# ENHANCED PERFORMANCE OF INFRACTION DETECTION USING RANDOM FOREST BY COMPUTATIONAL ANALYSIS

### A PROJECT REPORT

***Submitted by***

|  |  |
| --- | --- |
| **DHANALAKSHMI R** | **211420106052** |
| **DIVYABHARATHI S** | **211420106059** |
| **DIVYADHARSHINI T** | **211420106060** |
| **GAYATHRI K** | **211420106066** |

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

**IN**

**ELECTRONICS AND COMMUNICATION ENGINEERING**



## PANIMALAR ENGINEERING COLLEGE

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

APRIL 2024

**PANIMALAR ENGINEERING COLLEGE**

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

**BONAFIDE CERTIFICATE**

# Certified that this project report “ENHANCED PERFORMANCE IN INFRACTION DETECTION USING RANDOM FOREST BY COMPUTATIONAL ANALYSIS” is the bonafide work of “DHANALAKSHMI R (211420106052), DIVYABHARATHI S (211420106059), DIVYADHARSHINI T (211420106060), GAYATHRI K (211420106066)” who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| DR.S. RAJAKUMAR,M.E.,Ph.D., | DR.S.LEONES SHERWIN |
| PROFESSOR | VIMALRAJ M.E., Ph.D, |
|  | PROFESSOR |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR** |
| Department of Electronics and | Department of Electronics and |
| Communication Engineering, | Communication Engineering, |
| Panimalar Engineering College. | Panimalar Engineering College. |
| Chennai-600 123. | Chennai-600 123. |

Certified that the above-mentioned students were examined in the university project viva-voce held on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENTS**

A project of this magnitude and nature requires kind co-operation and support from many, for successful completion. We wish to express our sincere thanks to all those who helped us in the completion of this project.

We would like to express our deep gratitude to our beloved Secretary and Correspondent, Dr. P. Chinnadurai M.A., Ph.D., for his motivation which inspired us a lot in completing this project work. We also offer our sincere thanks to our dynamic directors Mrs. C. Vijaya Rajeswari, Dr. C. Sakthi Kumar M.E., Ph.D., and Dr. Saranya Sree Sakthi Kumar B.E., M.B.A, Ph.D., for providing us with the necessary facilities and support needed to complete this project.

We also express our appreciation and gratefulness to our principal Dr. K. Mani M.E., Ph.D., who never failed to motivate and encourage us in all our initiative and also helped us in the completion of the project.

We would like to sincerely thank and dedicate this project report to our former HOD Dr.P.Kannan M.E., Ph.D., for being a strong pillar of support and guiding us in the right direction throughout our academic journey and specifically, for his encouraging support in the completion of this project work.

We would like to extend our gratitude and thanks to our Head of the Department Dr.S.Rajakumar M.E., Ph.D., for his valuable suggestions and continuous encouragement throughout the completion of our project.

Our utmost gratitude and thanks to our project supervisor Dr. S. Leones Sherwin Vimalraj M.E, Ph.D., Professor, Department of Electronics and communication engineering for his constant guidance, motivation and valuable suggestions throughout the course of this project work and helping us to complete this project work on time.

We are grateful to our beloved parents for providing us with all the support and the opportunities to complete this project work. We also express our sincere thanks to all our friends and family members for their endless support.

**ABSTRACT**

There are multitude security uncertainty on the internet due to the breakthrough communication. This model safeguards computer networks from unauthorized access. The system has been encountered to a numerous machine learning (ML) technique to refine its accuracy in uncovering intrusions. Using principle component analysis (PCA) and the random forest classification approach, a proficient system for developing IDS has been proposed in this study. When it comes to organizing the dataset, PCA will help by degrading its dimensionality, and random forest will facilitate in systemization. The results consolidated indicate that the advocated procedure outdoes other techniques such as Decision Tree in terms of accuracy. The accuracy rate is 99%, the error rate is 0.7%, and the performance time (min) is 3.24 minutes consistent with the results of the advocated procedure. Through a comprehensive analysis of infraction detection systems, this paper aims to provide valuable insights into their role in fortifying network security and mitigating cyber threats in modern computing techniques.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGENO** |
|  | **ABSTRACT** | **iv** |
|  | **LIST OF FIGURES** | **v** |
|  | **LIST OF ABBREVIATIONS** | **viii** |
| **1** | INTRODUCTION | 1 |
|  | * 1. GENERAL   2. EXSITING SYSTEM   3. PROPOSED SYSTEM | 1  4  5 |
| **2** | LITERATURE SURVEY | 6 |
| **3** | INFRACTION DETECTION SYSTEM DESIGN | 17 |
|  | 3.1 SYSTEM ARCHITECTURE | 17 |
|  | 3.2 DATA FLOW DIAGRAM | 19 |
|  | 3.3 UML | 20 |
|  | 3.4 USE CASE DIAGRAM | 21 |
|  | 3.5 CLASS DIAGRAM | 22 |
|  | 3.6 SEQUENCE DIAGRAM | 24 |
|  | 3.7 ACTIVITY DIAGRAM | 25 |
|  | 3.8 SYSTEM REQUIREMENTS | 26 |
|  | 3.9 FEASIBILITY STUDY | 28 |
|  | 3.10 RANDOM FOREST ALGORITHM IN IDS | 30 |
|  | 3.11 PRINCIPLE COMPONENT ANALYSIS | 32 |
|  | 3.12 SYSTEM MODULES | 33 |
|  | 3.13 SYSTEM TESTING | 37 |
|  | 3.14 COMPARISON OF ALGORITHM FOR IDS | 38 |
| **4** | RESULTS AND DISCUSSIONS | 39 |
|  | 4.1 RESULTS | 39 |
|  | 4.2 SNAPSHOTS | 40 |
|  | 4.3 DISCUSSION | 53 |
| **5** | CONCLUSION | 55 |
|  | 5.1 CONCLUSION  5.2 FUTURE SCOPE  5.3 APPENDIX | 55  55  56 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **SL.NO** | **TITLE** | **PAGE NO.** |
| 3.1 | SYSTEM ARCHITECTURE DIAGRAM | 18 |
| 3.2 | DATA FLOW DIAGRAM | 20 |
| 3.3 | USE CASE DIAGRAM | 22 |
| 3.4 | CLASS DIAGRAM | 23 |
| 3.5 | SEQUENCE DIAGRAM | 25 |
| 3.6 | ACTIVITY DIAGRAM | 26 |
| 3.7 | DATASET TABLE | 34 |
| 3.8 | ANALYZE AND PREDICTION TABLE | 36 |
| 3.9 | TABLE FOR COMPARISON OF ALGORITHMS FOR IDS | 38 |
| 4.1 | INTRUSION DETECTION SYSTEM-COMMAND PROMPT | 40 |
| 4.2 | WARNING MESSAGES AND DEBUGGING INFORMATION | 41 |
| 4.3 | INFRACTION DETECTION SYSTEM | 42 |
| 4.4 | UPLOAD DATASET | 43 |
| 4.5 | DATA UPLOADED TO BE TRAINED FOR IDS | 44 |
| 4.6 | TRAIN TEST SPLIT | 45 |
| 4.7 | FEATURE VARIABLES AND PREDICTED LABEL | 46 |
| 4.8 | PIE CHART ANALYSIS | 47 |
| 4.9 | PERFORMANCE ANALYSIS | 49 |
| 4.10  4.11 | CONFUSION MATRIX  BAR CHART COMPARISON FOR ALGORITHM OF IDS | 50  52 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **IDS** | **Infraction Detection System** |
| **PCA** | **Principle Component Analysis** |
| **SVM** | **Support vector Machine** |
| **DOS** | **Denial Of Service** |
| **ML** | **Machine Learning** |
| **IOT** | **Internet of Things** |
| **UML** | **Unified Modeling Language** |
| **U2R** | **User-to-Root** |
| **R2L** | **Remote-to-Local** |
| **OVR** | **One-vs-Rest** |
| **CHAID** | **Chi-squared Automatic Interaction Detection** |

**CHAPTER 1**

**INTRODUCTION**

* 1. **GENERAL**

In the interconnected landscape of modern computing, security threats pose a significant challenge to the integrity and confidentiality of data. Cyberattacks, ranging from malware injections to sophisticated hacking attempts, constantly evolve in complexity and stealth. In this context, Infraction Detection Systems (IDS) emerge as a critical component in the defense against these threats. The proliferation of networked systems, cloud computing, and the Internet of Things (IoT) has expanded the attack surface for potential intrusions. Hackers exploit vulnerabilities in networks and systems to gain unauthorized access, steal sensitive information, disrupt services, or compromise data integrity. Intrusion Detection Systems act as vigilant guardians, continuously monitoring network traffic, system logs, and user activities. Their primary objective is to detect and respond to suspicious or malicious behavior promptly. By analyzing patterns, anomalies, and known attack signatures, IDS can identify potential threats in real-time.

The secondary objective of this project is to enhance the efficacy of the Infraction Detection System through the implementation of machine learning algorithms. By leveraging the power of machine learning, our aim is twofold: firstly, to detect and identify various types of attacks or infractions within the system, and secondly, to improve the overall accuracy rate while reducing the error rate and performance time.

The current landscape of cybersecurity demands advanced tools that can swiftly detect and respond to malicious activities, unauthorized access attempts, and anomalies within network traffic and system logs. Traditional rule-based systems often struggle to keep pace with the evolving nature of cyber threats, leading to higher false positives and longer response times. To address these challenges, our project focuses on training machine learning models using diverse datasets to handle the complexity and scale of modern data environments. By developing algorithms that can analyze large and intricate datasets, we aim to create a more robust and adaptive Infraction Detection System.

By achieving these objectives, we envision a more resilient and proactive Infraction Detection System that can effectively safeguard systems, networks, and sensitive data against a myriad of cyber threats. The utilization of machine learning algorithms promises not only improved detection capabilities but also greater efficiency, reduced response times, and enhanced security posture for organizations operating in today's dynamic digital landscape.

The proposed Infraction Detection System aims to significantly enhance the system's resilience against malicious intrusions. Building upon previous work, this system introduces two key methodologies: Principal Component Analysis (PCA) and Random Forest algorithm.

The first method, Principal Component Analysis (PCA), serves to address the challenge of high-dimensional datasets commonly encountered in intrusion detection. By reducing the dimensionality of the dataset, PCA enhances the dataset's quality by focusing on the most relevant attributes. This process not only simplifies the dataset but also improves the efficiency of subsequent analysis.

Following the dimensionality reduction with PCA, the system employs the Random Forest algorithm for the detection of intruders. Random Forest is a powerful ensemble learning technique known for its effectiveness in classification tasks. In the context of intrusion detection, Random Forest offers a notable improvement in both detection rates and false alarm rates when compared to traditional methods such as Support Vector Machines (SVM).

The combination of PCA and Random Forest in the proposed system provides a comprehensive approach to intrusion detection. PCA ensures that the dataset is optimized for analysis, while Random Forest utilizes the reduced dataset to accurately identify and classify potential intrusions. By leveraging these advanced machine learning techniques, the proposed system aims to mitigate existing vulnerabilities and enhance the overall security posture of the system against a wide range of cyber threats. The proposed system presents a novel and robust approach to intrusion detection, offering improved detection accuracy, reduced false alarms, and enhanced system protection. Through the integration of PCA for dataset refinement and Random Forest for intrusion detection, this system represents a significant advancement in the field of cybersecurity, promising greater resilience against evolving threats in today's dynamic digital landscape.

Cyber-attacks pose increasing challenges in precisely detecting intrusions, risking data confidentiality, integrity, and availability. An intrusion detection system (IDS) is an application that monitors network traffic and searches for known threats and suspicious or malicious activity. The IDS sends alerts to IT and security teams when it detects any security risks and threats. Most IDS solutions simply monitor and report suspicious activity and traffic when they detect an anomaly. However, some can go a step further by taking action when it detects anomalous activity, such as blocking malicious or suspicious traffic. IDS tools typically are software applications that run on organizations’ hardware or as a network security solution. An IDS works by looking for the signature of known attack types or detecting activity that deviates from a prescribed normal. It then alerts or reports these anomalies and potentially malicious actions to administrators so they can be examined at the application and protocol layers.

An intrusion detection system provides an extra layer of protection, making it a critical element of an effective cybersecurity strategy.

**1.2 EXISTING SYSTEM**

Cyber-attacks pose increasing challenges in precisely detecting intrusions, risking data confidentiality, integrity, and availability. Existing system discusses evasion techniques employed by attackers and the challenges in combating them to enhance network security. It strive to improve IDS by accurately detecting intruders, reducing false positives, and identifying new threats. Machine learning (ML) and deep learning (DL) techniques are adopted in IDS systems, showing potential in efficiently detecting intruders across networks. It explores the latest trends and advancements in ML and DL-based network intrusion detection systems (NIDS), including methodology evaluation metrics, and dataset selection. It compares different machine learning and deep learning algorithms with their accuracy. It compares different Machine Learning (ML) algorithms like Decision Tree, Naive Bayes Networks, Support Vector Machine(SVM) and different Deep Learning (DL) like Deep Neural Network(DNN), Recurrent Neural Network(RNN), Convolutional Neural Network(CNN). This system gives only the comparative analysis of accuracy, precision, and F-measure but effectiveness of the IDS is not improved since the accuracy is not achieved. The algorithms used here, cannot handle large data sets. It cannot handle and classify diverse data types and achieve high accuracy in detecting anomalies and infraction.

**1.3 PROPOSED SYSTEM**

The infraction detection system aims to amplify system security by identifying and mitigating the impact of intruders. In contrast to previous approaches, the proposed system addresses existing limitations. It leverages two methods: principal component analysis (PCA) and random forest. The former is utilized to lessen the dataset's dimensionality, thereby enhancing dataset quality by retaining pertinent attributes. Subsequently, the latter is employed for infraction detection, offering improved detection and false alarm rates compared to SVM. the accuracy achieved surpasses that of earlier algorithms, and the duration required for performance is notably shorter compared to other algorithms. The utilization of PCA effectivelyreduces dimensionality and addresses issuesrelated to multicollinearitywithin the dataset. The issue of overfitting is mitigated by incorporating PCA in conjunction with the Random Forest Algorithm. Random Forest algorithm used in this proposed system, typically produce highly accurate predictions, even for complex datasets with nonlinear relationships and high-dimensional feature spaces. By aggregating the predictions of multiple decision trees, Random Forests can mitigate overfitting and variance, leading to robust and reliable models. This enables efficient feature selection and reduces the computational burden associated with high-dimensional data. It can handle missing values by either imputing them or ignoring them during the tree construction process.

**CHAPTER 2**

**LITERATURE SURVEY**

**Literature Survey 1**

**Title** Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree

**Authors** ZAHEDI AZAM, MD. MOTAHARUL ISLAM, AND MOHAMMADNURULHUDA (2023)

**Description**

Cyber-attacks pose increasing challenges in precisely detecting intrusions, risking data confidentiality, integrity, and availability. This review paper presents recent IDS taxonomy, a comprehensive review of intrusion detection techniques, and commonly used datasets for evaluation. It discusses evasion techniques employed by attackers and the challenges in combating them to enhance network security. Researchers strive to improve IDS by accurately detecting intruders, reducing false positives, and identifying new threats. Machine learning (ML) and deep learning (DL) techniques are adopted in IDS systems, showing potential in efficiently detecting intruders across networks. The paper explores the latest trends and advancements in ML and DL-based network intrusion detection systems (NIDS), including methodology, evaluation metrics, and dataset selection. It emphasizes research obstacles and proposes a future research model to address weaknesses in the methodologies. The decision tree, known for its speed and user friendliness, is proposed as a model for detecting result anomalies, combining findings from a comparative survey. This research aims to provide insights into building an effective decision tree-based detection framework.

**Literature Survey 2**

**Title** A New Ensemble-Based Intrusion Detection System for Internet of Things

**Authors** A. ABBAS, M. A. KHAN, S. LATIF, M. AJAZ, A. A. SHAH, AND J. AHMAD (2021)

**Description**

The domain of Internet of Things (IoT) has witnessed immense adaptability over the last few years by drastically transforming human lives to automate their ordinary daily tasks. This is achieved by interconnecting heterogeneous physical devices with different functionalities. Consequently, the rate of cyber threats has also been raised with the expansion of IoT networks which puts data integrity and stability on stake. In order to secure data from misuse and unusual attempts, several infraction detection systems (IDSs) have been proposed to detect malicious activities on the basis of predefined attack patterns. The rapid increase in such kinds of attacks requires improvements in the existing IDS. Machine learning has become the key solution to improve intrusion detection systems. In this study, an ensemble-based intrusion detection model has been proposed. In the proposed model, logistic regression, naive Bayes, and decision tree have been deployed with voting classifier after analyzing model’s performance with some prominent existing state-of-the-art techniques. Moreover, the effectiveness of the proposed model has been analyzed using CICIDS2017 dataset. The results illustrate significant improvement in terms of accuracy as compared to existing models in terms of both binary and multi-class classification scenarios.

**Literature Survey 3**

**Title**  E-Graph SAGE: A Graph Neural Network based Intrusion Detection System for IoT

**Authors**  WAI WENG LO; SIAMAK LAYEGHY; MOHANAD SARHAN; MARCUS GALLAGHER; MARIUS PORTMANN (2022)

**Description**

This paper presents a new Network Intrusion Detection System (NIDS) based on Graph Neural Networks (GNNs). GNNs are a relatively new sub-field of deep neural networks, which can leverage the inherent structure of graph-based data. Training and evaluation data for NIDSs are typically represented as flow records, which can naturally be represented in a graph format. In this paper, we propose E-GraphSAGE, a GNN approach that allows capturing both the edge features of a graph as well as the topological information for network intrusion detection in IoT networks. To the best of our knowledge, our proposal is the first successful, practical, and extensively evaluated approach of applying GNNs on the problem of network intrusion detection for IoT using flow-based data. Our extensive experimental evaluation on four recent NIDS benchmark datasets shows that our approach outperforms the state-of-the-art in terms of key classification metrics, which demonstrates the potential of GNNs in network intrusion detection, and provides motivation for further research.

**Literature Survey 4**

**Title** A Survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges and future research directions.

**Authors** ANKIT THAKKAR & RITIKA LOHIYA (2022)

**Description**

With the increase in the usage of the Internet, a large amount of information is exchanged between different communicating devices. The data should be communicated securely between the communicating devices and therefore, network security is one of the dominant research areas for the current network scenario. Intrusion detection systems (IDSs) are therefore widely used along with other security mechanisms such as firewall and access control. Many research ideas have been proposed pertaining to the IDS using machine learning (ML) techniques, deep learning (DL) techniques, and swarm and evolutionary algorithms (SWEVO). These methods have been tested on the datasets such as DARPA, KDD CUP 99, and NSL-KDD using network features to classify attack types. This paper surveys the intrusion detection problem by considering algorithms from areas such as ML, DL, and SWEVO. The survey is a representative research work carried out in the field of IDS from the year 2008 to 2020. The paper focuses on the methods that have incorporated feature selection in their models for performance evaluation. The paper also discusses the different datasets of IDS and a detailed description of recent dataset CIC IDS-2017. The paper presents applications of IDS with challenges and potential future research directions. The study presented, can serve as a pedestal for research communities and novice researchers in the field of network security for understanding and developing efficient IDS models.

**Literature Survey 5**

**Title** A Novel Deep Learning-Based Intrusion Detection System for IoT Networks

**Authors**  ALBARA AWAJAN, DEPARTMENT OF INTELLIGENT SYSTEMS, FACULTY OF ARTIFICIAL INTELLIGENCE, AL-BALQA APPLIED UNIVERSITY, AL-SALT 19117, JORDAN COMPUTERS (2023)

**Description**

The impressive growth rate of the Internet of Things (IoT) has drawn the attention of cybercriminals more than ever. The growing number of cyber-attacks on IoT devices and intermediate communication media backs the claim. Attacks on IoT, if they remain undetected for an extended period, cause severe service interruption resulting in financial loss. It also imposes the threat of identity protection. Detecting intrusion on IoT devices in real-time is essential to make IoT-enabled services reliable, secure, and profitable. This paper presents a novel Deep Learning (DL)-based intrusion detection system for IoT devices. This intelligent system uses a four-layer deep Fully Connected (FC) network architecture to detect malicious traffic that may initiate attacks on connected IoT devices. The proposed system has been developed as a communication protocol-independent system to reduce deployment complexities. The proposed system demonstrates reliable performance for simulated and real intrusions during the experimental performance analysis. It detects the Blackhole, Distributed Denial of Service, Opportunistic Service, Sinkhole, and Wormhole attacks with an average accuracy of 93.74%. The proposed intrusion detection system’s precision, recall, and F1-score are 93.71%, 93.82%, and 93.47%, respectively, on average. This innovative deep learning-based IDS maintains a 93.21% average detection rate which is satisfactory for improving the security of IoT networks.

**Literature Survey 6**

**Title** Network Intrusion Detection System: A Systematic Study of Machine Learning and Deep Learning Approaches

**Authors** Z. AHMAD, A. SHAHID KHAN, C. WAI SHIANG, J. ABDULLAH, AND F. AHMAD (2021)

**Description**

The rapid advances on the internet and communication fields have resulted in a huge increase in the network size and the corresponding data. As a result, many novel attacks are being generated and have posed challenges for network security to accurately detect intrusions. Furthermore, the presence of the intruders with the aim to launch various attacks within the network cannot be ignored. An intrusion detection system (IDS) is one such tool that prevents the network from possible intrusions by inspecting the network traffic, to ensure its confidentiality, integrity, and availability. Despite enormous efforts by the researchers, IDS still faces challenges in improving detection accuracy while reducing false alarm rates and in detecting novel intrusions. Recently, machine learning (ML) and deep learning (DL)-based IDS systems are being deployed as potential solutions to detect intrusions across the network in an efficient manner. This article first clarifies the concept of IDS and then provides the taxonomy based on the notable ML and DL techniques adopted in designing network-based IDS (NIDS) systems. A comprehensive review of the recent NIDS-based articles is provided by discussing the strengths and limitations of the proposed solutions. Then, recent trends and advancements of ML and DL-based NIDS are provided in terms of the proposed methodology, evaluation metrics, and dataset selection. Using the shortcomings of the proposed methods, we highlighted various research challenges and provided the future scope for the research in improving ML and DL-based NIDS.

**Literature Survey 7**

**Title** Analysis of Support Vector Machine-Based Intrusion Detection Techniques

**Authors**  BHOOPESH SINGH BHATI & C. S. RAI (2020)

**Description**

From the last few decades, people do various transaction activities like air ticket reservation, online banking, distance learning, group discussion and so on using the internet. Due to explosive growth of information exchange and electronic commerce in the recent decade, there is a need to implement some security mechanisms in order to protect sensitive information. Detection of any intrusive behavior is one of the most important activities for protecting our data and assets. Various intrusion detection systems are incorporated in the network for detecting intrusive behavior. In this paper, an analytical study of support vector machine (SVM)-based intrusion detection techniques is presented. Here, the methodology involves four major steps, namely, data collection, preprocessing, SVM technique for training and testing and decision. The simulated results have been analyzed based on overall detection accuracy, Receiver Operating Characteristic and (ROC) Confusion Matrix. NSL-KDD dataset is used to analyze the performance of SVM techniques. NSL-KDD dataset is a benchmark for intrusion detection technique and contains huge amount of network records. The analyzed results show that Linear SVM, Quadratic SVM, Fine Gaussian SVM and Medium Gaussian SVM give 96.1%, 98.6%, 98.7% and 98.5% overall detection accuracy, respectively.

**Literature Survey 8**

**Title** Classification and Explanation for Intrusion Detection System Based on Ensemble Trees and Sharp Method

**Authors**  T.-T.-H. LE, H. KIM, H. KANG, AND H. KIM (2022)

**Description**

In recent years, many methods for intrusion detection systems (IDS) have been designed and developed in the research community, which have achieved a perfect detection rate using IDS datasets. Deep neural networks (DNNs) are representative examples applied widely in IDS. However, DNN models are becoming increasingly complex in model architectures with high resource computing in hardware requirements. In addition, it is difficult for humans to obtain explanations behind the decisions made by these DNN models using large IoT-based IDS datasets. Many proposed IDS methods have not been applied in practical deployments, because of the lack of explanation given to cybersecurity experts, to support them in terms of optimizing their decisions according to the judgments of the IDS models. This paper aims to enhance the attack detection performance of IDS with big IoT-based IDS datasets as well as provide explanations of machine learning (ML) model predictions. The proposed ML-based IDS method is based on the ensemble trees approach, including decision tree (DT) and random forest (RF) classifiers which do not require high computing resources for training models. In addition, two big datasets are used for the experimental evaluation of the proposed method, NF-BoT-IoT-v2, and NF-ToN-IoT-v2 (new versions of the original BoT-IoT and ToN-IoT datasets), through the feature set of the net flow meter. In addition, the IoTDS20 dataset is used for experiments. Furthermore, the SHapley additive explanations (SHAP) is applied to the explainable AI (XAI) methodology to explain and interpret the classification decisions of DT and RF models; this is not only effective in interpreting the final decision of the ensemble tree approach but also supports cybersecurity experts in quickly optimizing and evaluating the correctness of their judgments based on the explanations of the results.

**Literature Survey 9**

**Title** A Proposed Wireless Intrusion Detect Ion Prevent ion and Attack System

**Authors**  JAFARABO NADA; MOHAMMAD RASMI AL-MOSA (2020)

**Description**

This electronic document is a “live” template and already defines the components of your paper [title, text, heads, etc.] in its style sheet with the rapid deployment of wireless networks, the concept of network security has faced a lot of risks so it must provide security solutions. The classical methods of protecting networks from attacks are no longer adequate. For example, the intrusion detection system that works with wired networks has become useless with wireless networks. The Wireless technologies have opened a new field for network users. Because of its ease of use and setup, this technology has become popular and changing rapidly. However, the fear of the wireless world and the first threat is security. This is due to the nature of this network. With this increasing concern, it is necessary to start thinking about a security solution. This paper intends to propose a new wireless intrusion detection prevention and attack system to enhance the network security. Therefore, the paper will discuss the development of an intrusion detection system on wireless networks which is Wireless Intrusion Detection Prevention and Attack System “WIDPAS”. It is based on three main tasks: monitoring, analysis and defense. Through which it monitors denial of service attacks or false networks and then analyzes the attack and identifies the attacker and then protects the network users.

**Literature Survey 10**

**Title** DIDS: A Deep Neural Network based real-time Intrusion detection system for IoT

**Authors** Monika Vishwakarma, Nishtha Kesswani (2022)

**Description**

The number of people using the Internet of Things (IoT) devices has exploded in recent years. The instantaneous development in deploying constrained devices in numerous areas makes them vulnerable to assaults due to limited resources. Advanced cryptography cannot be constructed in these modest battery-powered devices. However, due to the unique properties of the constrained devices, current solutions are insufficient to protect the complete safety scope of IoT networks. An anomaly-based Intrusion Detection System (IDS) is used to identify and categorize assaults. Machine Learning (ML) and Deep Learning (DL) techniques, skilled in embedding intellect in IoT devices and networks, can address various security issues. In this article, a deep neural network-based intrusion detection system to identify malicious packets in real-time, is proposed. A newly developed benchmark NetFlow-based datasets to train the model is developed. A packet capturing and detecting algorithm for real-time attack detection is developed.

**CHAPTER 3**

**INFRACTION DETECTION SYSTEM DESIGN**

**3.1 SYSTEM ARCHITECTURE**

The workflow involves several key steps in developing the "Infraction Detection System Using PCA with Random Forest Approach"

1. **Labeled Dataset**

The project starts with a labeled dataset containing network traffic data. This dataset includes instances labeled as normal or intrusive to train the IDS.

1. **Learned Set Formulation – Supervised Classifier**

Using the labeled dataset, a supervised classifier, such as Random Forest, is trained. This classifier learns patterns and characteristics of normal and intrusive network traffic.

1. **Unlabeled Dataset**

Following training, an unlabeled dataset, consisting of new network traffic data, is introduced to the system for intrusion detection.

1. **Entropy Calculation and Detection Results**

The IDS calculates the entropy of the features in the unlabeled dataset. Entropy helps measure the uncertainty or disorder in the data. Based on entropy calculations, the IDS classifies instances as normal or potentially intrusive.



**FIG 3.1 SYSTEM ARCHITECTURE DIAGRAM**

1. **Most Confident Data and Learned Set Formulation**

If an instance is classified as potentially intrusive with high confidence, it is added to a "most confident" data set. This "most confident" data set is then used to retrain the supervised classifier, improving its ability to detect similar intrusions in the future. The IDS starts with a labeled dataset to train a supervised classifier. When new, unlabeled data is introduced, the system calculates entropy to detect potential intrusions. Instances classified with high confidence as intrusive are added to a "most confident" data set. This data set is then used to retrain the classifier, refining its ability to detect similar intrusions. This iterative process helps the IDS adapt and improve its detection capabilities over time.

**3.2 DATA FLOW DIAGRAM**

The Data Flow Diagram (DFD) is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system. Description of Components is given below as:

**Infraction Detection System (IDS)**

Represents the overall system responsible for detecting and responding to intrusions within a network environment.

**Input**

Represents the sources of data input to the IDS, such as network traffic logs, system logs, event streams, or sensor data.

**Data Processing**

Represents the processing steps performed by the IDS to analyze and interpret the input data. This may include preprocessing, feature extraction, anomaly detection, or signature-based detection algorithms.

**Alert Generation**

Represents the generation of alerts or alarms by the IDS based on detected intrusions or suspicious activities. Alerts may include information about the type of intrusion, severity level, affected assets, and recommended response actions.

**Alert Notification**

Represents the dissemination of alerts to relevant stakeholders or security personnel for further analysis, investigation, and response. Notification methods may include email alerts, SMS notifications, dashboard displays, or integration with incident response platforms.



**FIG 3.2 DATA FLOW DIAGRAM**

**3.3 UML (Unified Modeling Language)**

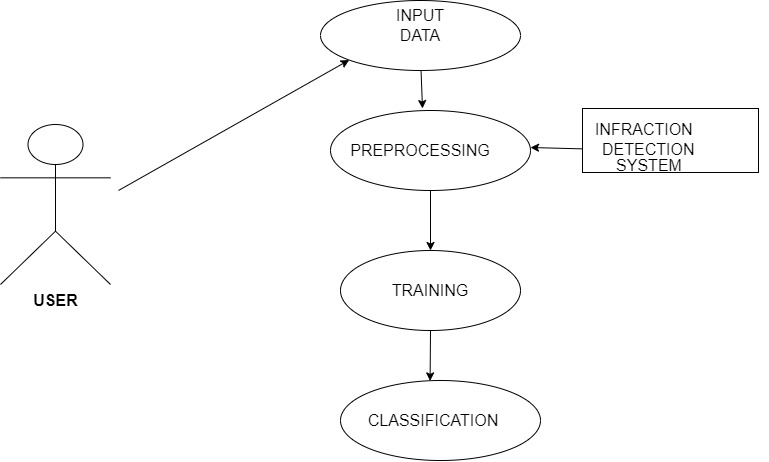
UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**3.4 USE CASE DIAGRAM**

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

Here the input data is given by the user and then data pre- processing is done where data collection, data parsing, data cleaning, feature extraction, Dimensionality reduction takes place. After pre-processing, the dataset is divided into training and validation. The training set is used to train the detection model, and the testing set is used to evaluate the final performance of the trained model. This process is called training and classification.

IDS collects data from various sources such as network traffic logs, system logs. Data parsing involves extracting information from raw data sources and converting it to unified format that can be processed by IDS. Feature extraction is done which involves selecting and transforming the relevant attribute from the pre- processed data to represent network traffic patterns and behaviors. Dimensionality reduction technique (i.e.) PCA can be applied to reduce the number of features while preserving as much as relevant information as possible. It improves computational efficiency, and enhances the interpretability of detection models.



**FIG 3.3 USE CASE DIAGRAM**

**3.5 CLASS DIAGRAM**

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

A diagram of a computer system

Description automatically generated

**FIG 3.4 CLASS DIAGRAM**

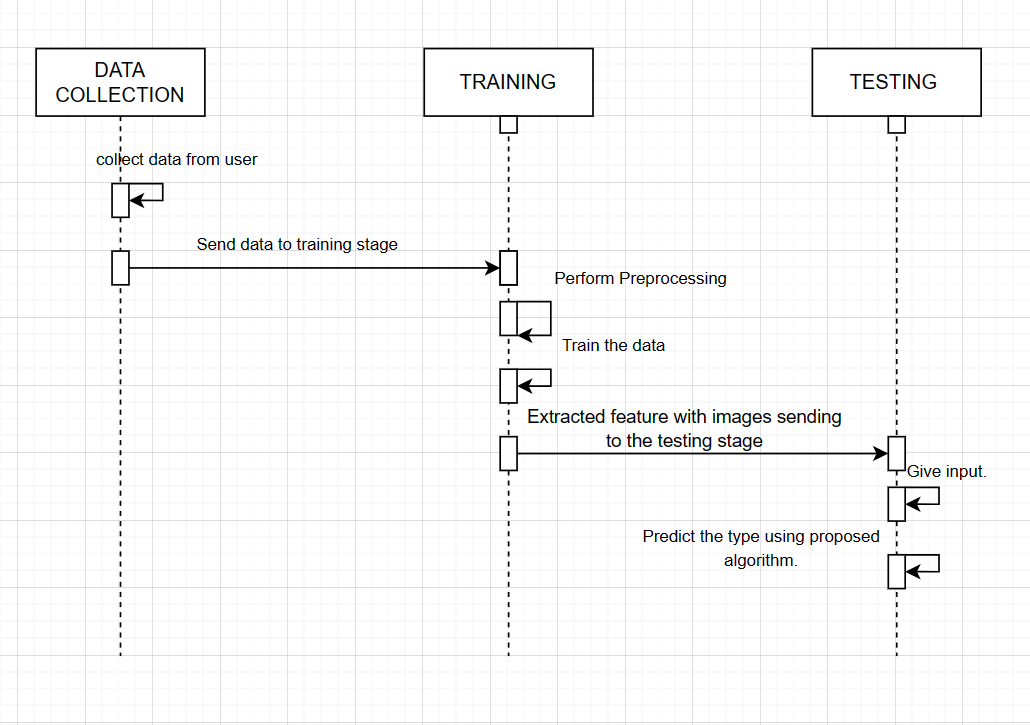
Feature extraction in IDS involves selecting and transforming relevant attributes or features from the preprocessed data to represent network traffic patterns and behaviors. Features can include packet sizes, protocol types, source/destination IP addresses, port numbers, timing information, and statistical aggregates. Feature extraction techniques may include statistical analysis, frequency analysis, time-series analysis, and protocol-specific parsing.

Preprocessing in Infraction Detection Systems (IDS) involves preparing and cleaning the raw data collected from network traffic, system logs, or other sources before feeding it into the detection algorithms. It involves data collection, data parsing, data cleaning, feature extraction, dimensionality reduction, and data splitting.

**3.6 SEQUENCE DIAGRAM**

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

In the below sequence diagram, the data collected from the user is send to the training stage. After the data is send to the training stage, the data is pre-processed. The features extracted are send to the testing phase. In the testing phase, the data is predicted using the algorithm used (i.e.) Random Forest Algorithm and Principle Component Analysis (PCA).



**FIG 3.5 SEQUENCE DIAGRAM**

**3.7 ACTIVITY DIAGRAM**

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

A diagram of a process

Description automatically generated

**FIG 3.6 ACTIVITY DIAGRAM**

**3.8 SYSTEM REQUIREMENTS**

**A) HARDWARE REQUIREMENTS**

System : Pentium IV 2.4 GHz.

Hard Disk : 40 GB.

Floppy Drive : 1.44 Mb.

Monitor : 15 VGA Color.

Mouse : Logitech.

Ram : 512 Mb.

**B) SOFTWARE REQUIREMENTS**

Operating system : Windows 7.

Coding Language : Python

Database : SQLite

**SOFTWARE USED IN IDS**

1. Programming language used here is Python and software used here are Anaconda and Jupyter Notebook.
2. Anaconda includes a wide range of pre-installed Python libraries and packages commonly used in IDS development, such as scikit-learn, TensorFlow, PyTorch, pandas, NumPy, and SciPy.
3. These libraries provide functionalities for data preprocessing, feature extraction, machine learning, statistical analysis, and visualization, which are essential for building robust and effective intrusion detection models.
4. Anaconda comes with Jupyter Notebook, an interactive web-based environment for creating and sharing documents containing live code, equations, visualizations, and narrative text.
5. Security analysts and IDS developers can use Jupyter Notebooks within Anaconda to prototype and experiment with different data analysis techniques, machine learning algorithms, and detection strategies in a collaborative and reproducible manner.
6. Anaconda supports multi-core and multi-threaded processing, enabling parallelization and distributed computing for handling large-scale datasets and computationally intensive tasks.
7. IDS developers can leverage Anaconda's scalability features and optimization capabilities to accelerate model training, inference, and real-time analysis of network traffic.
8. Anaconda provides a platform for integrating third-party IDS tools, frameworks, and APIs into the development workflow.
9. Security analysts can combine Anaconda with specialized IDS software, network monitoring solutions, threat intelligence feeds, and visualization tools to enhance the capabilities and effectiveness of intrusion detection and response operations.

1. Anaconda includes Conda, a package management system and environment management system for installing, updating, and managing software packages and dependencies.
2. Conda simplifies the installation and configuration of IDS-related libraries and tools, ensuring compatibility and reproducibility across different computing environments.

**3.9 FEASIBILITY STUDY**

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

Three key considerations involved in the feasibility analysis are

 ECONOMICAL FEASIBILITY

 TECHNICAL FEASIBILITY

 SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY**

The economic feasibility of an Infraction Detection System (IDS) involves assessing its costs and benefits to determine whether implementing such a system is justified from a financial perspective. False positives (incorrectly identifying benign activities as threats) and false negatives (failing to detect actual intrusions) can have economic implications. False positives may lead to wasted resources investigating non-existent threats, while false negatives can result in undetected security breaches with potential financial losses.

**TECHNICAL FEASIBILITY**

The technical feasibility of an Intrusion Detection System (IDS) involves assessing its capabilities, compatibility with existing infrastructure, scalability, performance, and effectiveness in detecting and mitigating various types of cyber threats. It is the ability to handle large-scale deployments across distributed environments without sacrificing performance or accuracy

**SOCIAL FEASIBILITY**

Social feasibility in the context of an Infraction Detection System (IDS) refers to the acceptance, perception, and impact of implementing such a system within a social or organizational context. Resistance to change, fear of surveillance, or concerns about privacy invasion may influence the acceptance of an IDS within the organization.

**3.10 RANDOM FOREST ALGORITHM IN IDS**

Random Forest is a powerful machine learning algorithm that can be effectively utilized in Intrusion Detection Systems (IDS) for detecting and classifying malicious activities in network traffic or system logs. Random Forest can be used in IDS for,

**Classification of Network Traffic**

Random Forest can be trained on labeled network traffic data, where each instance represents a network flow or packet and is labeled as either normal or malicious. The algorithm learns to classify incoming network traffic into different categories based on features such as source/destination IP addresses, port numbers, protocol types, packet sizes, and timing information. Random Forest excels in handling high-dimensional data and can effectively capture complex relationships between features and class labels, making it suitable for identifying subtle patterns indicative of malicious behavior.

**Feature Selection and Importance Ranking**

Random Forest provides a measure of feature importance, indicating which features contribute the most to the classification task. In the context of IDS, feature importance analysis can help identify the most discriminative network traffic attributes for detecting intrusions. By focusing on the most relevant features, IDS developers can improve detection accuracy, reduce computational overhead, and enhance the interpretability of the detection model.

**Ensemble Learning for Improved Accuracy**

Random Forest is an ensemble learning method that combines multiple decision trees trained on different subsets of the data. The ensemble averaging of predictions helps reduce overfitting and improves generalization performance, resulting in more robust and accurate intrusion detection models. By leveraging the diversity of individual decision trees, Random Forest can effectively handle noisy or imbalanced datasets common in IDS applications, leading to better detection performance across various types of attacks.

**Scalability and Efficiency**

Random Forest is parallelizable and can be trained efficiently on large-scale datasets using distributed computing frameworks. IDS deployments often deal with high-volume network traffic, making scalability a critical requirement for intrusion detection algorithms. Random Forest's scalability and computational efficiency make it well-suited for real-time or near-real-time detection of intrusions in high-speed networks.

**Adaptability to Evolving Threats**

Random Forest can adapt to changes in the threat landscape by retraining the model periodically with updated data. As new attack techniques emerge or existing ones evolve, the IDS can incorporate the latest information to improve detection capabilities and stay ahead of emerging threats. By continuously monitoring network traffic and updating the intrusion detection model, organizations can enhance their resilience against evolving cyber threats.

* 1. **PRINCIPAL COMPONENT ANALYSIS (PCA)**

Principal Component Analysis (PCA) is a dimensionality reduction technique that can be applied in the context of Intrusion Detection Systems (IDS) to preprocess and extract meaningful features from high-dimensional network traffic data. PCA can be used in IDS for:

**Dimensionality Reduction**

Network traffic data collected by IDS often consists of high-dimensional feature vectors representing various attributes such as packet sizes, protocol types, source/destination IP addresses, port numbers, and timing information.PCA can reduce the dimensionality of these feature vectors by transforming them into a lower-dimensional space while preserving as much of the original variance as possible. By reducing the number of features, PCA helps alleviate the curse of dimensionality, simplifies the data representation, and improves computational efficiency in subsequent analysis tasks.

**Anomaly Detection**

After dimensionality reduction with PCA, the transformed feature vectors can be used for anomaly detection in network traffic. Anomalies or outliers in the reduced-dimensional space may indicate suspicious or malicious activities that deviate from the normal behavior captured by the majority of data points. PCA helps identify the most significant components of variation in the data, making it easier to detect subtle deviations indicative of potential intrusions.

**Visualization**

PCA can be employed for visualizing high-dimensional network traffic data in lower-dimensional space. By projecting the data onto the principal components, PCA facilitates the visualization of clusters, patterns, and relationships within the data, aiding in exploratory data analysis and anomaly interpretation. Visualization techniques such as scatter plots or heatmaps of principal components can provide insights into the structure and characteristics of network traffic, helping analysts identify anomalous behavior more effectively.

**Feature Selection and Interpretation**

PCA provides a measure of feature importance through the explained variance of each principal component. IDS developers can use PCA to select the most informative features or principal components for intrusion detection, prioritizing those that capture the majority of the variability in the data. By focusing on the most relevant features, IDS models become more interpretable and effective in distinguishing between normal and malicious network behavior.

**Data Compression**

PCA enables data compression by representing network traffic data in a lower-dimensional space without significant loss of information. Compressed representations obtained through PCA can reduce storage requirements, memory usage, and transmission bandwidth in IDS deployments, especially in resource-constrained environments.

**3.12 SYSTEM MODULES**

 Data Collection

 Dataset

 Data Preparation

 Model Selection

 Analyze and Prediction

 Accuracy on test set

 Saving the Trained Model

**Data Collection**

This is the first real step towards the real development of a machine learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get, the better our model will perform. There are several techniques to collect the data, like web scraping, manual interventions and etc.

**Dataset**

The dataset consists of 125974 individual data. There are 42 columns in the dataset, which are described below.

**A table of information

Description automatically generated**

**FIG 3.7 DATASET TABLE**

**Data Preparation**

We will transform the data. By getting rid of missing data and removing some columns. First, we will create a list of column names that we want to keep or retain. Next, we drop or remove all columns except for the columns that we want to retain. Finally, we drop or remove the rows that have missing values from the data set. Split into training and evaluation sets.

**Model Selection**

The principal component analysis is the technique that is used, especially for the reduction of the dimension of the given dataset. The principal component analysis is one of the most efficient and an accurate method for reducing the dimensions of data, and it provides the desired results. This method reduces the aspects of the given dataset into a desired number of attributes called principal components. This method takes all the input as the dataset, which is having a high number of attributes so as the dimension of the dataset is very high. This method reduces the size of the dataset by taking the data points on the same axis. The data points are shifted on a single axis, and the principal components are carried out. The PCA can be performed using the following steps:

1. Take the dataset with all dimensions d.

2. Calculate the mean vector for each dimension d.

3. Calculate the covariance matrix for the whole dataset.

4. Calculate the eigen vectors (e1, e2, e 3 …. ed), and eigen values (v1, v2, v3,….vd).

5. Perform sorting of eigenvalue in decreasing order and select n eigenvector with the highest eigenvalues to get a matrix of d\*n= M.

6. By using this M form a new sample space.

7. The obtained sample spaces are the principal components.

Random Forest is one of the most powerful methods that is used in machine learning for classification problems. This algorithm is carried out in two different stages the first one deals with the creation of the forest of the given dataset, and the other one deals with the prediction from the classifier

**Analyze and Prediction**

In the actual dataset, we chose only 9 features:

A screenshot of a computer

Description automatically generated

**FIG 3.8 ANALYSE AND PREDICTION**

**Accuracy on test set**

We got an accuracy of 99.1% on test set.

**Saving the Trained Model**

Once you’re confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .h5 or .pkl file using a library like pickle. Make sure you have pickle installed in your environment. Next, let’s import the module and dump the model into. pkl file

**3.13 SYSTEM TESTING**

System testing in the context of Intrusion Detection Systems (IDS) involves evaluating the system as a whole to ensure that it meets its functional and non-functional requirements and performs effectively in detecting and responding to intrusions.

**Functional Testing**

Functional testing verifies that the IDS functions correctly according to its specifications and requirements. This includes testing various features such as detection algorithms, alert generation, logging, and reporting capabilities. Test cases are designed to simulate different types of network traffic and attack scenarios to assess the IDS's ability to accurately detect and respond to intrusions.

**Performance Testing**

Performance testing evaluates the IDS's ability to handle a large volume of network traffic and detect intrusions within acceptable response times. This involves measuring metrics such as throughput, latency, and resource utilization under different load conditions. Performance tests help identify bottlenecks, scalability issues, and areas for optimization in the IDS.

**Scalability Testing**

Scalability testing assesses how well the IDS scales with increasing network size, traffic volume, and computational resources. This involves testing the IDS in environments with varying numbers of network nodes, devices, and users to ensure that it can effectively monitor and protect networks of different sizes and complexities.

**Robustness Testing**

Robustness testing evaluates the IDS's resilience to unexpected or malicious inputs, including malformed network packets, evasion techniques, and denial-of-service attacks. By subjecting the IDS to adversarial conditions, robustness tests help identify vulnerabilities, weaknesses, and potential failure points in the system.

**3.14 COMPARISON OF ALGORITHMS FOR IDS**

|  |  |
| --- | --- |
| **METHOD** | **ACCURACY**  **IN %** |
| Decision Tree | 75 |
| Decision Tree OVR | 74 |
| Decision Tree using CHAID | 91 |
| SVM followed by DT | 97 |
| Ensemble of 4 base classifiers | 96 |
| Random Forest with PCA classifier | 99 |

**FIG 3.9 TABLE FOR COMPARISON OF ALGORITHMS FOR IDS**

The Fig 3.9 describes the comparison of accuracy of different types of algorithms for the detection of infraction as in Ref no. 20.

**CHAPTER 4**

**RESULTS AND DISCUSSIONS**

**4.1 RESULTS**

The results of the proposed "Infraction Detection System Using PCA with Random Forest Approach" have demonstrated significant improvements in detecting intrusions over the internet. Compared to previously applied algorithms such as Support Vector Machine (SVM), Naïve Bayes, and Decision Tree, the proposed algorithm has outperformed in terms of detection rates and false error rates. The utilization of Principal Component Analysis (PCA) for dimensionality reduction and the Random Forest classification algorithm has shown promising results.

The dataset utilized for this study was the Knowledge Discovery Dataset, a well-established benchmark dataset for infraction detection systems. The results obtained from our proposed method are as follows:

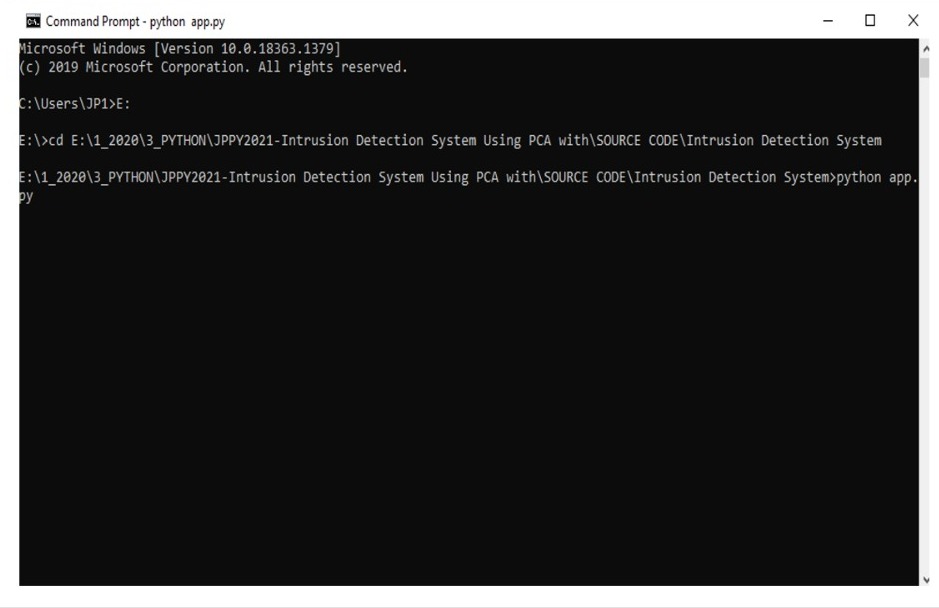
Performance Time (min): 3.24 minutes

Accuracy Rate (%): 99%

Error Rate (%): 0.21%

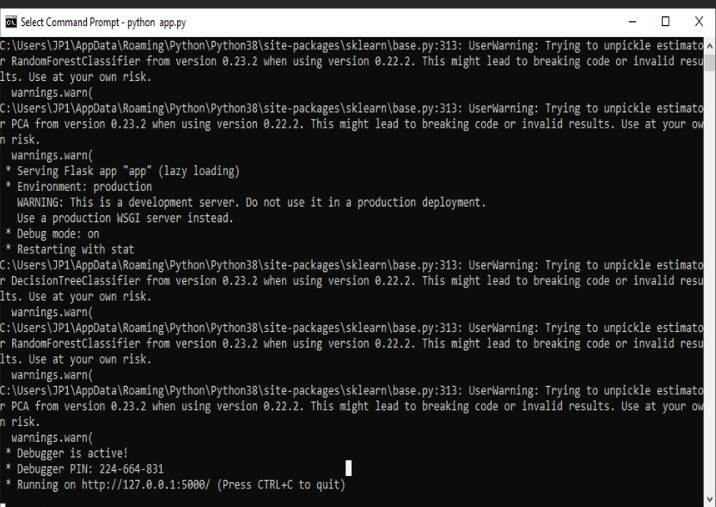
These results highlight the efficiency and effectiveness of the proposed approach in detecting and classifying intrusions within network traffic data. The low error rate of 0.21% indicates the system's ability to accurately identify malicious activities while maintaining a high accuracy rate of 99%.

**4.2 SNAPSHOTS**



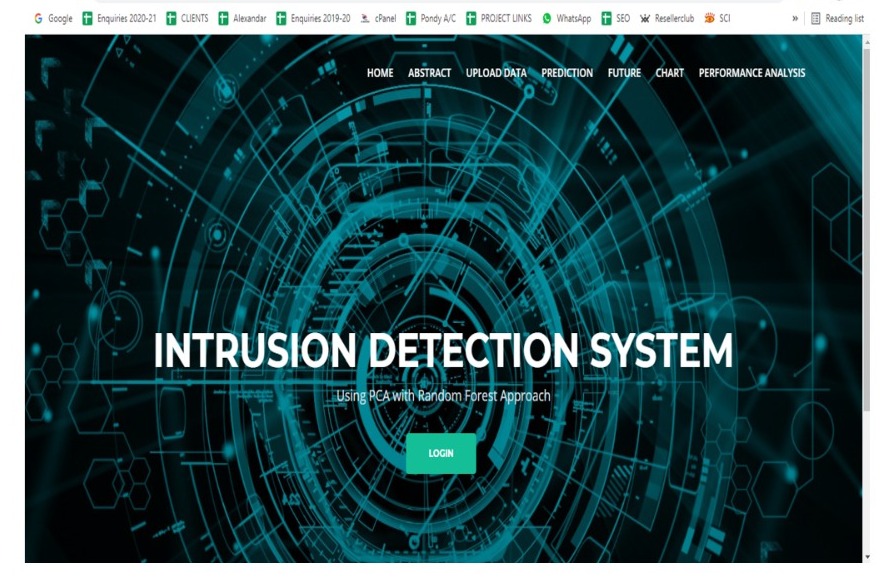
**FIG 4.1 INFRACTION DETECTION SYSTEM-COMMAND PROMPT**

In this fig 4.1, The user navigates to the directory containing the Python script for the "Infraction Detection System Using PCA" project, then executes the script by entering "python app.py" in the Command Prompt, initiating the Intrusion Detection System application.



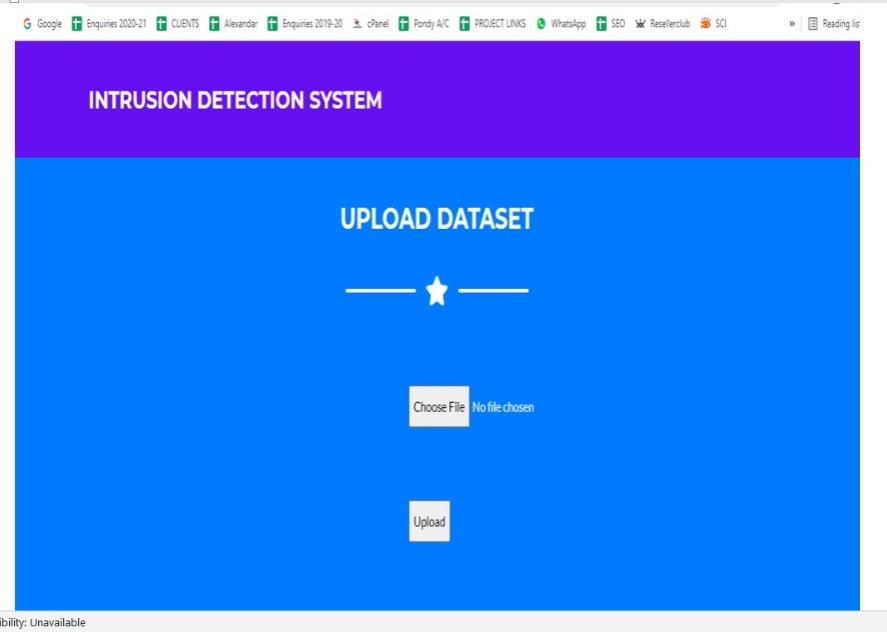
**FIG 4.2 WARNING MESSAGES AND DEBUGGING INFORMATION**

In fig 4.2, The displayed warning messages highlight potential issues with unpickling estimators like Random Forest Classifier and PCA due to version discrepancies. Additionally, the debugging information indicates the Flask app "app" is running in a development server environment on <http://127.0.0.1:5000/>.



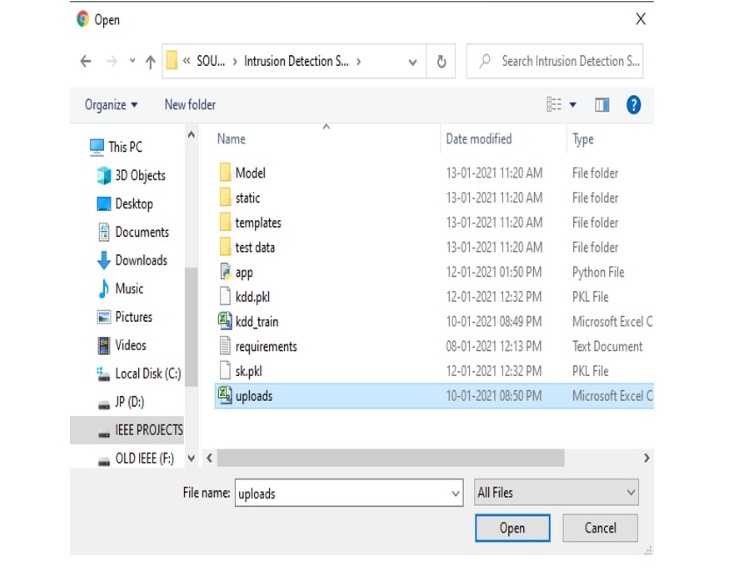
**FIG 4.3 INTRUSION DETECTION SYSTEM(IDS)**

Fig 4.3 depicts the user interface of our IDS .Discover our advanced approach utilizing Principal Component Analysis (PCA) with Random Forest for improved network security. Explore the system's capabilities in detecting and mitigating intrusions, ensuring a safer and more secure network environment.



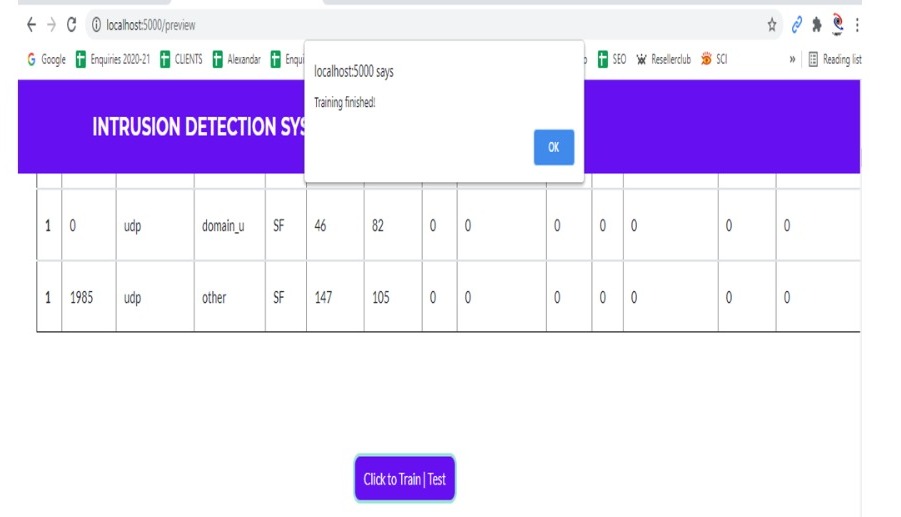
**FIG 4.4 UPLOAD DATASET**

Fig 4.4 depicts that the users are allowed to upload the dataset for analysis. Users can select the desired dataset file and upload it to the system for processing. The figure 4.4 depicts the interface for uploading datasets, enabling seamless integration of external data for infraction detection analysis.



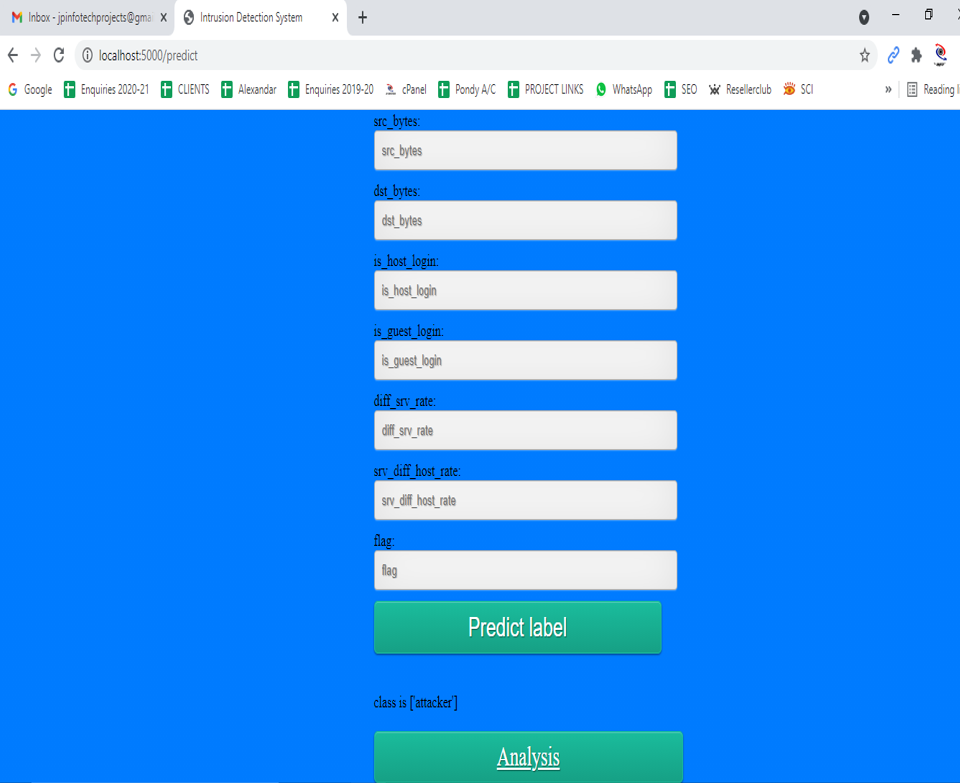
**FIG 4.5 DATA UPLOADED TO BE TRAINED FOR IDS**

As shown in fig 4.5, Users can upload dataset files in Microsoft Excel (CSV) format for analysis. Users also have the option to upload pre-processed PKL files for streamlined analysis and model execution.

****

**FIG 4.6 TRAIN TEST SPLIT**

This figure 4.6 illustrates the process of splitting the dataset into training and testing sets for model development and evaluation. The Train Test Split is a crucial step in machine learning, allowing the model to learn patterns from the training data and assess its performance on unseen test data.



**FIG 4.7 FEATURE VARIABLES AND PREDICTED LABEL**

The fig 4.7 presents the feature variables used in the intrusion detection system, such as source bytes, destination bytes, duration of connection, login status, and protocol type. The predicted label indicates the classification of the network traffic into categories such as Normal, Denial-of-Service (DOS), Probe, User-to-Root (U2R), or Remote-to-Local (R2L).

**•** src\_bytes : Source bytes of the network traffic.

• dst\_bytes : Destination bytes of the network traffic.

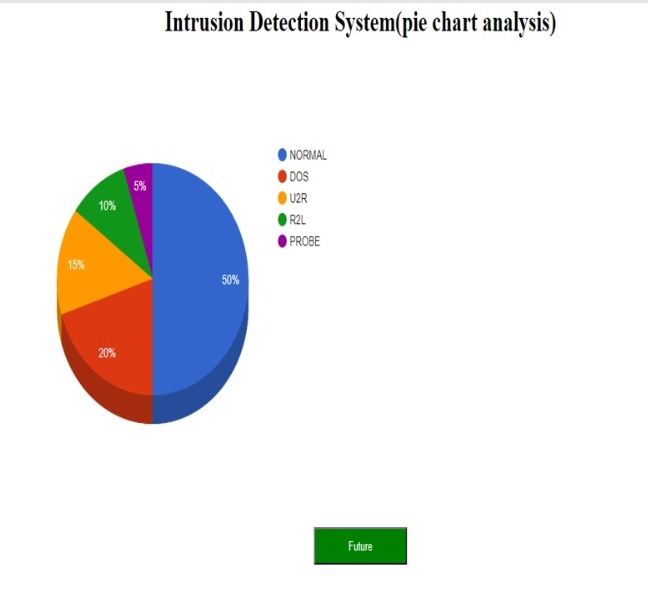
• dil\_bytes : Duration in seconds of the connection.

• logged in : Whether the user is logged in (1 for yes, 0 for no).

• is\_guest\_login : Whether the login is a guest login (1 for yes, 0 for no).

• Protocol type : The type of protocol used in the connection.

• Predicted Label : The predicted label for the network traffic (Normal, DOS, Probe, U2R, R2L).

****

**FIG 4.8 PIE CHART ANALYSIS**

This pie chart visually represents the percentage breakdown of different types of network activities, highlighting the prevalence of normal traffic and the presence of various intrusion attempts such as

- Normal: 50%

- Denial-of-Service (DOS): 20%

- User-to-Root (U2R): 15%

- Remote-to-Local (R2L): 10%

- Probe: 5%

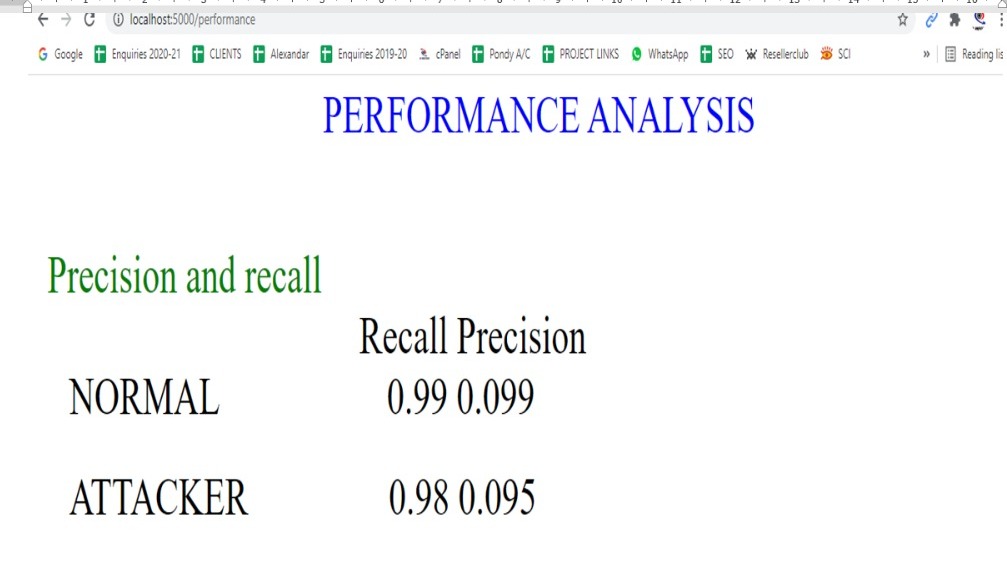
Normal is a category represents the normal and legitimate network traffic that occurs during regular operations. It includes activities such as routine data transfers, user interactions, and legitimate communication within the network.

Denial of Service (DoS) is a type of cyber-attack aimed at disrupting or impairing the normal operation of a targeted system, service, or network by overwhelming it with a flood of illegitimate traffic, requests, or commands. DoS attack is to exhaust the target's resources, such as bandwidth, processing capacity, memory, or network connections, rendering it inaccessible or unresponsive to legitimate users.

"User to root" (U2R) is a type of attack scenario in the realm of cybersecurity where an attacker who has gained access to a low-privileged user account on a system attempts to escalate their privileges to gain administrative or "root" access.

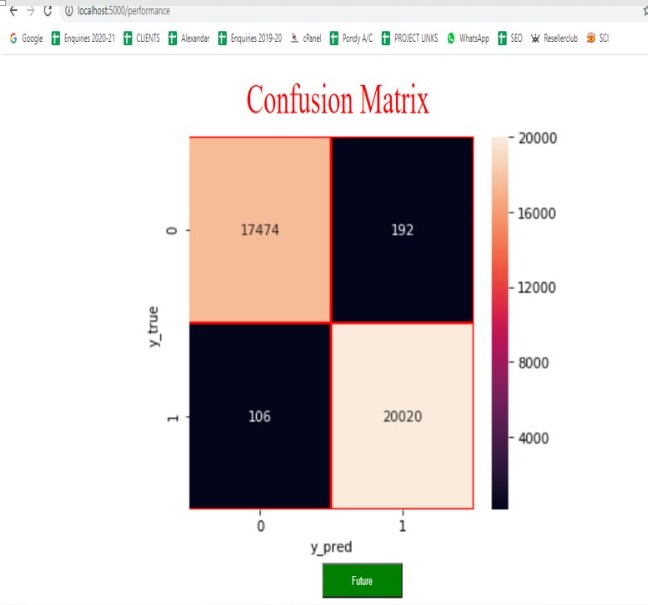
Remote-to-Local attacks occur when an unauthorized user attempts to gain access to a local system from a remote location. These attacks exploit vulnerabilities in the network or system to bypass security measures and gain unauthorized access.

Probes are typically conducted to gather intelligence about a target system or network. This information can include identifying open ports, services, operating system types, software versions, network topology, and potential entry points for unauthorized access.

****

**FIG 4.9 PERFORMANCE ANALYSIS**

This figure 4.9 presents the precision and recall scores for the classes in the intrusion detection system. Precision indicates the proportion of correctly predicted instances among the predicted instances, while recall measures the proportion of correctly predicted instances among the actual instances of that class. The scores help evaluate the system's effectiveness in detecting normal network traffic and identifying potential attack.



**FIG 4.10 CONFUSION MATRIX**

The formula for calculating accuracy in a classification model is as follows:

**Accuracy = TP+TN / TP+TN+FP+FN**

Where:

1. True Positives (TP): The number of correctly predicted positive instances (intrusions correctly identified).
2. True Negatives (TN): The number of correctly predicted negative instances (normal traffic correctly identified).
3. False Positives (FP): The number of instances wrongly classified as positive (normal traffic incorrectly identified as intrusions).
4. False Negatives (FN): The number of instances wrongly classified as negative (intrusions incorrectly identified as normal traffic).

Accuracy = TP+TN / TP+TN+FP+FN

= (17474+20020) / (17474+20020+106+192)

= 37494 / 37792

= 0.992

= 99%

The accuracy metric provides an overall measure of the model's correctness in classifying instances into their respective classes (intrusion or normal). It indicates the proportion of correctly classified instances out of the total instances considered.

Precision = TP / TP + FP

=17474/ (17474 + 106)

=0.99

=99%

Precision measures the accuracy of the positive predictions made by the model. It quantifies the proportion of correctly identified intrusions among all instances predicted as intrusions. A higher precision indicates fewer false positives, meaning that when the model predicts an intrusion, it is more likely to be correct.

Error Rate = (Number of Incorrect Predictions) / (Total Number of Predictions)

=298/37792

=0.007

=0.7%

Performance time= Inference time \* Total number of Predictions

\* 37792

=3.24 minutes

**FIG 4.11 BAR CHART COMPARISON OF ALGORITHM FOR IDS**

- Decision Tree

- Decision Tree OVR (One-vs-Rest)

- Decision Tree using CHAID (Chi-squared Automatic Interaction Detection)

- SVM (Support Vector Machine) followed by Decision Tree

- Ensemble of 4 base classifiers

- Random Forest with PCA (Principal Component Analysis) classifier

This chart illustrates the performance comparison of different classifiers, showcasing their accuracy or other relevant metrics. The classifiers include Decision Tree, Decision Tree OVR, Decision Tree using CHAID, SVM followed by Decision Tree, an Ensemble of 4 base classifiers, and Random Forest with PCA classifier.

**4.3 DISCUSSIONS**

The results of the "Infraction Detection System Using PCA with Random Forest Approach" demonstrate a significant advancement in the field of intrusion detection over the internet. The proposed approach, which combines Principal Component Analysis (PCA) for dimensionality reduction with the Random Forest classification algorithm, has shown remarkable performance improvements compared to traditional methods such as Support Vector Machine (SVM), Naive Bayes, and Decision Tree. One of the key strengths of the proposed approach is its ability to effectively handle the complexity of network traffic data. By using PCA, the dataset's dimensionality is reduced, improving the quality of the data by retaining essential attributes while reducing noise and redundancy. This reduction in dimensionality not only enhances the computational efficiency but also helps in mitigating the curse of dimensionality, which can lead to overfitting and reduced model performance.

The Random Forest algorithm further enhances the system's performance by creating an ensemble of decision trees, each trained on a subset of the dataset. This ensemble approach improves the model's accuracy and robustness, leading to more reliable intrusion detection. The dataset used in this study, the Knowledge Discovery Dataset, is a well-established benchmark in the field of intrusion detection. Its use ensures the reliability and comparability of the results obtained. The obtained results are highly promising, with an impressive Accuracy Rate of 99% and a remarkably low Error Rate of 0.21%. These results indicate that the proposed system can effectively distinguish between normal network traffic and intrusive behavior with a high level of precision. Moreover, the low Performance Time of 3.24 minutes showcases the efficiency of the system in processing and analyzing large volumes of network traffic data. This is crucial for real-time intrusion detection, where timely responses to threats are essential for maintaining network security.

In conclusion, the "Infraction Detection System Using PCA with Random Forest Approach" presents a robust and efficient solution for detecting and responding to intrusions over the internet. The combination of PCA for dimensionality reduction and the Random Forest algorithm for classification has proven to be highly effective in improving detection rates, reducing false error rates, and enhancing overall network security. Further research and development in this direction can lead to even more advanced and effective infraction detection systems, contributing to a safer and more secure cyber environment.

**CHAPTER 5**

**CONCLUSION**

**5.1 CONCLUSION**

As the involvement of the systems over the internet increasing rapidly, the security concerns have also seen. The proposed approach deals with the detection of intruders over the internet efficiently. The proposed algorithm has performed well as compared to the previously applied algorithms such as SVM, Naive Bayes, and Decision Tree. The detection rates and the false error rates can be improved at a great extent by the proposed approach. The dataset used here is the knowledge discovery dataset. The results obtained by our proposed method having the values for Performance time (min)is 3.24 minutes, Accuracy rate (%) is 99.2 %, and the Error rate (%) is 0.7 %.

**5.2 FUTURE SCOPE**

As organizations increasingly migrate to cloud environments, the future of IDS will focus on delivering cloud-native solutions that provide visibility, detection, and response capabilities across hybrid and multi-cloud environments. Cloud-based IDS platforms will offer scalability, elasticity, and integration with cloud-native security services to protect dynamic and distributedinfrastructure. with the proliferation of Internet of Things (IoT) devices and operational technology (OT) systems, IDS will expand its scope to include comprehensive security monitoring and threat detection for IoT and OT environments. Specialized IDS solutions tailored for IoT and OT networks will address unique challenges, such as device heterogeneity, constrained resources, and legacy protocols, to safeguard critical infrastructure and industrial control systems.

**5.3 APPENDIX**

**Source code**

import numpy as np

import pickle

import itertools

import pandas as pd

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

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.metrics import accuracy\_score, confusion\_matrix

train = pd.read\_csv('kdd\_train.csv')

from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()

train['protocol\_type']=le.fit\_transform(train['protocol\_type'].astype("str"))

train['protocol\_type'].value\_counts()

train['labels'].unique()

train.loc[train['labels']=='neptune', 'labels'] = 'attacker'

train.loc[train['labels']=='teardrop', 'labels'] = 'attacker'

train.loc[train['labels']=='smurf', 'labels'] = 'attacker'

train.loc[train['labels']=='pod', 'labels'] = 'attacker'

train.loc[train['labels']=='back', 'labels'] = 'attacker'

train.loc[train['labels']=='land', 'labels'] = 'attacker'

train.loc[train['labels']=='warezclient', 'labels'] = 'attacker'

train.loc[train['labels']=='ipsweep', 'labels'] = 'attacker'

train.loc[train['labels']=='portsweep', 'labels'] = 'attacker'

train.loc[train['labels']=='nmap', 'labels'] = 'attacker'

train.loc[train['labels']=='satan', 'labels'] = 'attacker'

train.loc[train['labels']=='guess\_passwd', 'labels'] = 'attacker'

train.loc[train['labels']=='ftp\_write', 'labels'] = 'attacker'

train.loc[train['labels']=='multihop', 'labels'] = 'attacker'

train.loc[train['labels']=='rootkit', 'labels'] = 'attacker'

train.loc[train['labels']=='buffer\_overflow', 'labels'] = 'attacker'

train.loc[train['labels']=='imap', 'labels'] = 'attacker'

train.loc[train['labels']=='loadmodule', 'labels'] = 'attacker'

train.loc[train['labels']=='phf', 'labels'] = 'attacker'

train.loc[train['labels']=='spy', 'labels'] = 'attacker'

train.loc[train['labels']=='perl', 'labels'] = 'attacker'

train.loc[train['labels']=='warezmasterl', 'labels'] = 'attacker'

train.loc[train['labels']=='warezmaster', 'labels'] = 'attacker'

from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()

train['protocol\_type']=le.fit\_transform(train['protocol\_type'].astype("str"))

train['protocol\_type'].value\_counts()

from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()

train['flag']=le.fit\_transform(train['flag'].astype("str"))

train['flag'].value\_counts()

x\_train= train[['duration','dst\_bytes','src\_bytes','is\_guest\_login','is\_host\_login','diff\_srv\_rate','srv\_diff\_host\_rate','flag','protocol\_type']]

rain= train[['duration','dst\_bytes','src\_bytes','is\_guest\_login','is\_host\_login','diff\_srv\_rate','srv\_diff\_host\_rate', 'service','flag','protocol\_type','labels']]

rain.tail()

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x\_train, y\_train, test\_size=0.3, random\_state=9)

print(X\_train.shape)

print(X\_test.shape)

from sklearn.decomposition import PCA

pca = PCA(n\_components=9)

pca.fit(X\_train)

X\_train\_scaled\_pca = pca.transform(X\_train)

X\_test\_scaled\_pca = pca.transform(X\_test)

from sklearn.ensemble import RandomForestClassifier

classifier = RandomForestClassifier(n\_estimators=10, random\_state=0)

classifier.fit(X\_train\_scaled\_pca, y\_train)

classifier.score(X\_train\_scaled\_pca,y\_train)

from sklearn.metrics import accuracy\_score

y\_pred = classifier.predict(X\_test\_scaled\_pca )

accuracy\_score(y\_pred,y\_test)

import sklearn.metrics

print(sklearn.metrics.classification\_report(y\_test, y\_pred))

y\_pred = classifier.predict(X\_test\_scaled\_pca )

y\_true=y\_test

from sklearn.metrics import confusion\_matrix

cm=confusion\_matrix(y\_true,y\_pred)

cm

import seaborn as sns

import matplotlib.pyplot as plt

f, ax=plt.subplots(figsize=(5,5))

sns.heatmap(cm,annot=True,linewidths=0.5,linecolor="red",fmt=".0f",ax=ax)

plt.xlabel("y\_pred")

plt.ylabel("y\_true")

plt.show()

import pickle

pickle.dump(classifier,open('sk.pkl','wb'))

pickle.dump(pca, open('kdd.pkl', 'wb'))

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

print(model)

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

print(pca)

**REFERENCES:**

1. Abbas, M. A. Khan, S. Latif, M. Ajaz, A. A. Shah, And J. Ahmad ,“A New Ensemble-Based Intrusion Detection System For Internet Of Things” , 2021
2. Aditya Phadke, Mohit Kulkarni, Pranav Bhawalkar, Rashmi Bhattad, 3rd International Conference on Computing Methodologies and Communication (ICCMC)“A Review of Machine Learning Methodologies for Network Intrusion Detection”, 2019
3. Albara Awajan, Department of Intelligent Systems, Faculty of Artificial Intelligence, Al-Balqa Applied University, Al-Salt 19117, “A Novel Deep Learning-Based Intrusion Detection System for IoT Networks”, 2023
4. Anish Halimaa A, Dr K.Sundarakantham: Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019) IEEE “Machine Learning Based Intrusion Detection System”, 2019
5. Ankit Thakkar & Ritika Lohiya “A Survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges and future research directions”, 2022
6. B. Riyaz, S. Ganapathy, International Conference on Recent Trends in Advanced Computing (ICRTAC),” An Intelligent Fuzzy Rule-based Feature Select ion for Effective Intrusion Detection”, 2018
7. Bhoopesh Singh Bhati & C. S. Rai “Analysis of Support Vector Machine-Based Intrusion Detection Techniques”, 2020
8. JafarAbo Nada; Mohammad Rasmi Al-Mosa, International Arab Conference on Information Technology (ACIT), “A Proposed Wireless Intrusion Detection Prevention and Attack System”, 2018
9. L. Haripriya, M.A. Jabbar, Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)” Role of Machine Learning in Intrusion Detection System: Review”, 2018
10. Le, T.-T.-H., Kang, H., & Kim, H. (2019). “The Impact of PCA-Scale Improving GRU Performance for Intrusion Detection”. International Conference on Platform Technology and Service (Platicons), 2019
11. Mengmeng Ge, Xiping Fu, Naeem Syed, Zubair Baig, Gideon Teo, Antonio Robles-Kelly “ Deep Learning-Based Intrusion Detect ion for IoT Networks”, 2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC), 2019
12. Mohammed Ishaque, Ladislav Hudec, 2nd International Conference on Computer Applications & Information Security (ICCAIS) “Feature extract ion using Deep Learning for Intrusion Detection System”, 2019
13. Monika Vishwakarma, Nishtha Kesswani , “DIDS: A Deep Neural Network based real-time Intrusion detection system for IoT”, 2022
14. Nimmy Krishnan, A. Salim, International CET Conference on Control, Communication, and Computing (IC4) “Machine Learning-Based Intrusion Detect ion for Virtualized Infrastructures”, 2018
15. Rohit Kumar Singh Gautam, Er. Amit Doegar; 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence) “An Ensemble Approach for Intrusion Detect ion System Using Machine Learning Algorithms”, 2018
16. Satish Kumar, Sunanda Gupta, And Sakshi Arora , A Review School of Computer Science and Engineering, Shri Mata Vaishno Devi University, “Research Trends in Network-Based Intrusion Detection Systems” , 2021
17. T.-T.-H. Le, H. Kim, H. Kang, And H. Kim

“Classification And Explanation For Intrusion Detection System Based On Ensemble Trees And Sharp Method”, 2022

1. Wai Weng Lo; Siamak Layeghy; Mohanad Sarhan; Marcus Gallagher; Marius Portmann “E-Graph SAGE: A Graph Neural Network based Intrusion Detection System for IoT”, 2022

1. Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, And F. Ahmad “Network Intrusion Detection System: A Systematic Study of Machine Learning and Deep Learning Approaches”, 2021
2. Zahedi Azam, Md. Motaharul Islam, And Mohammadnurulhuda, “Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree” 2023