OPTIMIZED CLOUD SECURITY FRAMEWORK HARNESSING HYBRID FEATURE SELECTION AND MACHINE LEARNING CLASSIFICATION FOR INTRUSION DETECTION

A PROJECT REPORT

Submitted by

CHENNA REDDY PARITHRAAN (211420104046)

MALIREDDY SAI RAKESH (211420104151)

MANCHIGANTI DEERAJ SURYA (211420104154)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MARCH 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **" OPTIMIZED CLOUD SECURITY FRAMEWORK HARNESSING HYBRID FEATURE SELECTION AND MACHINE LEARNING CLASSIFICATION FOR INTRUSION DETECTION"** is the bonafide work of **"CHENNA REDDY PARITHRAAN(211420104046), MALIREDDY SAI RAKESH (211420104151), MANCHIGANTI DEERAJ SURYA(211420104154)"** who carried out the project work under my supervision.

Signature of the HOD with date

Dr L.JABASHEELA M.E., Ph.D.,
PROFESSOR AND HEAD,

Department of Computer Science and Engineering,
Panimalar Engineering College,
Chennai – 123

Signature of the Supervisor with date

MRS C.JACKULIN,M.E
ASSISTANT PROFESSOR

Department of Computer Science and Engineering,
Panimalar Engineering College,
Chennai – 123

Certified that the above candidate(s) were examined in the End Semester Project Viva-Voce Examination held on ……..

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **CHENNA REDDY PARITHRAAN (211420104046), MALIREDDY SAI RAKESH (211420104151), MANCHIGANTI DEERAJ SURYA (211420104154)** hereby declare that this project report titled **"OPTIMIZED CLOUD SECURITY FRAMEWORK HARNESSING HYBRID FEATURE SELECTION AND MACHINE LEARNING CLASSIFICATION FOR INTRUSION DETECTION"**, under the guidance of **MRS C. JACKULIN, M.E** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

CHENNA REDDY PARITHRAAN (211420104179)

MALIREDDY SAI RAKESH (211420104151)

MANCHIGANTI DEERAJ SURYA(211420104154)

ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D**., for his fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We wish to express our deep gratitude to our Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHIKUMAR, M.E., Ph.D., an Dr. SARANYASREE SAKTHIKUMAR, B.E., M.B.A., Ph.D.,** for graciously affording us the essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

We express our heartfelt thanks to **Dr. L. JABASHEELA, M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to express our sincere thanks to **MRS C.JACKULIN,M.E,** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

CHENNA REDDY PARITHRAAN(211420104179)

MALIREDDY SAI RAKESH(211420104151)

MANCHIGANTI DEERAJ SURYA(211420104154)

ABSTRACT

The focus of cloud computing nowadays has been reshaping the digital epoch, in which clients now face serious concerns about the security and privacy of their data hosted in the cloud, as well as increasingly sophisticated and frequent cyber attacks. Therefore, it has become imperative for both individuals and organizations to implement a Intrusion Detection System capable of monitoring packets in the network, distinguishing between benign and malicious behavior, and detecting the type of attacks. Furthermore, training ML models on unbalanced datasets show a rising FPR and a lowering detection rate presented an improved cloud Intrusion Detection System designed by incorporating the synthetic minority over-sampling technique to address the data issue, and for feature selection, we propose to use a hybrid approach that includes three techniques: Information Gain, Chi-Square and Particle Swarm Optimization. Finally, RF model is utilized for detecting and classifying various types of attacks. The simulation results significantly outperform other methodologies proposed in the related work in terms of different evaluation metrics.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF TABLE	ix
	LIST OF ABBREVATION	x
1.	**INTRODUCTION**	**1**
	1.1 Overview	1
	1.2 Scope of the project	1
	1.3 Objective of the project	2
	1.4 Problem Definition	2
2.	**LITERATURE SURVEY**	**3**
3.	**SYSTEM ANALYSIS**	**7**
	3.1 Existing System	7
	3.2 Proposed System	7
	3.3 Feasibility Study	8
	3.4 System Requirements	8
	3.5 Software Description	9
4.	**SYSTEM DESIGN**	**13**
	4.1 ER Diagram	13
	4.2 Work Flow Diagram	14
	4.3 Usecase Diagram	15
	4.4 Class Diagram	16
	4.5 Activity Diagram	17
	4.6 Sequence Diagram	18
	4.7 Collaboration Diagram	19
5	**SYSTEM ARCHITECTURE**	**20**
	5.1 Algorithm and Techniques	21
	5.2 Module Description	22
	5.2.1 Data Pre-Processing	22
	5.2.2 Data Visualization	23
	5.2.3 Extra-Tree Classifier	24

6	**PERFORMANCE ANALYSIS**	**26**
7	**CONCLUSION**	**28**
	7.1 Results and Discussion	28
	7.2 Conclusion and Future Enhancements	28
	APPENDICES	**29**
	A.1 SDG GOALS	29
	A.2 SOURCE CODE	30
	A.3 SCREENSHOT	39
	A.4 CONFERENCE PAPER	44
	A.5 PLAGIARISM REPORT	51
	REFERENCES	**53**

LIST OF FIGURES

FIG NO	TITLE	PAGE NO
4.1	ER Diagram	13
4.2	Work Flow Diagram	14
4.3	Usecase Diagram	15
4.4	Class Diagram	16
4.5	Activity Diagram	17
4.6	Sequence Diagram	18
4.7	Collaboration Diagram	19
5.1	System Architecture	20
5.2.2	Data Validation	24

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
6.1	Normal	26
6.2	R2L Remote to Local	26
6.3	DoS Denial of Service	27
6.4	Probe	27

LIST OF ABBREVATION

KEYWORD	ABBREVATION
ABS	Artificial Bee Colony
AFSO	Artificial Fish Swarm Optimization
CNN	Convolutional NeuralNetwork
CS	Chi-Square
FAR	False Rate Alarm
IDS	Intrusion Detection System
IG	Information Gain
IPS	Intrusion Protection System
PoS	Particle Swarm Optimization
PSO	Particle Swarm Optimization
QoS	Quality of Service
RF	Random Forest
RNN	Recurrent Neural Network
SET	Stacking Ensemble Technique

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

Cloud computing has become an integral part of modern IT infrastructure, offeringscalability and flexibility. However, the increasing reliance on cloud services also attracts malicious activities and cyber threats. In this context, an effective Intrusion Detection System (IDS) is crucial to safeguard cloud environments. This paper presentsan improved design for a Cloud Intrusion Detection System using a hybrid approach for feature selection and a machine learning classifier. The proposed system leverageslabel encoding, correlation analysis, and the Extra Tree algorithm to enhance the accuracyand efficiency of intrusion detection.

The aim of this study is to enhance the efficacy of Cloud Intrusion Detection Systems by proposing an optimized design that integrates a hybrid feature selection approach with a machine learning classifier. The goal is to improve the accuracy and efficiency of intrusion detection in cloud environments, addressing the challenges posed by diverse feature types and ensuring robust protection against cyber threats

Machine learning is to predict the future from past data. Machine learning (ML) is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development ofComputer Programs that can change when exposed to new data and the basics of Machine Learning, implementation of a simple machine learning algorithm using python. Process of training and prediction involves use of specialized algorithm.

1.2 SCOPE OF THE PROJECT

The scope for this project is to develop an intrusion detection system that will improve the security of home network as that is the potential user of this system. The objective of this project is to investigate the methods needed to detect any unauthorizedaccess into a home networking system. IDS is a detective device designed to detect malicious including policy-violating actions. An IPS is primarily a preventive device designed not only to detect but also to block malicious actions As Cloud is an integral part of modern IT infrastructure, offering scalability and

flexibility. However, the increasing reliance on cloud services also attracts malicious activities and cyber threats. In this context, an effective IDS is crucial tosafeguard cloud environments. The paper presents an improved design for a Cloud Intrusion Detection System using a hybrid approach for feature selection and a machine learningclassifier. The proposed system leverages label encoding, correlation analysis, and the Extra Tree algorithm to enhance the accuracy and efficiency of intrusion detection.

1.3 OBJECTIVE OF THE PROJECT

This analysis aims to observe which features are most helpful in predicting thepresence of the attack and also the type of attack. The rapid advancement of technology gave us information in an instance. Network connection is vital in personal usage as with this connection we may gain an extra edge in knowledge information. With this advancement come a few problems such as spam, virus etc.Therefore, a solution is needed to prevent those attacks before it happens.

1.3.1 To develop an intrusion detection system for Windows-based Operating system

1.3.2 To prevent abuse or overload from bandwidth and denial of Service attacks

1.3.3 To monitor the traffic flow for any malicious activities of a network in real time

1.4 PROBLEM DEFINTION

Developing an intrusion detection system, identify Unauthorized access into a home networking system. Without a good detection system, a computer network will be access by an unauthorized individual. This individual may do harm to others by stealing other people's data not to mention confidentialinformation would be compromise. A DoS attacks may also occur. Because protecting the network from any potential attack is of utmost importance. The cloud employs a variety of strategies, such as firewalls, IPS, IDSs, etc., to addressnumerous security issues.

CHAPTER 2

LITERATURE SURVEY

INTRODUCTION

A literature review is a body of text that aims to review the critical points ofcurrent knowledge on and/or methodological approaches to a particular topic. It issecondary sources and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time. Its goal is to bring the reader up to date with current literature on a topic and forms thebasis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis

An Improved Design for a Cloud Intrusion detection System Using HybridFeatures Selection Approach With ML Classifier

Author: Mhamad Bakro, IEEE, Rakesh Ranjan Kumar, IEEE, Amerah IEEE.

Year: 2023

With the widespread use of cloud computing by people and businesses, security in the cloud is of the utmost concern. The suggested intrusion detection system aims to distinguishes machine learning models from deep learning models develop a modelthat would leverage increased intrusion detection's accuracy by combining the strengths of each employed feature selection algorithm (information gain, chi- square, and particle swarm optimization). The proposal displayed its power to identify multiple attack.

OPTCLOUD: An Optimal Cloud Service Selection Framework Using QoSCorrelation Lens

Author: R.R.Kumar,A.Tomar,M.Shameem,M.NAlam

Year: 2022

Finding the best cloud service for cloud users is a challenge if there are many QoS criteria. In general, most of the QoS criteria are correlated and are ignored by the existing works. The proposed work differs in many ways from the existing researchworks. First, we reduced the number of QoS criteria to simplify the process of selecting cloud services. Secondly, it removes the correlation between different QoScriteria and produces more authentic selection results.

Efficient Intrusion Detection System in the Cloud Using Fusion FeatureSelection approaches and an ensemble classifier.

Author: M.Bakro, R.R Kumar, A.A Alabrah, Z.Ashraf, S.K Bisoy

Year: 2023

The proposed work differs in many ways from the existing research works. First, wereduced the number of QoS criteria to simplify the process of selecting cloud services. Secondly, it removes the correlation between different criteria, authentic selection results. This contribution provides a new framework for the cloud serviceselection process. The proposed scheme demonstrates its feasibility and efficiency through a series of experiments with real datasets. Finally, we make a comparison with the other method to show that the proposed methodology outperforms them.

Machine Learning Methods for Cyber Security Intrusion Detection: Datasetsand Comparative Study

Author: I.F.Kilincer, F.Ertam, A.Sengur

Year: 2023

Improving an intrusion detection system is something which is challenging. The detection rate of an NIDS is affected by the number of features. The key task of datamining and ML techniques aim at improving the detection accuracy and reducing the positive false rate for an NIDS. The latter model is based on the UNSW-NB15dataset. The proposed feature selection model is based on the PSO, GWO, FFA andGA bio-inspired algorithms and MI. In the case of bio-inspired algorithms, PSO FFAreduces the number of the selected features to 21 features; and GA reduces the number of the selected features to 23 features.

Improving the Classification Effectiveness of Intrusion Detection SystemUsing Multiverse Optimization

Author: Y.Yang, K.Zheng ,C.Wu and Y.Yang

Year: 2022

This paper, We proposed a hybrid intrusion detection alert system using a highly scalable framework on common highly scalable framework on commodity hardwareserver which has the capability to analyze the network and host –level activities. Theframework employed distributed deep learning model with DNNs for handling andanalyzing very large scale data in real-time. Our proposed architecture is able to perform better than previously implemented classical machine learning classifier inboth HIDS and NIDS.

A Tree-Based Stacking Ensemble Technique with Feature Selection forNetwork Intrusion Detection .

Author: M.Rashid, J.Kamruzzaman, T.Imam, S.Wibowo

Year: 2022

In this research, we introduce a tree-based stacking ensemble technique (SET) andtest the effectiveness of the proposed model on two intrusion datasets (NSL-KDDand UNSW-NB15). We further enhance incorporate feature selection techniques toselect the best relevant features with the proposed SET. A comprehensive performance analysis shows that our proposed model can better identify the normaland anomaly traffic in network than other existing IDS models. This implies the potentials of our proposed system for cybersecurity in Internet of Things (IoT) and large scale networks.

An Optimization Method for intrusion detection classification model BasedonDeep Belief Network

Author: P.Wei, Y.Li, Z.Zhang, T.Hu, D.Liu

Year: 2022

DBN is a deep learning model widely used in speech recognition, image decognition and other fields. The proposed methods is to optimize the DBN networkstructure in the range of limitied hidden layers. If the maximum number of hidden layers in the DBN is to set too large the training time will have a greater impact onthe fitness,DBN is set too large, the training time will have a greater impact on thefitness, which is not conductive to find a well _funtioning DBN-IDNmodel. Basedon the DBN'S network structure optimized by our algorithm, the weights or threshold parameters on the DBN network structure are optimized to further optimize the DBN- IDS model, Its good classification performance makes.

Swarm Intelligence Inspired Intrusion Detection System – Systematic Literature Review

Author: M.H Nasir, S.A.Khan, M.M Khan and M.Fatima

Year: 2022

We aim to accomplish a cavernous understanding of the swarm intelligenceapproaches with respect to their application to intrusion detection to make the detection process more efficient and robust. Swarm intelligence approaches such as Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), and Artificial Fish Swarm Optimization (AFSO) have been widely used for solving manyoptimization problems. In this study, we focus on SI approaches that are applied to

improve or optimize traditional intrusion detection process. including intrusion classification, feature selection and weight/parameter optimization of ML-based classifiers.

Improved Security in cloud using sandpiper and extended equilibrium deeptransfer learning based intrusion detection

Author: G.Sreelatha, A.V.Babu, D.Midhunchakkaravarthy

Year: 2022

The proposed cloud IDS effectively classify whether the network traffic behavior is normal or attack. The proposed system is executed in python using the UNSW-NB15 dataset, and NSL-KDD dataset. The various evaluation metrics are used to show the efficiency of the proposed method and compared to the existing works. The simulation results show that the proposed method can able to detect intrusionswith a high detection rate and a low false alarm rate (FAR) than other approaches.

Hybrid Intrusion Detection Using mapreduce based black widow optimizedconvolutional Long Short-Term Memory Neural Networks.

Author: P.R Kanna and P.Santhi

Year: 2022

Deep Learning (DL) is a class of ML algorithms that overcomes the slow training problem. The development of DL algorithms has increased the computational capabilities in various applications in multiple research domains. Recent IDS modelshave been built upon the effective DL algorithms namely Convolutional CNN, RNN,Long Short – Term Memory neural networks, The DL algorithms process large, multifarious and high dimensional network.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

We present an improved cloud IDS designed by incorporating the synthetic minority over-sampling technique address the imbalanced data issue, and forfeature selection, we propose to use a hybrid approach that includes three techniques: IG, CS, and PSO. Finally, RF model is utilized for detecting and classifying various types of attacks.

Disadvantages

- They did not use any machine learning algorithms.
- Accuracy was low.
- They did not build a deploy model.

3.2 PROPOSEED SYSTEM

We proposed a system where the feature selection process is enhanced by employing label encoding to transform categorical data into numerical format, making suitable for machine learning algorithms. This ensures that the classifers can effectively process diverse types of feature selection present in cloud security datsets, A correlation analysis is performed to identify and eliminate redundant features. This step aids in reducing dimensionally ,enhance the efficiency of the IDS and improving the interpretability of results. Correlation analysis helps identify relationships between features that contribute significantly to the detection of intrusions.

Advantages

- We build a framework-based user-friendly application.
- We use multiple machine learning algorithms for train data.
- Accuracy was improved.

3.3 FEASIBILTY STUDY

3.3.1 Data Wrangling

In this section of the report will load in the data, check for cleanliness, andthen trim and clean given dataset for analysis. Make sure that the document stepscarefully and justify for cleaning decisions.

3.3.2 Data Collection

The data set collected for predicting given data is split into Training set and Testset. Generally, 7:3 ratios are applied to split the Training set and Test set.

3.3.3 Preprocessing

The data which was collected might contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed so as to improve theefficiency of the algorithm. The outliers have to be removed and also variable conversion need to be done.

3.4 SYSTEM REQUIREMENTS

Requirements are the basic constrains that are required to develop a system. Requirements are collected while designing the system. The following arethe requirements that are to be discussed.

1. Functional requirements
2. Non-Functional requirements
3. Environment requirements
    - Hardware requirements
    - software requirements

3.4.1 Functional Requirements

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system. The following details to follow the special libraries like sk-learn,

pandas, numpy, matplotlib and seaborn.

3.4.2 Non-Functional Requirements

Process of functional steps,

1. Problem Statement
2. Preparing data
3. Evaluating algorithms
4. Improving results
5. Prediction the result

3.4.3 Environmental Requirements

1. Software Requirements:

Operating System	:	Windows 10 (64 bit)
Software	:	Python 3.7
Tools	:	Anaconda (Jupyter Note Book IDE)

2. Hardware requirements:

Processor	:	Pentium IV/III
Hard disk	:	minimum 100 GB
RAM	:	minimum 4 GB

3.5 SOFTWARE DESCRIPTION

3.5.1 Machine Learning Introduction:

Machine learning is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence.Machine learning algorithms build a mathematical model based on sample data, knownas "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide varietyof applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the

task.Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data miningis a field of study within machine learning, and focuses on exploratory data analysis through learning. In its application across business problems, machine learning is also referred to as predictive analytics.

Machine learning tasks

Machine learning tasks are classified into several broad categories. In supervised learning, the algorithm builds a mathematical model from a set of data that contains both the inputs and the desired outputs. For example, if the task were determining whether an image contained a certain object, the training data for a supervised learningalgorithm would include images with and without that object (the input), and each image would have a label (the output) designating whether it contained the object. In special cases, the input may be only partially available, or restricted to special feedback.Semi algorithms develop mathematical models from incomplete training data, where aportion of the sample input doesn't have labels.

Classification algorithms and regression algorithms are types of supervised learning. Classification algorithms are used when the outputs are restricted to a limited set of values. For analgorithm that identifies spam emails, the output would be the prediction of either "spam" or "not spam", represented by the Boolean values true and false. Regression algorithms are named for their continuous outputs, meaning they may have any value within a range. Examples of a continuous value are the temperature, length, or price ofan object.

In unsupervised learning, the algorithm builds a mathematical model from a set of data thatcontains only inputs and no desired output labels. Unsupervised learning algorithms are used to find structure in the data, like grouping or clustering of data points. Unsupervised learning can discover patterns in the data, and can group the inputs into categories, as in feature learning. Dimensionality reduction is the process of reducing the number of "features", or inputs, in a set of data. Other specialized algorithms in machine learning include topic modeling, where the computer program is given a set ofnatural language documents and finds other documents.

Active learning algorithms access the desired outputs (training labels) for a limited setof inputs based on a budget and optimize the choice of inputs for which it will acquire training labels. When used interactively, these can be presented to a human user for labeling. Reinforcement learning algorithms are given feedback in the form of positive or negative reinforcement in a dynamic environment and are used in autonomous vehicles or in learning to play a game against a human opponent. Other specialized algorithms in machine learning include topic modeling, where the computer program is given a set of natural language documents and finds other documents that cover similar topics. Machine learning algorithms can be used to find the unobservable probability density function in density estimation problems. Meta learning algorithms learn their own inductive bias based on previous experience. In developmental robotics, robot learning algorithms generate their own sequences of learning experiences, also knownas a curriculum, to cumulatively acquire new skills through self-guided exploration and social interaction with humans. These robots use guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

Types of learning algorithms

The types of machine learning algorithms differ in their approach, the type of data theyinput and output, and the type of task or problem that they are intended to solve.

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, andconsists of a set of training examples An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.

Supervised learning algorithms include classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value withina range. Similarity learning is an area of supervised machine learning closely related function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification.

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of

unsupervised learningis in the field of density estimation in statistics, though unsupervised learning encompasses otherdomains involving summarizing and explaining data features.

Cluster analysis is the assignment of a set of observations into subsets (called clusters) sothat observations within the same cluster are similar according to one or more pre designated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data,often defined by some similarity metric and evaluated, for example, by internalcompactness, or the similarity between members of the same cluster, and separation, the difference between clusters. Other methods are based on estimated density and graph connectivity.

Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Many machine-learning researchers have found that unlabeled data, when used in conjunctionwith a small amount of labeled data, can produce a considerable improvement in learning accuracy.

CHAPTER 4

SYSTEM DESIGN

4.1 ER DIAGRAM

An entity relationship diagram (ERD), also known as an entity relationshipmodel, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts, or events within that system. An ERD is a data modeling technique that can help define business processes and be used as the foundation for a relational database. Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a referral point, should any debugging or business process re-engineering be needed later.

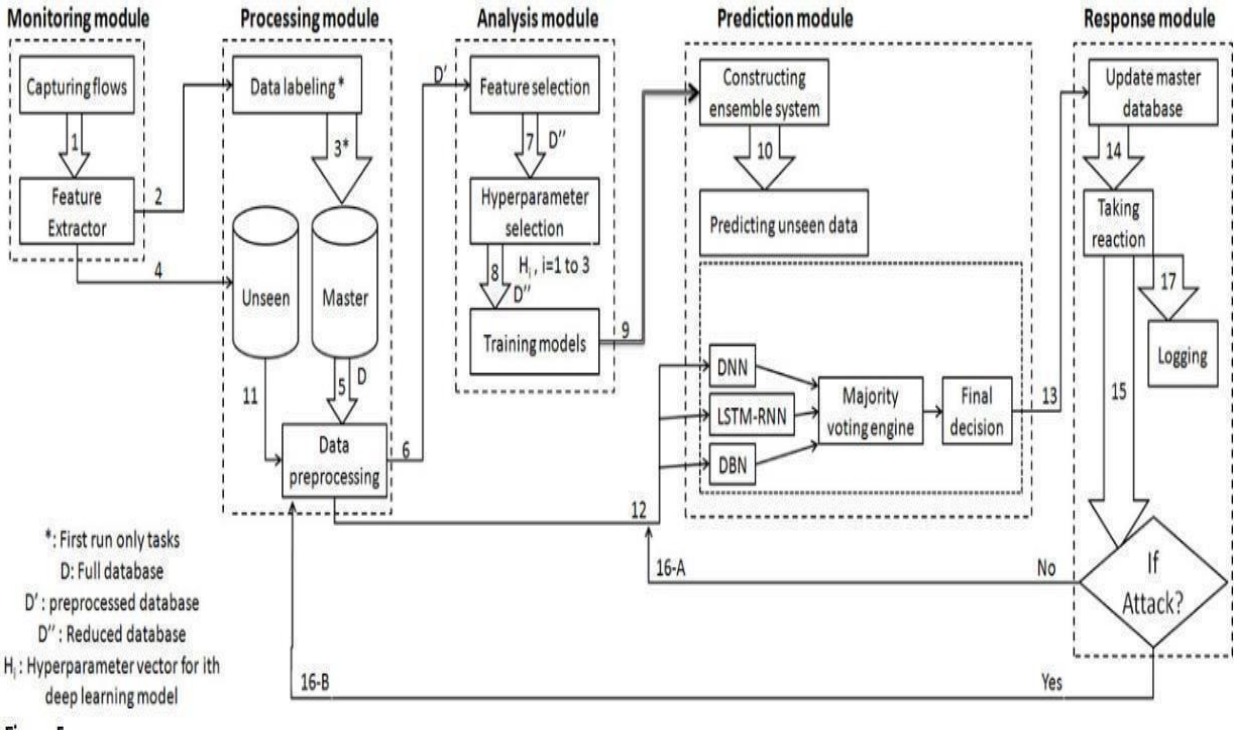


Fig 4.1 ER diagram for Cloud Intrusion Detection

4.2 WORK FLOW DIAGRAM

A Data-Flow Diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. There are several notations for displaying data-flow diagrams. For each data flow,at least one of the endpoints (source and / or destination) must exist in a process. The refined representation of a process can be done in another data -flow diagram,which subdivides this process into sub-processes. The dataflow diagram is part of thestructured-analysis modelling tools. When using UML, the activity diagram typically takes over the role of the data-flow diagram. A special form of data-flowplan is a site-oriented data-flow plan.

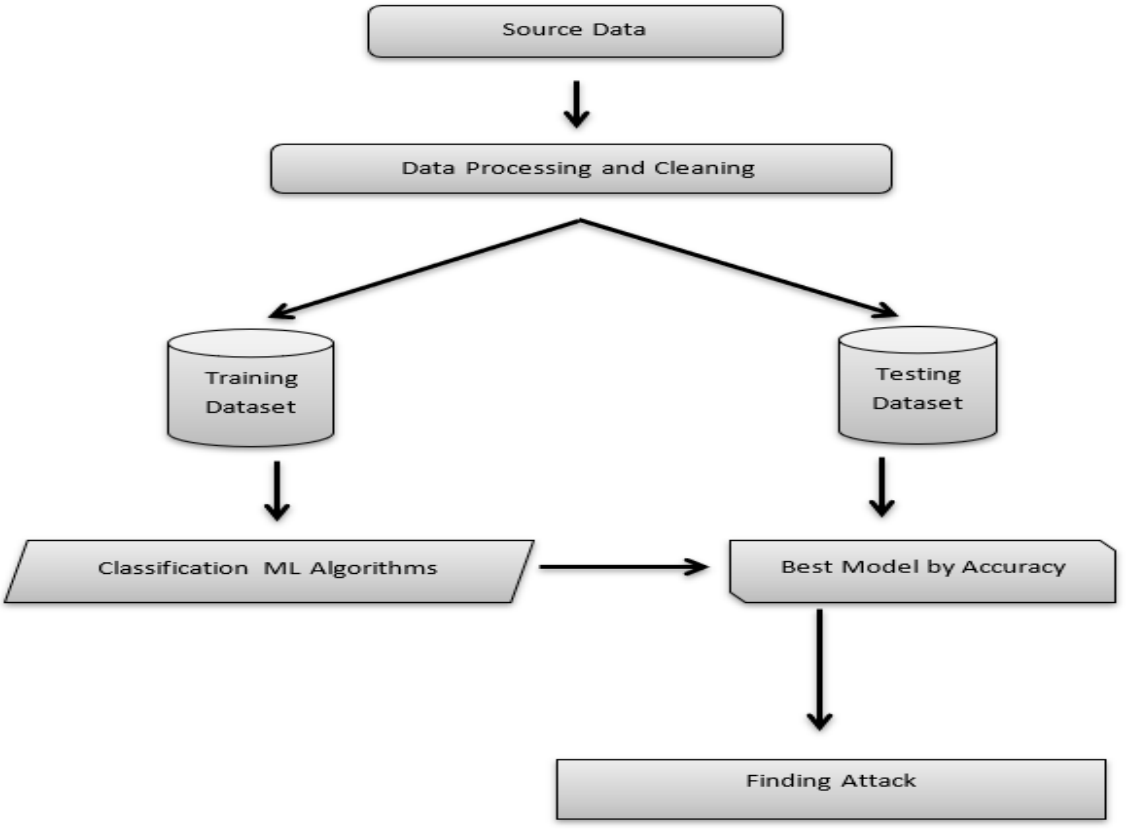


Fig 4.2 Workflow diagram for Cloud Intrusion Detection

4.3 USECASE DIAGRAM

Use case diagrams are considered for high level requirement analysis ofa system. So when the requirements of a system are analyzed the functionalities arecaptured in use cases. So, it can say that uses cases are nothing but the system functionalities written in an organized manner.

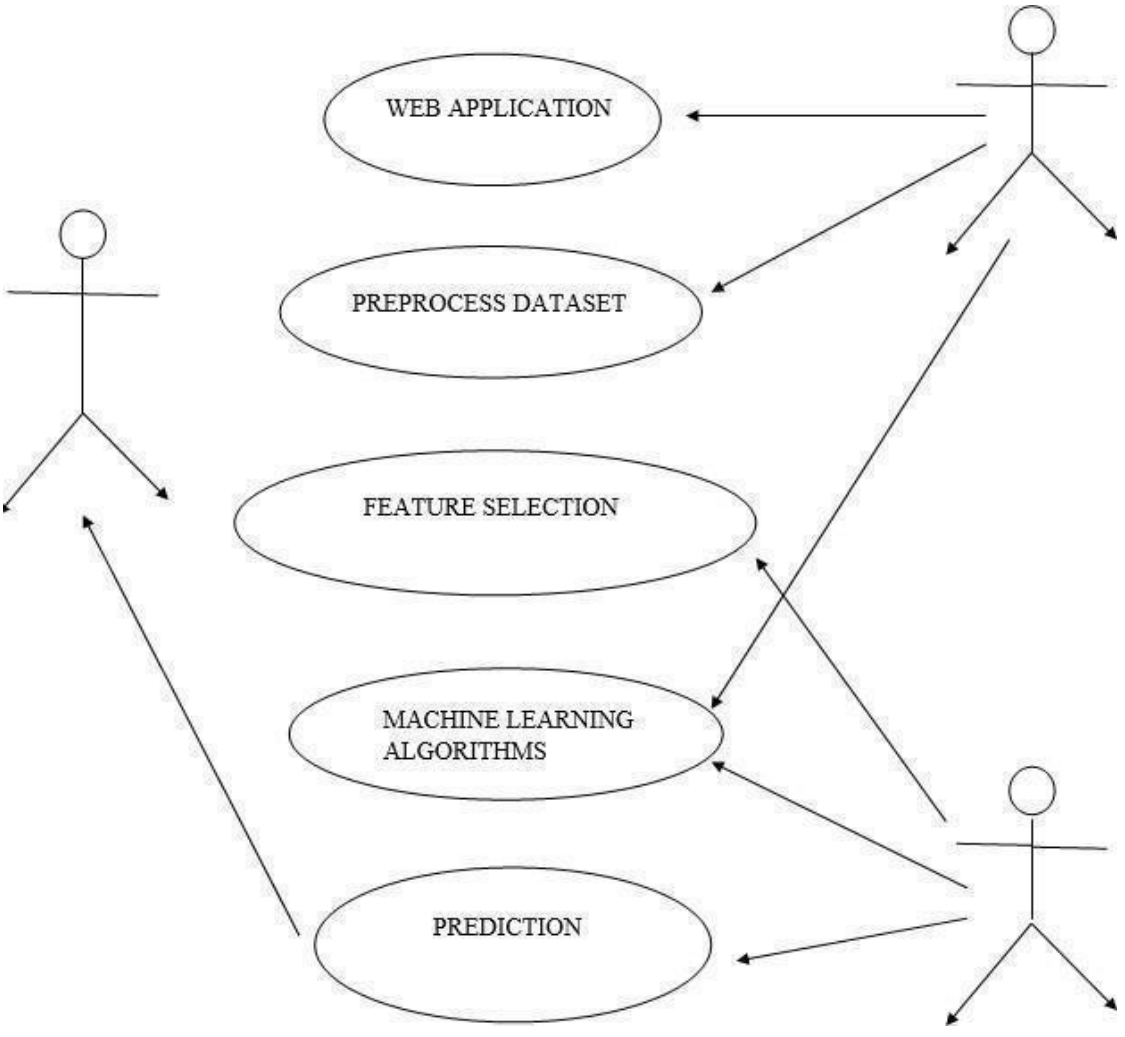


Fig 4.3 Usecase diagram for Cloud Intrusion Detection

4.4 CLASS DIAGRAM

Class diagram is basically a graphical representation of the static viewof the system and represents different aspects of the application. So a collection of class diagrams represent the whole system. The name of the class diagram should bemeaningful to describe the aspect of the system. Each element and their relationships should be identified in advance Responsibility (attributes and methods) of each classshould be clearly identified for each class minimum number of properties should bespecified and because, unnecessary properties will make the diagram complicated. Use notes whenever required to describe some aspect of the diagram and at the endof the drawing it should be understandable to the developer/coder. Finally, before making the final version, the diagram should be drawn on plain paper and rework asmany times as possible to make it correct.

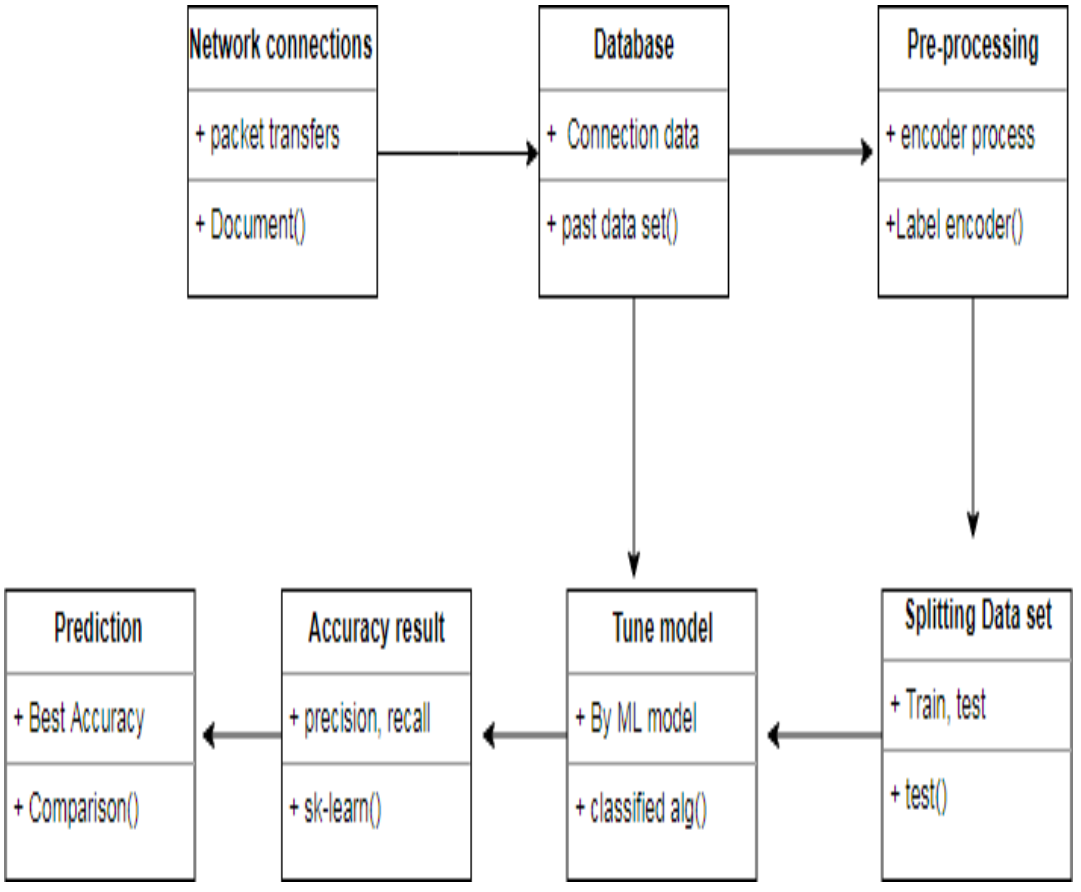


Fig 4.4 Class diagram for Cloud Intrusion Detection

4.5 ACTIVITY DIAGRAM

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part. It doesnot show any message flow from one activity to another. Activity diagram is sometime considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.

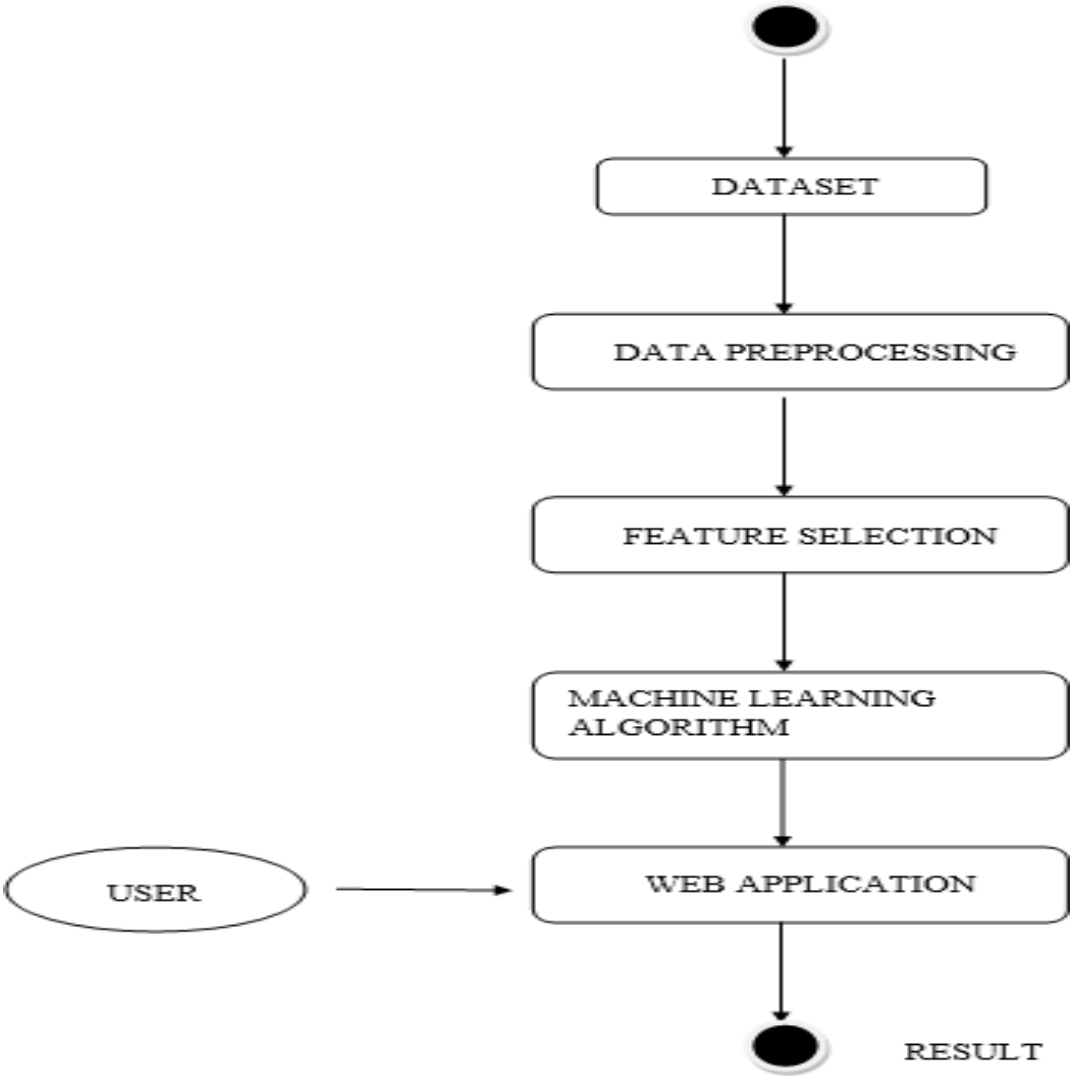


Fig 4.5 Activity diagram for Cloud Intrusion Detection

4.6 SEQUENCE DIAGRAM

Sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modelling, which focuses on identifying the behaviour within your system. Other dynamic modelling techniques include activity diagramming, communication diagramming, timing diagramming, and interaction overview diagramming. Sequence diagrams, along with class diagramsand physical data models are in my opinion the most important design-level models for modern business application development.

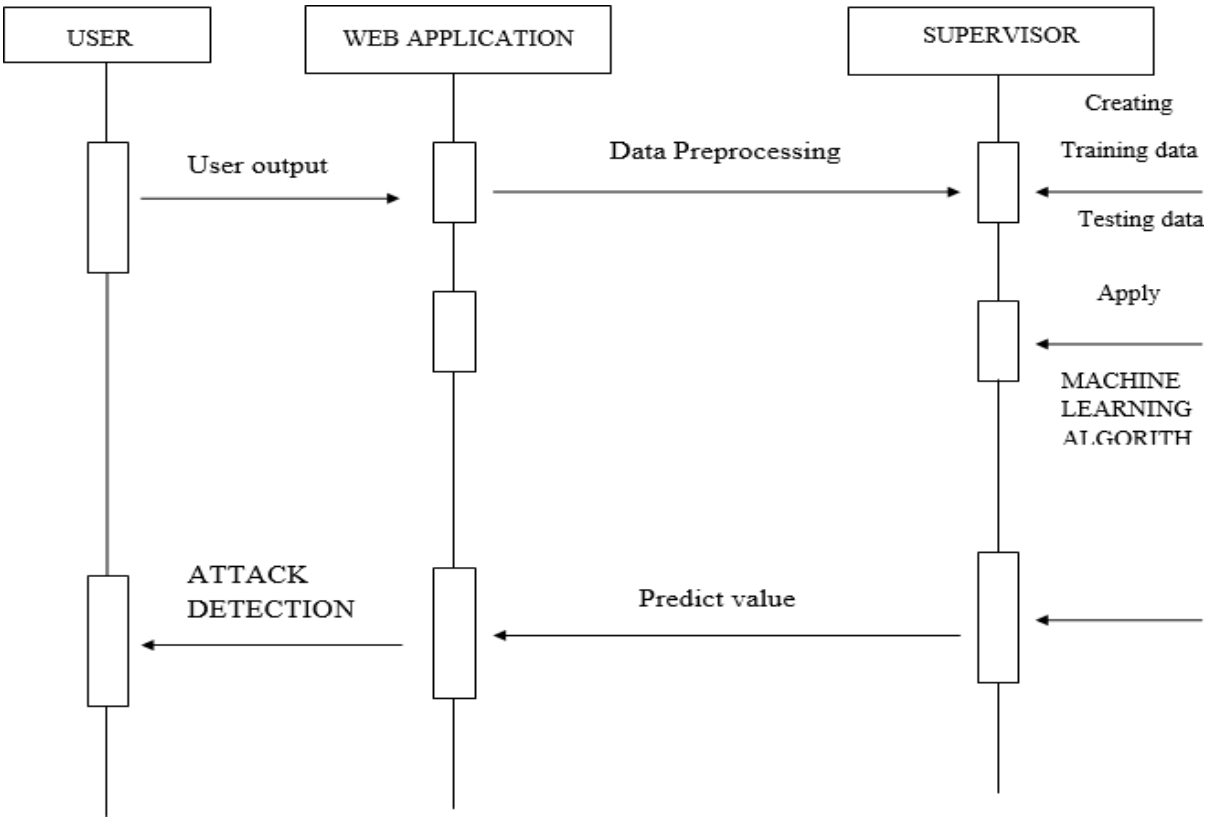


Fig 4.6 Sequence diagram for Cloud Intrusion Detection

4.7 COLLABORATION DIAGRAM

A collaboration diagram is a type of visual presentation that shows how various software objects interact with each other within an overall IT architecture and howusers (like doctor or patient) can benefit from this collaboration. A collaboration diagram often comes in the form of a visual chart that resembles a flow chart. It can show, briefly, how a single piece of software complements other parts of a greater system.

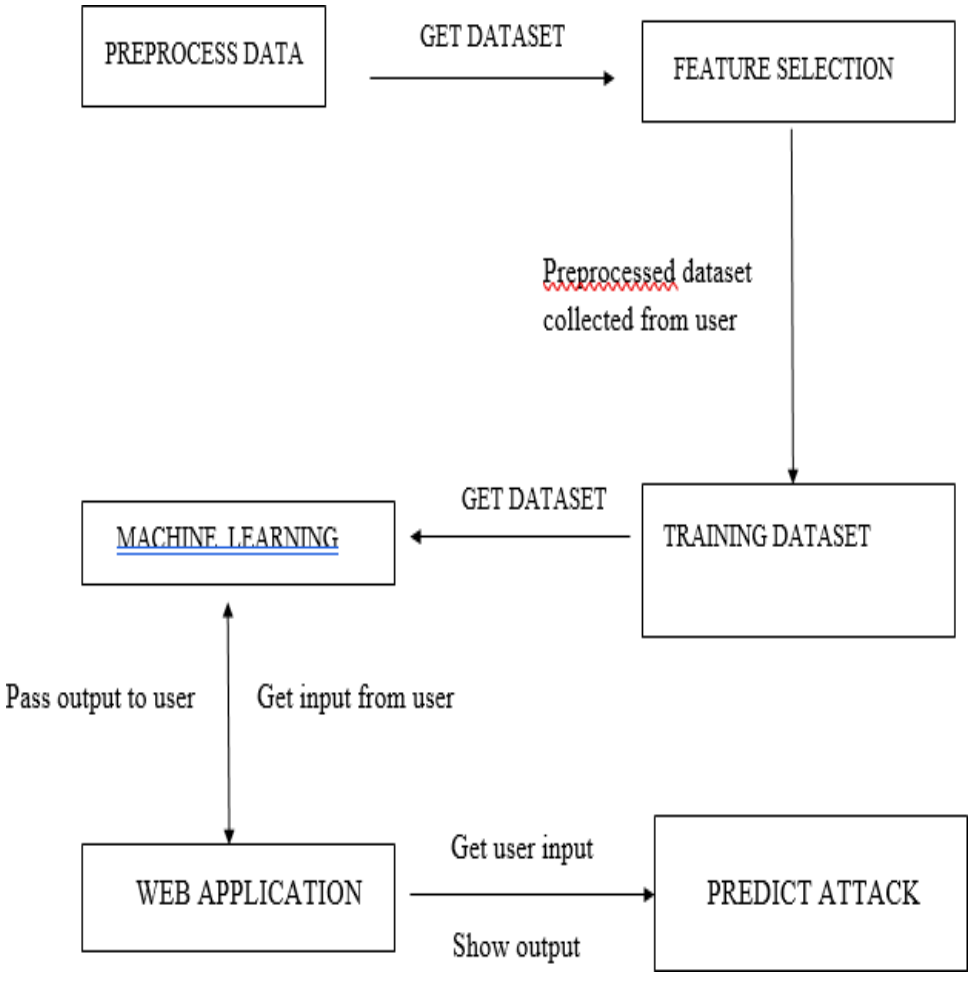


Fig 4.7 Collaboration diagram for Cloud Intrusion Detection

CHAPTER 5

SYSTEM ARCHITECTURE

A System architecture is the conceptual model that defines the structure ,behavior and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structure and the behavior of the system. A system architecture can consist of system components andthe sub-systems developed that will work together to implement the overall system.There have been efforts to formalize languages to describe system architecture.

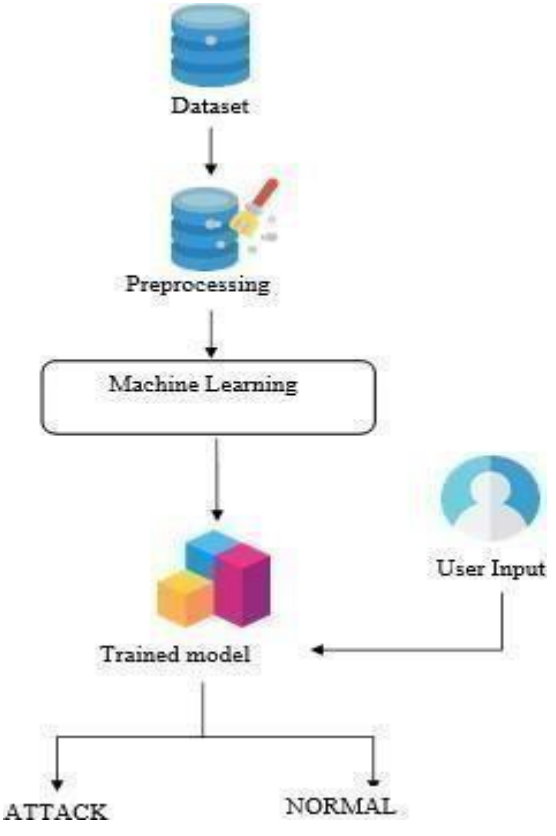


Fig 5.1 Architecture diagram for Cloud Intrusion Detection

5.1 ALGORITHM AND TECHNIQUES

In machine learning and statistics, classification is a supervisedlearning approach in which the computer program learns from the data input given toit and then uses this learning to classify new observation. This data set may simplybe bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc. In Supervised Learning, algorithms learn from labelled data. After understanding the data, the algorithm determines which label should be given to new data based on pattern and associating the patterns to the unlabeled new data.

Used Python Packages:

sklearn:

- In python, sklearn is a machine learning package which include a lot of ML algorithms.
- Here,we are using some of its modules like train_test_split, Decision Tree Classifier or Logistic Regression and accuracy_score.

NumPy:

- It is a numeric python module which provides fast maths functions for calculations.
- It is used to read data in numpy arrays and for manipulation purpose.

Pandas:

- Used to read and write different files.
- Data manipulation can be done easily with data frames.

Matplotlib:

- Data visualization is a useful way to help with identify the patterns from given dataset.
- Data manipulation can be done easily with data frames.

Comparing Algorithm with prediction in the form of best accuracy result

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Pythonwith scikit- learn. It can use this test harness as a template

on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that youhave created. When have a new dataset, it is a good idea to visualize the datausing different techniques to look at the data from different perspectives. The same idea applies to model selection. You should use several different waysof looking at the estimated accuracy of your machine learning algorithms to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance, and other properties of the distribution of model accuracies.

In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

5.2 MODULE DESCRIPTION

5.2.1 Module 1 - Data Pre-processing

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representativeof the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value,duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a modelfit on the training dataset while tuning model hyper parameters.

The evaluation becomes more biased as skill on the validation datasetis incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers use this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process ofdata identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model. A number of different data cleaning tasks using Python's Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values

and it able to more quickly clean data. It wants to spend less time cleaning data,and more time exploring and modeling.

Some of the sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It's important to understandthese different types of missing data from a statistics point of view. The type ofmissing data will influence how to deal with filling in the missing values

and to detect missing values, and do some basic imputation and detailed statistical approach for dealing with missing data. Before, joint into code, it'simportant to understand the sources of missing data. Here are some typical reasons why data is missing.

Variable identification with Uni-variate, Bi-variate and Multi-variate analysis:

- import libraries for access and functional purpose and read the given dataset
- General Properties of Analyzing the given dataset
- Display the given dataset in the form of data frame
- show columns
- shape of the data frame
- To describe the data frame
- Checking data type and information about dataset
- Checking for duplicate data
- Checking Missing values of data frame
- Checking unique values of data frame
- Checking count values of data frame
- Rename and drop the given data frame
- To specify the type of values

5.2.2 Module 2 - Data Validation

Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluatingthe missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skillwhile tuning model's and procedures that you can use to make the best use ofvalidation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc

Module Diagram

Fig 5.2.2 Data Validation

5.2.3 Module 3 - Extra Tree Classifier Algorithm

Tree based models have increased in popularity over the last decade, primarily dueto their robust nature. Tree-based models can be used on any type of data(categorical/continuous), can be used on data that is not normally distributed, andrequire little if any data transformations (can handle missing value/scale issues etc.) While Decision Trees and Random Forest are often the go to tree-based models, a lesser known one is Extra Trees.

Similar to Random Forests, Extra Trees is an ensemble ML approach that trains numerous decision trees and aggregates the results from the group of decision trees tooutput a prediction. However, there are few differences between Extra Trees and Random Forest.

Random Forest uses to select different variations of the trainingdata to ensure decision trees are sufficiently different. However, Extra Trees uses theentire dataset to train decision trees. As such, to ensure sufficient differences between individual decision trees, it Randomly Selects the values at which to splita feature and create child nodes. In contrast, in a Random Forest, we use an algorithm to greedy search and select the value at which to split a feature. Apart from these two differences, Random Forest and Extra Trees are largely the same. So what effect do these changes have?

Using the entire dataset (which is the default setting and can bechanged) allows Extra Trees to reduce the bias of the model. However, the randomization of the feature value at which to split, increases the bias and variance. The paper that introduced the Extra Trees model conducts a bias-variance analysisof different tree based models. From the paper we see on most classification and regression tasks (six were analyzed) Extra Trees have higher bias and lower variance than Random Forest. However, the paper goes on to say this is because therandomization in extra trees works to include irrelevant features into the model. Assuch, when irrelevant feature excluded, say via a feature selection pre- modelling step In terms of computational cost, Extra Trees is much faster than Random Forest.This is because Extra Trees randomly selects the value at which to split features,instead of the greedy algorithm used in Random Forest.

Why we should use Extra Tree Algorithm?

Random Forest remains the go-to ensemble tree based model (with recent competition from XBoost Models). However, from our prior discussion on thedifferences between Random Forest and Extra Trees, we see that Extra Trees havevalue, especially when computational cost is a concern. Specifically, when building models that have substantial feature engineering/feature selection pre-modelling steps, and computational cost is an issue Extra Trees would be a good choice over other ensemble tree-based models.

How to build an Extra Trees Model?

Extra Trees can be used to build classification model or regression models and is available via Scikit-learn. For this tutorial, we will cover the classification model,but the code can be used for regression with minor tweaks (i.e., switching from Extra Trees Classifier to Extra Trees Regressor)

The detailed list of parameters for the Extra Trees Model can be found on The Extra Trees Research paper calls out three key parameters explicitly, with the following statement.: The parameter K, nmin and M have different effects: K determines the strength of the attribute selection process, nmin the strength of averaging output noise, and M the strength of the variance reduction of the ensemble model aggregation.

Let's look at these parameters more closely from the implementation perspective.

- K is the max_feature in Scikit-learn documentation and refers to the number of features to be considered at each decision node. The higher the value of K, more features are considered at each decision node, and hence lower the bias of the model.However, too high a value of K reduces randomization, negating the effect of the ensemble.

- nmin maps to min_sample_leaf, and is a minimum number of samples required to be at a leaf node. The higher its value, the less likely the model is to overfit. Smallernumbers of samples result in more splits and a deeper, more specialized tree.

- M maps to n_estimators, and is a number of trees in the forest. The higher its value,the lower the variance of the model.

CHAPTER 6

PERFORMANCE ANALYSIS

TEST CASE AND REPORT

Normal:

The System is said to be normal when it is free from performing malicious activities against computer systems, devices, networks, applications by any threat actors who employ malicious activities.

srv_count	serror_rat	srv_serror	rerror_rat	srv_rerror	same_srv	diff_srv_r	srv_diff_h	dst_host_	dst_host_	dst_host_	dst_host_	dst_host_	dst_host_	dst_host_	dst_host_	dst_host_	target	Attack Type
8	0	0	0	0	1	0	0	9	9	1	0	0.11	0	0	0	0	normal.	normal
8	0	0	0	0	1	0	0	19	19	1	0	0.05	0	0	0	0	normal.	normal
8	0	0	0	0	1	0	0	29	29	1	0	0.03	0	0	0	0	normal.	normal
6	0	0	0	0	1	0	0	39	39	1	0	0.03	0	0	0	0	normal.	normal
6	0	0	0	0	1	0	0	49	49	1	0	0.02	0	0	0	0	normal.	normal
6	0	0	0	0	1	0	0	59	59	1	0	0.02	0	0	0	0	normal.	normal
2	0	0	0	0	1	0	1	1	69	1	0	1	0.04	0	0	0	normal.	normal
5	0	0	0	0	1	0	0	11	79	1	0	0.09	0.04	0	0	0	normal.	normal
8	0	0	0	0	1	0	0	8	89	1	0	0.12	0.04	0	0	0	normal.	normal
8	0	0	0	0	1	0	0	8	99	1	0	0.12	0.05	0	0	0	normal.	normal
18	0	0	0	0	1	0	0	18	109	1	0	0.06	0.05	0	0	0	normal.	normal
1	0	0	0	0	1	0	0	28	119	1	0	0.04	0.04	0	0	0	normal.	normal
11	0	0	0	0	1	0	0	38	129	1	0	0.03	0.04	0	0	0	normal.	normal
4	0	0	0	0	1	0	0	4	139	1	0	0.25	0.04	0	0	0	normal.	normal
1	0	0	0	0	1	0	0	14	149	1	0	0.07	0.04	0	0	0	normal.	normal
11	0	0	0	0	1	0	0	24	159	1	0	0.04	0.04	0	0	0	normal.	normal
2	0	0	0	0	1	0	0	34	169	1	0	0.03	0.04	0	0	0	normal.	normal
12	0	0	0	0	1	0	0	44	179	1	0	0.02	0.03	0	0	0	normal.	normal
8	0	0	0	0	1	0	0.25	54	189	1	0	0.02	0.03	0	0	0	normal.	normal
7	0	0	0	0	1	0	0	64	199	1	0	0.02	0.03	0	0	0	normal.	normal

Fig 6.1 Normal

R2L- Remote to Local:

This returns to a type of attack where an unauthorized user attempts to gain access toa system remotely from a network to explot vulnerabilities and escalate privileges toaccess local resource

0	2	2	0	0	0	0	1	0	0	4	14	1	0	1	0.21	0	0	0	0	warezclie	r2l
0	1	1	0	0	0	0	1	0	0	5	15	1	0	1	0.2	0	0	0	0	warezclie	r2l
0	2	2	0	0	0	0	1	0	0	6	16	1	0	1	0.19	0	0	0	0	warezclie	r2l
0	1	1	0	0	0	0	1	0	0	7	17	1	0	1	0.18	0	0	0	0	warezclie	r2l
0	2	2	0	0	0	0	1	0	0	8	18	1	0	1	0.17	0	0	0	0	warezclie	r2l
0	1	1	0	0	0	0	1	0	0	1	19	1	0	1	0.21	0	0	0	0	warezclie	r2l
0	2	2	0	0	0	0	1	0	0	2	20	1	0	1	0.2	0	0	0	0	warezclie	r2l
0	1	1	0	0	0	0	1	0	0	3	21	1	0	1	0.19	0	0	0	0	warezclie	r2l
0	2	2	0	0	0	0	1	0	0	4	22	1	0	1	0.18	0	0	0	0	warezclie	r2l
0	1	1	0	0	0	0	1	0	0	5	23	1	0	1	0.17	0	0	0	0	warezclie	r2l
0	2	2	0	0	0	0	1	0	0	6	24	1	0	1	0.17	0	0	0	0	warezclie	r2l
0	1	1	0	0	0	0	1	0	0	7	25	1	0	1	0.16	0	0	0	0	warezclie	r2l
0	2	2	0	0	0	0	1	0	0	8	26	1	0	1	0.15	0	0	0	0	warezclie	r2l
0	1	1	0	0	0	0	1	0	0	1	27	1	0	1	0.19	0	0	0	0	warezclie	r2l
0	2	2	0	0	0	0	1	0	0	2	28	1	0	1	0.18	0	0	0	0	warezclie	r2l
0	1	1	0	0	0	0	1	0	0	3	29	1	0	1	0.17	0	0	0	0	warezclie	r2l
0	2	2	0	0	0	0	1	0	0	4	30	1	0	1	0.17	0	0	0	0	warezclie	r2l
0	1	1	0	0	0	0	1	0	0	5	31	1	0	1	0.16	0	0	0	0	warezclie	r2l
0	2	2	0	0	0	0	1	0	0	6	32	1	0	1	0.16	0	0	0	0	warezclie	r2l
0	1	1	0	0	0	0	1	0	0	7	33	1	0	1	0.15	0	0	0	0	warezclie	r2l
0	2	2	0	0	0	0	1	0	0	8	34	1	0	1	0.15	0	0	0	0	warezclie	r2l
0	1	1	0	0	0	0	1	0	0	1	35	1	0	1	0.17	0	0	0	0	warezclie	r2l
0	2	2	0	0	0	0	1	0	0	2	36	1	0	1	0.17	0	0	0	0	warezclie	r2l

Fig 6.2 R2L- Remote to Local

DoS – Denial of Service

A type of cyber attack that aims to make a machine or network resource unavailable toits intented users by overwhelming it excessive traffic or requests, thereby denying access to legitimate users

0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	510	510	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	510	510	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos
0	511	511	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	smurf.	dos

Fig 6.3 DoS- Denial of Service

Probe:

A probe attack involves scanning or probing a network or system to gather information such as open ports, vulnerabilities or weaknesses, often a precursor to amore significant attack.

0	511	1	0.23	0	0.77	1	0	1	0	255	1	0	1	0	0	0.23	0	0.77	1	satan.	probe
0	511	1	0.23	0	0.77	1	0	1	0	255	1	0	1	0	0	0.19	0	0.81	1	satan.	probe
0	511	1	0.23	0	0.77	1	0	1	0	255	1	0	1	0	0	0.15	0	0.85	1	satan.	probe
0	511	1	0.23	0	0.77	1	0	1	0	255	1	0	1	0	0	0.15	0	0.85	1	satan.	probe
0	511	1	0.23	0	0.77	1	0	1	0	255	1	0	1	0	0	0.15	0	0.85	1	satan.	probe
0	511	1	0.23	0	0.77	1	0	1	0	255	1	0	1	0	0	0.15	0	0.85	1	satan.	probe
0	511	1	0.23	0	0.77	1	0	1	0	255	1	0	1	0	0	0.15	0	0.85	1	satan.	probe
0	511	1	0.23	0	0.77	1	0	1	0	255	1	0	1	0	0	0.15	0	0.85	1	satan.	probe
0	509	1	0.24	0	0.76	1	0	1	0	255	1	0	1	0	0	0.16	0	0.84	1	satan.	probe
0	509	1	0.23	0	0.77	1	0	1	0	255	1	0	1	0	0	0.16	0	0.84	1	satan.	probe
0	509	1	0.21	0	0.79	1	0	1	0	255	1	0	1	0	0	0.15	0	0.85	1	satan.	probe
0	509	1	0.19	0	0.81	1	0	1	0	255	1	0	1	0	0	0.11	0	0.89	1	satan.	probe
0	509	1	0.17	0	0.83	1	0	1	0	255	1	0	1	0	0	0.08	0	0.92	1	satan.	probe
0	509	1	0.16	0	0.84	1	0	1	0	255	1	0	1	0	0	0.04	0	0.96	1	satan.	probe
0	509	1	0.17	1	0.83	0	0	1	0	255	1	0	1	0	0	0.02	1	0.98	0	satan.	probe
0	509	1	0.18	1	0.82	0	0	1	0	255	1	0	1	0	0	0.06	1	0.94	0	satan.	probe
0	509	1	0.2	0	0.8	1	0	1	0	255	1	0	1	0	0	0.09	0	0.91	1	satan.	probe
0	509	1	0.22	1	0.78	0	0	1	0	255	1	0	1	0	0	0.13	1	0.87	0	satan.	probe
0	509	1	0.23	0	0.77	1	0	1	0	255	1	0	1	0	0	0.15	0	0.85	1	satan.	probe
0	509	1	0.23	0	0.77	1	0	1	0	255	1	0	1	0	0	0.15	0	0.85	1	satan.	probe
0	509	1	0.23	0	0.77	1	0	1	0	255	1	0	1	0	0	0.15	0	0.85	1	satan.	probe
0	509	1	0.23	0	0.77	1	0	1	0	255	1	0	1	0	0	0.15	0	0.85	1	satan.	probe
0	509	1	0.23	0	0.77	1	0	1	0	255	1	0	1	0	0	0.15	0	0.85	1	satan.	probe

Fig 6.4 Probe

CHAPTER 7

CONCLUSION

7.1 RESULTS AND DISCUSSIONS

The initial phases in the analytical process were data cleansing and processing, missingvalue detection, and lastly model construction and assessment. High accuracy scores will be used to determine who has the best accuracy on public tests. The probable outcome of any network penetration can be discovered with the use of this software.

7.2 CONCLUSION AND FUTURE ENHANCEMENTS

The analytical process started from data cleaning and processing , missing value, exploratory analysis and finally model building and evaluation . The best accuracy onpublic test set is higher accuracy score will be find out by comparing each algorithm with type of all WSN Attacks for future prediction Results by finding best connections.This brings some of the following insights about diagnose the network attack of each new connection. To presented a Prediction model with the aid of artificial Intelligenceto improve over human accuracy and provide with the scope of early detection. It canbe inferred from this model that , area analysis and use of machine learning techniqueis useful in developing prediction models that can helps to network sectors reduce thelong process of diagnosis and eradicate any human error.

- Cloud sector want to automate and detecting the attacks of transfers from eligibility process (real time) based on the connection detail.
- To automate this process by show the prediction result in web application or desktop application at cloud.

APPENDICES

A.1 SDG GOALS

Goal 0

Industry, Innovation, and Infrastructure our project contributes to promoting inclusive sustainable industrialization, and fostering innovation.

Goal 1

Peace, Justice, and Strong Institutions - By enhancing security and reducing cyber threats, your framework contributes to building strong institutions and promoting peace and justice.

Goal 2

Partnerships for the Goals - Collaboration and partnerships are essential for achieving sustainable development, and your project likely involves cooperation between different stakeholders.

These goals reflect the broader impact of your work beyond technical advancements, highlighting its potential to contribute to a more secure, innovative, and collaborative world.

A.2 SOURCE CODE

```
import pandas as
pd import numpy
as np import
seaborn as sns
import matplotlib.pyplot as plt
data=pd.read_csv('./dataset/kddcup.data_10_percent_cor
rected') data
data['normal.'].unique()
f=open("./dataset/kddcup_names.csv",'r')
x=f.read()
print(x) cols=['duration','protocol_type','service','flag','src_bytes','dst_bytes','land','wrong_fragment',
'urgent','hot','num_failed_logins','logged_in','num_compromised','root_shell','su_attempted',
'num_root','num_file_creations','num_shells','num_access_files','num_outbound_cmds','is_host_login',
'is_guest_login','count','srv_count','serror_rate','srv_serror_rate',']
data.columns=c
olsdata
f1=open(r"./dataset/training_attack_types
",'r')x1=f1.read()
print(x
1)
l=x1.spl
it(
)l
len
(l)
d={}
for i in
range(0,44,2):
```

```
d[l[i]]=l[i+1]
d['normal']='normal'
data['target_attack_type']=data['target'].apply(lambda r:d[r[:-1]])
data.isnull().sum()
data['target_attack_type'].unique()
data['target_attack_type'].value_co
unts()
data['label']= data['target_attack_type'].apply(lambda x: 0 if x == 'normal' else 1)
data['label'].value_counts()
value=data['label'].value_counts()
plt.pie(value.values.tolist(), labels=['cyber_attack','Safe'],
autopct='%.2f%%')plt.legend()
plt.show()
value1=data['target_attack_type'].value_cou
nts()labels = ['dos','Normal', 'probe', 'r2l',
'u2r']
plt.pie(value1.values.tolist(), labels=labels, autopct='%.2f%%')
plt.show()
data.info()
data.describ
e()
data=data.drop(['label'],axis=1
)data
data.to_csv("./dataset.csv",index=False)
```

Visualization

```
import pandas as
pdimport numpy
as np
data=pd.read_csv('./dataset.csv')data
```

```
data['target'].value_counts()

data.shape

data

data['protocol_type'].value_c

ounts()

data['target'].value_counts()

data['Attack

Type'].value_counts()

X=data.drop(['target','At

tackType'],axis=1)X

Y=data['Attack

Type']Y

from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.15,

random_state=111)import seaborn as sns

import matplotlib.pyplot

as plt

plt.figure(figsize=(20,20)

) cor=X_train.corr()

sns.heatmap(cor,annot=True,cmap=plt.cm.CMR

map_r)plt.show()

def correlation(

X_train,threshold):

col_corr=set()

corr_matrix=X_train.corr()

for i in

range(len(corr_matrix.columns)):for j
```

```
in range(i):
if
abs(corr_matrix.iloc[i,j])>threshold
:colname=corr_matrix.columns[i]
col_corr.add(colname)
   return col_corr
corr_features=correlation(X_train
,0.7)corr_features
data = data.drop(corr_features, axis=1)

data
X=data.drop(['target','At
tack
Type'],axis=1)Y=data['Attack
Type']X
dd=pd.read_csv('./testingdatas.
csv')dd
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.15,
random_state=111)X_train
X_test
fromsklearn.ensemble import ExtraTreesClassifier
etc = ExtraTreesClassifier(min_samples_split=7, random_state=111)
etc.fit(X_train,Y_train)
ET=etc.score(X_train,Y_train
) ETC =
etc.score(X_test, Y_test)
print('Score:{}'.format(E
```

```
TC))etc
import joblib
joblib.dump(etc,"etc_model.
pkl")data

 data['Attack
Type'].value_cou
nts()
data[data['Attack
Type']==0]
data[data['Attack
Type']==2]
data[data['Attack
Type']==3]
data[data['Attack Type']==4]
from flask import Flask
,render_template,request,jsonify,sessionfrom flask import
Flask, render_template, url_for, request importpandas as
pd
import joblib
#from sklearn.feature_extraction.text import
CountVectorizer#from sklearn.naive_bayes import
MultinomialNB
#import sqlite3 as
sql#import
base64
#from sklearn.preprocessing import
LabelEncoder#from flask_bootstrap import
Bootstrap
```

```
import numpy as np
#from sklearn.utils import
shuffleimport os
from flask import Flask, render_template, request,
url_for,send_from_directoryimport os
#import tensorflow as tf
#from geo import
getTweetLocationapp = Flask(
name )
#app.secret_key = 'any random string'
#PEOPLE_FOLDER = os.path.join('static',
'people_photo')data=pd.read_csv('./testingdatas.csv'
) model=joblib.load(open('./etc_model.pkl','rb'))
@app.route('/')
def home():
 return render_template('index.html')
@app.route('/data', methods=['GET',
'POST'])defindex():
 if request.method == 'POST':
number = request.form['number']
number=int(number)
# Assuming you have a dataset 'data' with columns and values

d1=data.iloc[number:numb
er+1,:]cl=data.iloc[1:1,:]
cl=cl.keys().to_list()
   l=len(cl)
print("#"*50,cl,len
```

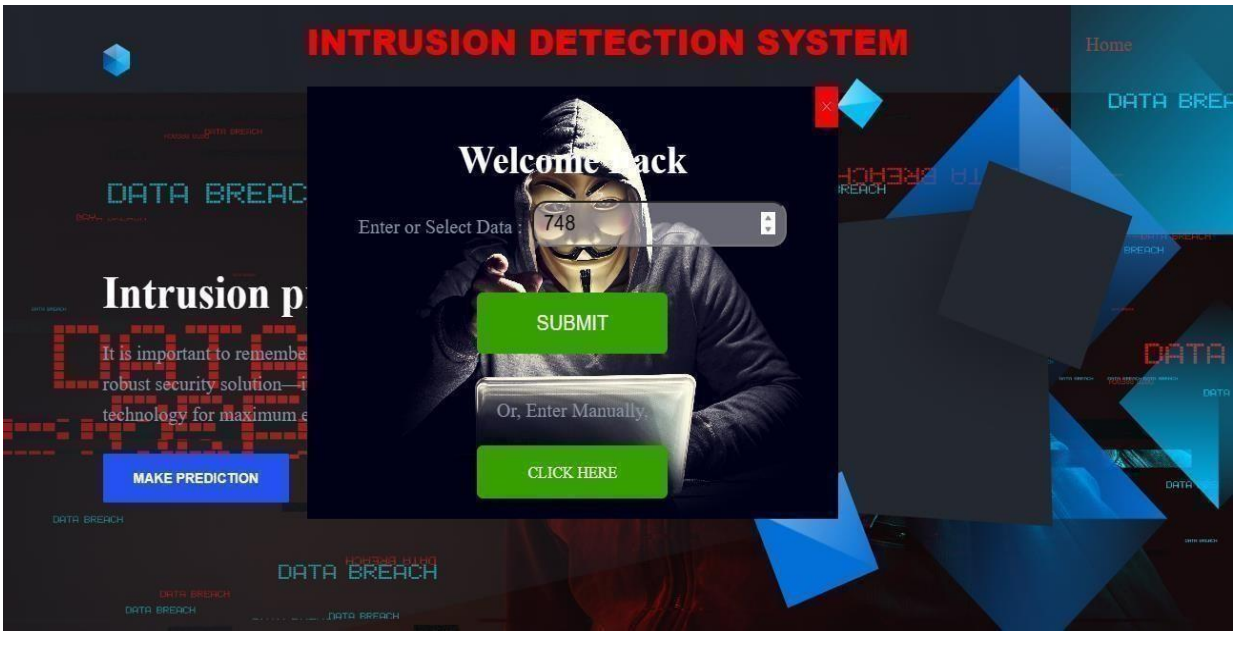
```
(cl)
d2=np.array(d1)
for i in d2:
d22=i
# Pass the dataset to the HTML template
   return render_template('data.html',
data=d22,columns=cl,number=number,l=l)@app.route('/result',methods
= ['POST'])
def result():
if request.method == 'POST':
duration =
request.form['duration']
protocol_type = request.form['protocol_type']
src_bytes = request.form['src_bytes'] dst_bytes
= request.form['dst_bytes']
land = request.form['land']
wrong_fragment =
request.form['wrong_fragment'urgent =
request.form['urgent']
hot = request.form['hot']
num_failed_logins = request.form['num_failed_logins']
logged_in = request.form['logged_in']
num_compromised
= request.form['num_compromised']root_shell
=request.form['root_shell']
      su_attempted = request.form['su_attempted']
num_file_creations =
```

```
request.form['num_file_creations']num_shells =
request.form['num_shells']
num_access_files = request.form['num_access_files']
num_outbound_cmds =
request.form['num_outbound_cmds']is_host_login =
request.form['is_host_login']
diff_srv_rate = request.form['diff_srv_rate']
srv_diff_host_rate = request.form['srv_diff_host_rate']
dst_host_count = request.form['dst_host_count']
dst_host_diff_srv_rate = request.form['dst_host_diff_srv_rate']
dst_host_srv_diff_host_rate =
request.form['dst_host_srv_diff_host_rate'input_data =
np.array([[duration,protocol_type,src_bytes,dst_bytes,land,
wrong_fragment,urgent,hot,num_failed_logins,logged_in,
num_compromised,root_shell,su_attempted,num_file_creations,
num_shells,num_access_files,num_outbound_cmds,is_host_login,
diff_srv_rate,srv_diff_host_rate,dst_host_count,
dst_host_diff_srv_rate,dst_host_srv_diff_host_rate]])
# Reshape the data to a 2D array (required by some
models)#input_data = input_data.values.reshape(1, -
1)
# Make predictions
result =
model.predict(input_data)#
Display the result
print(result)
return render_template('result.html',result=result)
@app.route('/input', )
```

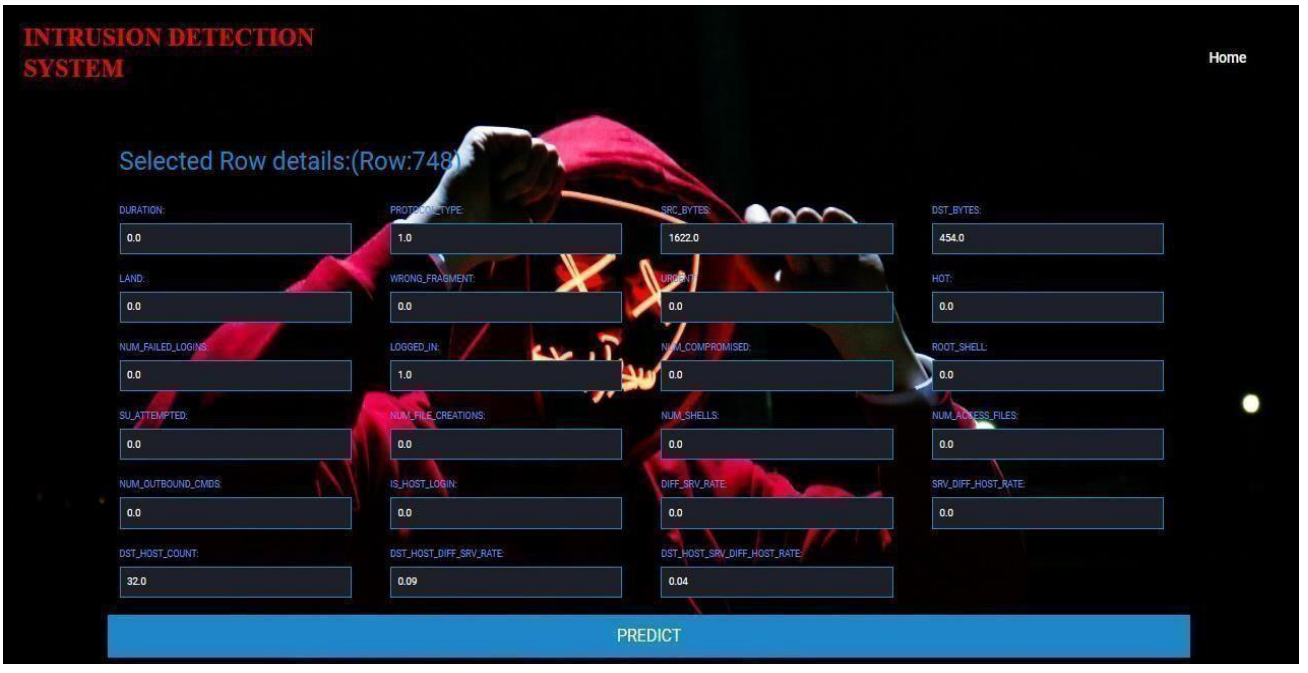
```
def input()
# Assuming you have a dataset 'data' with columns and
valuesdata=pd.read_csv('./testingdatas.csv')
cl=data.iloc[1:1,:
]
cl=cl.keys().to_li
st(
)l=len(cl)
# Pass the dataset to the HTML template
 return render_template('input.html',
columns=cl,l=l)if__name____== ' main ':
app.run(debug = True )
```

A.3 SCREENSHOTS

Home Page



Login Page



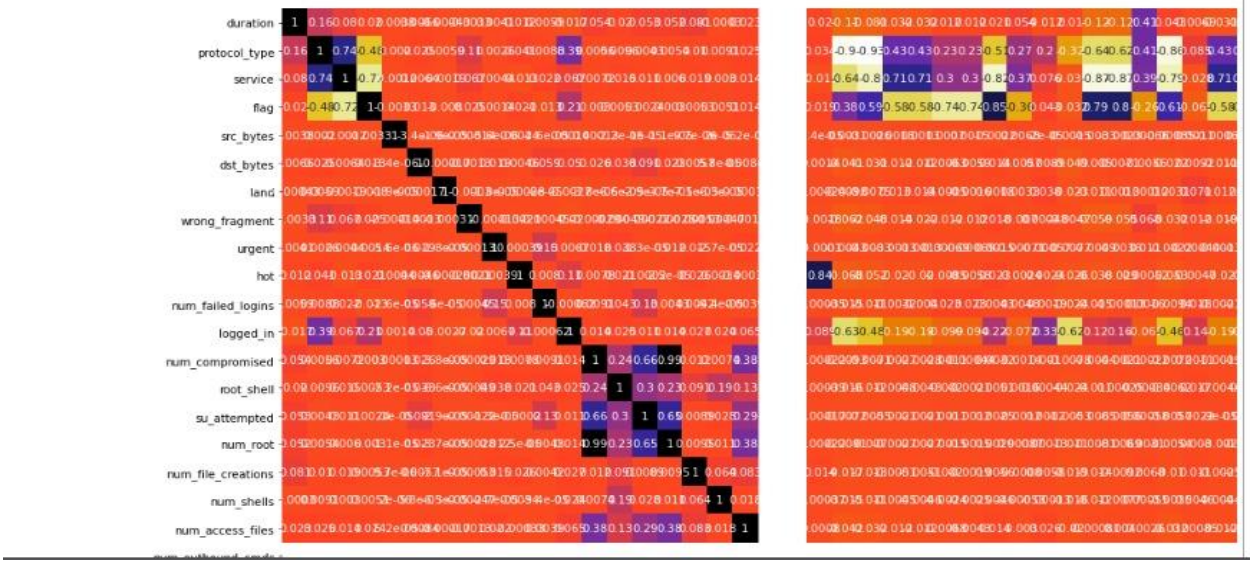
Data Collection

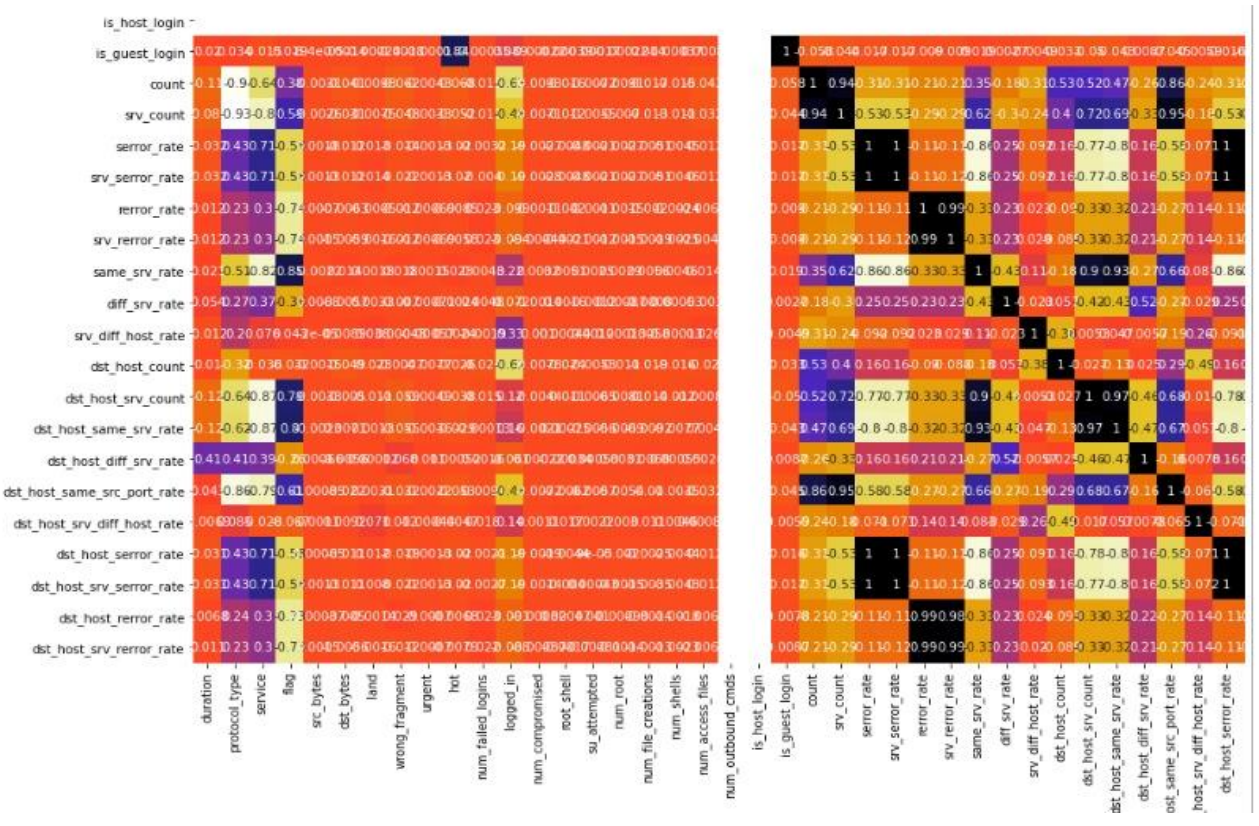
Result Page 1

Result Page 3

Result Page 4

```
In [18]: import seaborn as sns
         import matplotlib.pyplot as plt
         plt.figure(figsize=(20,20))
         cor=X_train.corr()
         sns.heatmap(cor,annot=True,cmap=plt.cm.CMRmap_r)
         plt.show()
```





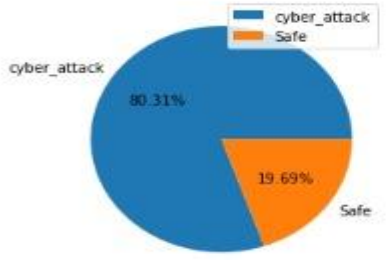
Accuracy Chart 1

```
In [20]: data['label']= data['target_attack_type'].apply(lambda x: 0 if x == 'normal' else 1)

In [21]: data['label'].value_counts()

Out[21]: 1    396743
         0     97277
         Name: label, dtype: int64

In [22]: value=data['label'].value_counts()
         plt.pie(value.values.tolist(), labels=['cyber_attack','Safe'], autopct='%.2f%%')
         plt.legend()
         plt.show()
```

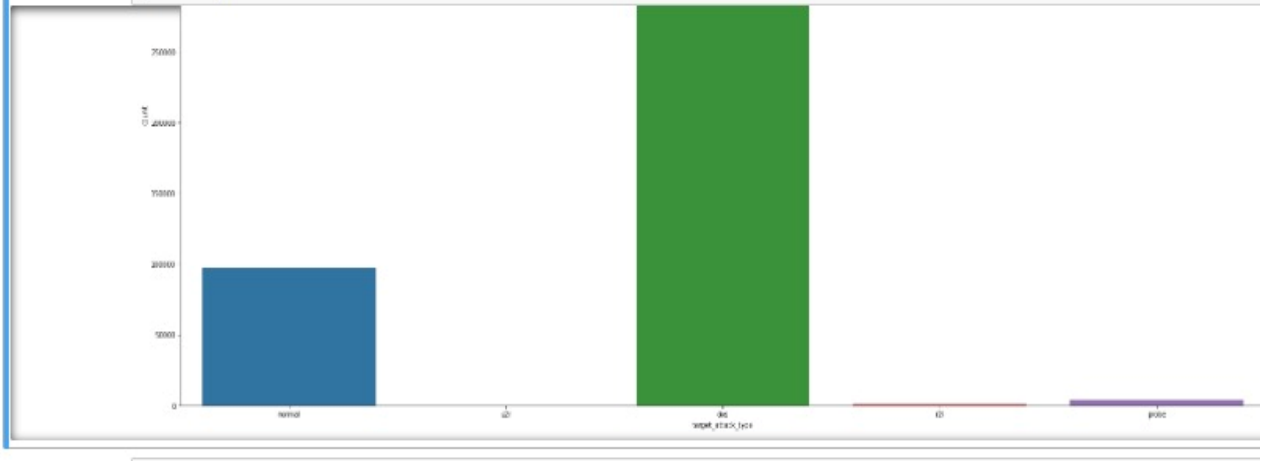


Accuracy Chart 2

```
In [25]: labels = ['dos','Normal', 'probe', 'r2l', 'u2r']

         plt.pie(value1.values.tolist(), labels=labels, autopct='%.2f%%')

         plt.show()
```



Accuracy Chart 3

Optimized Cloud Security Framework Harnessing Hybrid Feature Selection and Machine Learning Classification for Intrusion Detection

1st Jackulin C
Assistant Professor
Department of CSE
Panimalar Engineering College
Chennai, India
chin.jackulin@gmail.com

2nd Chenna Reddy Parithraan
UG Scholar
Department of CSE
Panimalar Engineering College
Chennai, India
parithraanreddy1267@gmail.com

3rd Manchiganti DeerajSurya
UG Scholar
Department of CSE
Panimalar Engineering College
Chennai, India
deerajsurya632@gmail.com

4th Malireddy Sai Rakesh
UG Scholar
Department of CSE
Panimalar Engineering College
Chennai, India
rakeshmalireddy143@gmail.com

***Abstract*— Network attacks pose a major danger to computer networks' integrity and security. Being able to recognize and prevent these threats is crucial to maintaining the security of a network environment. Supervised machine learning algorithms are now helpful tools for forecasting network assaults because of their ability to examine enormous amounts of network data and identify patterns suggestive of hostile activity. We present a comprehensive analysis of unsupervised machine learning techniques for network attack prediction. Once the data is gathered, we preprocess it by eliminating significant knowledge and organizing it so that machine learning algorithms can use it. We assess these algorithms' performance. To understand the foundational patterns and traits of network attacks, we have to look into the comprehensible nature the trained models. This allows network administrators to understand the nature of attacks and develop appropriate defences strategies. Additionally, we discuss the challenges and limitations associated with the application of supervised machine learning techniques in the domain of network attack prediction, such as the need for real-time analysis and the emergence of sophisticated evasion techniques.**

Keywords— Network attack, Security, Strategy, Supervised Learning.

I. INTRODUCTION

.

Using historical data to forecast the future is the aim of machine learning. Machine learning (ML), a subset of artificial intelligence (AI), allows computers to learn without the need for explicit programming. The creation of computer programs that can adapt to new data and applying the concepts of machine learning, like using Python to develop a simple machine learning algorithm, are the primary concerns of machine learning. Particular algorithms are used for training and prediction. Particular algorithms are used for training and prediction. An algorithm utilizes the training data in order to anticipate what will happen to new test data after it has been received. Three main classifications can be used to categorize machine learning. Supervised learning, unsupervised learning, and incentive-based learning are the three categories of learning. Both the input data and the accompanying labeling are provided for supervised learning. The steps listed below must be followed in order to determine the optimal clustering of the input data. In summary, through dynamic interactions with its surrounding environment, reinforcement learning learns to perform effectively through the use of both positive and negative feedback. Data scientists look for patterns in Python that provide informative information through implementing an assortment of machine learning techniques. These many algorithms can mostly be divided into two categories:

supervised and unsupervised learning, based on the way they "learn" from the data in order to provide predictions. Classification is the process of predicting the class of a data piece. Classes can also be called labels, objectives, or classifications Segmentation modeling for prediction is the process of predicting a mapping function from discrete output variables (Y) to input variables (X). Both multi-class and bi-class data collection is possible (for example, specifying if the message is spam or not, or whether the recipient is a man or a woman). Classification difficulties can be found in speech recognition, handwriting recognition, biometric identification, document categorizing, and other fields. In statistics and machine learning, classification is a type of supervised learning in which a computer program learns from the data it is fed and uses that knowledge to classify future observations. Network breaches can be identified by using supervised learning to classify network traffic as benign or malicious. While methods for unsupervised learning can be used to find undiscover Algorithms based on machine learning are able to recognize known exploits and additionally zero-day threats. The most effective framework has been found by combining algorithmic methods for supervised learning with methods for choosing features to account for detection success rate. The algorithm measures its accuracy using the loss function and makes modifications until the error is appropriately decreased.

II. PROPOSED METHODOLOGY

The recommendation we make was to employ a system and a machine learning technique to create the project. Recently, artificial intelligence and machine learning are becoming progressively significant for the continued expansion and development of many enterprises. In an attempt to strengthen their sense of safety, we attempted to use machine learning procedures. The project's goal is to create a thread that alerts security so that it may be closed before catastrophically damaging the organization or a person. We compile the historical documents of the incidents which happened in these

eras. Our machine learning system investigated for patterns in those datasets using the data that was generated. The machine can use data previously collected to project the outcome of the instance after distinguishing particular characteristics. To do so, we can take advantage of a replacement algorithm. We can achieve exceptional precision for that. We pronounce our framework to be good considering its high values of accuracy.

Data Collection:

Two separate sets of data a Training set and a Test combination make up the data set that was made use of for generating the prediction. The Training set and Test set are usually divided using 7:3 ratios. The Data Model, which was developed using logistic, Random Forest, Decision Tree, and Support Vector Classifier (SVC) algorithms, is exposed to the training set. The correctness of the test results serves as a starting point for a test set prediction.

Table 01: Comparison between proposed model using RF and other ML models

Dataset	Feature Selection	Selected Features	Total No. of Features	ACC	P	R	F	FAR
UNSW-NB15	Without	All features except No. 48	48	96.49	96.58	96.49	96.49	0.161
	IG	1,2,3,4,6,7,8,9,10,11,15,16,18,23,24,32,37,49	18	98.02	98.11	98.02	98.08	0.129
	CS	1,2,3,4,6,10,11,14,15,16,19,20,21,22,23,24,27,29,30,33,34,35,37,39,41,42,43,44,45,46,47,49	32	97.61	97.79	97.61	97.73	0.156
	PSO	1,2,3,4,5,6,7,8,9,10,11,15,16,18,23,24,29,30,31,32,37,45,46,47,49	25	97.99	98.07	97.99	98.05	0.141
	IG-CS-PSO	1,2,3,4,5,6,7,8,9,10,11,15,16,18,23,24,29,30,32,37,46,49	21	98.39	98.54	98.39	98.46	0.046
Kyoto	Without	All features except No. 18	23	97.32	96.87	97.32	96.50	0.122
	IG	1,2,3,4,5,6,8,10,13,14,17,19,20,21,22,23	16	98.96	98.99	98.96	98.95	0.053
	CS	6,8,10,12,13,14,17,21,24	9	98.97	99.01	99.08	99.11	0.012
	PSO	1,2,3,4,8,10,14,17,19,20,21,22,24	13	99.08	99.13	99.08	99.11	0.012
	IG-CS-PSO	1,2,3,4,10,14,19,20,21,22	10	99.25	99.27	99.25	99.26	0.008

Processing:

The obtained data may have missing values, leading to inconsistencies. In order to improve algorithm performance and yield better results, preprocessing data is necessary. In addition to removing the outliers, the variable conversion must be accomplished.

Constructing a model for Classification:

The expected assault, The following factors affect the prediction model's effectiveness when using the Random Forest Algorithm: It produces better results in the classification task. It is effective at handling outliers and irrelevant variables in addition to mixed discrete, continuous, and categorical data preparation. It also yields a very customizable out-of-bag estimate error that has shown objectivity throughout multiple tests.

III SYSTEM ARCHITECTURE

In machine learning and statistics, supervised learning—in which a computer program learns from the data it is fed—is used to classify new observations. This data collection could be multi-class in addition to bi-class (e.g., determining whether the recipient is male or female, or indicating if the email is spam).

Such as biometric identification, speech recognition, handwriting recognition, document categorization, and other classification-related problems. Using tagged data, alphabets are learned using supervised learning. The algorithm recognizes patterns in the unlabeled data to determine which label to apply to new data once it has a firm grasp of the existing data.

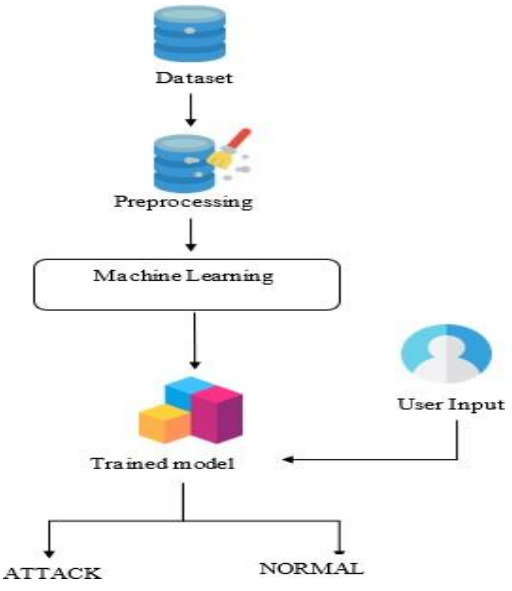


Figure 1: Architecture Diagram

Regularly comparing the effectiveness of multiple machine learning algorithms is required. This will show why it's crucial to create a test harness using scikit-learn to evaluate

different Python machine learning algorithms. Using this test harness as a basis to add additional and other algorithms for comparison, you can construct your own machine learning challenges.

Table 02: Outcome of the proposed model

Model	ACC	R	FAR	Training Time	Testing Time
UNSW-NB15 / IG-CS-PSO / 21 Features					
LR	97.11	97.11	0.291	124.741 s	0.029s
SVM	97.83	97.83	0.221	16,823.152 s	1549.362 s
DT	98.12	98.12	0.167	88.021 s	0.014 s
XGBoost	98.38	98.38	0.162	2360.165 s	0.530 s
RF	98.39	98.39	0.046	328.511 s	0.361 s
Kyoto / IG-CS-PSO / 10 Features					
LR	93.82	93.82	9.101	15.758 s	0.006 s
SVM	95.96	95.96	5.229	11,609.284 s	823.682 s
DT	98.97	98.97	0.082	8.777	0.024
XGBoost	99.19	99.19	0.061	134.846	0.498
RF	99.25	99.25	0.008	64.273	0.386 s

The performance characteristics of each model willvary. You can find out how accurate each model might be on unidentified data by using resampling techniques like cross validation. From your developed suite of models, it must be able to select one or two of the best models using these estimations.

When working with a fresh dataset, it's a good idea to use many ways to evaluate and analyze the data from different perspectives. The same logic appliesto model selection. You should assess the estimatedaccuracy of each machine learning algorithm using a variety of methods in order to decide which algorithm—or algorithms—to use in the end. To show the variance, average accuracy, and other features of the model accuracy distribution, a number of visualization techniques can be applied.The next section goes over exactly how to use scikit-learn to do that in Python. An important first step in facilitating an equitable comparison of machine learning algorithms is to enforce standardization in test harness evaluation. To achieve this criteria, all algorithms must be evaluated consistently using thesame set of data.

Extra Tree Algorithm:

Tree-based models have gained popularity during the past 10 years, mostly because of their resilience. Tree-based models can be used to non-normally distributed data as well as any type of continuous or categorical data.

They also require little to no data transformation theycan manage issues with dimensions and missing values, for example .Extra Tree is a lesser-known tree-based model, whereas Decision Trees and Random Forest are frequently the models of choice.Extra Tree, an ensemble machine learning technique, works similarly to Random Forests .

Extra Trees may decrease the unfavorable outcome ofthe model by using the full dataset, which is thestandard setting that can be altered. However, bias andvariation are increased by randomly selecting the feature value at which to split. A bias-variance study of several tree-based models is carried out in the paperthat first proposed the Extra Trees model. The most classification and regression tasks (six were assessed), Extra Trees outperform Random Forest in terms of bias and variance. However, the paper goes on to say that this is because the model incorporates irrelevant attributes due to the randomization process in the extra trees. Thus, Extra Trees get a bias score similar to what happens when features that aren't important are removed, like in a feature selection pre-modelling process.

In terms of computational cost, Extra Trees is much faster than Random Forest. This is because Extra Trees randomly selects the value at which to split features, instead of the greedy algorithm used in Random Forest.

Table 03: Evaluation metrics for several categories within the UNSW-NB15 dataset

CATEGORY	ACC	P	R	F	FAR
Analysis	99.91	55.66	78.57	63.77	0.065
Backdoor	99.93	54.84	73.91	62.96	0.048
DOS	99.80	87.00	81.90	84.37	0.082
Exploits	99.76	93.28	93.03	93.15	0.119
Fuzzers	99.88	94.68	92.74	93.70	0.050
Generic	99.91	99.64	99.37	99.50	0.034
Normal	1.00	1.00	1.00	1.00	0.000
Reconnaissance	99.92	93.41	91.90	92.65	0.037
Shellcode	99.97	56.90	89.19	69.47	0.028
Worms	1.00	66.67	85.71	75.00	0.003

Random Forest is still the recommended ensemble tree-based model; however, XGBoost Models has just joined the competition. Nonetheless, as we have demonstrated from our prior explanation of the differences between Random Forest and Extra Trees, ExtraTrees are helpful, especially when computational cost is a primary consideration. Specifically, when computing costs are an issue during massive feature engineering/feature selection pre-modelling operations, Extra Trees would be a preferable choice over other ensemble tree-based models.

Scikit-learn provides Extra Trees, which can be used to create regression or classification models. Although the classification model will be covered in this lesson,the code may also be used for regression with a few little adjustments (i.e., switching from Extra Trees Classifier to Extra Trees Regressor).

The detailed list of parameters for the Extra Trees Model can be found on The calls out three keyparameters explicitly, with the following statement.*"The parameters K, nmin and M have different effects: K determines the strength of the attribute selection process, nmin the strength of averaging output noise, and M the strength of the variance reduction of the ensemble model aggregation."*

Let's look at these parameters more closely from the implementation perspective.In the Scikit-learn manual, K is the max_feature. and describes the quantity of features that each decision node must take into account. More features are taken into account ateach decision node when K is greater,

which reduces the model's bias. Nevertheless, if K is set too large, randomization is diminished, which counteracts the ensemble's effect. nmin is the least number of samples needed to be at a leaf node, and it corresponds to min_sample_leaf. The likelihood of the model overfitting decreases with increasing value. A deeper, more specialized tree and more splits are the outcome of using fewer samples. M represents the number of trees in the forest and maps to n_estimators. The model's variancedecreases as its value increases.

The algorithm has been evaluated using the K-fold cross validation process. In order to guarantee that the same splits to the training data have to be carried out and that each technique is properly evaluated, it is crucial to set it up with the same random seed. To build a machine learning model, install the Scikit-Learn libraries prior to running the comparison technique.

Preprocessing, a decision tree classifier with a tree,a linear model using logistic regression, the K-Fold method for cross-validation, and an ensemble approach using the random forest method are all required by the library package. It's also essential todivide the train set from the test set. comparative precision to predict the result.

IV. MODULE DESCRIPTION

Data Pre-Processing

The error rate of the Machine Learning (ML) model is generated by validation techniques, and it approximates the true dataset error rate as nearly as is practical. If a sufficient amount of representative data is collected, it might not be essential to apply the validation procedures. In real-world situations, it is typical to work with data samples that may not be a true representation of the population in a given dataset. to determine whether the data is of the integer or float data type, and to find any duplicate or missing values. the data sample that provides an unbiased evaluation of a model fit on the training dataset and is used to adjust the model hyperparameters. as the procedure for addressing the composition, content,

The assessment becomes more skewed when more expertise from the validation dataset is included into the model design. Though this is standard procedure, a particular model is evaluated using the validation set. The hyperparameters of the model are modified by machine learning specialists using this data. The act of collecting and analyzing data, together with addressing the organization, content, and quality of the data, can lead to a laborious to-do list. Throughout the data identification process, your knowledge of your data and its properties will be useful in helping you choose the best algorithm for building your model.

The Pandas module in Python can be used for a variety of data cleansing tasks. It is faster at cleaning data and concentrates on missing values—possibly the hardest work in data cleaning. It would prefer to devote more time in research and modeling and less time in data cleaning. A few of the publications have only inconsiderate errors. Sometimes the absence of data could have deeper causes. It is imperative to comprehend the various kinds of missing data from a statistical standpoint.

Here are some typical reasons why data is missing:

- -User forgot to fill in a field.
- -. Data was lost while transferringmanually from a legacy database.
- - There was a programming error.
- - User chose not to fill out a fieldtied to their beliefs about how the results would be used or interpreted.

Data Validation:

Exporting the required dataset and loading library packages. The variable identification needs to be examined using data shape and data type analysis inorder to assess missing and duplicate values. When measuring model skill during model tuning, a validation dataset is a subset of data that was not included in the training phase.

Table 04: Comparing the effectiveness of previous research with the proposed approach based on the FAR for every class in the UNSW-NB15

Class	**[24] DR,2017**	**[12]** DR,2017	**[09]** DR,2018	[20] DR,2019	[34] DR,2019	[36] DR,2020	Our Proposal Detection Rate
Normal	97.38	96.30	70.30	82.00	94.70	100.00	100.00
Analysis	69.83	00.90	17.40	1.34	0.00	12.13	78.57
Backdoor	70.44	02.10	16.00	00.00	33.55	63.94	73.91
DOS	84.81	35.70	69.30	0.44	97.80	12.63	81.90
Exploits	95.61	72.80	60.70	57.14	00.17	89.44	93.03
Fuzzers	87.50	28.90	60.70	40.30	00.00	83.86	92.74
Generic	97.81	97.60	96.50	61.21	57.70	97.33	99.37
Reconnaissances	83.80	80.60	83.70	24.89	04.50	66.61	91.90
Shellcode	58.20	29.10	69.30	00.85	00.00	36.51	89.19
Worms	38.24	75.00	90.90	00.00	00.00	24.64	85.71

It also includes methods for making the best use of testand validation datasets while evaluating the model. Rename the provided dataset, remove a column, or otherwise tidy up the data in order to assess the univariate, bivariate, and multivariate processes. various datasets will require various cleaning techniques. Finding and removing errors and abnormalities is the primary goal of data cleaning in order to enhance the value of data in analytics.

Extra Tree Algorithm:

The ensemble learning technique Extra Trees Classifier is essentially based on decision trees. Similar to Random Forest, Extra Trees Classifier randomizes some choices and data subsets in order toreduce overfitting and overlearning from the data. With two significant exceptions it does not bootstrap data, which means it samples without replacement andit splits nodes based on random splits rather than optimal splits, Extra Trees functions similarly to Random Forest in that it constructs numerous trees and splits nodes using random subsets of features. In conclusion, Extra Trees creates several trees with bootstrap set to False by default, sampling without replacing nodes and splitting them according to random splits between a random subset of the characteristics chosen at each node. Randomness in Extra Trees originates from the randomsplits of all observations rather than from bootstrapping data. The acronym Extra Trees comes from "Extremely Randomized Trees."

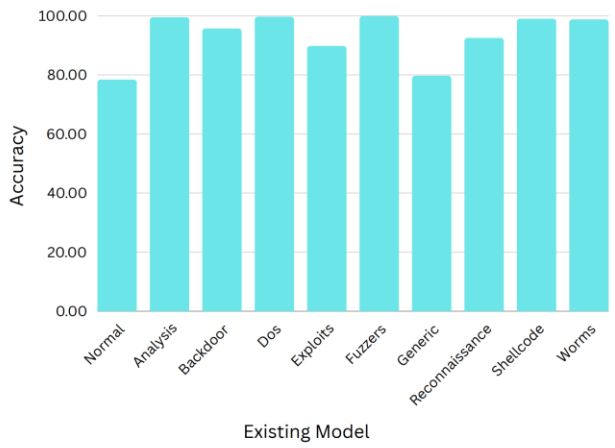


Figure 2: Accuracy in Existing Model

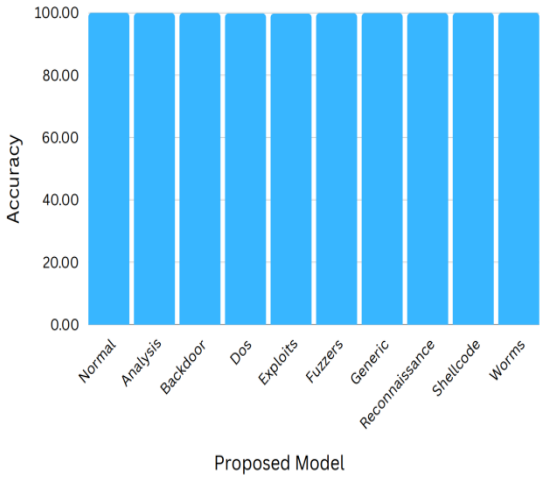


Figure 3: Accuracy in Proposed Model

Table 05: Performance comparison of the proposed model based on the DR for each class in the UNSW-NB15 with previous research

Class	[24]FAR,2017	[42]FAR, 2019	[18]FAR, 2019	Our Proposal
				False Alarm Rate
Normal	0.106	0.001	0.299	0.000
Analysis	0.000	0.789	0.000	0.065
Backdoor	0.000	0.822	0.000	0.048
DOS	0.023	5.478	0.000	0.082
Exploits	0.073	1.391	0.000	0.119
Fuzzers	0.017	1.321	0.000	0.050
Generic	0.005	0.519	0.155	0.034
Reconnaissance	0.004	1.438	0.000	0.037
Shellcode	0.002	0.435	0.000	0.028
Worms	0.000	0.062	0.000	0.003

Decision Trees:

Decision trees serve as the Random Forest Classifier's central component. A decision tree is a straightforward structure that resembles a tree and uses a series of binary choices to categorize data points. A decision based on a particular feature is represented by each internal node of the tree, and a class label in the case .

Random Sub sampling:

Every decision tree in the forest is built using a different random subset of the training set. This process is known as "bagging" or "bootstrap sampling." By doing this, the likelihood of the individual trees getting overfit to the training set is reduced.

Robustness:

Random forests are widely known for their durability and ability to handle complex or noisy information. Since the ensemble of trees reduces variance, they are less prone to overfit than single decision trees.

Ensemble Learning:

A Random Forest is created by assembling many decision trees. Rather than relying just on one decision tree, the algorithm builds many decision trees, each with a slightly different subset of the training data and attributes, for prediction purposes

Accuracy: This metric represents the proportion of cases out of all instances that have been successfully recognized.

$$\text{Accuracy}(ACC) = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision: This measure evaluates the proportion of successfully anticipated assaults compared to the total number of attack instances

$$\text{Precision}(P) = \frac{TP}{TP + FP} \quad (2)$$

Recall: The ratio of cases that were accurately classified as assaults to all instances that were really attacked is described by this metric.

$$\text{Recall}(R) = \text{Detection Rate}\,(DR) = \text{Sensitivity}\,(S) = \text{True Positive Rate}\,(TPR) = \frac{TP}{TP + FN} \quad (3)$$

F-measure: This metric evaluates a system's performance by taking into account both recall and precision.

$$\text{F measure}(F) = \frac{2}{1/\text{Precision} + 1/\text{Recall}} \quad (4)$$

False Alarm Rate: This is the proportion of assault incidents among all real, normal cases that were incorrectly forecasted.

$$FAR = \text{False Positive Rate}\,(FPR) = \frac{FP}{FP + TN} \quad (5)$$

Feature scaling: A technique for normalizing and converting all feature values into a predetermined range is called feature scaling. Since it removes the skewed characteristics of higher numbers, it is a crucial step. Standardization, also known as the Z-score, and normalization, often dubbed min-max scaling and frequently yielding satisfying results, are the two most widely used methods for feature scaling. We used the min-max method, which is described in the following way:

$$X = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (6)$$

where X_{min} and X_{max} show the minimum and maximum values of feature H.

Practical Swarm Optimization PSO

Assume that the variables z_1, z_2, and w represent inertia weight, social learning, and cognitive learning, respectively. Furthermore, Gb_i is the particle i's global position, while Pb_i is the particle i's personal best position. Let n_1 and n_2 be two random numbers. Therefore, the following are the main guidelines for modifying the position and speed of each particle:

$$A_i\,(t+1) = A_i\,(t) + F_i\,(t+1) \quad (7)$$

$$F_i\,(t+1) = wF_i\,(t) + z_1 n_1\,(Pb_i - A_i\,(t)) + z_2 n_2\,(Gbi_i - A_i\,(t)) \quad (8)$$

Feature Importance: Random Forests are a useful tool for understanding the relative importance of different traits since they highlight the features that have the biggest impact on predictions. This might help you choose features or understand the importance of different factors in your dataset.

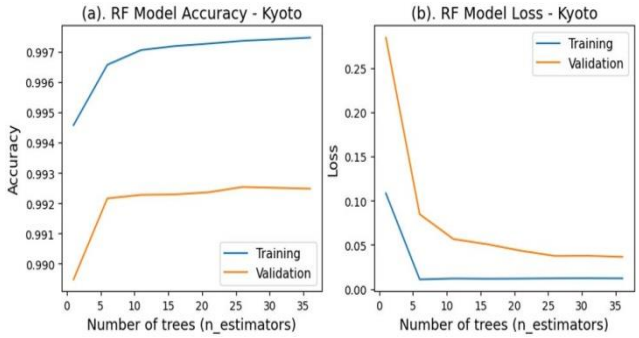


Figure 4 : Model accuracy and loss - UNSW-NB15

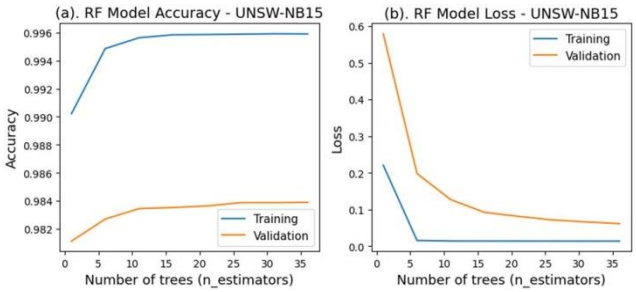


Figure 5:: Model accuracy and loss - Kyoto

V CONCLUSION

Model building and evaluation, missing value analysis, exploratory analysis, and data processing and cleaning were the first steps in the analytical process. When all strategies are contrasted against the various Attack types for future prediction outcomes, the connections with the best performance will yield the highest accuracy score on the public testset. Among these are You can use the following information to determine which newly found connection is the target of a cloud attack. provided a prediction model that can identify changes more accurately and sooner than a person would be able to, with the help of AI. Based on this model, it can be concluded that area analysis and machine learning approaches are helpful in developing prediction models that help cloud sectors shorten waiting times.

REFERENCES

[1] Mhamad Bakro,Rakesh Ranjan Kumar,Amerah Alabrah, "An Improved Design for a Cloud intrusion detection systen I In hybrid an Features Selection Approach with MLalgorithm Classifier",IEEE 29 June 2023.

[2] R. Kumar, A. Tomar, M. Shameem, and M. N Alam, "OPTCLOUD: An optimal cloud with service selection framework using QoS, of the Correlationlens"Comput.Intell.Neurosci.,Vol .2022,pp, 1-16,May 2022.

[3] M.Bakro,R.R Kumar,A.A Alabrah,Z.Ashraf, S.K .Bisoy,"Efficient intrusion detectionsyste In the cloud using fusion feature selection Approaches and an ensemble classifier",vol. 12 No.11,p.2427,May 2023.

[4] I.F Killincer, F.Ertam, A.Sengur, "Machine Le- arning methods for cyber security intrusion detection:Datasets and Comparative Study",vol 188,Apr 2021

[5] Y.Yang,K.Zheng, C.Wu, "Improving the Classi- Fication effectiveness of intrusion detection by Using improved conditional variational Network,vol.19,no.11,p.2528,Jun.2019

[6] M.Rashid , J.Kamruzzaman, T. Imam, S.Wibowo And S.Gordon ," A Tree-based stacking ensem- ble technique with feature selection for network intrusion detection",vol.52,no.9,pp.9768-9781,jul 2022

[7] P.Wei, Y.Li,Z.Zhang,T. Hu,Z.Li and D.Liu," An Optimization method for intrusion Detection Sys Tem model based on deep belief network",Vol.7, pp.87593-87605,2022.

[8] M.H Nasir, S.A. Khan, M.M. Khan, M.Fatima, "Swarm intelligence inspired intrusion detection System-A systematic literature review",vol.205 Mar. 2022.

[9] G.Sreelatha,A.V Babu and D.Midhunchakkarava Thy, "Improved security in cloud using sandpiper And extended equilibrium deep transfer learning Based Intrusion deection",vol.194,May 2022.

[10] P.R.Kanna and P.Santhi, "Hybrid intrusion detec tion using MapReduce based black widow optimiz ed convolutional long short term memory neural networks", vol. 194,May 2022.

[11] J. Zhang, Y. Ling, X. Fu, X. Yang . g, "Model of the intrusion detection system based on the integration of spatial temporal features",vol 89, Feb 2020.

[12] S.M Kasongo and Y.Sun, "Performance analysis Of Intrusion detection system using a Machine feature selection method on the data",vol 89, Feb 2020

[13] S.M. Hosseini Bamakan, H.Wang, and Y.Shi, "Ramp loss k-support vector classification and Regression; a robust and spare multi class appro ch to the system",vol.126,Jun 2017.

[14] V.Hajisalem and S.Babaie, " A hybrid intrusion detection system based on ABC-AFC algorithm for misuse and anamaly detection" ,vol 136,May 2018.

[15] N.Marir,H.Wang,G.Feng,B.Li and M.Jia, "Distri Buted abnormal bahaviour detection approach Based on a deep-belief network and esemble SVM USING Spark",Vol.6,2018.

[16] I.SumaiyaThaseen,J.Saira Banu, "A Integrated Intrusion Detection System using correlation Based attribute selection and artificial neural Network",vol. 32,no. 2,Feb 2021.

[17] C.Khammassi and S.Krichen, " NSGA2 wrapp Er approach for feature selection in network int Rusion detection system using correlation-base d attribute selection and artificial neural networ k,"vol 172,May 2020.

[18] A.S "Almogren,Intrusion detection edge of thi ngs computing",vol 137,pp.259-265,Mar 2020.

[19] J.Gao,S.Chai, B.Zhang , Y.Xia, "Research on Network intrusion detection based on and incremental extreme learning machine and adaptive principal component analysis",Vol.12 no. 7, Mar 2019.

[20] P.Dahiya and D.K Srinivastava,A comparative Evolution of unsupervised tecniques for effect Ive network intrusion detection Hadoop",Vol 906,2018.

[21] P. Mishra, E.S.Pilli, "Out-Vm monitoring for Malicious network packet detection in cloud , Jan 2017.

[22] S.M.Kasongo , Y.Sun",A Deep learning met Hod with wrapper based feature extraction for Wireless intrusion detection system",vol,92 May 2020.

[23] R.K Malaiya,D.Kwon,S.C.Suh,H.Kim, "A emp Irical evaluation of deep learning for network Anomaly detection",vol 7,2019.

[24] Y.Shen,K.Zhen,C.Wu, "A ensemble method Based on selectio using bat algorithm for intr Usion detection", Vol 4,Apr. 2018.

[25] R.Singh,H.Kumar,R.K Singla, " Intrusion dete ction system using network traffic profiling and online sequential extreme learning machine",vol 42,Dec 2015.

[26] R.Chitrakar and C.Huang, "Selection of candidat ESupport vectors in incremental SVM for netw Ork intrusion detection",vol.45, Sep 2014.

[27] R.Vinaya Kumar,M.Alazab , "Deep Learning app Roach for intelligent intrusion detection system Vol 7, pp 41525-41550,2019.

[28] B.A Tama,M.Comuzzi and K.Rhee, "Two stage a Classifier ensemble for intelligent anomaly base Intrusion detection",vol 7,pp.94497-94507,2019.

[29] Z.Ahmad,A.S Khan,C.W Shing, "Network Intrusi On detection system: A systematic study of mach ine learning and deep learning approaches,Vol. 32,Jan 2021.

[30] M.A.Akbar, M.Shameem,S.Mahmood, "Prioritizat Ion based taxonomy of cloud based outsorce Development challenges:Fuzzy AHP analysis," Vol. 12,no 11,p.2427,May 2023.

[31] K.Jiang,W.Wang,A.Wang and H.Wu, "Network Intrusion detection combined hybrid sampling the deep hierarchical network",Vol 8,pp.32464- 32476,2020.

[32] O.Almomani, "A feature selection model for netw ork intrusion detection system combined hybrid Sampling system based on PSO,GWO algorithm" Vol.12,no.6.pp.1-20,2020.

[33] A.I Saleh,F.M Talaat and L.M.Labib, "A hybrid in trusion detection system based on prioritized k-ne arest neighbors and optimized SVM",Vol.51,No.3 Mar 2019.

[34] O.Almomani, "A feature selection model for netw ork intrusion detection system combined hybrid Sampling system based on PSO,GWO algorithm" Vol.12,no.6.pp.1-20,2020.

[35] A.I Saleh,F.M Talaat and L.M.Labib, "A hybrid in trusion detection system based on prioritized k-ne arest neighbors and optimized SVM",Vol.51,No.3 Mar 2019.

[36] I.Benmessahel,K.Xie and M.Chellal, "A New evolutionary neural networks based on intrusion detect ion System using multiverse optimization,Vol 48, no.8, Aug 2018.

[37] R.R Kumar,M.Shameem,R.Khanam,C.Kumar "A Hybrid evaluation framework for QoS based Service selection and ranking in cloud consisting environment." Oct 2018.

[38] M.Bakro, S.K.BISOY, A.K Patel and M.A Naal, "Hybrid blockchain enabled security in cloud using storage infrastructure using ECC and AES algo."2020

[39] M.Bakro,S.KBisoy,A.K.Patel, "Performance Analysis of cloud computing encryption algo rithms",Vol 202,pp.357-367, 2021.

[40] N.Moustafa.J.Slay and G.Creech, "Novel geo Metric area analysis technique for anomaly Detection using trapezoidal area estimation On large-scale networks",IEEE,Vol.5,no 4, Dec 2019.

[41] T.Janarthanan and S.Zargari, "Feature select Ion in UNSW-NB15 and KDDCUO'99 data Sets",,IEEE,Vol 8,Jun 2017.

[42] S.M.Hosseini Bamakan,H.Wang and Y.Shi, "Ramp loss k-support vector classification Regression; a robust and spare multi class Approach to the intrusion detection system" Vol.126,pp 113-126,Jun 2017.

A.5 Plagiarism Report:

Result

This document contains 11% plagiarism. This means that the author has copied datafrom public sources when writing this work.

Analysis

Result	11%
Document title	conference-rak.docx
Content hash	9a624ac66fd9103003029a3871d4877a
Date	2024-03-23 07:09:02
Check time	20 seconds
Character count	10,000
Special character count	18
Word count	1,438
Number of plagiarized words	153

Plagiarism sources

https://discussions.apple.com/thread/254820929

https://in.indeed.com/career-advice/career-development/dear-sir…

https://wac.colostate.edu/repository/writing/guides/businesslett… https://wit-ie.libguides.com/c.php?g=693702&p=4975713

https://www.coursesidekick.com/health-science/2547183

https://stackoverflow.com/questions/9794985/config-error-this-co… https://www.linkedin.com/pulse/machine-process-human-generated-d… https://www.linkedin.com/pulse/anticipate-failure-even-your-test… https://www.trendmicro.com/en_ie/ciso/23/d/hybrid-cloud-security… https://www.edps.europa.eu/data-protection/data-protection/gloss… https://www.citizensinformation.ie/en/employment/employment-righ… https://stackoverflow.com/questions/9990242/alert-dialog-from-th… https://ijisrt.com/assets/upload/files/IJISRT21JUL1121.pdf https://help.twitter.com/en/using-x/create-a-thread

REFERENCES

[1] Mhamad Bakro, Rakesh Ranjan Kumar, Amerah Alabrah, Zubair Ashraf, Md Nadeem Ahmed,Mohammad Shameem, And Ahmed Abdelsalam, “An Improved Design for a Cloud IntrusionDetection System Using Hybrid Features Selection Approach With ML Classifier”, IEEE, 26 June2023.

[2] M. Bakro, R. R. Kumar, S. K. Bisoy, M. O. Addas, and D. Khamis, ‘‘Developing a cloud intrusiondetection system with filter-based features selection techniques and SVM classifier,’’ in Proc. Int.Conf. Comput., Commun. Learn., vol. 1729. Cham, Switzerland: Springer, 2023, pp.15–26, doi:10.1007/978-3-031-21750-0_2.

[3] M. Bakro, R. R. Kumar, A. A. Alabrah, Z. Ashraf, S. K. Bisoy, N. Parveen, S. Khawatmi, andA. Abdelsalam, ‘‘Efficient intrusion detection system in the cloud using fusion feature selectionapproaches and an ensemble classifier,’’ Electronics, vol. 12, no. 11, p. 2427, May 2023,doi:10.3390/electronics12112427.

[4] R. R. Kumar, A. Tomar, M. Shameem, and M. N. Alam, ‘‘OPTCLOUD: An optimal cloud service selection framework using QoS correlation lens,’’ Comput. Intell. Neurosci., vol. 2022,pp. 1–16, May 2022, doi: 10.1155/2022/2019485.

[5] R. R. Kumar, M. Shameem, and C. Kumar, ‘‘A computational framework for ranking predictionof cloud services under fuzzy environment,’’Enterprise Inf. Syst., vol. 16, no. 1, pp. 167–187, Jan. 2022, doi: 10.1080/17517575.2021.1889037.

[6] M. Bakro, S. K. Bisoy, A. K. Patel, and M. A. Naal, ‘‘Hybrid blockchain enabled security in cloudstorage infrastructure using ECC and AES algorithms,’’ in Blockchain based Internet of Things.Singapore: Springer, 2022, pp. 139–170, doi: 10.1007/978-981-16-9260-4_6.

[7] G. Sreelatha, A. V. Babu, and D. Midhunchakkaravarthy, ‘‘Improved security in cloud using sandpiper and extended equilibrium deep transfer learning based intrusion detection,’’ Cluster Comput., vol. 25, no. 5, pp. 3129–3144, Oct. 2022, doi: 10.1007/s10586-021-03516-9.

[8] P. R. Kanna and P. Santhi, ‘‘Hybrid intrusion detection using MapReduce based black widow optimized convolutional long short-term memory neural networks,’’ Expert Syst. Appl., vol.194, May 2022, Art. no. 116545,10.1016/j.eswa.2022.116545.

[9] I. Sumaiya Thaseen, J. Saira Banu, K. Lavanya, M. Rukunuddin Ghalib, and K selection andartificial neural network,’’ Trans. Emerg. Telecommun. Technol., vol.32, no. 2, pp. 1–15, Feb. 2021, doi:10.1002/ett.4014.

[10] H. Zhang, J.-L. Li, X.-M. Liu, and C. Dong, ‘‘Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection,’’ Future Gener. Comput. Syst., vol. 122, pp. 130–143, Sep. 2021, doi:10.1016/j.future.2021.03.024.

[11] J. Zhang, Y. Ling, X. Fu, X. Yang . g, "Model of the intrusion detection system based on the integration of spatial temporal features",vol 89, Feb 2020.

[12] S.M Kasongo and Y.Sun, "Performance analysis of Intrusion detection system using a Machine feature selection method on the data",vol 89,Feb 2020

[13] S.M. Hosseini Bamakan, H.Wang, and Y.Shi , "Ramp loss k-support vector classification and Regression; a robust and spare multi class approch to the system",vol.126,Jun 2017.

[14] V.Hajisalem and S.Babaie, " A hybrid intrusion detection system based on ABC- AFC algorithm for misuse and anamaly detection" ,vol 136,May 2018.

[15] N.Marir,H.Wang,G.Feng,B.Li and M.Jia, "Distrbuted abnormal bahaviour detection approach Based on a deep-belief network and esemble SVM USING Spark",Vol.6,2018.

[16] I.SumaiyaThaseen,J.Saira Banu, "A Integrated Intrusion Detection System using correlation Based attribute selection and artificial neural network",vol. 32,no. 2,Feb 2021.

[17] C.Khammassi and S.Krichen, " NSGA2 wrap Er approach for feature selection in network I Intrusion detection system using correlation-based attribute selection and artificial neural network,"vol 172,May 2020.

[18] A.S "Almogren,Intrusion detection edge of things computing",vol 137,pp.259- 265,Mar 2020.

[19] J.Gao,S.Chai, B.Zhang , Y.Xia, "Research on Network intrusion detection based on and incremental extreme learning machine and adaptive principal component analysis",Vol no. 7, Mar 2019.

[20] P.Dahiya and D.K Srinivastava,A comparative Evolution of unsupervised tecniques for effective network intrusion detection Hadoop",Vol 906,2018.

[21] S.M.Kasongo , Y.Sun",A Deep learning Hod with wrapper based feature extraction for Wireless intrusion detection system",vol,9, May 2020.

[22] H. Zhang,J.Li,X.M , "Multi dimensional feature fusion and stacking ensemble machine forNetwork intrusion detection system",vol .122,Sep 2021