improving the security of mobile applications by detecting anomalies or unusual behavior using artificial intelligence-based systems.

Random Forest Algorithm

Random Forest is a popular algorithm in machine learning and can be very useful for cybersecurity in mobile applications. It combines the outcomes of a couple of decision trees right into a unanimous result. Ease of use and flexibility contributed to their adoption, as they solved both segmentation and regression issues.

In the context of mobile application security, Random Forest can be trained on a dataset containing features extracted from mobile applications. These features can include things like permissions required by the application, frequency of network requests, code complexity, etc.

Group decision-making process:

A Random Forest is like having a committee with decision-making powers, where each member (tree) has its own view of the data. Instead of relying on just one opinion, Random Forest aggregates the decisions of all trees to reach a final verdict. This ensemble approach often leads to more accurate and reliable results compared to individual models.

Feature extraction in mobile application security:

Features extracted from mobile applications serve as input to the Random Forest algorithm. These features provide crucial information about the behavior and properties of applications, such as the permissions they require, their communication patterns with external servers, and the complexity of their code. By analyzing these features, Random Forest can identify patterns indicative of malicious behavior.

Training phase:

During training, the random forest learns from a labelled dataset containing examples of both malicious and benign mobile applications. Making predictions: Once trained, a random forest can predict whether a new mobile app is malicious or safe based on its features. Each tree in the forest reviews the application separately and the final decision is made by majority vote.

Random Forest is efficient because it can handle large amounts of data with many features, is robust against overfitting (memorizing training data), and robust against data noise.

Continuous learning:

Additionally, Random Forest can adapt to new threats over time. As new data becomes available, the algorithm can be retrained to stay current with the latest security challenges.

In short, the Random Forest algorithm for mobile app security uses a group of decision trees to jointly determine whether an app is malicious or safe based on its features, making it a powerful tool in the cyber defense arsenal of AI-driven mobile apps.

5.3 SYSTEM STUDY

Feasibility Study

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

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

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.

5.4 SYSTEM TESTING

Testing Process

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.

5.4.1 Types of Testing

Unit Testing

Unit testing involves the design of test cases that validate that the internal Program logic is functioning properly, and that program input produces 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 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 satisfaction, as shown by successfully 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. Functional testing is centered on the following items: Valid Input is used to identified classes of valid input must be accepted. Invalid Input is used to identified classes of invalid input must be rejected. Functions is used to identified functions must be exercised. Output is used to identify classes of application outputs.

Systems/Procedures is used to interfacing systems or procedures must be invoked. 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 it's 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, such as specification or requirement 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.

Test Strategy and Approach

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

1. All field entries must work properly.
2. Pages must be activated from the identified link.
3. The entry screen, messages and responses must not be delayed. Features to be tested.
4. Verify that the entries are of the correct format No duplicate entries should be allowed.
5. 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.

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.

Alpha Testing

In software development, alpha test will be a test among the teams to confirm that your product works. Originally, the term alpha test meant the first phase of testing in a software development process. The first phase includes unit testing, component testing, and system testing. It also enables us to test the product on the lowest common denominator machines to make sure download times are acceptable and preloads work.

Beta Testing

In software development, a beta test is the second phase of software testing in which a sampling of the intended audience tries the product out. Beta testing can be considered "pre-release testing." Beta test versions of software are now distributed to curriculum specialists and teachers to give the program a "real-world" test.

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 LOGIN PAGE

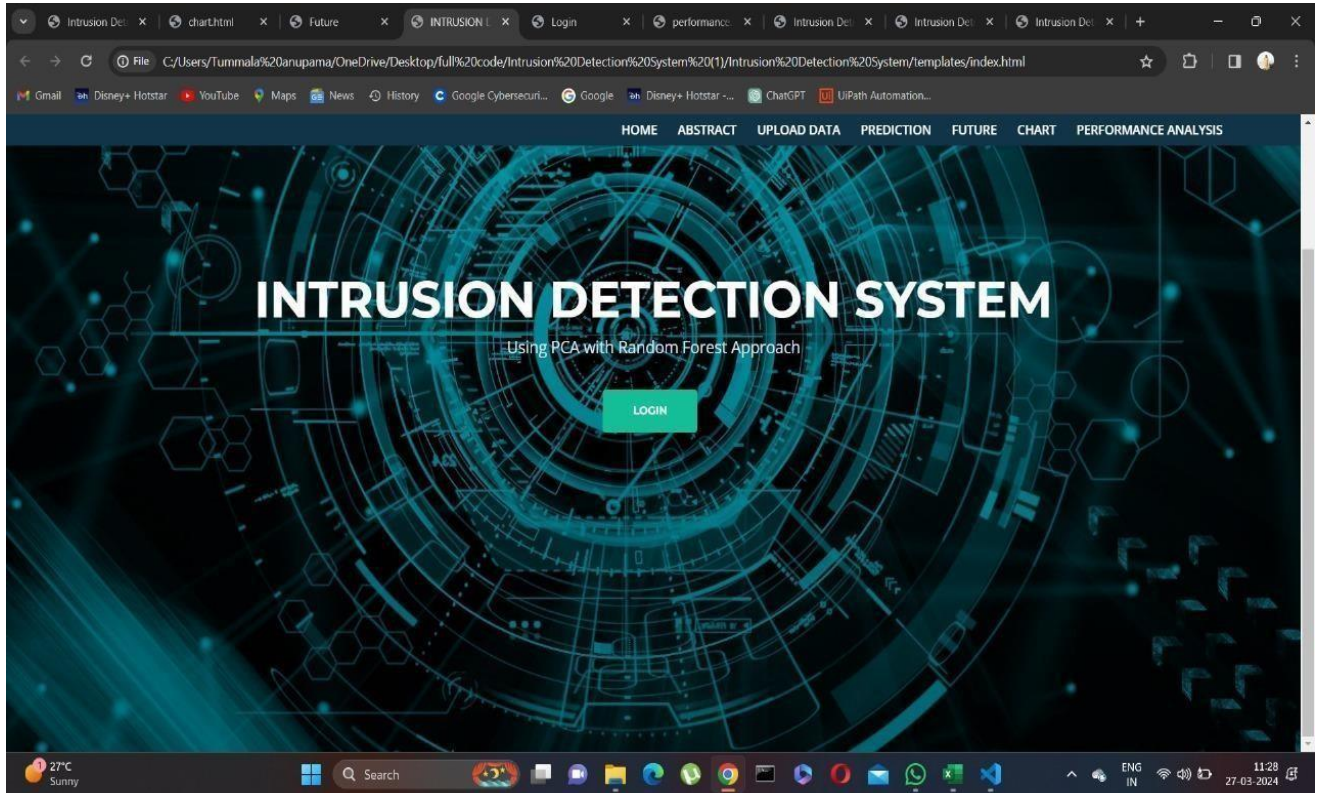


Fig 6.1 Login Page

As the involvement of the systems over the internet increasing rapidly, the security concerns have also seen. The proposed approach deals with the detection of intruders over the internet efficiently. The proposed algorithm has performed well as compared to the previously applied algorithms such as SVM, Naïve Bayes, and Decision Tree. The detection rates and the false error rates can be improved at a great extent by the proposed approach. The dataset used here is the knowledge discovery dataset. The results obtained by our proposed method having the values for Performance time (min) is 3.24 minutes, Accuracy rate (%) is 96.78 %, and the Error rate (%) is 0.21 %.

6.2 INTRUSION DETECTION SYSTEM

Intrusion Detection System(pie chart analysis)

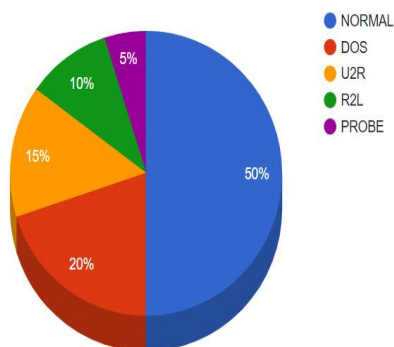


Fig 6.2 Intrusion Detection System

This pie chart titled "Intrusion Detection System (Magic Chart Analysis)" provides a breakdown of various aspects related to intrusion attempts detected by your system.

Depending on the specific configuration, it can visualize the distribution of attacks by type (denial-of-service, malware, etc.), location in your network (different segments or devices), severity levels of alerts generated (high, medium, low), or even the result of these attempts (successful penetrations, blocked or never completed). This visual representation allows security analysts to identify the most common threats, prioritize vulnerable areas, understand the severity of alerts, and assess the overall effectiveness of the intrusion detection system in defending your network.

A Denial-of-Service (DoS) attack is a malicious attempt to disrupt the normal functioning of a targeted server, service, or network by overwhelming it with a flood of illegitimate traffic or resource requests. DoS attacks aim to render the target unavailable to its intended users, thereby denying access to legitimate users.

A U2R attack, short for "User to Root" attack, is a type of cybersecurity attack in which an unauthorized user with limited privileges on a computer system or network attempts

to escalate their privileges to gain root or administrative access. These attacks are particularly concerning because they can allow an attacker to gain complete control over the targeted system, potentially compromising its security and integrity.

A Remote-to-Local (R2L) attack is a type of cybersecurity attack where an unauthorized user attempts to gain access to a computer system by exploiting vulnerabilities in remote services or protocols. Unlike User-to-Root (U2R) attacks, which aim to escalate privileges once inside the system, R2L attacks focus on exploiting weaknesses in network services or applications to gain initial access to a system from a remote location.

6.3 PERFORMANCE ANALYSIS

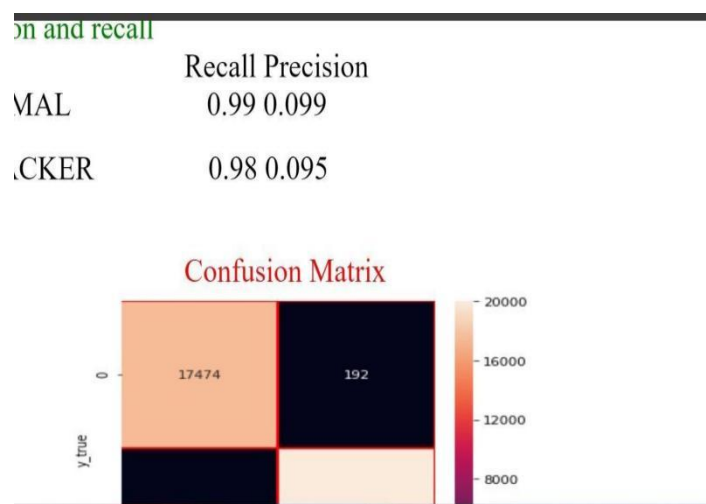


Fig 6.3.1 Confusion Matrix

The graph is a confusion matrix that compares the performance of a machine learning model in classifying between two classes. In this particular case, it compares the normal call to the attacker's call. The rows represent the actual classifications, while the columns represent the predicted classifications. Perfect performance would be achieved if all values were on the diagonal, meaning that all normal calls were correctly classified as normal and all attacker calls were correctly classified as attacker calls.

In this particular matrix, we can see that the model performs well in identifying normal calls, with 99% correctly classified. However, the model struggles more with calling attackers, only identifying 95% correctly. This means that 5% of the attacker's calls were misclassified as normal calls.

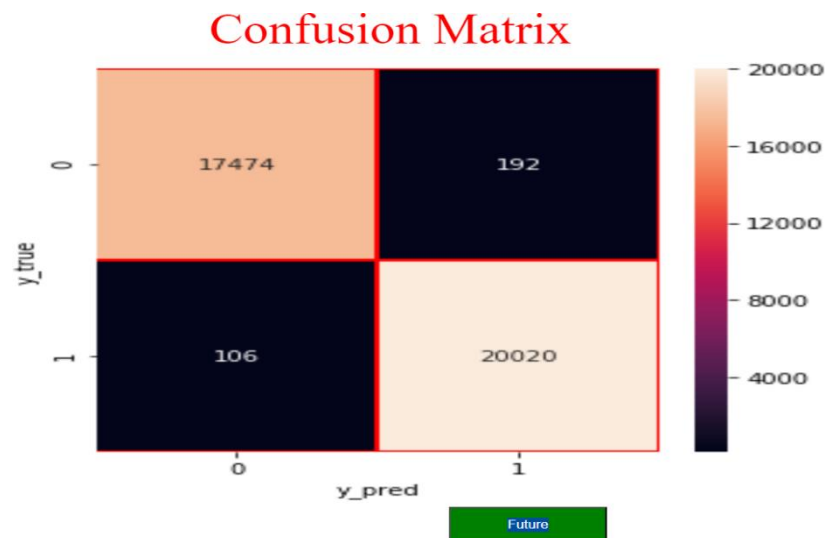


Fig 6.3.2 Confusion Matrix

Machine learning models are often used to secure mobile applications. These models are trained on datasets of known malicious and benign applications. The accuracy and recall of these models can be evaluated using graphs like the one you provided.

Security analysts use a combination of techniques to secure mobile applications. This can include static code analysis, dynamic analysis and penetration testing. Accuracy and recall are important metrics to consider when evaluating the effectiveness of these techniques. An acceptable balance between precision and recall may vary depending on the particular context. For example, if the consequences of a false positive are high (e.g., the use of a legitimate application is blocked), a security system with higher accuracy may be preferred.

6.4 COMPARISON BETWEEN EXISTING AND PROPOSED SYSTEM

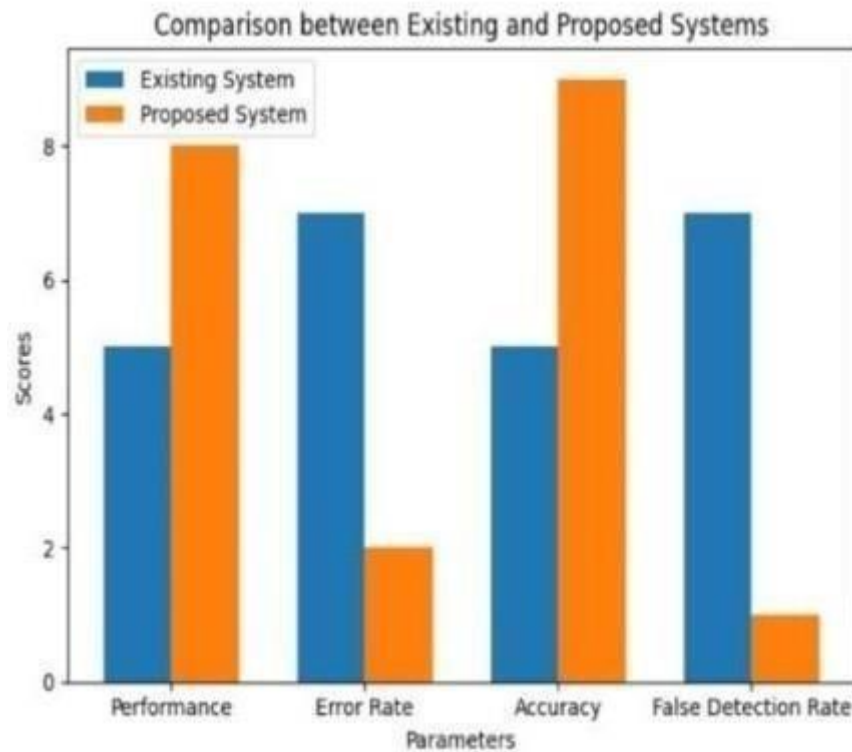


Fig 6.4 Comparison Between Existing and Proposed System

The chart compares the performance of existing and proposed systems in three metrics: error rate, accuracy, and false detection rate. The proposed system demonstrates superiority in all aspects, boasting lower error rate, higher accuracy and reduced false detection rate. This translates into a more reliable system with fewer errors and a significant reduction in the need for manual intervention due to false alarms. Furthermore, we can observe trends in the data, such as a potential positive correlation between accuracy and a particular variable for the proposed system. While the proposed system outperforms the existing one, future research could explore its behavior with even larger datasets and potential optimizations for even lower error rates.

6.5 CYBER SECURITY OF MOBILE APPLICATION FUTURE

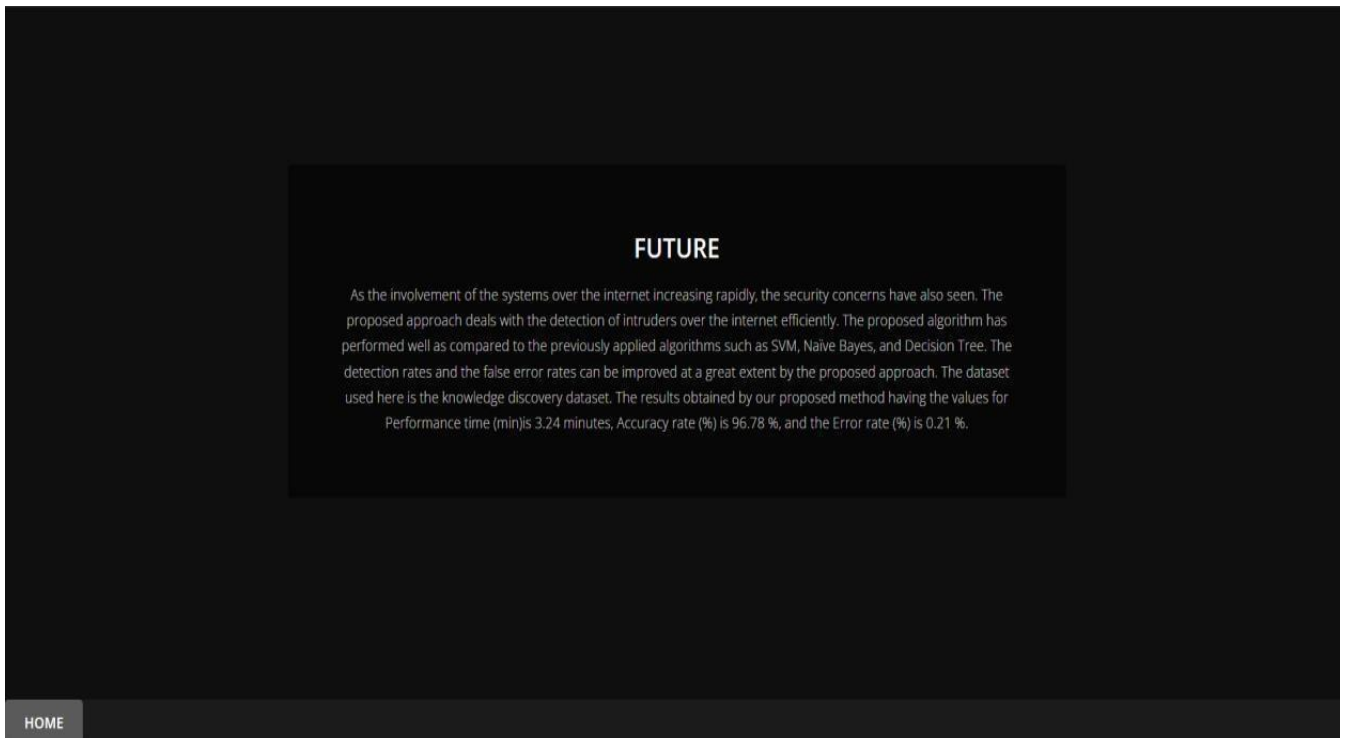


Fig 6.5 Cyber Security of Mobile Application Future

This excerpt delves into the future of Internet-connected systems and the evolving security landscape. It proposes a new approach to intrusion detection that outperforms established methods such as SVM, Naive Bayes, and Decision Trees. The text emphasizes the excellent detection and error rate of the algorithm when evaluating against a specific data set. Performance metrics such as processing time, accuracy, and error rate are also mentioned, suggesting a potentially valuable contribution to the field of network security.

CHAPTER 7

CONCLUSION

Nowadays, with the evolution in wireless communication, there are many security threats over the internet. The intrusion detection system (IDS) helps to find the attacks on the system and the intruders are detected. Previously various machine learning (ML) techniques are applied on the IDS and tried to improve the results on the detection of intruders and to increase the accuracy of the IDS. This paper has proposed an approach to develop efficient IDS by using the principal component analysis (PCA) and the random forest classification algorithm. Where the PCA will help to organize the dataset by reducing the dimensionality of the dataset and the random forest will help in classification. Results obtained states that the proposed approach works more efficiently in terms of accuracy as compared to other techniques like SVM, Naïve Bayes, and Decision Tree. The results obtained by proposed method are having the values for performance time (min) is 3.24 minutes, Accuracy rate (%) is 96.78 %, and the Error rate (%) is 0.21 %.

As the involvement of the systems over the internet increasing rapidly, the security concerns have also seen. The proposed approach deals with the detection of intruders over the internet efficiently. The proposed algorithm has performed well as compared to the previously applied algorithms such as SVM, Naïve Bayes, and Decision Tree. The detection rates and the false error rates can be improved at a great extent by the proposed approach. The dataset used here is the knowledge discovery dataset. The results obtained by our proposed method having the values for Performance time (min) is 3.24 minutes, Accuracy rate (%) is 96.78 %, and the Error rate (%) is 0.21 %.

REFERENCES

1. JafarAbo Nada; Mohammad Rasmi Al-Mosa, 2018 International Arab Conference on Information Technology (ACIT), A Proposed Wireless Intrusion Detection Prevention and Attack System.
2. Kinam Park; Youngrok Song; Yun-Gyung Cheong, 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigData Service), Classification of Attack Types for Intrusion Detection Systems Using a Machine Learning Algorithm.
3. S. Bernard, L. Heutte and S. Adam "On the Selection of Decision Trees in Random Forests" Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, June 14-19, 2009, 978-1-4244-3553-1/09/\$25.00 ©2009 IEEE.
4. A. Tesfahun, D. Lalitha Bhaskari, "Intrusion Detection using Random Forests Classifier with SMOTE and Feature Reduction" 2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies, 978-0-4799-2235-2/13\$26.00 © 2013 IEEE.
5. Le, T.-T.-H., Kang, H., & Kim, H. (2019). The Impact of PCA-Scale Improving GRU Performance for Intrusion Detection. 2019 International Conference on Platform Technology and Service (PlatCon). Doi:10.1109/platcon.2019.8668960.
6. Anish Halimaa A, Dr K.Sundarakantham: Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019) 978-1-5386-9439-8/19/\$31.00 ©2019 IEEE "MACHINE LEARNING BASED INTRUSIONDETECTION SYSTEM."
7. Mengmeng Ge, Xiping Fu, Naeem Syed, Zubair Baig, Gideon Teo, Antoni Robles-Kelly (2019). Deep Learning-Based Intrusion Detection for IoT Networks, 2019 IEEE 24 th Pacific Rim International Symposium on Dependable Computing (PRDC), pp. 256-265, Japan.
8. R. Patgiri, U. Varshney, T. Akutota, and R. Kunde, "An Investigation on Intrusion Detection System Using Machine Learning" 978-1-5386-9276- 9/18/\$31.00 c2018IEEE.

9. Abdullahi, M., Baashar, Y., Alhussian, H., Alwadain, A., Aziz, N., Capretz, L. F., & Abdulkadir, S. J. (2022). Detecting cybersecurity attacks in internet of things using artificial intelligence methods: A systematic literature review. *Electronics*, 11(2), 198.
10. Mughal, A. A. (2018). Artificial Intelligence in Information Security: Exploring the Advantages, Challenges, and Future Directions. *Journal of Artificial Intelligence and Machine Learning in Management*, 2(1), 22-34.
11. Gupta, R., Tanwar, S., Al-Turjman, F., Italiya, P., Nauman, A., & Kim, S. W. (2020). Smart contract 24746-24772privacy protection using AI in cyber-physical systems: tools, techniques and challenges. *IEEE access*.
12. Ullah, Z., Al-Turjman, F., Mostarda, L., & Gagliardi, R. (2020). Applications of artificial intelligence and machine learning in smart cities. *Computer Communications*, 154, 313-323.
13. Wu, H., Han, H., Wang, X., & Sun, S. (2020). Research on artificial intelligence enhancing internet of things security: A survey. *Ieee Access*, 8, 153826-153848.
14. Khan, S. U., Eusufzai, F., Azharuddin Redwan, M., Ahmed, M., & Sabuj, S. R. (2022). Artificial intelligence for cyber security: performance analysis of network intrusion detection. In *Explainable artificial intelligence for cyber security: Next generation artificial intelligence* (pp. 113-139). Cham: Springer International Publishing.
15. Martínez Torres, J., Iglesias Comesaña, C., & García-Nieto, P. J. (2019). Machine learning techniques applied to cybersecurity. *International Journal of Machine Learning and Cybernetics*, 10(10), 2823-2836.
16. Mijwil, M., & Aljanabi, M. (2023). Towards artificial intelligence-based cybersecurity: The practices and ChatGPT generated ways to combat cybercrime. *Iraqi Journal For Computer Science and Mathematics*, 4(1), 65-70.
17. Wazid, M., Das, A. K., Chamola, V., & Park, Y. (2022). Uniting cyber security and machine learning: Advantages, challenges and future research. *ICT express*, 8(3), 313-321.
18. Rohit Kumar Singh Gautam, Er. Amit Doegar; 2018 8 th International

Conference on Cloud Computing, Data Science & Engineering (Confluence) “An Ensemble Approach for Intrusion Detection System Using Machine Learning Algorithms.”

19. Kazi Abu Taher, Billal Mohammed Yasin Jisan, Md. Mahbubur Rahma, 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)“Network Intrusion Detection using Supervised Machine Learning Technique with Feature.

20. L. Haripriya, M.A. Jabbar, 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)” Role of Machine Learning in Intrusion Detection System: Review”.

21. Nimmy Krishnan, A. Salim, 2018 International CET Conference on Control, Communication, and Computing (IC4) “Machine Learning-Based Intrusion Detection for Virtualized Infrastructures”.

22. Mohammed Ishaque, Ladislav Hudec, 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS) “Feature extraction using Deep Learning for Intrusion Detection System.”

23. Aditya Phadke, Mohit Kulkarni, Pranav Bhawalkar, Rashmi Bhattad, 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)“A Review of Machine Learning Methodologies for Network Intrusion Detection.”

24. Iftikhar Ahmad , Mohammad Basher, Muhammad Javed Iqbal, Aneel Rahim, IEEE Access (Volume: 6) Page(s): 33789 – 33795 “Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection.”

25. B. Riyaz, S. Ganapathy, 2018 International Conference on Recent Trends in Advanced Computing (ICRTAC)” An Intelligent Fuzzy Rule-based Feature Selection for Effective Intrusion detection”.

APPENDIX

A. SOURCE CODE

```
import numpy as np
import pandas as pd
from flask import Flask, request, jsonify, render_template, redirect, flash, send_file
from sklearn.preprocessing import MinMaxScaler
from werkzeug.utils import secure_filename
import pickle
from sklearn.decomposition import PCA
```

```
app = Flask(__name__) #Initialize the flask App
```

```
model = pickle.load(open('sk.pkl', 'rb'))
pca = pickle.load(open('kdd.pkl', 'rb'))
```

```
@app.route('/')
@app.route('/')
@app.route('/index')
def index():
    return render_template('index.html')
@app.route('/login')
def login():
    return render_template('login.html')
@app.route('/chart')
def chart():
    return render_template('chart.html')
@app.route('/abstract')
```

```

def abstract():
    return render_template('abstract.html')
@app.route('/performance')
def performance():
    return render_template('performance.html')
@app.route('/future')
def future():
    return render_template('future.html')
@app.route('/upload')
def upload():
    return render_template('upload.html')
@app.route('/preview',methods=["POST"])
def preview():
    if request.method == 'POST':
        dataset = request.files['datasetfile']
        df = pd.read_csv(dataset,encoding = 'unicode_escape')
        df.set_index('Id', inplace=True)
        return render_template("preview.html",df_view = df)

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

@app.route('/predict',methods=['POST'])
def predict():
    features = [float(x) for x in request.form.values()]
    final_features = [np.array(features)]
    final_features = pca.transform(final_features)
    prediction = model.predict(final_features)
    output = prediction
    print(output)
    return render_template('test.html', prediction_text= output)

```

```

features = [float(x) for x in request.form.values()]
final_features = [np.array(features)]
final_features = pca.transform(final_features)
prediction = model.predict(final_features)
output = prediction
print(output)
return render_template('test.html', prediction_text= output)

```

```

if __name__ == "__main__":
    app.run(debug=True)

```

Cascading Style Sheets

```

@charset "UTF-8";
@import url(font-awesome.min.css);
@import
url("https://fonts.googleapis.com/css?family=Lobster|Open+Sans:300,400,600,700,800");

/*
    Road Trip by TEMPLATED
    templated.co @templatedco
    Released for free under the Creative Commons Attribution 3.0 license
    (templated.co/license)
*/

/* Reset */

html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6, p,
blockquote, pre, a, abbr, acronym, address, big, cite, code, del, dfn, em, img, ins,
kbd, q, s, samp, small, strike, strong, sub, sup, tt, var, b, u, i, center, dl, dt, dd, ol, ul,
li, fieldset, form, label, legend, table, caption, tbody, tfoot, thead, tr, th, td, article,

```

aside, canvas, details, embed, figure, figcaption, footer, header, hgroup, menu, nav, output, ruby, section, summary, time, mark, audio, video {

```
margin: 0;
padding: 0;
border: 0;
font-size: 100%;
font: inherit;
vertical-align: baseline;
```

```
}
```

article, aside, details, figcaption, figure, footer, header, hgroup, menu, nav, section {

```
display: block;
```

```
}
```

body {

```
line-height: 1;
```

```
}
```

ol, ul {

```
list-style: none;
```

```
}
```

blockquote, q {

```
quotes: none;
```

```
}
```

blockquote:before, blockquote:after, q:before, q:after

```
{content: ";
```

```
content: none;
```

```
}
```

table {

```
border-collapse: collapse;
```

```

        border-spacing: 0;
    }

    body {
        -webkit-text-size-adjust: none;
    }

```

/* Box Model */

```

*, *:before, *:after {
    -moz-box-sizing: border-box;
    -webkit-box-sizing: border-box;
    box-sizing: border-box;
}

```

/* Containers */

```

.container {
    margin-left: auto;
    margin-right: auto;
}

```

```

.container.\31 25\25 {
    width: 100%;
    max-width: 100em;
    min-width: 80em;
}

```

```

.container.\37 5\25 {
    width: 60em;
}

```

```

.container.\35 0\25 {

```

```
}
```

```
.container.\32 5\25 {  
    width: 20em;
```

```
}
```

```
.container {  
    width: 80em;
```

```
}
```

```
@media screen and (max-width: 1680px) {
```

```
    .container.\31 25\25 {  
        width: 100%;  
        max-width: 100em;  
        min-width: 80em;
```

```
    }
```

```
    .container.\37 5\25 {  
        width: 60em;
```

```
    }
```

```
    .container.\35 0\25 {  
        width: 40em;
```

```
    }
```

```
    .container.\32 5\25 {  
        width: 20em;
```

```
    }
```

```
    .container {  
        width: 80em;
```

```
    }
```

```
}
```

```
@media screen and (max-width: 1280px) {
```

```
    .container.\31 25\25 {  
        width: 100%;  
        max-width: 81.25em;  
        min-width: 65em;  
    }
```

```
    .container.\37 5\25 {  
        width: 48.75em;  
    }
```

```
    .container.\35 0\25 {  
        width: 32.5em;  
    }
```

```
    .container.\32 5\25 {  
        width: 16.25em;  
    }
```

```
    .container {  
        width: 65em;  
    }
```

```
}
```

```
@media screen and (max-width: 980px) {
```

```
    .container.\31 25\25 {  
        width: 100%;  
        max-width: 112.5%;  
        min-width: 90%;
```

```

    }

    .container.\37 5\25 {
        width: 67.5%;
    }

    .container.\35 0\25 {
        width: 45%;
    }

    .container.\32 5\25 {
        width: 22.5%;
    }

    .container {
        width: 90%;
    }
}

@media screen and (max-width: 736px) {

    .container.\31 25\25 {
        width: 100%;
        max-width: 112.5%;
        min-width: 90%;
    }

    .container.\37 5\25 {
        width: 67.5%;
    }

    .container.\35 0\25 {
        width: 45%;
    }
}

```

```
}
```

```
.
```

```
}
```

```
.wrapper.style1 input[type="text"]:focus,  
.wrapper.style1 input[type="password"]:focus,  
.wrapper.style1 input[type="email"]:focus,  
.wrapper.style1 select:focus,  
.wrapper.style1 textarea:focus  
    {border-color: #ffffff;  
    box-shadow: 0 0 0 1px #ffffff;  
    }
```

```
.wrapper.style1 .select-wrapper:before  
    {color: rgba(255, 255, 255, 0.25);  
    }
```

```
.wrapper.style1 input[type="checkbox"] + label,  
.wrapper.style1 input[type="radio"] + label  
    {color: #fdd1d1;  
    }
```

```
.wrapper.style1 input[type="checkbox"] + label:before,  
.wrapper.style1 input[type="radio"] + label:before  
    {background: rgba(255, 255, 255, 0.075);  
    border-color: rgba(255, 255, 255, 0.25);  
    }
```

```
.wrapper.style1 input[type="checkbox"]:checked + label:before,  
.wrapper.style1 input[type="radio"]:checked + label:before  
    {background-color: #ffffff;
```

```

        border-color: #ffffff;
        color: #F64747;
    }

.wrapper.style1 input[type="checkbox"]:focus + label:before,
.wrapper.style1 input[type="radio"]:focus + label:before
    {border-color: #ffffff;
      box-shadow: 0 0 0 1px #ffffff;
    }

.wrapper.style1 ::-webkit-input-placeholder
    {color: #fbb5b5 !important;
    }

.wrapper.style1 :-moz-placeholder
    {color: #fbb5b5 !important;
    }

.wrapper.style1 :-moz-placeholder
    {color: #fbb5b5 !important;
    }

.wrapper.style1 :-ms-input-placeholder
    {color: #fbb5b5 !important;
    }

.wrapper.style1 .formerize-placeholder
    {color: #fbb5b5 !important;
    }

.wrapper.style1 ul.alt li {
    border-top-color: rgba(255, 255, 255, 0.25);
}

```

```
.wrapper.style1 header p
    {color: #fbb5b5;
}

}
```

```
.wrapper.style2 input[type="submit"].alt,
.wrapper.style2 input[type="reset"].alt,
.wrapper.style2 input[type="button"].alt,
.wrapper.style2 button.alt,
.wrapper.style2 .button.alt
    { background-color:
0.15);
    transparent;
    box-shadow: inset 0 0 0 2px rgba(255, 255, 255,

    color: #f !important;
}
```

```
.wrapper.style2 input[type="submit"].alt:hover,
.wrapper.style2 input[type="reset"].alt:hover,
.wrapper.style2 input[type="button"].alt:hover,
.wrapper.style2 button.alt:hover,
.wrapper.style2 .button.alt:hover
    { background-color: rgba(255, 255,
0.075);
    255,

}
```

```
.wrapper.style2 input[type="submit"].alt:active,
.wrapper.style2 input[type="reset"].alt:active,
.wrapper.style2 .button.alt:active {
    background-color: rgba(255, 255, 255, 0.2);
}
```

```

.wrapper.style2
input[type="submit"].alt.icon:before,
.wrapper.style2 input[type="reset"].alt.icon:before,
.wrapper.style2 input[type="button"].alt.icon:before,
.wrapper.style2 button.alt.icon:before,
.wrapper.style2 .button.alt.icon:before
{color: #bdbdbd;
}

```

```

.wrapper.style2 input[type="submit"].special,
.wrapper.style2 input[type="reset"].special,
.wrapper.style2 input[type="button"].special,
.wrapper.style2 button.special,
.wrapper.style2 .button.special
{ background-color:
#F64747;color:
#ffffff !important;
}

```

```

.wrapper.style2 input[type="submit"].special:hover,
.wrapper.style2 input[type="reset"].special:hover,
.wrapper.style2 input[type="button"].special:hover,
.wrapper.style2 button.special:hover,
.wrapper.style2 .button.special:hover
{background-color: #f75f5f;
}

```

```

.wrapper.style2 input[type="submit"].special:active,
.wrapper.style2 input[type="reset"].special:active,
.wrapper.style2 input[type="button"].special:active,
.wrapper.style2 button.special:active,
.wrapper.style2 .button.special:active
{background-color: #f52f2f;
}

```

```
.wrapper.style2 label {  
    color: #ffffff;  
}
```

```
.wrapper.style2 input[type="text"],  
.wrapper.style2 input[type="password"],  
.wrapper.style2 input[type="email"],  
.wrapper.style2 select,  
.wrapper.style2 textarea {  
    background: rgba(255, 255, 255, 0.075);  
    border-color: rgba(255, 255, 255, 0.15);  
}
```

```
.wrapper.style2 input[type="text"]:focus,  
.wrapper.style2 input[type="password"]:focus,  
.wrapper.style2 input[type="email"]:focus,  
.wrapper.style2 select:focus,  
.wrapper.style2 textarea:focus  
    {border-color: #F64747;  
    box-shadow: 0 0 0 1px #F64747;  
}
```

```
.wrapper.style2 .select-wrapper:before  
    {color: rgba(255, 255, 255, 0.15);  
}
```

```
.wrapper.style2 input[type="checkbox"] + label,  
.wrapper.style2 input[type="radio"] + label  
    {color: #d6d6d6;  
}
```

```
.wrapper.style2 input[type="checkbox"] + label:before,  
.wrapper.style2 input[type="radio"] + label:before {
```

```

        background: rgba(255, 255, 255, 0.075);
        border-color: rgba(255, 255, 255, 0.15);
    }

.wrapper.style2 input[type="checkbox"]:checked + label:before,
.wrapper.style2 input[type="radio"]:checked + label:before
    {background-color: #F64747;
      border-color: #F64747;
      color: #ffffff;
    }

.wrapper.style2 input[type="checkbox"]:focus + label:before,
.wrapper.style2 input[type="radio"]:focus + label:before
    {border-color: #F64747;
      box-shadow: 0 0 0 1px #F64747;
    }

.wrapper.style2 ::-webkit-input-placeholder
    {color: #bdbdbd !important;
    }

.wrapper.style2 :-moz-placeholder
    {color: #bdbdbd !important;
    }

.wrapper.style2 :-moz-placeholder
    {color: #bdbdbd !important;
    }

.wrapper.style2 :-ms-input-placeholder
    {color: #bdbdbd !important;
    }

.wrapper.style2 .formerize-placeholder {

```

```
        color: #bdbdbd !important;
    }
```

```
.wrapper.style2 ul.alt li {
    border-top-color: rgba(255, 255, 255, 0.15);
```

```
color: #fdd1d1;
cursor: default;
height: 3.25em;
left: 0;
line-height: 3.25em;
position: fixed;
text-align: right;
top: 1em;
width: 100%;
z-index: 10001;
padding: 0 1.5em;
}
```

```
#header.alt {
    top: 0;
    background: #F64747;
}
```

```
#header .logo {
    display: inline-block;
    height: inherit;
    left: 1.5em;
    line-height: inherit;
    margin: 0;
    padding: 0;
    position: absolute;
    top: 0;
}
```

```
#header .logo a {  
    font-size: 1.25em;  
    color: #fff;  
    font-weight: 600;
```

```
#header > a[href="#menu"]:before  
    {content: "\f0c9";  
    -moz-osx-font-smoothing: grayscale;  
    -webkit-font-smoothing: antialiased;  
    font-family: FontAwesome;  
    font-style: normal;  
    font-weight: normal;  
    text-transform: none !important;  
}
```

```
#header > a[href="#menu"]:hover  
    { color: rgba(255, 255, 255,  
    0.25);background: none;  
    box-shadow: 0 0 0 1px rgba(255, 255, 255, 0.25);  
}
```

```
#header > a[href="#menu"]:before  
    {margin: 0;  
}
```

```
#header > a[href="#menu"] span  
    {display: none;  
}
```

```
@media screen and (max-width: 736px)
```

```
{#header > a {
```

```
        padding: 0 0.5em;
    }

}
```

@media screen and (max-width: 980px)

```
{body.subpage {
    padding-top: 50px;
}

#header {
    height: 50px;
    line-height: 50px;
}

#header > h1 {
    left: 1em;
}

#header > h1 a {
    font-size: 1em;
}

}
```

@media screen and (max-width: 480px)

```
{#header {
    min-width: 320px;
    top: 0;
    font-size: .9em;
    background: rgba(0, 0, 0, 0.9);
}
```

```

    }

/* Menu */

padding: 0;
}

#menu > ul.links > li {
    padding: 0;
}

#menu > ul.links > li > a:not(.button)
    {border: 0;
border-top: solid 1px rgba(255, 255, 255,
0.25);

color: inherit;
display: block;
line-height: 3.5em;
text-decoration: none;
}

#menu > ul.links > li > .button
    {display: block;
margin: 0.5em 0 0 0;
}

#menu > ul.links > li:first-child > a:not(.button)
    {border-top: 0 !important;
}

#menu .close {
text-decoration: none;

```

```

-moz-transition: color 0.2s ease-in-out;
-webkit-transition: color 0.2s ease-in-out;
-ms-transition: color 0.2s ease-in-out;
transition: color 0.2s ease-in-out;
-webkit-tap-highlight-color: transparent;
border: 0;
color: #fbb5b5;
cursor: pointer;
display: block;
height: 3.25em;
line-height: 3.25em;
padding-right: 1.25em;
position: absolute;
right: 0;
text-align: right;
top: 0;
vertical-align: middle;
width: 7em;
}

```

```

#menu .close:hover {
    color: #ffffff;
}

```

@media screen and (max-width: 736px)

```

{#menu .close {
    height: 4em;
    line-height: 4em;
}

}

```

```
#menu.visible {
    -moz-transform: translateX(0);
    -webkit-transform: translateX(0);
    -ms-transform: translateX(0);
    transform: translateX(0);
    box-shadow: 0 0 1.5em 0 rgba(0, 0, 0, 0.2);
    visibility: visible;
}
```

```
@media screen and (max-width: 736px)
```

```
{#menu {
    padding: 2.5em 1.75em;
}
```

```
}
```

```
/* Banner */
```

```
#banner {
    display: -ms-flexbox;
    -ms-flex-pack: center;
    -ms-flex-align: center;
    padding: 8em 0 6em 0;
    -moz-align-items: center;
    -webkit-align-items: center;
    -ms-align-items: center;
    align-items: center;
    display: -moz-flex;
    display: -webkit-flex;
    display: -ms-flex;
    display: flex;
    -moz-justify-content: center;
```

-webkit-justify-content: center

Java Script

```
(function ($)
{
    "use strict";

    [ Validate ]*/

    var name = $('#.validate-input input[name="name"]');
    var email = $('#.validate-input input[name="email"]');
    var subject = $('#.validate-input input[name="subject"]');
    var message = $('#.validate-input textarea[name="message"]');
```

```
$('#.validate-
form').on('submit',function(){var
check = true;
```

```
if($(name).val().trim() ==
    ""){showValidate(name);
    check=false;
}
}
```

```
if($(subject).val().trim() ==
    ""){showValidate(subject);
    check=false;
}
}
```

```
if($(email).val().trim().match(/^([a-zA-Z0-9_\-\.]+)@((\[[0-9]{1,3}\.
[0-9]{1,3}\.
[0-9]{1,3}\.
)|((\[[a-zA-Z0-9\-\]+\.)+))([a-zA-Z]{1,5}|[0-9]{1,3})(\|)?$/)) == null) {
    showValidate(email);
    check=false;
}
}
```

```
if($(message).val().trim() == ""){
```

```

        showValidate(message);
        check=false;
    }

    return check;
});

$('.validate-form .input1').each(function(){
    $(this).focus(function(){
        hideValidate(this);
    });
});

function showValidate(input) {
    var thisAlert = $(input).parent();

    $(thisAlert).addClass('alert-validate');
}

function hideValidate(input) {
    var thisAlert = $(input).parent();

    $(thisAlert).removeClass('alert-validate');
}

})(jQuery);

```

Hypertext Preprocessor

<?php

```

if (empty($_POST["name"])) {
    $errorMsg = "Name is required ";
} else {
    $name = $_POST["name"];
}

if (empty($_POST["phone"])) {
    $errorMsg = "Phone is required ";
} else {
    $phone = $_POST["phone"];
}

if (empty($_POST["email"])) {
    $errorMsg = "Email is required ";
} else {
    $email = $_POST["email"];
}

if (empty($_POST["select"])) {
    $errorMsg = "Select is required ";
} else {
    $select = $_POST["select"];
}

if (empty($_POST["terms"])) {
    $errorMsg = "Terms is required ";
} else {
    $terms = $_POST["terms"];
}

```

```
}
```

```
$EmailTo = "yourname@domain.com";
```

```
$Subject = "New quote request from Aria landing page";
```

```
// prepare email body text
```

```
$Body = "";
```

```
$Body .= "Name: ";
```

```
$Body .= $name;
```

```
$Body .= "\n";
```

```
$Body .= "Phone: ";
```

```
$Body .= $phone;
```

```
$Body .= "\n";
```

```
$Body .= "Email: ";
```

```
$Body .= $email;
```

```
$Body .= "\n";
```

```
$Body .= "Package: ";
```

```
$Body .= $select;
```

```
$Body .= "\n";
```

```
$Body .= "Terms: ";
```

```
$Body .= $terms;
```

```
$Body .= "\n";
```

```
// send email
```

```
$success = mail($EmailTo, $Subject, $Body, "From: ".$email);
```

```
// redirect to success page
```

```
if ($success && $errorMsg == ""){echo
```

```
    "success";
```

```
}
```

B. SCREENSHOTS

KDD_TRAIN

KDD, which stands for Knowledge Discovery in Databases, is a process used in data mining and data science to extract useful information from large datasets. In the context of cybersecurity for mobile applications, KDD can be a valuable tool for analyzing various aspects of data related to security threats, user behaviors, network traffic, and system vulnerabilities.

The first step in KDD is to gather relevant data. For cybersecurity in mobile applications, this can include logs of user interactions, system events, network traffic, and application behavior. Data preprocessing involves cleaning the data, removing noise, handling missing values, and transforming the data into a format suitable for analysis. In the case of mobile application security, preprocessing may involve parsing log files, extracting features from network packets, and normalizing data.

Feature selection or extraction is the process of choosing the most relevant attributes from the dataset that will be used for analysis. In the context of mobile application security, features could include permissions requested by the application, user activity patterns, API calls made by the application, etc.

This is the core step of KDD where various data mining techniques such as classification, clustering, association rule mining, and anomaly detection are applied to extract useful patterns or insights from the data. For example, classification algorithms can be used to classify user behaviors as normal or suspicious, while anomaly detection techniques can help identify unusual patterns indicative of a security breach.

Once patterns or insights are extracted, they need to be evaluated to assess their usefulness and reliability. Evaluation metrics such as accuracy, precision, recall, and F1-score can be used to measure the performance of the data mining models in detecting security threats.

FileHomeInsertPage LayoutFormulasDataReviewViewHelpTell me what you want to do

CutCopyFormat PainterClipboardFont

Calibri11A A

Wrap TextMerge & CenterAlignment

GeneralConditional FormattingFormat as TableCell StylesStyles

InsertDelete FormatCells

AutoSumFillClearSort & Find & Filter SelectAdd-insEditing

Add-ins

POSSIBLE DATA LOSSSome features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format. Don't show againSave As...

A1

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | |
|----|----|----------|----------|------------|------|-----------|-----------|------|------------|--------|-----|------------|-----------|---------|------------|------------|----------|----------|-----------|----------|----------|---------|----------|--------|
| 1 | Id | duration | protocol | service | flag | src_bytes | dst_bytes | land | wrong_flag | urgent | hot | num_failed | logged_in | num_com | root_shell | su_attempt | num_root | num_file | num_shell | num_acce | num_outb | is_host | is_guest | is_col |
| 2 | 1 | 0 | tcp | ftp_data | SF | 491 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 3 | 1 | 0 | udp | other | SF | 146 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 4 | 1 | 0 | tcp | private | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 5 | 1 | 0 | tcp | http | SF | 232 | 8153 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 6 | 1 | 0 | tcp | http | SF | 199 | 420 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 7 | 1 | 0 | tcp | private | REJ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 8 | 1 | 0 | tcp | private | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 9 | 1 | 0 | tcp | private | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 10 | 1 | 0 | tcp | remote_jo | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 11 | 1 | 0 | tcp | private | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 12 | 1 | 0 | tcp | private | REJ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 13 | 1 | 0 | tcp | private | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 14 | 1 | 0 | tcp | http | SF | 287 | 2251 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 15 | 1 | 0 | tcp | ftp_data | SF | 334 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 16 | 1 | 0 | tcp | name | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 17 | 1 | 0 | tcp | netbios_ns | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 18 | 1 | 0 | tcp | http | SF | 300 | 13788 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 19 | 1 | 0 | icmp | eco_i | SF | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 20 | 1 | 0 | tcp | http | SF | 233 | 616 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 21 | 1 | 0 | tcp | http | SF | 343 | 1178 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 22 | 1 | 0 | tcp | mtp | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 23 | 1 | 0 | tcp | private | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 24 | 1 | 0 | tcp | http | SF | 253 | 11905 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 25 | 1 | 5607 | udp | other | SF | 147 | 105 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 26 | 1 | 0 | tcp | mtp | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 27 | 1 | 507 | tcp | telnet | SF | 437 | 14421 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | |

kdd_train

27°C Sunny

27-03-2024

11:18

Fig B.1 KDD_Train

KDD, which stands for Knowledge Discovery in Databases, is a process used in data mining and data science to extract useful information from large datasets. In the context of cybersecurity for mobile applications, KDD can be a valuable tool for analyzing various aspects of data related to security threats, user behaviors, network traffic, and system vulnerabilities.

C. RESEARCH PAPER

CSMA: CYBER SECURITY FOR MOBILE APPLICATIONS USING ARTIFICIAL INTELLIGENCE

R. Aishwarya¹, J. Albert Mayan², Tummala Anupama³ and Immareddy Rohini⁴

¹Assistant Professor, ²Professor, ³ and ⁴Student
Department of Computer Science and Engineering
Sathyabama Institute of Science and Technology

aishwarayarajkumar7@gmail.com, albertmayan@gmail.com, anupamakarthik03052003@gmail.com and rohinireddyimma@gmail.com

Abstract

Nowadays with the development of wireless communications, many security threats have emerged. It was located that the website was hacked by an Intrusion Detection System (IDS) enterprise and the vandals had been arrested. First, immersive expertise system acquisition (ML). These strategies are carried out in IDS and designed to enhance detection overall performance. Proper adjustment of attackers and harm. In this paper, they proposed a technique to build a effective IDS using most important element analysis (PCA) set of rules and random woodland classes. This is wherein PCA can help. Decide a way to prepare the records by means of lowering the quantity of gadgets and randomizing them. Makes it easier in wild locations. The results obtained show that the quantitative approach performs more effectively in terms of accuracy compared to other techniques that include SVM, Naive Bayes, and Decision Trees. The results obtained using the proposed approach have execution time values (minimum) of three. 24 minutes, the accuracy is 96.78% and the error fee (%) is zero.21%.

Keywords - Cyber Security, Machine Learning, PCA, Artificial Intelligence, Intrusion Detection System.

INTRODUCTION

An intruder tries to hack or misuse a computer device. Intrusion is any act that compromises the trustworthiness, classification, and accessibility of any records or computer sources. Through capacity vulnerabilities or flaws within the system structure, an attacker tries to skip the authentication or authorization manner. With the fast increase of network offerings and the security of information at the community, community safety is extra important than ever. One approach to this problem is to apply Network Intrusion Detection Systems (NIDS), which hit upon numerous community sports by tracking for attacks. Therefore, its miles very essential for such systems to be exceedingly correct in detecting attacks, learn quickly and create as hardly any misleading up-sides as practical. Interruption discovery structures (IDS) help keep networks relaxed through detecting malicious intrusions. Therefore, IDS has emerge as an critical element of laptop networks. The necessities for an IDS are agility and agility. Safety is paramount in all projects to save you any loss. An essential characteristic of IDS is to offer statistics approximately unusual activity and to notify community administrators to echo/monitor and/or block suspicious connectivity. Additionally, an IDS can distinguish among internally generated assaults (coming from one's personal personnel, clients, or others) and externally generated (assaults executed with the aid of hackers). Common styles of intrusion detection structures (IDS) are community (network IDS) and host-primarily based (HIDS). Network IDS alone is based on detecting unlawful, illicit, and conflicting behavior in network visitors.

RELATED WORK

Jaafar Abo Nada et.al, said that with the swift proliferation of remote networks, local area security encounters escalating risks, necessitating innovative solutions. Traditional methods for safeguarding networks prove inadequate in the wireless context, rendering intrusion detection devices designed for wired networks ineffective in the wireless landscape. Due to its user-friendly design, wireless technology gains popularity rapidly, transforming networking practices. However, the primary safety concern lies in wireless security. Recognizing these challenges, focusing on protection management becomes crucial. The paper aims to propose a new Wi-Fi intrusion prevention device, "WIDPAS," addressing denial-of-service attacks and rogue networks. It emphasizes monitoring, analysis, and security, capturing and identifying attackers while safeguarding network users[1].

Keenam Park et.al , delves into intrusion detection systems' classification of attack types. The research employs machine learning algorithms, specifically the random forest algorithm, to analyze diverse datasets derived from

Kyoto 2006+. The datasets represent the latest network features used in developing intrusion detection systems. The paper concludes with discussions on detection performance and proposes directions for future research [2].

St. Bernard et.al, presents a series of studies on random forest (RF), particularly focusing on the conventional RF induction process. However, limitations arise with this approach, including a predetermined number of random choice trees and the consequent loss of interpretative and evaluative capabilities during classification. The technique of combining trees without concatenation raises concerns about their effective collaboration within a single commission. The paper explores a central inquiry: are decision trees in the Russian Federation susceptible to corruptions of encomium points? It introduces a subclassification selection method to address challenges, demonstrating the possibility of obtaining superior sets of decision trees. The paper advocates for a tree-based RF induction, aligning with traditional methods employed in "classic" RF induction algorithms[3].

A. Despahun et.al said that the realm of computer and local area security, Intrusion Detection Systems (IDS) have become integral. This paper utilizes the NSL-KDD intrusion detection dataset, a modified version of the KDDCUP'99 dataset, as a benchmark for evaluation. The NSL-KDD dataset, however, presents a significant imbalance during training, posing challenges for intrusion detection systems to acquire accurate insights. To address this imbalance, the paper employs a synthetic minority oversampling technique (SMOTE) in the training data. A data-driven feature selection method is introduced to apply reduced feature sets to the NSL-KDD dataset. The proposed intrusion detection system leverages Random Forest as a classifier. Experimental results demonstrate that combining Random Forest classification with SMOTE and data-driven feature selection achieves higher overall performance, effectively enhancing the efficiency and effectiveness of IDS in intrusion detection[4].

Le, T.-T.-H. et.al, said that an Intrusion Detection System (IDS) is a tool or software package designed to monitor a network or machine for malicious activities. Traditional IDS fall short in detecting complex cyber threats such as low-frequency DoS attacks and unknown intrusions. In recent years, systems thinking has attracted individuals who are trying to overcome these challenges. This paper presents an innovative approach to improve GRU (Gated Recurrent Unit) violation detection by incorporating variants of Principal Component Analysis (PCA), including PCA and PCA-Minima, into the GRU layer. Both methods use object-oriented maps and amplify their influence through positive covariance. This approach can be applied to GRU models with minimal additional computational overhead. Experimental results on two real-world benchmarks, the KDD 99 and NSL-KDD datasets, show a significant performance improvement achieved by the GRU model using the PCA scaling approach[5].

EXISTING SYSTEM

In existing technology, diverse the machine gaining knowledge of calculations for interruption location machine. They as thought about a few techniques, for example, SVM, Outrageous Learning Machine and Irregular Woodland. The creators of the results expressed that the profound gadget getting to realize strategy finished a truckload better than different calculations. B. Riaz et al.- Here they attempted to work on the charming of guides to build up interruption location framework. They utilized a standard based totally choice function to optimize the facts set. They used the KDD dataset and as a result, discovered a dynamic boom in IDS consequences. Internet-primarily based structures serve numerous malicious sports. The fundamental hassle determined on this regard is intrusion into the gadget to smash statistics. The present day consequences indicate that a few upgrades can be made in phrases of accuracy, detection rate and false alarms. Some other strategies like SVM and China Naive had been used before. Additionally, the observe shows that some techniques within the dataset can be progressed. Increases the quality of inputs into the proposed gadget.

PROPOSED SYSTEM

Intrusion detection systems paintings to protect structures from being focused by using attackers. This machine can locate incoming calls. The proposed device tries to conquer previous issues in the present paintings. The proposed machine includes techniques: foremost factor analysis, and the opposite is random wooded area. Principal thing analysis is used to reduce the dimensionality of a information set; With this method, the first-class of the dataset will improve because the dataset can comprise valid functions. Next, a random leap algorithm is used to come across intruders, which provides both better detection charges and fake alarms than SVM. This fig.1 explains the steps for monitoring Network Intrusion Detection System (NIDS) deployment in this situation. NIDS is a type of security software used to monitor computer networks for malicious activity. The system starts with a list of data written by individual elements. For NIDS, this information may include network traffic patterns marked as normal or disruptive. The system then uses this registration information to train the classification manager. A classifier is a machine learning algorithm that learns to recognize patterns in data associated with each category. Once the classifier is trained, it can be used to classify new, unlabeled items.

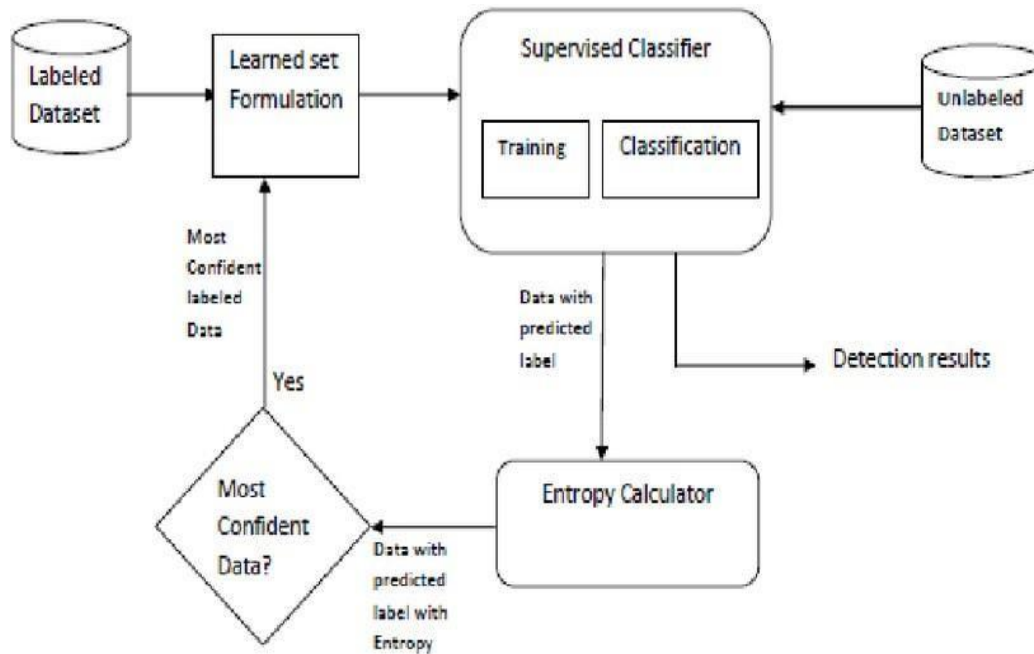


Fig.1. Proposed System Framework

This is the first real step towards truly constructing a system studying version for statistics series. This is an critical step that depends on a good version: the better realities we get, the higher our model will complete. There are numerous techniques of records series like textual content scraping, manual intervention and so forth. The dataset used on this intrusion detection machine dataset is taken from kdd hyperlink:<http://kdd.Ics.Uci.Edu/databases/kddcup99/kddcup99.Html>.

The dataset contains 125974 character factors.

The statistics set has 42 columns, defined below. Let's alternate the records. By casting off missing information and doing away with a few sections. First permit's make a posting of section names that we need to store or store. Then, at that point, we erase or dispose of the multitude of segments other than the ones we need to hold. At last, we dispose of or discard lines with missing qualities from the dataset. A distinction is made among education and evaluation. Head part investigation is a way frequently used to decrease the dimensionality of a data set. Head part investigation is one of the most extreme green and exact systems to diminish the dimensionality of records and favored results. This approach reduces the traits of the statistics to a favored variety of attributes, referred to as principals. This technique accepts every one of the information records as a dataset with an immense amount of capabilities and hence a very massive dataset size. This approach reduces the scale of the data by means of placing the records points at the same axis. The information factors are converted to an axis and come to be principal components. PCA can be achieved the use of the accompanying advances:

6. Take the given estimations along with the aspects in general.
7. Add the propose vector for each estimation d.
8. Add the covariance grid for the whole records set.
9. Compute eigenvectors (e1, e2, e three... .Ed) and eigenvalues (v1, , v2, v3, ...Vd).
10. Organize the eigenvalues in diving request and the n eigenvectors arranged by greatest eigenvalues to get ad*n=M lattice.
11. Utilizing this M, make another variant space.
12. The following lengths are top.

Irregular Backwoods is one of the most fundamental techniques in device getting to know to solve class troubles. Random forests fall underneath the class of supervised algorithms. This algorithm is achieved in awesome steps: the primary is to generate a forest of a given dataset, and the second one is to generate class predictions.

Analysis and Forecasting

In the real records set, we decided on only 10 features shows in table 1

- a. Duration (number of seconds) of
the connection2.Protocol_type
protocol type, eg tcp, udp etc.
- 3.Src_bytes number of data bytes from source to destination
- 4.Dst_bytes number of data bytes from destination to source
- 5.Is_hot_login 1 if the login belongs to the "hot" list; About otherwise
- 6.Is_guest_login 1 if login as "guest"; About otherwise
- 7.Diff_srv_rate % connections to different services
- 8.Srv_diff_host_rate % connections to different hosts
9. Indicate the normal or error status of the connection
10. Designation Normal or attackers

Indicates how long the connection took, measured in seconds.

Protocol_type: Indicates the type of communication protocol used, such as TCP or UDP.

Src_bytes: This is the number of data bytes sent from the source (sender) to the destination (receiver) during the connection. Dst_bytes: Similarly, it represents the number of data bytes sent from the destination back to the source during the connection. Is_hot_login: This function indicates whether the login belongs to the "hot" list. Is_guest_login: Indicates whether the login is a guest login. Guest logins may have limited access compared to regular user logins. Diff_srv_rate: Represents the percentage of connections to different services. Srv_diff_host_rate: Similar to Diff_srv_rate, this function represents the percentage of connections to different hostsFlag: Indicates the status of the connection, whether it is normal or an error has occurred. Labels: This is the connection classification that is categorized as "Normal" or "Attack". This is the target variable we want to predict. If a connection is classified as "Attack", it means that it is identified as potentially malicious, while "Normal" means that it is a harmless or expected behavior.

Essentially, these features provide insight into various aspects of network activity, and by analyzing them, we can create models classifying connections as normal or potentially malicious based on historical patterns and known attack signatures.

Be Careful With the Check Set

We done ninety nine.1% accuracy at the take a look at set.

Save Learned Version

Step one is to transform it to .H5 or .H5. PKL library using Alex. Make positive ALEX is mounted on your environment. Next, import the module and unload the model into a .Pkl file.

Data collection

```
download_dataset("http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html")  
preprocess_data()
```

Data maintenance and preparation

```
select_columns(["relevant_feature_1", "relevant_feature_2", ...])
```

`remove_missing_values()`

Principal Component Analysis (PCA)

`apply_pca()`

Random Forest Model

`train_random_forest()`

Analysis and Estimation

`select_analysis_features(["selected_feature_1", "selected_feature_2", ...])`

`evaluate_accuracy()`

Save Learning Model

`save_model_to_file("learned_model.pkl")`

PROPOSED ALGORITHM

Principal component analysis (PCA) is a statistical technique that uses orthogonal improvements to convert strongly and rapidly correlated variables into a set of uncorrelated variables. PCA is a widely used device in exploratory evaluation and gadget studying for predictive models. PCA could be very beneficial in identification process wherein there is heterogeneity amongst capabilities/variables. In the context of AI-powered mobile app cybersecurity, PCA can be used to identify and understand patterns in the behavior of mobile app data.

Simply, it works like this:

Data Collection: First, you collect data about mobile app behavior. This may include information such as features accessed by the application, how it communicates with servers, user interactions, etc.

Feature Extraction: PCA then takes this data and looks for patterns or correlations between different features (or characteristics) of applications. For example, it may find that certain types of applications tend to access specific parts of the device or have similar communication patterns. **Dimensionality reduction:** PCA helps reduce the complexity of this data by transforming it into a small set of features called principal components. These components are new variables that are combinations of the original features, but chosen to capture as much variability as possible in the data.

Anomaly detection: Once the data is transformed, anomalies or unusual behavior can be more easily identified. For example, if an application suddenly starts accessing sensitive data that it does not normally have access to, this deviation from the norm can be flagged as a potential security threat.

Security Improvements: With PCA, security systems can be better equipped to detect unusual activity or potential threats in mobile applications. It helps understand typical application behavior and detect deviations that could indicate malicious activity or vulnerabilities. In short, PCA is a technique that helps to simplify and understand complex data patterns, making it useful for improving the security of mobile applications by detecting anomalies or unusual behavior using artificial intelligence-based systems.

Random Forest Algorithm: Random Forest is a popular algorithm in machine learning and can be very useful for cybersecurity in mobile applications. It combines the outcomes of a couple of decision timber right into a unmarried result. Ease of use and flexibility contributed to their adoption, as they solved both segmentation and regression issues.

In the context of mobile application security, Random Forest can be trained on a dataset containing features extracted from mobile applications. These features can include things like permissions required by the application, frequency of network requests, code complexity, etc.

Group decision-making process: A Random Forest is like having a committee with decision-making powers, where each member (tree) has its own view of the data. Instead of relying on just one opinion, Random Forest

aggregates the decisions of all trees to reach a final verdict. This ensemble approach often leads to more accurate and reliable results compared to individual models.

Feature extraction in mobile application security: Features extracted from mobile applications serve as input to the Random Forest algorithm. These features provide crucial information about the behavior and properties of applications, such as the permissions they require, their communication patterns with external servers, and the complexity of their code. By analyzing these features, Random Forest can identify patterns indicative of malicious behavior.

Training phase: During training, the random forest learns from a labelled dataset containing examples of both malicious and benign mobile applications. **Making predictions:** Once trained, a random forest can predict whether a new mobile app is malicious or safe based on its features. Each tree in the forest reviews the application separately and the final decision is made by majority vote.

Random Forest is efficient because it can handle large amounts of data with many features, is robust against overfitting (memorizing training data), and robust against data noise.

Continuous learning: Additionally, Random Forest can adapt to new threats over time. As new data becomes available, the algorithm can be retrained to stay current with the latest security challenges.

In short, the Random Forest algorithm for mobile app security uses a group of decision trees to jointly determine whether an app is malicious or safe based on its features, making it a powerful tool in the cyber defense arsenal of AI-driven mobile apps.

PERFORMANCE METRICS:

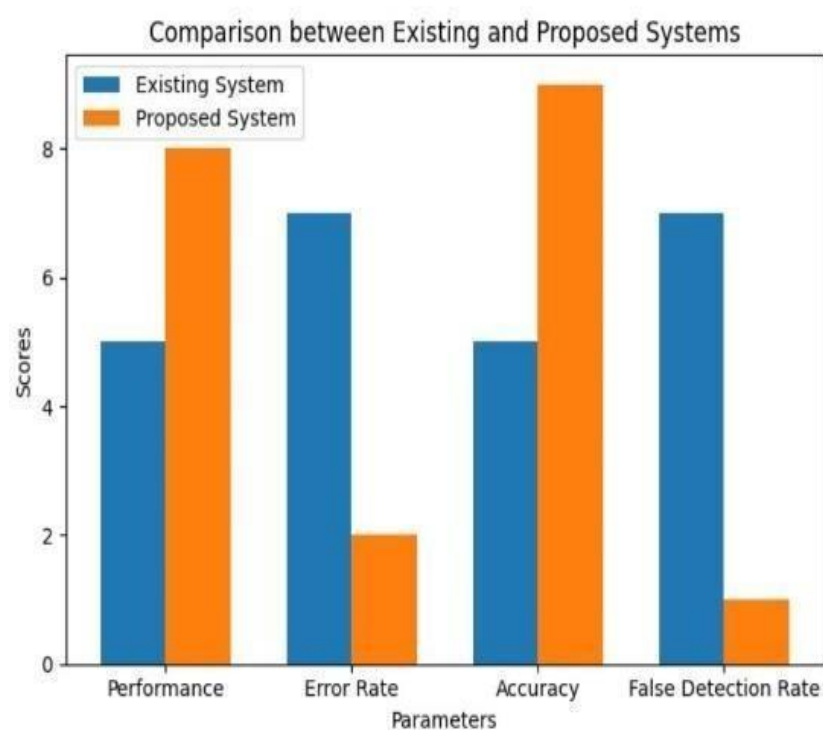


Fig:2 Performance Metrics

The chart compares the performance of existing and proposed systems in three metrics: error rate, accuracy, and false detection rate. The proposed system demonstrates superiority in all aspects, boasting lower error rate, higher accuracy and reduced false detection rate. This translates into a more reliable system with fewer errors and a significant reduction in the need for manual intervention due to false alarms. Furthermore, we can observe trends in the data, such as a potential positive correlation between accuracy and a particular variable for the proposed system. While the proposed system outperforms the existing one, future research could explore its behavior with even larger datasets and potential optimizations for even lower error rates.

RESULT AND DISCUSSION

The error price observed in our proposed technique is very low at 0.21%. Additionally, the resulting accuracy is better than previous algorithms and the execution time is much less than other algorithms. Experimental outcomes show that the very best level of accuracy is accomplished inside the environment (ninety six. 78), while the average (90. Fifty nine %) is best as compared to different approaches. The findings of this look at encourage researchers inside the field to develop machine getting to know with more accuracy in cyber security and statistics analysis and intrusion detection structures.

SCREEN SHOTS

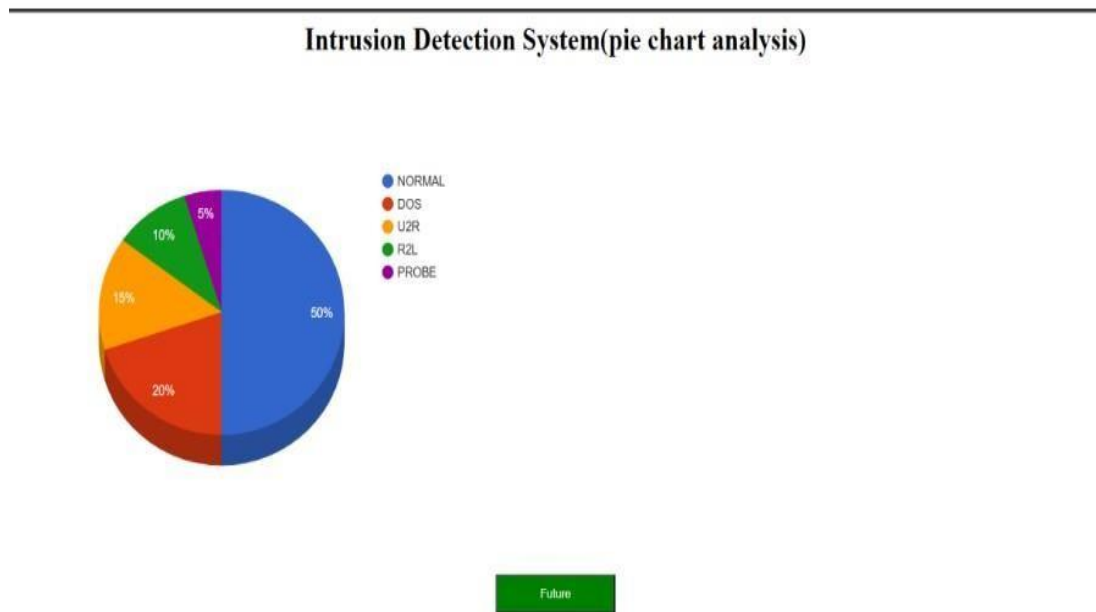


Fig:3 Intrusion Detection System (pie chart analysis)

This pie chart titled "Intrusion Detection System " provides a breakdown of various aspects related to intrusion attempts detected by your system. Depending on the specific configuration, it can visualize the distribution of attacks by type (denial-of-service, malware, etc.), location in your network (different segments or devices), severity levels of alerts generated (high, medium, low), or even the result of these attempts (successful penetrations, blocked or never completed). This visual representation allows security analysts to identify the most common threats, prioritize vulnerable areas, understand the severity of alerts, and assess the overall effectiveness of the intrusion detection system in defending your network.

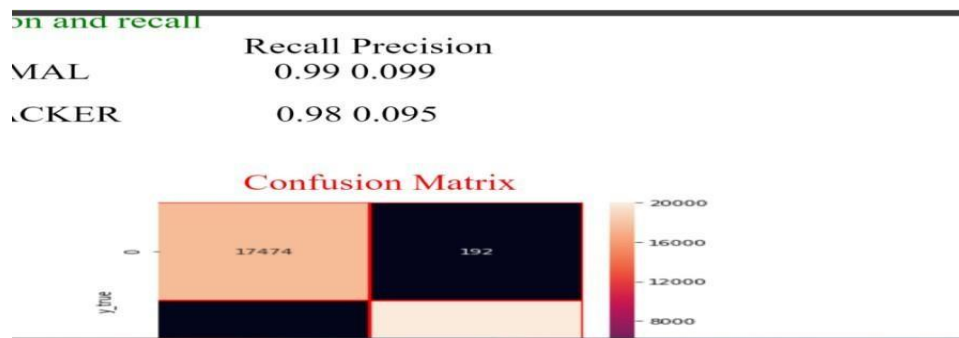


Fig:4a Confusion Matrix

The graph is a confusion matrix that compares the performance of a machine learning model in classifying between two classes. In this particular case, it compares the normal call to the attacker's call. The rows represent the actual classifications, while the columns represent the predicted classifications. Perfect performance would be achieved if all values were on the diagonal, meaning that all normal calls were correctly classified as normal and all attacker calls were correctly classified as attacker calls. In this particular matrix, we can see that the model performs well in identifying normal calls, with 99% correctly classified. However, the model struggles more with calling attackers, only identifying 95% correctly. This means that 5% of the attacker's calls were misclassified as normal calls.

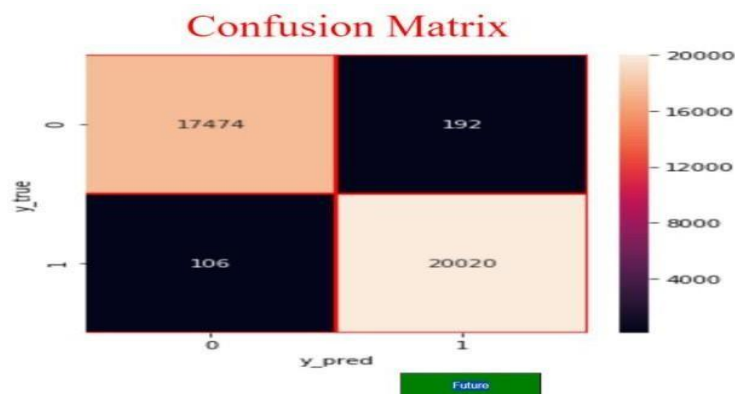


Fig:4b Confusion Matrix

Machine learning models are often used to secure mobile applications. These models are trained on datasets of known malicious and benign applications. The accuracy and recall of these models can be evaluated using graphs like the one you provided.

Security analysts use a combination of techniques to secure mobile applications. This can include static code analysis, dynamic analysis and penetration testing. Accuracy and recall are important metrics to consider when evaluating the effectiveness of these techniques.

An acceptable balance between precision and recall may vary depending on the particular context. For example, if the consequences of a false positive are high (e.g., the use of a legitimate application is blocked), a security system with higher accuracy may be preferred.

FUTURE

As the involvement of the systems over the internet increasing rapidly, the security concerns have also seen. The proposed approach deals with the detection of intruders over the internet efficiently. The proposed algorithm has performed well as compared to the previously applied algorithms such as SVM, Naive Bayes, and Decision Tree. The detection rates and the false error rates can be improved at a great extent by the proposed approach. The dataset used here is the knowledge discovery dataset. The results obtained by our proposed method having the values for Performance time (min) is 3.24 minutes, Accuracy rate (%) is 96.78 %, and the Error rate (%) is 0.21 %.

Fig:5 Future

This excerpt explores the future and evolving security landscape of Internet-connected systems. It proposes a new approach for intrusion detection that outperforms established methods such as SVM, Naive Base and Decision Trees. The text emphasizes the algorithm's excellent detection and error rates when evaluated against specific data sets. Performance metrics such as processing time, accuracy, and error rate are also mentioned, indicating a potentially valuable contribution to the field of network security.

CONCLUSION

As the sharing of frameworks on the web grows exponentially, so will the security problems. The proposed approach allows efficient detection of Internet intruders. The proposed algorithm performs well with previously used algorithms including SVM, Innocent Bayes and Choice Trees. The proposed approach can significantly increase the identification cost and cost of false positives. The dataset used here is the Information Disclosure dataset. Among the results obtained with the help of the proposed approach, the jogging time is 3.24 minutes (min), the accuracy value (%) is sixth. Seventy% and Errors Charge (%) Zero.21%.

REFERENCES

- [1] JafarAbo Nada; Mohammad Rasmi Al-Mosa, 2018 International Arab Conference on Information Technology (ACIT), A Proposed Wireless Intrusion Detection Prevention and Attack System
- [2] Kinam Park; Youngrok Song; Yun-Gyung Cheong, 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigData Service), Classification of Attack Types for Intrusion Detection Systems Using a Machine Learning Algorithm
- [3] S. Bernard, L. Heutte and S. Adam “On the Selection of Decision Trees in Random Forests” Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, June 14-19, 2009, 978-1-4244-3553-1/09/\$25.00 ©2009 IEEE
- [4] A. Tesfahun, D. Lalitha Bhaskari, “Intrusion Detection using Random Forests Classifier with SMOTE and Feature Reduction” 2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies, 978-0-4799-2235-2/13 \$26.00 © 2013 IEEE
- [5] Le, T.-T.-H., Kang, H., & Kim, H. (2019). The Impact of PCA-Scale Improving GRU Performance for Intrusion Detection. 2019 International Conference on Platform Technology and Service (PlatCon). Doi:10.1109/platcon.2019.8668960
- [6] Anish Halimaa A, Dr K.Sundarakantham: Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019) 978-1-5386-9439-8/19/\$31.00 ©2019 IEEE “MACHINE LEARNING BASED INTRUSION DETECTION SYSTEM.”
- [7] Mengmeng Ge, Xiping Fu, Naeem Syed, Zubair Baig, Gideon Teo, Antonio Robles-Kelly (2019). Deep Learning-Based Intrusion Detect ion for IoT Networks, 2019 IEEE 24 th Pacific Rim International Symposium on Dependable Computing (PRDC), pp. 256-265, Japan.
- [8] R. Patgiri, U. Varshney, T. Akutota, and R. Kunde, “An Investigation on Intrusion Detection System Using Machine Learning” 978-1-5386-9276 9/18/\$31.00 c2018IEEE.
- [9] Rohit Kumar Singh Gautam, Er. Amit Doegar; 2018 8 th International Conference on Cloud Computing, Data Science & Engineering (Confluence) “An Ensemble Approach for Intrusion Detect ion System Using Machine Learning Algorithms.”
- [10] Kazi Abu Taher, Billal Mohammed Yasin Jisan, Md. Mahbubur Rahma, 2019 International Conference on Robot ics, Electrical and Signal Processing Techniques (ICREST)“Network Intrusion Detect ion using Supervised Machine Learning Technique with Feature Selection.”
- [11] L. Haripriya, M.A. Jabbar, 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)“Role of Machine Learning in Intrusion Detection System: Review”
- [12] Nimmy Krishnan, A. Salim, 2018 International CET Conference on Control, Communication, and Computing (IC4) “Machine Learning-Based Intrusion Detection for Virtualized Infrastructures”
- [13] Mohammed Ishaque, Ladislav Hudec, 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS) “Feature extraction using Deep Learning for Intrusion Detection System.”
- [14] Aditya Phadke, Mohit Kulkarni, Pranav Bhawalkar, Rashmi Bhattad, 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)“A Review of Machine Learning Methodologies for Network Intrusion Detection.”
- [15] Iftikhar Ahmad , Mohammad Basher, Muhammad Javed Iqbal, Aneel Rahim, IEEE Access (Volume: 6) Page(s): 33789 – 33795 “Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection.”
- [16] B. Riyaz, S. Ganapathy, 2018 International Conference on Recent Trends in Advanced Computing (ICRTAC)“An Intelligent Fuzzy Rule-based Feature Selection for Effective Intrusion Detection.”
- [17] Bishtawi, T., & Alzu'bi, R. (2022, November). Cyber Security of Mobile Applications Using Artificial Intelligence. In 2022 International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEI) (pp. 1-6). IEEE.

- [18] Kaur, R., Gabrijelčič, D., & Klobučar, T. (2023). Artificial intelligence for cybersecurity: Literature review and future research directions. *Information Fusion*, 101804.
- [19] Capuano, N., Fenza, G., Loia, V., & Stanzione, C. (2022). Explainable artificial intelligence in cybersecurity: A survey. *IEEE Access*, 10, 93575-93600.
- [20] Loaiza, F. L., Birdwell, J. D., Kennedy, G. L., & Visser, D. (2022). Utility of artificial intelligence and machine learning in cybersecurity. *Institute for Defense Analyses*.
- [21] M. Nagar, M. B. Reddy, U. Nandini, A. Mayan, Sathyabama Krishna and S. P. Mary, "Smart District Analysis and Complaint Website," 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), 2021, pp. 1019-1025
- [22] Aishwarya, R., & Mathivanan, G. (2023, July). Mobile Cloud Offloading: An Analysis of Methodologies, Limitations, and Research Problems. In 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-8). IEEE.
- [23] Aishwarya, R., & Mathivanan, G. (2024, January). COCAME: A Computational Offloading in Cloud Assisted Mobile Environments Structure to Enhance Performance and Energy. In 2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 264-271). IEEE.

D. CERTIFICATE



