

DEFENSE MECHANISM FOR PROACTIVE DETECTION OF COMPROMISED SYSTEMS

A PROJECT REPORT

Submitted to

Amrita Vishwa Vidyapeetham

in partial fulfillment for the award of the degree of

**BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND
ENGINEERING (CYBER SECURITY)**

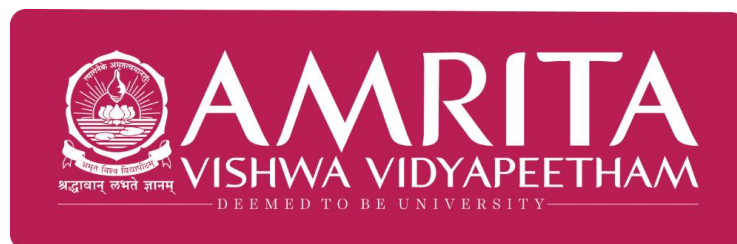
By

PANCHADARLA SHIVA POOJITHA (Reg. No. CH.EN.U4CYS20056)

PODILE SRUTHI (Reg. No. CH.EN.U4CYS20059)

Supervisor

Dr. K. DEEPAK



AMRITA SCHOOL OF COMPUTING, CHENNAI

AMRITA VISHWA VIDYAPEETHAM

CHENNAI – 601103

April 2024



BONAFIDE CERTIFICATE

Certified that this project report **“DEFENSE MECHANISM FOR PROACTIVE DETECTION OF COMPROMISED SYSTEMS”** is the bonafide work of **“PANCHADARLA SHIVA POOJITHA (Reg. No. CH.EN.U4CYS20056)** and **PODILE SRUTHI (Reg. No. CH.EN.U4CYS20059)”**, who carried out the project work under my supervision.

SIGNATURE

Dr. SOUNTHARRAJAN S
CHAIRPERSON

Associate Professor
Department of CSE
Amrita School of Computing
Chennai-601103

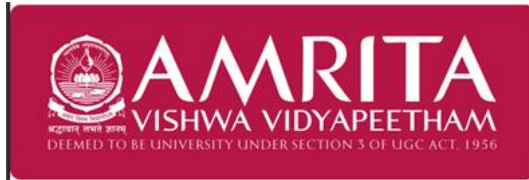
SIGNATURE

Dr. K. DEEPAK
SUPERVISOR

Assistant Professor
Department of CSE
Amrita School of Computing
Chennai-601103

INTERNAL EXAMINER

EXTERNAL EXAMINER



**SCHOOL OF
COMPUTING
CHENNAI**

DECLARATION BY THE CANDIDATES

I declare that the report entitled **“DEFENSE MECHANISM FOR PROACTIVE DETECTION OF COMPROMISED SYSTEMS”** submitted by me for the degree of Bachelor of Technology is the record of the project work carried out by me under the guidance of **“DR. K. DEEPAK”** and this work has not formed the basis for the award of any degree, diploma, associateship, fellowship, titled in this or any other University or other similar institution of higher learning.

SIGNATURE

PODILE SRUTHI

Reg. No. CH.EN.U4CYS20059

Department of CSE (CYS)

SIGNATURE

PANCHADARLA SHIVA POOJITHA

Reg. No. CH.EN.U4CYS20056

Department of CSE (CYS)

ABSTRACT

In the field of Network Intrusion Detection Systems (NIDS), reliable threat identification is critical to guaranteeing network infrastructure security. This study addresses the critical requirement for robust NIDS by utilizing modern machine learning techniques. Specifically, we suggest using Random Forest (RF) and Linear Support Vector Machine (SVM) models to detect and classify network intrusions. To improve model performance, we choose features using Recursive Feature Elimination (RFE) and tune models using Optuna hyperparameter optimization. The datasets chosen for training and evaluation include the US Air Force LAN, which achieves around 98% accuracy, and UNSW-NB15, a well-known collection of network traffic data, which achieves approximately 86% accuracy. Testing, on the other hand, is done with the NSL-KDD dataset and real-time data streams. Our study uses RF and Linear SVM models with adjusted hyperparameters and selected features to obtain outstanding performance in detecting various forms of network intrusions, ensuring the robustness and dependability of NIDS in real-world scenarios.

ACKNOWLEDGEMENT

The satisfaction that accompanies successful completion of any task would be incomplete without mention of people who made it possible, and whose constant encouragement and guidance have been source of inspiration throughout the course of this project work.

We offer our sincere pranams at the lotus feet of “AMMA”, **MATA AMRITANANDAMAYI DEVI** who showered her blessing upon us throughout the course of this project work.

We owe our gratitude to **Dr. V. JAYAKUMAR**, Principal and **Mr. I. B. MANIKANTAN**, Director, Amrita School of Computing, Chennai.

We thank **Dr. SOUNTHARRAJAN S**, Chairperson-CSE and **Dr. A. G. SREEDEVI**, former Program Head-CYS, Amrita School of Computing, Chennai for their support and inspiration.

It is a great pleasure to express our gratitude and indebtedness to our project guide **Dr. K DEEPAK**, Department of Computer Science and Engineering, Amrita School of Computing, Chennai for his valuable guidance, encouragement, moral support, and affection throughout the project work.

Accept my endless gratitude to my Project Coordinator **Dr. S. UDHAYAKUMAR**, Department of Computer Science and Engineering (Cyber Security), Amrita School of Computing, Chennai, for the constant source of inspiration.

We would like to thank express our gratitude to project panel members for their suggestions, encouragement, and moral support during the process of project work and all faculty members for their academic support. Finally, we are forever grateful to our parents, who have loved, supported, and encouraged us in all our endeavors.

PANCHADARLA SHIVA POOJITHA
(Reg. No. CH.EN.U4CYS20056)

PODILE SRUTHI
(Reg. No. CH.EN.U4CYS20059)

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	Abstract	iv
	List of Tables	viii
	List of Figures	ix
	List of Symbols and Abbreviations	x
1	INTRODUCTION	1
	1.1. Introduction	1
	1.1.1. Introduction to Machine Learning	1
	1.1.2. Network Intrusion Detection Systems	1
	1.1.3. Need for Machine Learning for Efficient NIDS	2
	1.2. Importance of NIDS for Assured Security	2
	1.2.1. Network Attacks as a Common Threat	3
	1.2.2. NIDS to Secure a Network	3
2	LITERATURE REVIEW	5
3	PROBLEM STATEMENT AND METHODOLOGY	10
	3.1. Problem Statement	10
	3.1.1. Identification of Problem	10
	3.1.2. Need for an Efficient Solution	11
	3.2. System Design and Methodology	11
	3.2.1. Methodology to Gather and Process Data	12
	3.2.2. System Architecture	12
	3.3. Integration of Recursive Feature Elimination to Train Random Forest and Linear Support Vector Machine Models	13
	3.3.1. RFE Feature Selection	13
	3.3.2. Training RF and SVM Models	14
4	DATASET INFORMATION	16

	4.1. UNSW-NB15	16
	4.2. US Air Force Lan Network Traffic Dataset	16
	4.3. NSL-KDD	17
5	EXPERIMENTAL WORK AND IMPLEMENTATION	18
	5.1. NIDS Application To Identify Network Attacks	18
	5.1.1. Testing the Model Using NSL-KDD	18
	5.1.2. Incorporating the Delete Functionality	20
	5.2. Flask Application For Nids	20
6	RESULTS AND ANALYSIS	24
	6.1. US Air Force Lan - Performance Metric Evaluation	24
	6.2. USNW-NB15 - Performance Metric Evaluation	26
	6.3. Making Predictions Based On Network Traffic Parameters	29
	6.4. Ablation Study	32
7	CONCLUSION AND SCOPE FOR FURTHER WORK	34
	References	35

LIST OF TABLES

TABLE NO.	TITLE	PAGE NO.
6.1	Ablation Study	32

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
3.1	System Architecture and Design	12
5.1	Registration Page	21
5.2	Login Page	21
5.3	Data Analysis Page	22
5.4	Feature Description Page	22
5.5	Network Traffic Input	23
6.1	Model Performance (Random Forest) 1	24
6.2	Classification Report (Random Forest) 1	24
6.3	Confusion Matrix (Random Forest) 1	25
6.4	Model Performance (Linear_SVM) 1	25
6.5	Classification Report (Linear_SVM) 1	25
6.6	Confusion Matrix (Linear_SVM) 1	26
6.7	Model Performance (Random Forest) 2	26
6.8	Classification Report (Random Forest) 2	27
6.9	Confusion Matrix (Random Forest) 2	27
6.10	Model Performance (Linear_SVM) 1	28
6.11	Classification Report (Linear_SVM) 1	28
6.12	Confusion Matrix (Linear_SVM) 1	29
6.13	Normal Traffic	30
6.14	User to Root Attack	30
6.15	DoS Attack	31
6.16	Probe Attack	31
6.17	Remote to Local Attack	32

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

NIDS	-	Network Intrusion Detection System
RF	-	Random Forest
SVM	-	Support Vector Machine
IDS	-	Intrusion Detection System
RFE	-	Recursive Feature Elimination
ML	-	Machine Learning
SIDS	-	Signature-based Intrusion Detection System
AIDS	-	Anomaly-based Intrusion Detection System
DoS	-	Denial of Service
DDoS	-	Distributed Denial of Service
RNN	-	Recurrent Neural Network
DNN	-	Deep Neural Network
Bi-LSTM	-	Bi-directional Long Short Term Memory
APT	-	Advanced Persistent Threats
XSS	-	Cross Site Scripting
PCI DSS	-	Payment Card Industry Data Security Standard
TTP	-	Tactics, Techniques and Procedures
HIPAA	-	Health Insurance Portability and Accountability Act
IoC	-	Indicators of Compromise
KNN	-	K-Nearest Neighbour

CHAPTER 1

INTRODUCTION

1.1. INTRODUCTION

An essential component of contemporary cybersecurity frameworks, a network intrusion detection system (NIDS) acts as a first line of protection against constantly changing cyberthreats. NIDS detects potentially dangerous activity, such as malware infections, unauthorized access, and intrusion attempts, by continually monitoring and analyzing network traffic. NIDS can efficiently identify both known and unknown threats by combining signature-based detection and anomaly detection techniques. This gives enterprises vital information about their network security posture. When a breach is discovered, NIDS quickly notifies security staff, allowing for quick action to control and lessen the impact.

1.1.1. Introduction to Machine Learning

At the forefront of the digital era, machine learning emerges as a disruptive catalyst that changes the way that data is processed, comprehended, and utilized. Based on artificial intelligence, this technology enables computers to learn from large datasets and adjust accordingly, radically changing how decisions are made in a variety of industries. Its ability to identify complex patterns and connections powers a variety of applications, from personalized recommendations on digital platforms to picture and speech recognition. Machine learning stimulates innovation in sectors like healthcare, where it propels diagnostic breakthroughs, and cybersecurity, where it strengthens defenses against ever-evolving threats, by automating tasks, optimizing procedures, and improving resource allocation efficiency. As we begin this basic investigation of machine learning, it is evident that its capacity to promote creativity, effectiveness, and data-driven decision-making has the potential to bring about a significant revolution in our society.

1.1.2. Network Intrusion Detection Systems

A paradigm shift in cybersecurity has occurred with the integration of machine learning (ML) into network intrusion detection systems (NIDS). NIDS can improve their capacity to quickly identify and address sophisticated cyberthreats by utilizing machine learning algorithms. ML gives NIDS the flexibility and capacity to learn from enormous volumes of network traffic data, which helps them identify new attack patterns and abnormalities that

conventional rule-based systems could overlook. This adaptive feature enhances the overall network security posture by reducing false positives and increasing detection accuracy. The combination of NIDS and ML technologies is becoming more and more important in strengthening defenses and protecting digital assets from bad actors as cyber threats continue to change.

1.1.3. Need for Machine Learning for Efficient NIDS

In today's dynamic cybersecurity landscape, the integration of Machine Learning (ML) into Network Intrusion Detection Systems (NIDS) is imperative for bolstering network defenses. ML algorithms excel in identifying patterns and anomalies within extensive datasets, allowing NIDS to detect previously unseen threats with precision. Through continuous learning and adaptation, ML-based NIDS can refine their detection capabilities over time, minimizing false positives and effectively distinguishing between legitimate network traffic and potential attacks. Moreover, ML techniques empower NIDS to handle the complexities of modern network environments, including encrypted traffic analysis and detection of sophisticated evasion techniques employed by adversaries. By harnessing the power of ML, organizations can proactively fortify their network infrastructure, enhancing their resilience against evolving cyber threats.

1.2. IMPORTANCE OF NIDS FOR ASSURED SECURITY

Network Intrusion Detection Systems (NIDS) hold significant importance in modern cybersecurity landscapes, serving as a crucial subject for research papers. They play a pivotal role in defending against diverse cyber threats by continuously monitoring and analyzing network traffic to detect and respond to potential security breaches. Beyond detection, NIDS contribute to preventing data breaches, safeguarding sensitive information, and maintaining network integrity. Additionally, they aid organizations in complying with industry standards and regulations related to data security. Research on NIDS allows exploration of their technological foundations, effectiveness in threat detection, evolving challenges, and integration of advanced techniques like machine learning and artificial intelligence to enhance capabilities. Recognizing the significance of NIDS in protecting digital assets and managing cyber risks is vital for organizations striving to establish robust cybersecurity defenses in an ever-changing threat environment.

1.2.1. Network Attacks as a Common Threat

Network attacks present significant challenges for organizations, leading to a range of detrimental consequences. Among these, data breaches pose a primary concern, where sensitive information like personal data, financial records, or proprietary business data is compromised. The ramifications of a data breach extend beyond financial losses, including reputational damage and a loss of customer trust. Furthermore, network attacks can disrupt vital services through Denial-of-Service (DoS) or Distributed Denial-of-Service (DDoS) attacks, resulting in revenue loss, reduced productivity, and harm to brand reputation. Financial losses are also a significant worry, involving theft of funds, fraudulent transactions, and extortion demands. Apart from immediate financial impacts, organizations encounter indirect costs related to incident response, remediation efforts, and regulatory fines. In essence, network attacks present a multifaceted threat, necessitating robust cybersecurity measures to mitigate risks and safeguard organizational assets.

Securing a network presents a multifaceted challenge, exacerbated by the dynamic and intricate nature of modern IT environments. The ever-changing threat landscape poses a continual stream of complex attacks, spanning from malware and phishing schemes to advanced persistent threats (APTs). Furthermore, the complexity of IT infrastructure, encompassing various operating systems, applications, and interconnected devices, introduces numerous potential vulnerabilities exploitable by attackers. Effectively managing security across this diverse landscape necessitates comprehensive visibility into network traffic, robust endpoint protection measures, stringent access controls, and ongoing monitoring for anomalous activities. Additionally, securing remote access and cloud-based services adds another layer of complexity, demanding effective authentication mechanisms and robust data encryption protocols. Striking a balance between security and usability further complicates the endeavor, as rigorous security measures may at times hinder user workflows and collaboration. Ultimately, securing a network demands a holistic approach, incorporating technological solutions, employee training, routine assessments, and a proactive stance against emerging threats to mitigate risks and safeguard critical assets.

1.2.2. NIDS to Secure a Network

Network Intrusion Detection Systems (NIDS) serve as vital components of contemporary cybersecurity strategies, offering a proactive approach to detecting a broad spectrum of network attacks. Through continuous monitoring of network traffic, NIDS analyze data packets in real-time, utilizing both signature-based and anomaly-based detection techniques to identify potential security threats. Signature-based detection involves comparing network

traffic against known patterns of malicious activity, enabling NIDS to identify and highlight suspicious behavior indicative of common attack vectors like malware infections, SQL injections, and cross-site scripting (XSS) attacks. In contrast, anomaly-based detection focuses on identifying deviations from established baseline behavior, enabling NIDS to detect novel or previously unseen threats, including zero-day exploits and insider attacks. By integrating these detection methods, NIDS offer organizations comprehensive coverage against various threats, empowering them to detect and respond to attacks promptly.

Moreover, NIDS significantly contribute to bolstering organizations' overall security stance by furnishing real-time alerts and actionable insights to security teams. Upon detecting suspicious activity, NIDS generate alerts that can be categorized based on severity and relevance, empowering security personnel to allocate their response efforts efficiently. Additionally, NIDS meticulously capture and log comprehensive information concerning detected threats, encompassing the attack's source and nature, impacted systems, and potential ramifications. This data facilitates post-incident analysis and forensic examination, proving invaluable in comprehending adversaries' tactics, techniques, and procedures (TTPs). Consequently, organizations can refine their security measures and proactively thwart future attacks.

Furthermore, NIDS assist organizations in achieving regulatory compliance and adhering to industry standards for data protection and network security. Various regulatory frameworks, such as the Payment Card Industry Data Security Standard (PCI DSS) and the Health Insurance Portability and Accountability Act (HIPAA), mandate the implementation of intrusion detection systems as part of security protocols. Through NIDS deployment, organizations can showcase their dedication to safeguarding sensitive data and preventing unauthorized access, thus mitigating the risk of facing substantial fines and penalties associated with non-compliance.

In summary, Network Intrusion Detection Systems play a crucial role in protecting organizational assets, identifying network attacks, and mitigating cybersecurity risks. By offering real-time threat detection, actionable insights, and compliance assurance, NIDS empower organizations to defend against ever-changing cyber threats and uphold the integrity, confidentiality, and availability of their networks and data assets. As cyber threats continue to advance and become more complex, the significance of NIDS as a fundamental cybersecurity solution cannot be emphasized enough.

CHAPTER 2

LITERATURE REVIEW

The research [1] explores the growing cybersecurity issues brought on by the speed at which technology is developing and the rise in cyberattacks. There has been a noticeable shift in the use of machine learning (ML) approaches inside intrusion detection systems (IDS) due to the realization that traditional signature-based intrusion detection methods are inadequate for identifying developing threats. The CICIDS2017 dataset, which is renowned for its thorough depiction of benign and attack traffic, is used in the study's evaluation of different machine learning methods in order to determine how well they categorize network traffic and distinguish between harmful and legitimate flows. Significantly, tree-based machine learning techniques such as PART, J48, and random forest perform better, especially when combined with feature selection strategies like CFS and using attributes from Zeek-derived tools. The study demonstrates the potential of decision tree algorithms as workable solutions for real-time threat detection and emphasizes the importance of machine learning (ML) in enhancing the effectiveness and accuracy of intrusion detection systems, particularly in flow-based scenarios.

By providing a taxonomy of contemporary intrusion detection systems (IDS) approaches in his work [2], Khraisat significantly advances the subject by addressing both signature-based intrusion detection systems (SIDS) and anomaly-based intrusion detection systems (AIDS). The study is a significant resource for scholars and practitioners alike by thoroughly examining recent research achievements and identifying regularly used datasets for assessment. It also tackles the important problem of attackers using evasion strategies to get around detection systems, highlighting the continued necessity for research projects focused at reducing such risks and fortifying computer systems against intrusion attempts. In light of changing cyberthreats, this literature review highlights the significance of developing IDS technologies to fortify cybersecurity defences.

The increasing frequency and intricacy of network assaults pose a significant obstacle to modern security infrastructure. Because almost every computing device is connected, there is always a chance of an infiltration, which calls for strong detection methods. Machine learning-based network intrusion detection (NID) systems have become a popular alternative in recent years, able to detect sophisticated attacks that are beyond the scope of conventional signature-based techniques. These systems do have high detection rates, but they also

frequently have elevated false alarm rates, which reduces their total efficacy. The proposed LuNet neural network [3], which mixes recurrent and convolutional architectures, aims to extract temporal and spatial information from network traffic data in order to meet this difficulty. Experimental results show that LuNet outperforms current methods in terms of detection capability and false alarm reduction, highlighting its potential to strengthen network security against dynamic attacks.

Network routers [4] are extremely vulnerable to a variety of malicious assaults, providing a considerable danger due to their critical role in data transfer. Exploiting these vulnerabilities might have serious effects, as routers are critical for routing and managing network traffic. Attackers can use compromised routers to carry out denial-of-service, spying, or man-in-the-middle assaults by modifying, diverting, or dropping packets. Recent research has focused on detecting routers that display erroneous packet forwarding behavior, with the goal of designing protocols capable of identifying and mitigating these hazards. This study advances the area by properly describing the problem and investigating the design space of protocols for detecting malicious router behavior. It also introduces a real protocol that provides practical and cost-effective deployment on a broad scale, helping to ongoing efforts to improve network security against emerging threats to network infrastructure.

The study [5] goes into the ongoing challenge of recognizing and protecting against anomalous attacks that can emerge unexpectedly. Despite substantial research in this area, the dynamic landscape of networks, characterized by new technologies and a plethora of linked devices, continues to provide challenges for intrusion detection. Machine learning appears to be a potential approach to addressing these difficulties, as it is adaptable and versatile enough to support varied network architectures. This paper investigates the challenges of anomaly detection in both traditional and next-generation networks, as well as the application of machine learning approaches within this framework. It provides insights into various machine learning approaches, highlighting their advantages and conducting comparative evaluations of different models, so improving our understanding of successful intrusion detection strategies in modern network environments.

Given the growing complexity and diversity of cyber threats, Intrusion Detection Systems (IDS) play an important role in guaranteeing cybersecurity in today's ICT systems. With the growth of sophisticated attacks and the difficulty in distinguishing between attack types, it is critical to add advanced techniques such as Deep Neural Networks (DNNs) into IDS frameworks. This paper contributes to the field by using DNNs [6] to predict assaults on Network Intrusion Detection Systems (N-IDS). The study compares the efficacy of a DNN

with a learning rate of 0.1 and training it on the KDDCup-'99' dataset to several conventional machine learning algorithms across multiple network designs. The results show that a three-layer DNN outperforms traditional machine learning approaches, showing the promise of deep learning techniques in improving the efficacy of intrusion detection systems in reducing cyber threats.

The vulnerability of ML/DL models to adversarial attacks significantly reduces the security of anomaly-based NIDS. While previous research has mostly focused on feature-space and white-box assaults, which may be ineffective in real-world circumstances, there is a vacuum in understanding realistic gray/black-box traffic-space [7] threats. To remedy this gap, this study conducts a comprehensive examination of such attacks in order to test the resilience of machine learning-based NIDS. It presents a viable, generic, and interpretable attack approach that can evade detection by various NIDSs by leveraging a variety of ML/DL models and feature sets. Furthermore, the study provides a defense mechanism to prevent these threats and demonstrates its efficacy in improving system resilience through extensive experimental testing.

The growing use of cloud technologies emphasizes the critical role of Network Intrusion Detection Systems (NIDS) in fighting against emerging cyber threats. With the increasing volume of network traffic, the effectiveness of NIDS is crucial, particularly in addressing developing attack vectors. Current NIDS systems often rely on either pattern matching or AI/ML-based anomaly detection algorithms, each with their own set of constraints. While pattern matching methods frequently produce large false positive rates, AI/ML-based systems such as KNN and SVM may face issues relating to limited feature sets and accuracy. To address these challenges, this study introduces a new deep learning model that combines Convolutional Neural Networks (CNNs) with Bi-directional Long Short-Term Memory (Bi-LSTM) networks [8] to capture both spatial and temporal characteristics of data. By leveraging publically available datasets such as NSL-KDD and UNSW-NB15, the proposed approach outperforms existing state-of-the-art NIDS based on machine learning and deep learning models.

Creating reliable classifiers from telemetry data to model network traffic is difficult due to the numerous and complex signals that defy perfect human interpretation. Acquiring sufficiently extensive and diverse training data from labeled samples can be prohibitively expensive. This paper approaches the challenge of detecting infected computers by analyzing their HTTP(S) traffic, which is often collected from network sensors like as proxy servers or firewalls, with little human intervention during the model training phase. The suggested

discriminative model makes judgments based on each computer's traffic over a given time period, learning discriminative patterns in traffic routed to specific servers. By relying on human judgments about the general status of the computer (infected or clean) rather than precise labeling, the model achieves excellent accuracy in identifying compromised systems. Implemented as a neural network with a specialized structure reflecting two stacked multi-instance problems [9], the model not only improves accuracy and learns from coarse labels, but it also automatically identifies server types commonly accessed by infected computers, thereby improving its overall threat detection performance.

This study [10] investigates the critical role of Intrusion Detection Systems (IDSs) in cybersecurity, a field riddled with issues such as high false positives. To address this issue, researchers turn to machine learning (ML) techniques. Studies using Recursive Feature Elimination (RFE) have proved its effectiveness in identifying relevant characteristics required for reliable detection. Furthermore, this study highlights the ability of Deep Neural Networks (DNNs) to discriminate between normal and attack traffic in binary classification, whereas Recurrent Neural Networks (RNNs) are well-suited to multi-class classification in IDS. Integrating RFE with DNNs and RNNs has resulted in significant improvements in accuracy and fewer false positives. Using benchmark datasets such as NSL-KDD, this study highlights the potential of ML-based IDSs in improving system security.

In this study [11], the efficiency of Intelligent Intrusion Detection Systems is strongly dependent on the availability of robust datasets. The NSL-KDD dataset is a refined version of the KDD-99 dataset, including high-quality data that accurately represents real-time settings. This research investigates the NSL-KDD dataset using analysis and experimentation to assess the usefulness of various classification methods in detecting network traffic anomalies. Furthermore, the study investigates the relationship between protocols in the network protocol stack and intruder attack patterns, offering insight on their interconnection. Using classification algorithms from the WEKA data mining tool, the study provides vital insights into the complex link between protocols and network threats.

Drawing on the insights offered by this review [12], an experiment is carried out to compare the performance of several machine learning algorithms on the KDD-99 Cup and NSL-KDD datasets. This study intends to address anomaly detection difficulties inside the Intrusion Detection System (IDS), including high false alarm rates and moderate accuracy. The experiment results show the efficacy of various methodologies in terms of accuracy, detection rate, and false alarm rate. By shining light on these findings, this study highlights potential pathways for improving intrusion detection skills and thereby strengthening

cybersecurity measures.

This study [13] emphasizes the efficiency of Intrusion Detection Systems (IDS) in fortifying security by efficiently spotting harmful activity in recent years. Anomaly detection, an important aspect of IDS, faces significant hurdles, including high false alarm rates and mediocre accuracy, which are linked to its failure to accurately detect all attack types. To address these difficulties, this study undertakes an experiment that compares the performance of various machine learning algorithms on the KDD-99 Cup and NSL-KDD datasets. The data reveal which strategy has greater accuracy and detection rate while maintaining a fair false alarm rate. This study gives light on how to improve anomaly detection, which is critical for boosting cybersecurity measures.

CHAPTER 3

PROBLEM STATEMENT AND METHODOLOGY

Within Network Intrusion Detection Systems (NIDS), a substantial obstacle involves effectively discerning legitimate cyber threats from harmless network activities, facilitating swift response and efficient resource allocation. This research presents a comprehensive strategy utilizing advanced machine learning methods to analyze network traffic patterns. Through the utilization of sophisticated algorithms, this strategy ensures precise categorization of intrusion severity levels while mitigating the impact of false positives, thereby augmenting the effectiveness of intrusion detection and response mechanisms in fortifying network security.

3.1. PROBLEM STATEMENT

The importance of early detection of compromises in computing devices, especially in critical information infrastructure, is highlighted by the widespread reliance on Indicators of Compromise (IoCs). However, current methods face limitations when encountering unknown IoCs, exposing a significant gap in non-IoC-based compromise detection strategies. To tackle this challenge, there is a pressing need to explore AI/ML models, harnessing their capabilities to innovate and develop robust techniques for early compromise detection across various devices such as systems, firewalls, routers, and networks. These advancements are essential for bolstering the resilience and security posture of critical infrastructure against evolving cyber threats.

3.1.1. Identification of Problem

In the domain of critical information infrastructure, the absence of Indicators of Compromise (IoCs) poses a significant challenge, prompting the exploration of innovative approaches to detect compromises across various devices. Conventional methods, which rely on IoCs, often struggle in such situations, emphasizing the urgent need for novel AI/ML models. These sophisticated models can proficiently analyze intricate data patterns and anomalies, facilitating robust detection of compromises even without predefined indicators. Through the utilization of advanced machine learning techniques, organizations can establish proactive detection mechanisms capable of identifying emerging threats and potential compromises across a diverse array of devices within critical infrastructure.

Traditional methods of compromise detection face challenges in keeping up with the rapidly evolving threat landscape, where attackers continuously develop new tactics to avoid detection. In this dynamic environment, the limitations of conventional approaches become increasingly evident, emphasizing the need for innovative AI/ML models. These models possess the ability to adapt and evolve alongside emerging threats, offering a more resilient defense mechanism against compromise. By addressing this gap in detection capabilities, organizations can significantly improve their overall security posture, utilizing early warnings provided by AI/ML models to mitigate risks and minimize the impact of undetected compromises on critical information infrastructure.

3.1.2. Need for an Efficient Solution

The increasing complexity of the threat landscape and the absence of predefined Indicators of Compromise (IoCs) highlight the urgent requirement for an effective solution to detect compromises on devices within critical information infrastructure. Traditional methods struggle to adapt to the evolving tactics of cyber attackers, emphasizing the necessity for innovative approaches. Advanced AI/ML models offer potential in delivering robust and adaptable compromise detection capabilities, enabling analysis of intricate data patterns and anomalies. By harnessing these state-of-the-art techniques, organizations can proactively identify and respond to emerging threats, safeguarding critical assets and reducing the risk of undetected compromises. Addressing this critical need is crucial for bolstering the overall security posture and resilience of critical information infrastructure against evolving cyber threats.

In addition to employing AI/ML models, enterprises must prioritize constant monitoring and threat intelligence exchange to boost their cyber defenses. Organizations can obtain significant insights into new threats and vulnerabilities by putting in place effective monitoring methods and cooperating with industry peers and security experts. This coordinated approach provides proactive threat mitigation measures and allows for timely responses to possible security issues. Furthermore, investing in employee training and cybersecurity awareness programs can enable personnel within firms to detect and report suspicious activity, thereby improving the overall resilience of vital information infrastructure against new cyber threats.

3.2. SYSTEM DESIGN AND METHODOLOGY

In the cybersecurity domain, Network Intrusion Detection Systems (NIDS) serve as

crucial defenses against constantly evolving cyber threats, necessitating robust methodologies for their design and implementation. This paper presents a comprehensive approach to NIDS development, focusing on the integration of two machine learning models, Linear Support Vector Machine (SVM) and Random Forest (RF). By utilizing the Air Force LAN dataset for training and the NSL-KDD dataset for testing, our methodology aims to empower a web-based NIDS capable of distinguishing normal network traffic from potentially malicious activities, including Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R), or Probe attacks. Through the amalgamation of these models and datasets, our system strives to provide effective real-time monitoring and classification of network traffic, thereby contributing to enhanced cybersecurity resilience in modern network infrastructures.

3.2.1. Methodology to Gather and Process Data

The methodology for collecting and analyzing data in the construction of a Network Intrusion Detection System (NIDS) begins with getting the NSL-KDD dataset, which is widely regarded as a benchmark dataset in network security. The NSL-KDD dataset is a comprehensive collection of labeled network traffic data that includes both normal activity and examples of various attack types such as Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R), and Probe assaults. This dataset is the primary resource for training and evaluating the effectiveness of the planned NIDS.

Once the NSL-KDD dataset has been received, preparation steps are conducted to prepare the data for further analysis and model training. Data preprocessing consists of numerous fundamental techniques, including data cleaning, normalization, feature selection, and categorical variable encoding. Missing values, outliers, and inconsistencies are addressed using data cleaning techniques, which ensure the dataset's integrity and trustworthiness. Normalization methods are then used to normalize the feature scales, which promotes uniformity and aids in the convergence of machine learning algorithms during training. Furthermore, feature selection approaches are used to discover and keep the most informative features while rejecting redundant or irrelevant ones, resulting in improved model performance and reduced computing complexity. Categorical variables are encoded in numerical representations so that they can be used in model training.

3.2.2. System Architecture

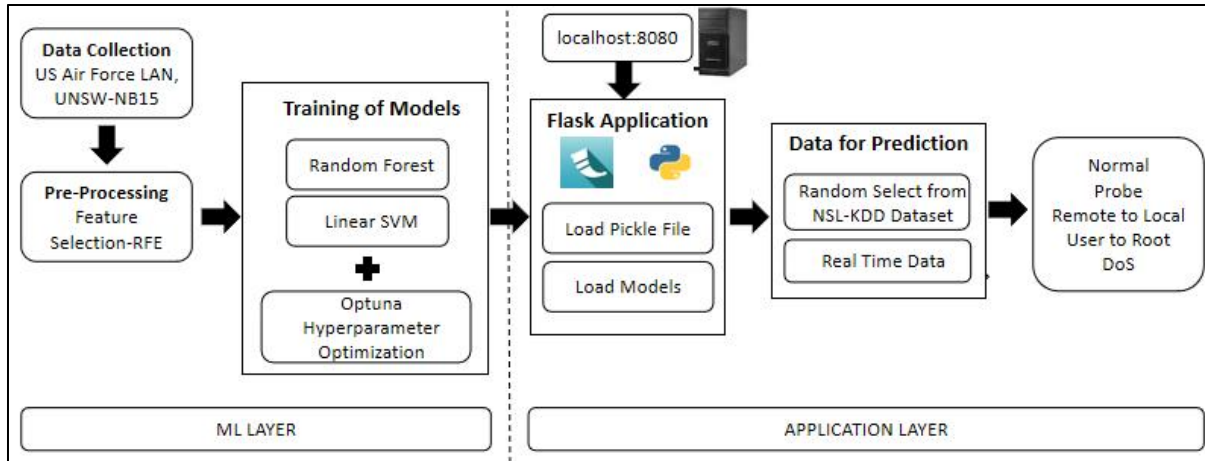


Fig. 3.1. System Architecture and Design

3.3. INTEGRATION OF RECURSIVE FEATURE ELIMINATION TO TRAIN RANDOM FOREST AND LINEAR SUPPORT VECTOR MACHINE MODELS

The integration of Recursive Feature Elimination (RFE) in training both Random Forest and Linear Support Vector Machine (SVM) models, augmented with Optuna hyperparameter optimization, epitomizes a methodical approach to model development. RFE systematically prunes less informative features, enhancing model efficiency. Optuna then fine-tunes hyperparameters to optimize model performance. This disciplined methodology ensures that each model is finely tuned, leveraging the most salient features for classification tasks. By synergizing RFE and Optuna, we strike a balance between feature relevance and parameter optimization, culminating in robust and accurate predictive models suited for diverse real-world applications.

3.3.1. RFE Feature Selection

The feature selection process employed Recursive Feature Elimination (RFE), a methodical technique aimed at identifying the most salient features for our predictive modeling endeavor. This approach is pivotal in enhancing the efficiency and interpretability of our model, ensuring that it is well-attuned to the underlying data dynamics.

Through RFE, we systematically pruned the feature set, iteratively removing less informative attributes while retaining those that contribute most significantly to the predictive power of our model. By iteratively training the model on subsets of features and assessing their impact on performance, we were able to discern a subset of features that best capture the essential patterns within the data.

This disciplined approach to feature selection not only streamlines the modeling process

but also mitigates the risk of overfitting by focusing on the most pertinent features. By prioritizing feature relevance over sheer quantity, we aim to construct a model that is both parsimonious and robust, capable of generalizing effectively to unseen data.

Furthermore, the utilization of RFE underscores our commitment to methodological rigor and empirical validation, as we seek to develop a model that is not only accurate but also interpretable and actionable. Through this principled approach to feature selection, we endeavor to distill the essence of our data into a concise yet informative set of features, laying the groundwork for a robust and reliable predictive model.

3.3.2. Training RF and SVM Models

The training process encompassed the development of both Random Forest and Linear Support Vector Machine (SVM) models, each tailored to address the classification task at hand. A systematic approach was adopted to ensure the effectiveness and reliability of both models, with due consideration given to data preprocessing, feature selection, model training, optimization, and evaluation. The datasets utilized, comprising Air Force LAN network traffic and UNSW-NB15, were meticulously examined for missing values and duplicates. Following this, categorical variables underwent label encoding to render them compatible with machine learning algorithms. Eliminating the 'id' column, which held no predictive value, ensured dataset cleanliness and model efficiency.

The initial phase involved loading the dataset from CSV files containing both training and testing data segments. To prepare the data for modeling, categorical variables underwent label encoding, facilitating the models' interpretation and learning from the data. This preprocessing step laid the foundation for subsequent modeling tasks.

Feature selection was conducted using Recursive Feature Elimination (RFE), a technique employed to identify the most relevant features for predictive modeling. The selected features were standardized to ensure consistency in scale across the dataset, thereby optimizing model performance and interpretability. With the dataset prepared and features selected, the models underwent training on the training dataset. For the Random Forest model, an ensemble learning technique known for its robustness in classification tasks, the training process focused on optimizing hyperparameters to maximize performance. Similarly, the Linear SVM model, known for its effectiveness in linearly separable data, underwent training with a focus on parameter tuning to enhance classification accuracy.

The optimization process was facilitated by Optuna, a hyperparameter optimization framework, which systematically explored the hyperparameter space to identify the optimal

configuration for each model. The best set of hyperparameters obtained through this process was utilized for subsequent model training, ensuring that both models were fine-tuned for optimal performance. Following training, comprehensive evaluation was conducted on both the training and testing datasets to assess the models' efficacy. Evaluation metrics such as accuracy, precision, recall, and F1 score were computed to quantify performance, providing insights into each model's predictive capabilities. Additionally, confusion matrices and classification reports were generated to offer deeper insights into model performance and potential areas for improvement.

To sum up, the Random Forest (RF) and Linear Support Vector Machine (SVM) models were trained with great care, using accepted best practices in machine learning. The datasets were first thoroughly cleaned, normalized, and feature encoded as part of the extensive data preprocessing that was done to get them ready for model training. Model performance and computational efficiency were optimized by selecting the most informative features through the use of techniques like Recursive Feature Elimination (RFE). Differentiated network traffic scenarios were trained using the US Air Force LAN and UNSW-NB15 datasets, and model parameters were optimized for improved performance using hyperparameter optimization via Optuna. The models' efficacy on unknown data was rigorously evaluated using measures such as accuracy, precision, recall, and F1 score.

CHAPTER 4

DATASET INFORMATION

The datasets that the model has been trained with are UNSW-NB15 and US Air Force LAN Network Traffic, and it has been tested against the NSL-KDD dataset. These are among the most widely used datasets for testing of machine learning algorithms that have been developed for network intrusion detection due to the varied data that is incorporated in them.

4.1. UNSW-NB15

The UNSW-NB15 dataset is a popular benchmark dataset in the field of network security, providing significant insights into network traffic and cyber risks. The UNSW-NB15 dataset, which consists of a varied variety of network traffic data gathered in a controlled environment, provides academics and practitioners with a comprehensive collection of labeled examples reflecting various forms of network assaults and regular activity. The dataset contains features such as source and destination IP addresses, protocol type, service, and flag indicators, allowing for extensive investigation of network behavior. With over two million cases and 49 attributes, the UNSW-NB15 dataset is a valuable resource for training and testing Network Intrusion Detection Systems (NIDS) and other cybersecurity solutions. Furthermore, the information includes labels indicating the presence of other sorts of attacks, such as DoS, DDoS, probing, and infiltration attacks, among others. With such detailed annotations, researchers can accurately analyze the efficacy of intrusion detection algorithms. Overall, the UNSW-NB15 dataset is critical to furthering network security research and improving the effectiveness of cybersecurity defenses against emerging attacks.

4.2. US AIR FORCE LAN NETWORK TRAFFIC DATASET

The US Air Force LAN network traffic dataset encapsulates a comprehensive portrayal of network traffic dynamics within a simulated US Air Force Local Area Network (LAN) environment. It represents a meticulously crafted emulation of real-world military network infrastructures, providing researchers with a controlled yet authentic platform for investigating network security and intrusion detection methodologies. Comprising raw TCP/IP dump data, the dataset faithfully captures the intricacies of network communication, offering insights into information exchange among interconnected devices. Deliberately exposed to a diverse range of intrusions and security threats, the dataset mirrors the dynamic

nature of cybersecurity challenges encountered in military LAN environments. With connections meticulously labeled as "Normal" or "Anomalous," and further annotated with specific attack types, the dataset facilitates detailed analysis and classification of security breaches, making it a crucial resource for advancing cybersecurity research and fortifying defense capabilities.

4.3. NSL-KDD

The NSL-KDD dataset is a well-known network security benchmark dataset, providing researchers with a broad collection of labeled instances to evaluate intrusion detection systems (IDS) and related cybersecurity solutions. The dataset contains approximately 125,000 instances and 41 attributes, providing a comprehensive picture of network traffic data, covering both routine activity and various sorts of cyber attacks. Notably, it contains attack categories like as Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R), and Probe assaults, allowing for a comprehensive assessment of IDS performance across a variety of threat scenarios. Each instance is labeled to identify whether it is normal or represents a certain form of attack, which allows for more precise evaluation of detection systems. The NSL-KDD dataset's uniform structure and comprehensive feature set make it an invaluable resource for network security researchers and practitioners looking to create, benchmark, and improve intrusion detection methods.

Using these datasets provides a broad and comprehensive foundation for constructing and evaluating Network Intrusion Detection models. By including these datasets, the proposed approach becomes more robust and capable of generalization, making the NIDS more reliable and efficient.

CHAPTER 5

EXPERIMENTAL WORK AND IMPLEMENTATION

RF and Linear SVM models for Network Intrusion Detection System (NIDS) applications are trained on datasets obtained from the US Air Force LAN and UNSW-NB15. These datasets offer a comprehensive archive of network traffic data, spanning both benign and harmful operations. Training on such different datasets allows the RF and Linear SVM models to learn detailed patterns and properties that indicate network intrusions. Following the training phase, these models' performance is evaluated and tested using the NSL-KDD dataset, a well-known benchmark for NIDS evaluation. This dataset contains a diverse set of assault scenarios, allowing for rigorous testing of trained models under real-world situations. The models' capacity to accurately detect and categorize network intrusions may be examined by validating them on NSL-KDD, which provides insights into their efficacy and generalizability. Finally, this rigorous training and testing technique ensures the reliability and efficacy of RF and Linear SVM models in improving network security and minimizing cyber threats.

5.1. NIDS APPLICATION TO IDENTIFY NETWORK ATTACKS

This integrated application seamlessly merges the power of RF and Linear SVM models, complemented by Recursive Feature Elimination (RFE) for feature selection and Optuna hyperparameter optimization, thereby fortifying its ability to discern intricate patterns within network traffic data. By amalgamating these sophisticated techniques, the application achieves heightened accuracy and efficiency in identifying various types of network attacks. The RF model, renowned for its ensemble learning approach, excels in capturing complex relationships within the data, while the SVM model provides robust classification capabilities, particularly in high-dimensional spaces. RFE strategically selects the most informative features, streamlining the model's learning process and enhancing interpretability, while Optuna fine-tunes model parameters, ensuring optimal performance. This holistic approach empowers the application to navigate the intricate landscape of network traffic parameters, facilitating real-time detection and classification of network attacks with precision and reliability, thereby bolstering network security measures and safeguarding against potential threats.

5.1.1. Testing the Model Using NSL-KDD

The algorithm employs the Python `'pickle'` module to load serialized testing data and machine learning models stored as `'.pkl'` and `'.sav'` files, respectively. The process of loading pickle files involves deserializing Python objects previously saved to disk using the `'pickle.dump()'` function. In this context, the testing data is loaded from a pickle file using the `'pd.read_pickle()'` function from the pandas library, converting the serialized DataFrame object back into its original format. Similarly, pre-trained machine learning models, specifically the Random Forest and Linear SVM models, are loaded using the `'pickle.load()'` function, enabling the retrieval of trained model objects. These models are typically saved during the training phase using the `'pickle.dump()'` function after model fitting. By loading the serialized model objects, the script gains access to the learned parameters and structure, facilitating their direct application to new data for prediction tasks. This streamlined approach to data and model loading via pickle serialization enhances code modularity and facilitates reproducibility in machine learning workflows, offering a seamless integration of trained models into the network intrusion detection process.

The algorithm incorporates an implementation for conducting network intrusion detection using machine learning techniques. Initially, it loads testing data from a pickle file named `"testing_df.pkl"` into a pandas DataFrame, likely containing features extracted from network traffic data. Subsequently, the script organizes this data into separate lists based on the type of network intrusion, including `'normal'`, `'dos'` (denial-of-service), `'r2l'` (remote-to-local), `'u2r'` (user-to-root), and `'probe'` attacks. Each list stores the feature vectors corresponding to instances of the respective attack category, preparing the data for model evaluation and prediction. Additionally, the script loads pre-trained machine learning models from serialized files using the `'pickle.load()'` function, including a Random Forest model stored as `"random_forest_model.sav"` and a Linear SVM model stored as `"Linear_SVM_model.sav"`.

The `'main()'` function encapsulates the core functionality of the script, orchestrating the process of predicting network intrusion types for random test samples. It randomly selects a test sample from the appropriate list based on the provided `'class_name'` parameter, representing the type of network intrusion. The selected sample is then used to generate predictions and probabilities using both the Random Forest and Linear SVM models. The algorithm prints the predicted intrusion types and corresponding probabilities, formatted for clarity, and returns the results. This comprehensive approach demonstrates the application of machine learning models in effectively detecting and classifying various types of network intrusions, enhancing the overall security posture of network environments.

5.1.2. Incorporating the Delete Functionality

The algorithm demonstrates a data preprocessing workflow for network intrusion detection using the NSL-KDD dataset loaded from a CSV file. Initially, the script utilizes the pandas library to read the CSV file into a DataFrame (`df`), setting the header to `None` to avoid treating the first row as column names. Subsequently, column names are assigned to the DataFrame based on the provided list of features (`columns`). The DataFrame is then processed to create a new column named "Class", which categorizes each sample into one of five attack types: 'dos' (denial-of-service), 'r2l' (remote-to-local), 'u2r' (user-to-root), 'probe', or 'normal'. This categorization is determined using predefined lists of attack names (`dos_attacks`, `r2l_attacks`, `u2r_attacks`, `probe_attacks`) and a helper function (`label_attack`) applied to each row of the DataFrame.

Additionally, the script drops redundant columns ("outcome" and "difficulty") from the DataFrame, as they are replaced by the newly created "Class" column. The resulting DataFrame (`df1`) contains a subset of features ('duration', 'protocol_type', 'service', 'flag', 'src_bytes', 'dst_bytes', 'land', 'wrong_fragment', 'logged_in', 'count', 'srv_count') along with the "Class" label. This processed DataFrame is then saved to a new CSV file named "file1.csv". Lastly, there is commented-out code intended to select random samples from the 'dos' class for further analysis, which could be useful for data exploration or model validation purposes. Overall, this script demonstrates a structured approach to data preprocessing for network intrusion detection tasks, facilitating subsequent analysis and model training.

5.2. FLASK APPLICATION FOR NIDS

This Flask application serves as a streamlined platform for Network Intrusion Detection System (NIDS) analysis, leveraging the Flask framework to deliver an intuitive user experience. The application comprises several basic web pages, each catering to specific functionalities related to NIDS evaluation and prediction. Users can seamlessly navigate through these pages, including the index, features, analysis, and model pages, to access relevant information and tools for network security assessment. The simplicity and clarity of the interface ensure accessibility for users of varying technical backgrounds, facilitating efficient interaction with the application.

Furthermore, the application facilitates real-time traffic type prediction through a straightforward submission process. Upon submitting the desired traffic type, the application utilizes a backend evaluation module to predict the corresponding attack category and

associated probabilities. Leveraging machine learning techniques, the application provides users with insightful predictions regarding potential network intrusions, empowering them to proactively address security threats. The structured presentation of results in the result.html template enhances user comprehension, enabling informed decision-making regarding network security measures. Overall, the Flask application embodies efficiency in delivering essential NIDS functionalities, contributing to bolstering network security for organizations and individuals alike.

Following is the user interface for the flask application. The web application incorporates a registration page, a login page, a data analysis page, a dataset feature extraction page, a web page that takes network traffic parameters as input, and a page that accurately predicts the type of network traffic identified.

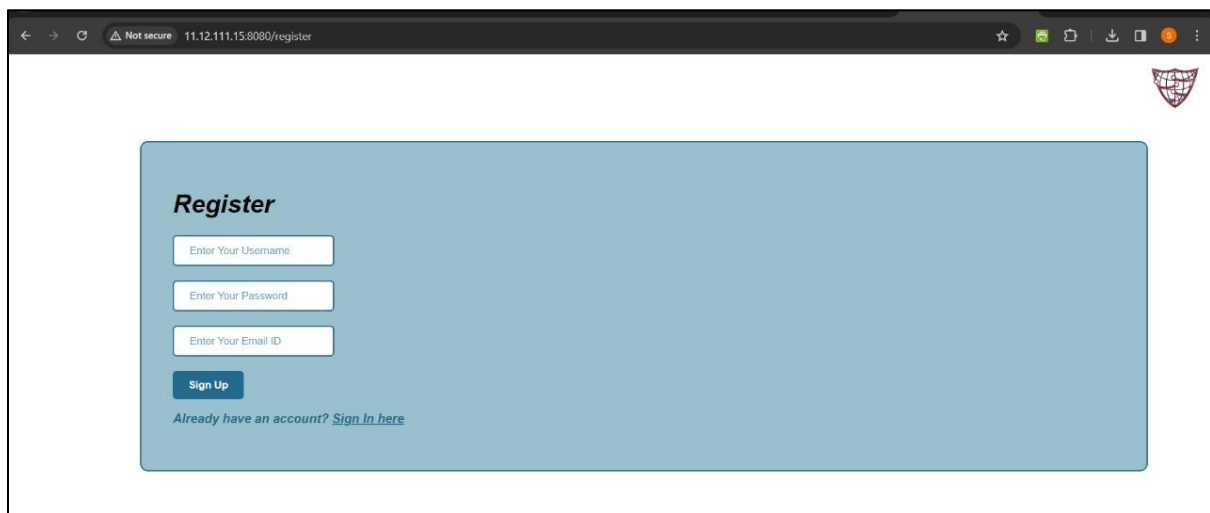
A screenshot of a web browser displaying the registration page. The browser's address bar shows "11.12.111.15:8080/register". The page features a light blue rectangular form with the heading "Register" in bold. Inside the form, there are three input fields labeled "Enter Your Username", "Enter Your Password", and "Enter Your Email ID". Below these fields is a dark blue "Sign Up" button. At the bottom of the form, there is a link that says "Already have an account? [Sign In here](#)". A small red shield logo is visible in the top right corner of the browser window.

Fig. 5.1. Registration Page

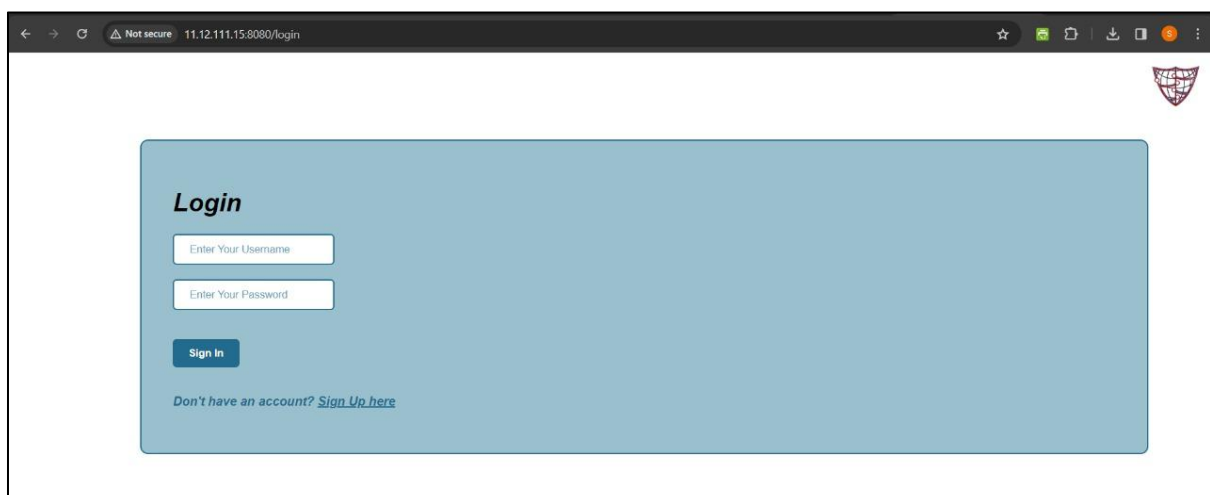
A screenshot of a web browser displaying the login page. The browser's address bar shows "11.12.111.15:8080/login". The page features a light blue rectangular form with the heading "Login" in bold. Inside the form, there are two input fields labeled "Enter Your Username" and "Enter Your Password". Below these fields is a dark blue "Sign In" button. At the bottom of the form, there is a link that says "Don't have an account? [Sign Up here](#)". A small red shield logo is visible in the top right corner of the browser window.

Fig. 5.2. Login Page

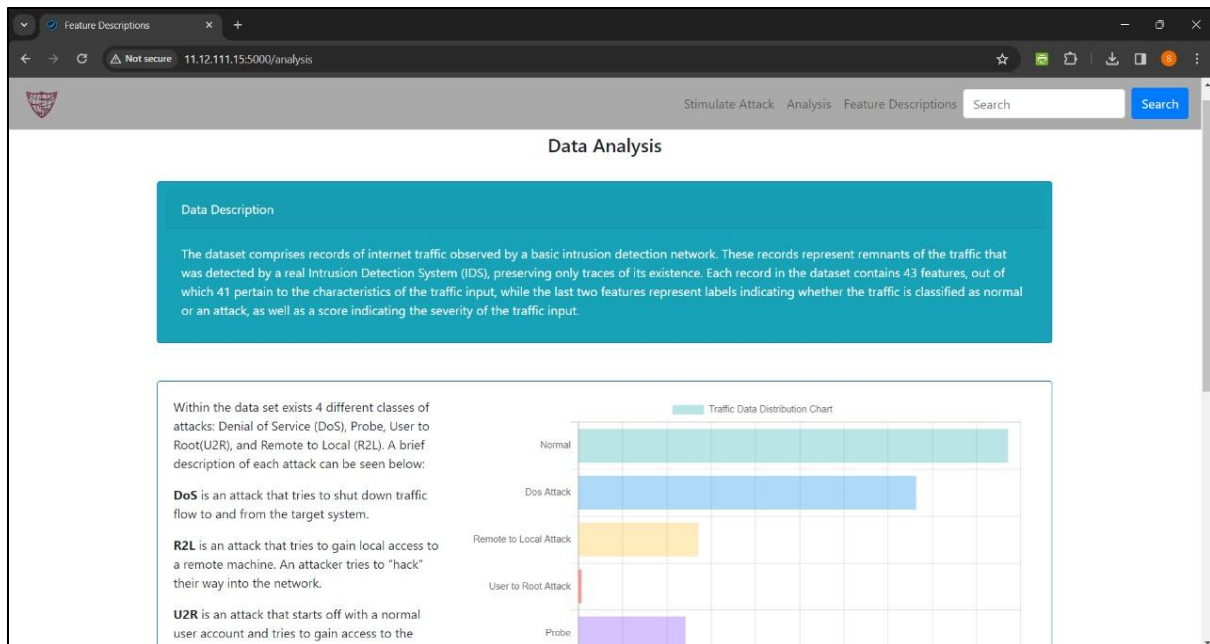


Fig. 5.3. Data Analysis Page

This page consists of information regarding the different types of attacks like DoS, R2L, U2R and probe. It includes the impact these attacks have, and the distribution of these attacks in the NSL-KDD Dataset.

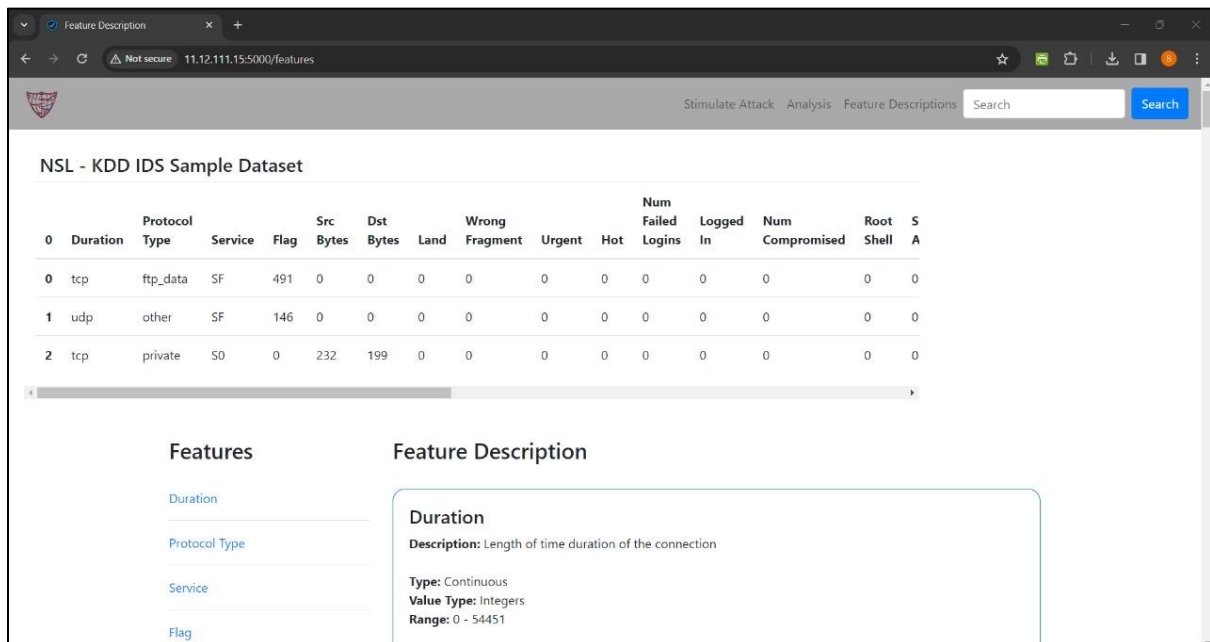


Fig. 5.4. Feature Description Page

This page reflects the different features that can be found in the NSL-KDD dataset, and the attributes associated with them. It also includes a short description of the attack, along with the type, value type, range, etc.

Stimulate an input traffic **Select** Traffic features

Duration Length of time duration of the connection (0 - 54451)	Protocol Type Protocol used in the connection
Service Destination network service used	Flag Flag Type
Src Bytes Number of data bytes transferred from source to dest	Dstn Bytes Number of data bytes transferred from dest to source
Logged In Login Status	Wrong Fragment Total number of wrong fragments in this connection
Same Destn Count Number of connections to the same destination	Same Port Count Number of connections to the same service (port number)

Reset Form **Submit**

NOTE: Only significant features are selected here, for other features visit: [Feature Descriptions](#)

Fig. 5.5. Network Traffic Input

This page includes various fields that take input parameters of network traffic. Once submitted, the resulting page predicts the type of traffic using both Linear_SVM and RF models.

CHAPTER 6

RESULTS AND ANALYSIS

Performance metrics are essential in assessing the performance of models and systems, guiding decision-making, and facilitating quantitative comparisons and improvements in various fields, from machine learning and business analytics to healthcare and beyond. The metrics that are considered to measure the performance of the model are accuracy, a classification report, and a confusion matrix. Accuracy is a performance metric in classification that measures the proportion of correctly predicted instances over the total number of instances in a dataset, often expressed as a percentage. A confusion matrix is a table used in machine learning to describe the performance of a classification model, displaying the true and predicted class values. A classification matrix is a visual representation of the performance of a classification model, typically showing true positive, true negative, false positive, and false negative values.

6.1. US AIR FORCE LAN - PERFORMANCE METRIC EVALUATION

For the dataset US Air Force LAN Network Traffic, the testing accuracy for the random forest model is 99.64. The following are the accuracy, classification report and confusion matrix for the same.

Metric	Value
Training Time	0.06738543510437012
Best Parameters	{'rf_n_estimators': 105, 'rf_max_depth': 19}
Train Score	1.0
Test Score	0.9962953162212226
Accuracy	0.9964276263561789
Precision	0.9964275906264386
Recall	0.9964276263561789
F1 Score	0.9964275909508982

Fig. 6.1. Model Performance (Random Forest) 1

Classification Report:				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	3498
1	1.00	1.00	1.00	4060
accuracy			1.00	7558
macro avg	1.00	1.00	1.00	7558
weighted avg	1.00	1.00	1.00	7558

Fig. 6.2. Classification Report (Random Forest) 1

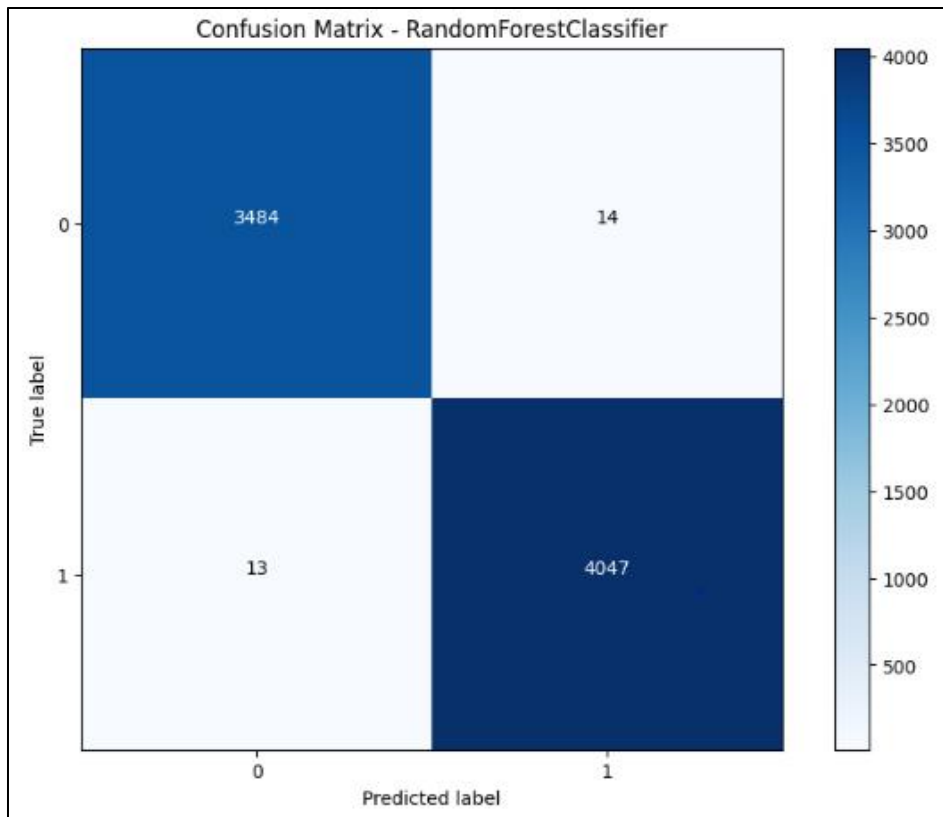


Fig. 6.3. Confusion Matrix (Random Forest) 1

For the dataset US Air Force LAN Network Traffic, the testing accuracy for the linear SVM model is 95.23. The following are the accuracy, classification report and confusion matrix for the same.

Metric	Value
-----	-----
Training Time	0.00308990478515625
Best Parameters	{'svc_C': 4.2478827485881485}
Train Score	0.9523080412838834
Test Score	0.9539560730351945
Accuracy	0.9523683514157184
Precision	0.9525234925387166
Recall	0.9523683514157184
F1 Score	0.9523215391663019

Fig. 6.4. Model Performance (Linear_SVM) 1

Classification Report:					
	precision	recall	f1-score	support	
0	0.96	0.94	0.95	3498	
1	0.95	0.97	0.96	4060	
accuracy			0.95	7558	
macro avg	0.95	0.95	0.95	7558	
weighted avg	0.95	0.95	0.95	7558	

Fig. 6.5. Classification Report (Linear_SVM) 1

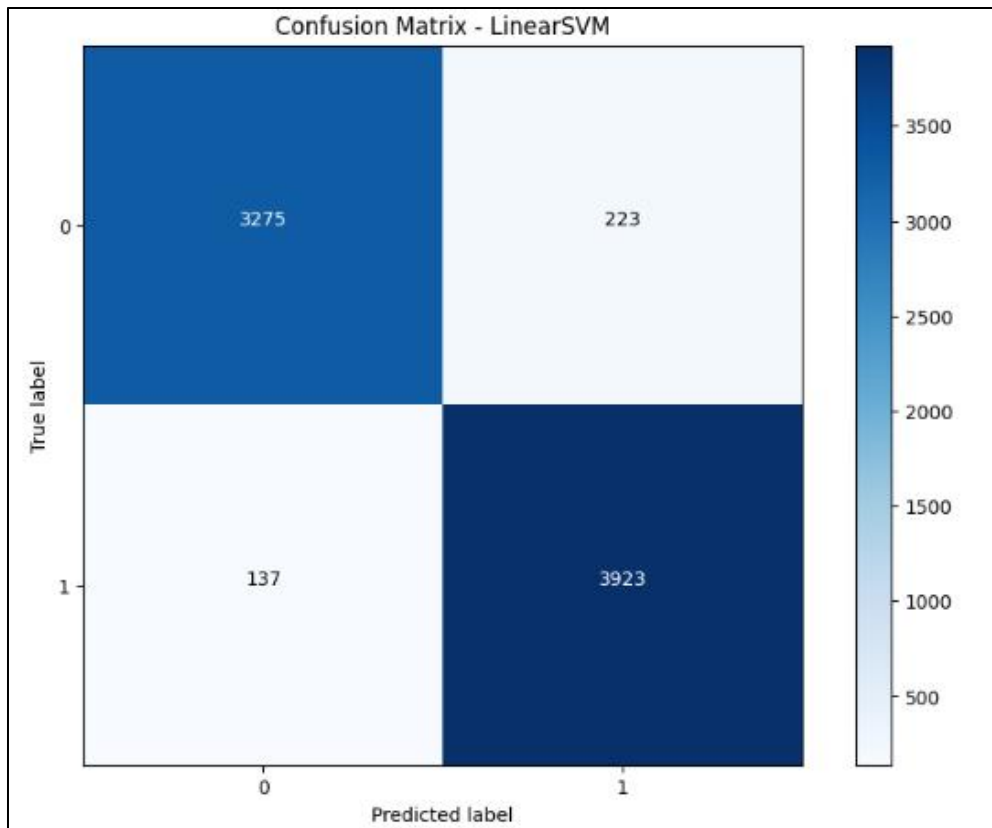


Fig. 6.6. Confusion Matrix (Linear_SVM) 1

6.2. UNSW-NB15 - PERFORMANCE METRIC EVALUATION

For the dataset UNSW-NB15, the testing accuracy for the random forest model is 89.51. The following are the accuracy, classification report and confusion matrix for the same.

Metric	Value
Training Time	0.3858523368835449
Best Parameters	{'rf_n_estimators': 12, 'rf_max_depth': 14}
Train Score	0.9176499167129373
Test Score	0.8982186234817814
Accuracy	0.8951417004048583
Precision	0.8940992793647795
Recall	0.8951417004048583
F1 Score	0.8926430131408059

Fig. 6.7. Model Performance (Random Forest) 2

Classification Report:				
	precision	recall	f1-score	support
0	0.39	0.08	0.13	205
1	0.24	0.03	0.05	174
2	0.44	0.48	0.46	1259
3	0.67	0.75	0.71	3314
4	0.78	0.80	0.79	1830
5	0.99	0.97	0.98	5660
6	1.00	1.00	1.00	11098
7	0.92	0.81	0.86	1035
8	0.62	0.65	0.63	110
9	0.62	0.33	0.43	15
accuracy			0.90	24700
macro avg	0.67	0.59	0.61	24700
weighted avg	0.89	0.90	0.89	24700

Fig. 6.8. Classification Report (Random Forest) 2

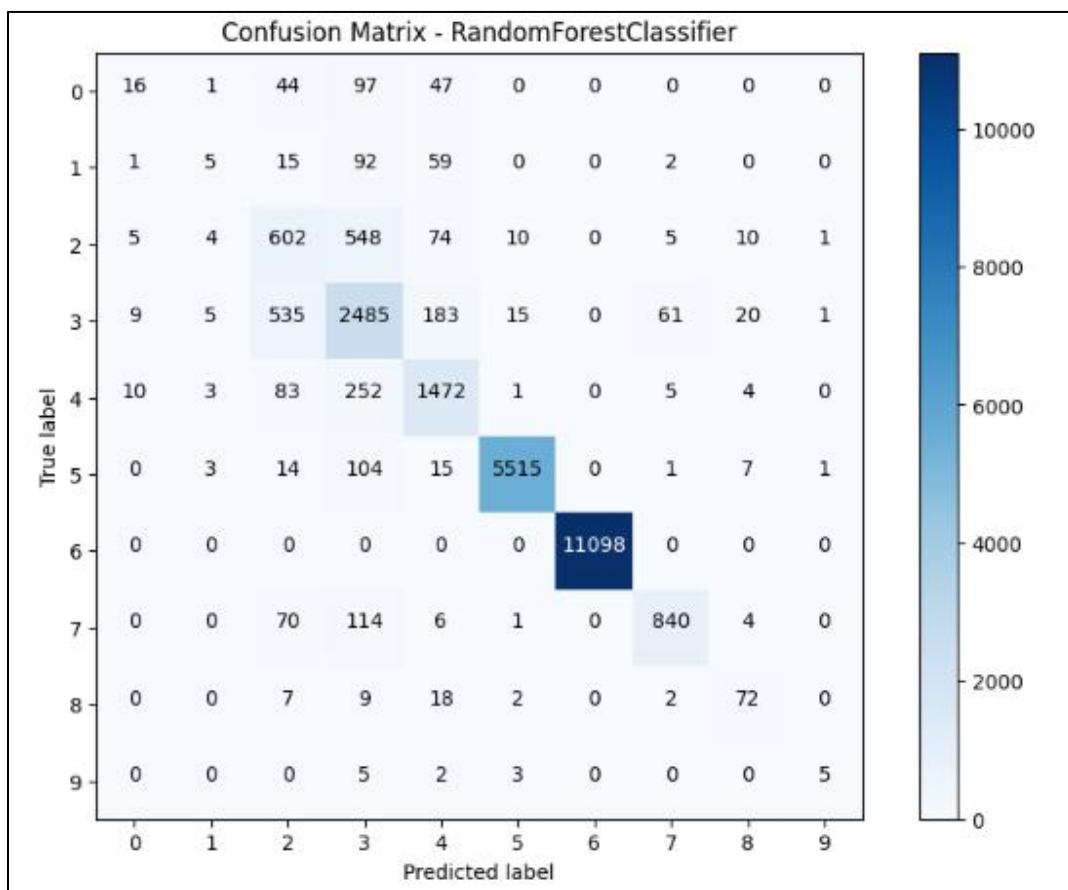


Fig. 6.9. Confusion Matrix (Random Forest) 2

For the dataset USNW-NB15, the testing accuracy for the linear SVM model is 85.53. The following are the accuracy, classification report and confusion matrix for the same.

Metric	Value
-----	-----
Training Time	17.639332056045532
Best Parameters	{'rf_n_estimators': 153, 'rf_max_depth': 11}
Train Score	0.9076554691837868
Test Score	0.8980566801619433
Accuracy	0.8553441295546559
Precision	0.8366792569939765
Recall	0.8553441295546559
F1 Score	0.837046276599436

Fig. 6.10. Model Performance (Linear_SVM) 2

Classification Report:					
	precision	recall	f1-score	support	
0	0.00	0.00	0.00	205	
1	0.00	0.00	0.00	174	
2	0.32	0.03	0.06	1259	
3	0.60	0.78	0.68	3314	
4	0.52	0.68	0.59	1830	
5	1.00	0.96	0.98	5660	
6	1.00	1.00	1.00	11098	
7	0.54	0.71	0.61	1035	
8	0.00	0.00	0.00	110	
9	0.00	0.00	0.00	15	
accuracy			0.86	24700	
macro avg	0.40	0.42	0.39	24700	
weighted avg	0.84	0.86	0.84	24700	

Fig. 6.11. Classification Report (Linear_SVM) 2

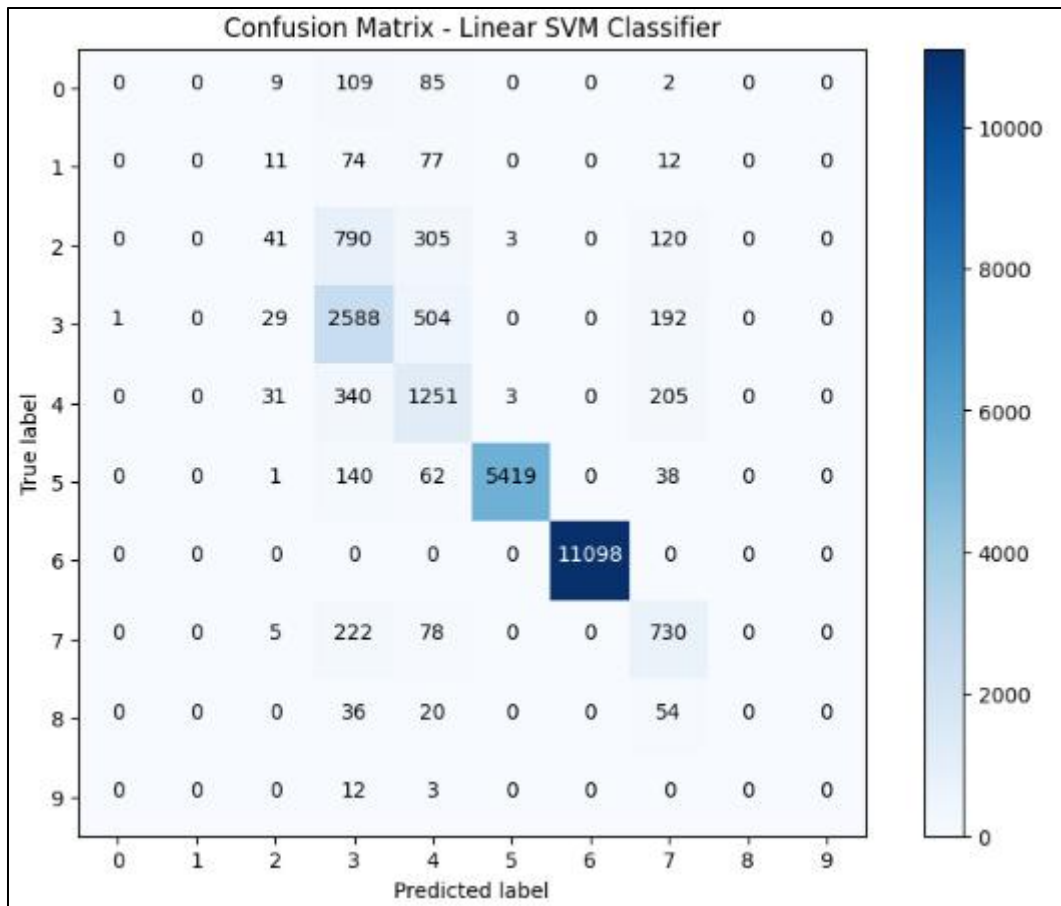


Fig. 6.12. Confusion Matrix (Linear_SVM) 2

6.3. MAKING PREDICTIONS BASED ON NETWORK TRAFFIC PARAMETERS

On giving various parameters as input, the model predicts the type of traffic as either normal or that of an attack. The different attacks that can be identified are DoS, Remote to Local, User to Root, and probe.

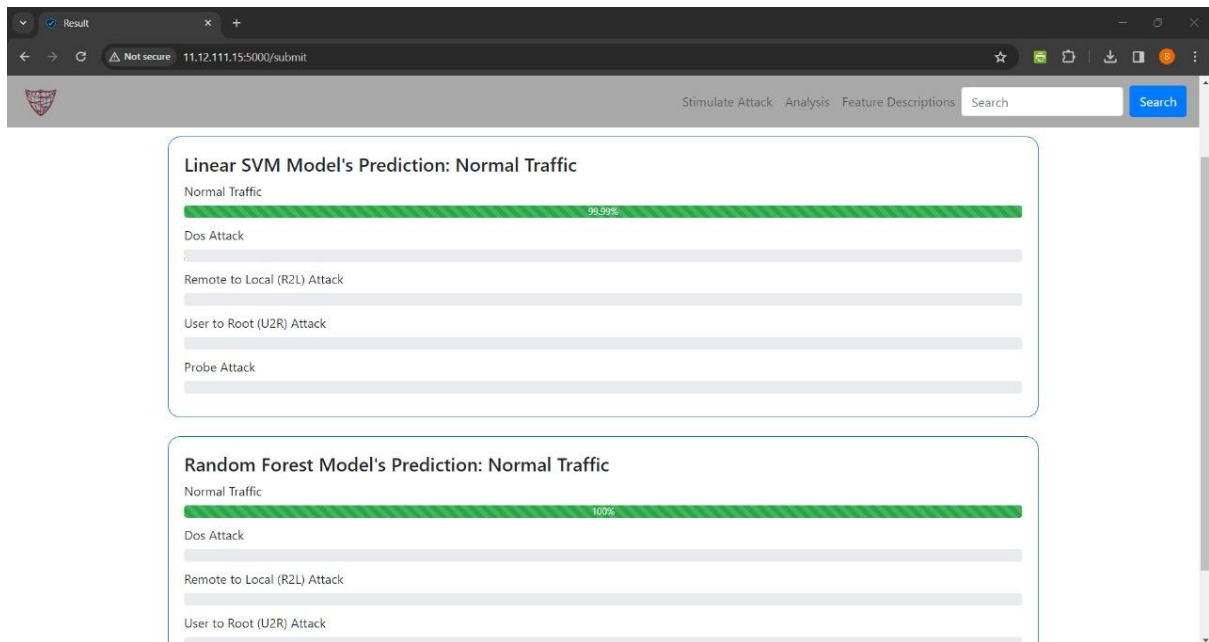


Fig. 6.13. Normal Traffic

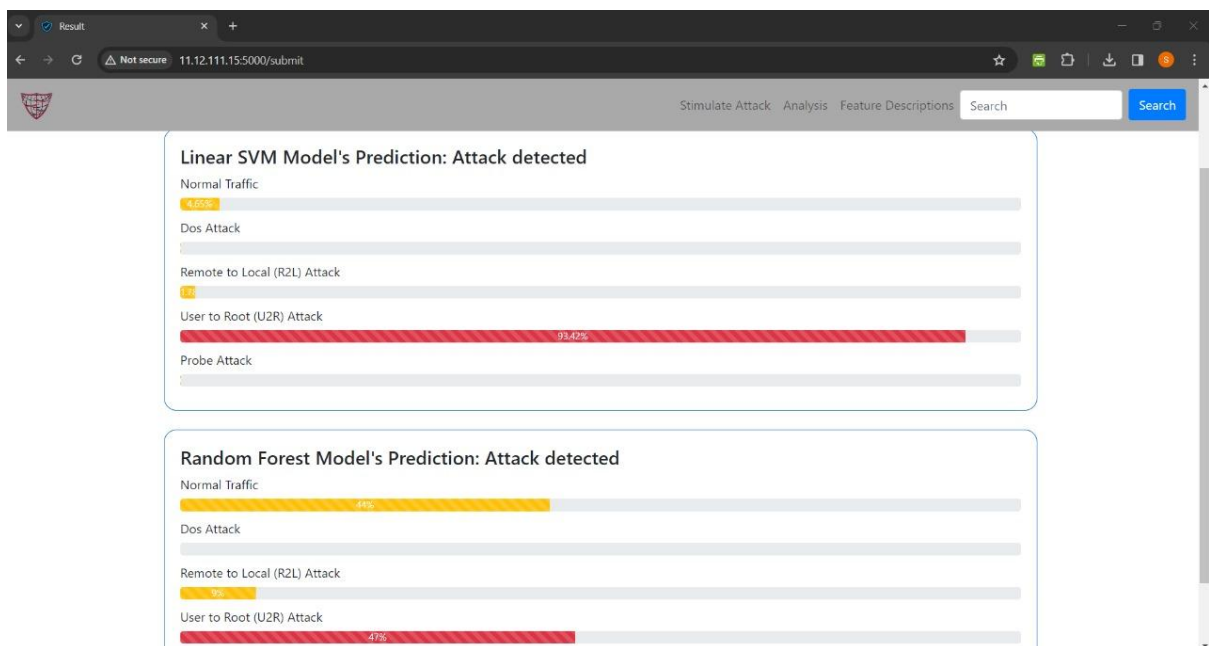


Fig. 6.14. User to Root Attack

A user-to-root (U2R) attack occurs when an unauthorized user gains root access by exploiting system vulnerabilities. Once accomplished, attackers can change system files, possibly compromising the entire machine.

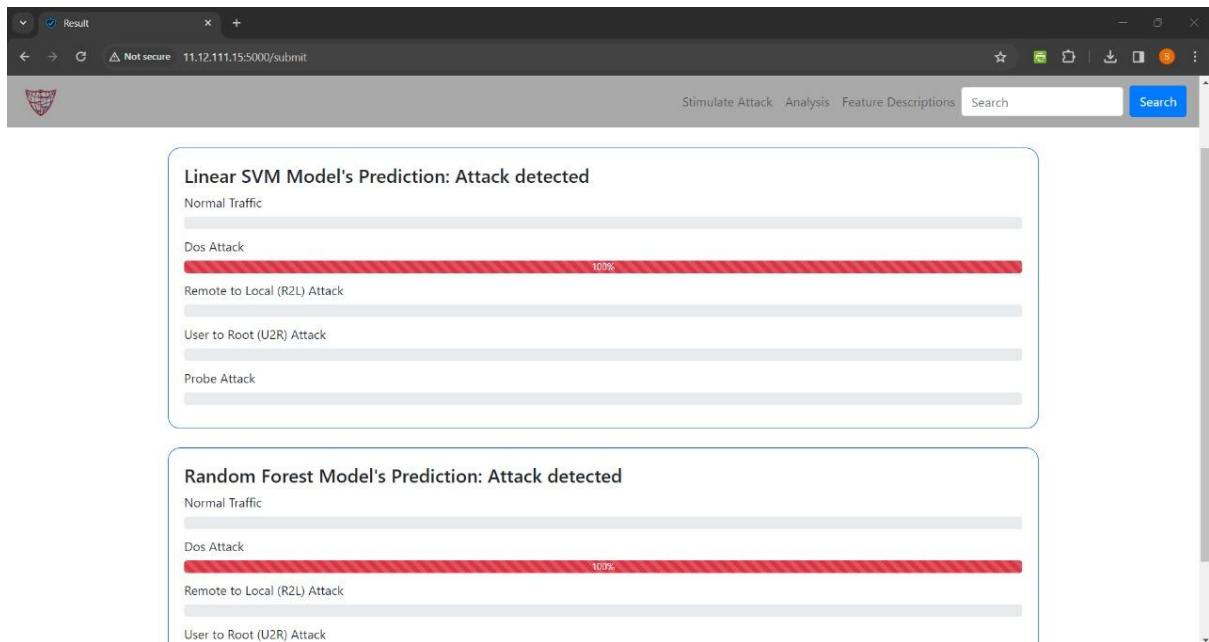


Fig. 6.15. DoS Attack

A Denial of Service (DoS) attack floods a system or network with too much traffic, making it unreachable to normal users. These assaults are designed to disrupt services, exhaust resources, and cause system breakdowns, resulting in downtime and financial losses for the targeted entities. Defending against DoS attacks necessitates strong network security and regular monitoring.

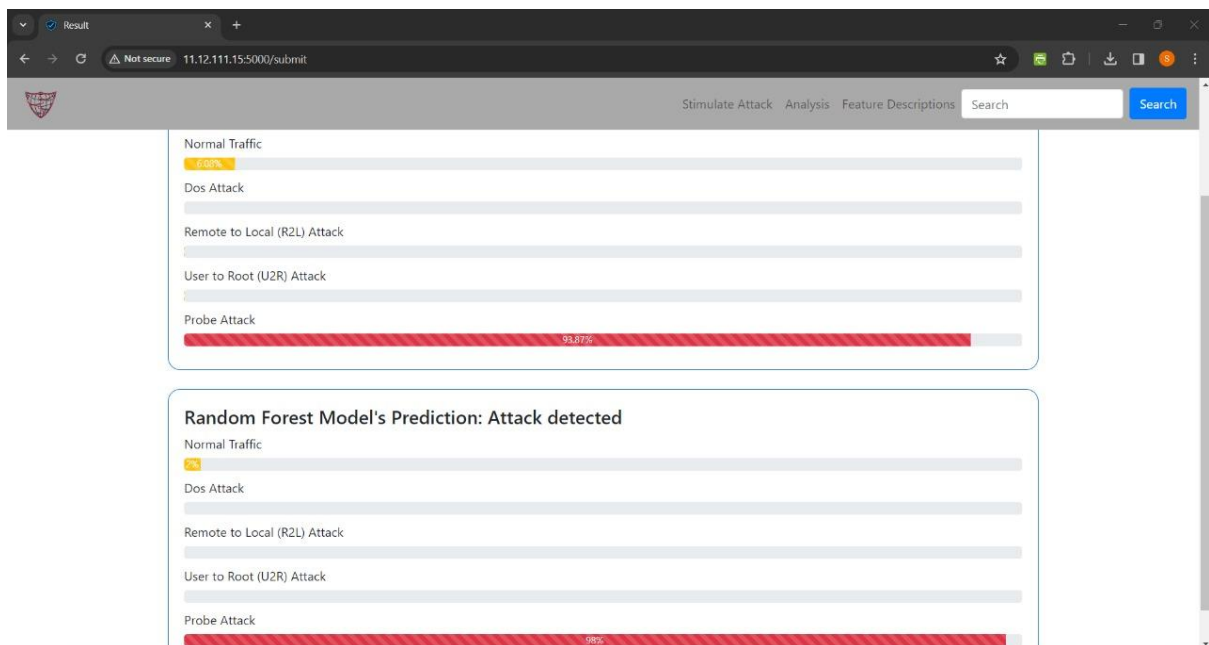


Fig. 6.16. Probe Attack

A probe attack is an illegal attempt to obtain information about a target system in order to uncover vulnerabilities through actions such as port scanning and network mapping.

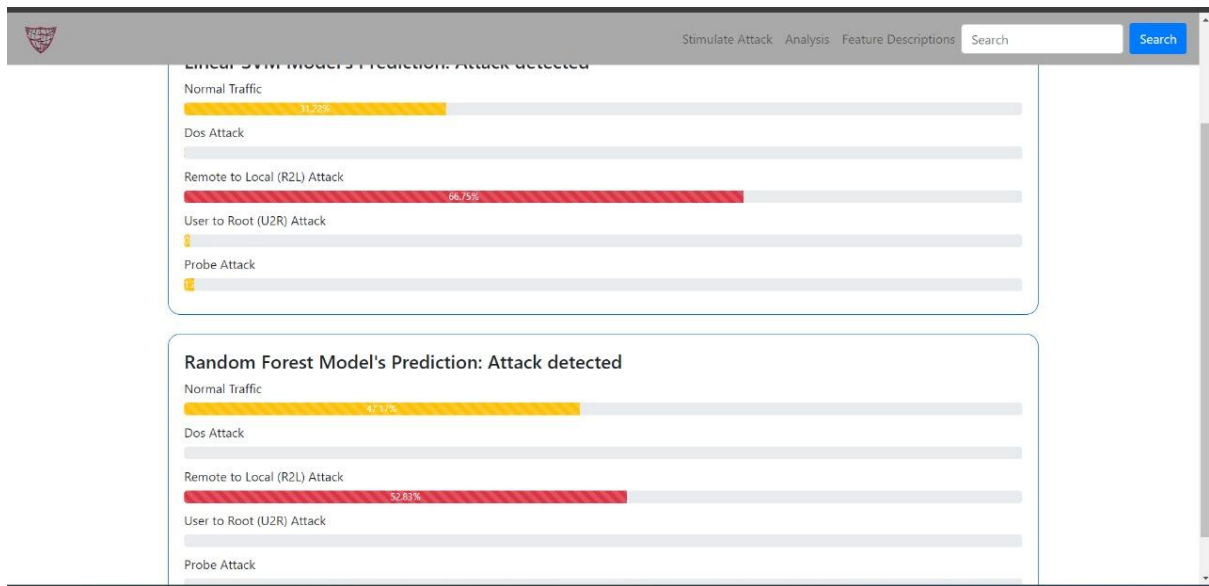


Fig. 6.17. Remote to Local Attack

A Remote-to-Local (R2L) attack is a cybersecurity issue in which an unauthorized person attempts to get access to a computer system from a remote location. These attacks use weaknesses in services or programs to obtain access and elevate privileges, possibly compromising sensitive data. Strong authentication systems and regular software updates to fix known vulnerabilities are effective defenses against R2L attacks.

6.4. ABLATION STUDY

The NIDS model has been trained against the datasets US Air Force LAN and USNW-NB15 to improve the efficiency with which it works, and to improve its accuracy. To achieve this, it has been trained using various models and pre-processing techniques like RF, Principal Component Analysis (PCA), and Auto Encoding (AE).

Following are the training and testing accuracies for the models Logistic Regression, K Nearest Neighbour, Decision Tress, Multi-layer Perceptron, Artificial Neural Networks, Recursive Neural Networks (RNN), RNN + LSTM (Long Short Term Memory) with various pre-processing techniques.

		Random Forest	PCA	Auto Encoding
Logistic Regression	Training	92.88	93.63	99.46
	Testing	92.33	93.30	99.31
K Nearest Neighbour	Training	98.31	99.41	99.08
	Testing	98.13	98.99	98.76

Decision Trees	Training	1	1	1
	Testing	99.40	99.40	99.40
Multi-layer Perceptron	Training	96.80	99.03	99.46
	Testing	96.94	98.82	99.31
Artificial Neural Networks	Training	96.73	97.81	94.04
	Testing	96.36	97.55	94.54
RNN	Training	81.73	98.24	95.07
	Testing	81.60	-	94.62
RNN+LSTM	Training	95.62	98.83	98.40
	Testing	95.68	-	-

Table 6.1. Ablation Study

CHAPTER 7

CONCLUSION AND SCOPE FOR FURTHER WORK

In summary, this research paper delves into the domain of Network Intrusion Detection Systems (NIDS), leveraging the US Air Force LAN and UNSW-NB15 datasets for training and the NSL-KDD dataset along with real-time data for testing. Through the utilization of Random Forest (RF) and Linear Support Vector Machine (SVM) models, coupled with Recursive Feature Elimination (RFE) and Optuna hyperparameter optimization, the study aims to advance the understanding of network intrusion detection. By selecting pertinent features and optimizing model parameters, the research endeavors to enhance the accuracy and efficiency of NIDS in real-world scenarios.

The development of a live predictions module represents a significant advancement, enabling real-time assessment of potential network intrusions. The practical applications of this system are extensive, ranging from enhancing network security to ensuring the integrity of critical systems. The research underscores the importance of meticulous data preprocessing, model selection, and hyperparameter tuning in achieving optimal performance.

However, further exploration into the inclusion of additional layers in the neural network, along with the investigation of an optimal number of features for feature extraction, could improve the accuracy of the NIDS models. Moreover, integrating the models into user-friendly web or mobile applications would enhance accessibility, allowing users to upload network traffic data for analysis, thereby facilitating proactive network security measures. This research contributes to the evolving field of network intrusion detection, laying the groundwork for future studies aimed at advancing the efficacy and usability of NIDS technologies.

REFERENCES

- [1] Maria Rodriguez, Alvaro Alesanco, Lorena Mehavilla, and Jose Garcia. *Evaluation of Machine Learning Techniques for Traffic Flow-Based Intrusion Detection*, in *Sensors* 2022, 22(23), 9326; <https://doi.org/10.3390/s22239326>.
- [2] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, Joarder Kamruzzaman, *Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges*, in *Cybersecurity*, Springer Open Access, 2019; <https://doi.org/10.1186/s42400-019-0038-7>.
- [3] Peilun Wu and Hui Guo, *LuNet: A Deep Neural Network for Network Intrusion Detection*, in arXiv: 1909.10031v2, 2019.
- [4] Alper Tugay Mizrak, Yu-Chung Cheng, Keith Marzullo and Stefan Savage, *Faith: Detecting and Isolating Malicious Routers*, 2005 International Conference on Dependable Systems and Networks (DSN'05), 2005; doi: 10.1109/DSN.2005.49.
- [5] S. Wang, J. F. Balarezo, S. Kandeepan, A. Al-Hourani, K. G. Chavez and B. Rubinstein, *Machine Learning in Network Anomaly Detection: A Survey*, in *IEEE Access*, vol. 9, pp. 152379-152396, 2021, doi: 10.1109/ACCESS.2021.3126834.
- [6] R. K. Vigneswaran, R. Vinayakumar, K. P. Soman and P. Poornachandran, *Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security*, 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, India, 2018, pp. 1-6, doi: 10.1109/ICCCNT.2018.8494096.
- [7] D. Han et al., *Evaluating and Improving Adversarial Robustness of Machine Learning-Based Network Intrusion Detectors*, in *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2632-2647, Aug. 2021, doi: 10.1109/JSAC.2021.3087242.
- [8] Jay Sinha and M. Manollas, *Efficient Deep CNN-BiLSTM Model for Network Intrusion Detection*, Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition, 2020; <https://doi.org/10.1145/3430199.3430224>.
- [9] Tomas Pevny and Petr Somol, *Discriminative Models for Multi-Instance Problems with Tree-Structure*, arXiv:1703.02868, 2017; <https://doi.org/10.48550/arXiv.1703.02868>.
- [10] Ekhlas K. Gbashi, *Intrusion Detection System for NSL-KDD Dataset Based on Deep Learning and Recursive Feature Elimination*, *Engineering and Technology Journal* Vol. 39 No. 7, 2021, doi: 10.3.684/etj.v39i7.1695.
- [11] L. Dhanabal, *A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms*, *International Journal of Advanced Research in Computer and*

Communication Engineering, 2015.

[12] Ravipati Rama Devi, *Intrusion Detection System Classification Using Different Machine Learning Algorithms on KDD-99 and NSL-KDD Datasets - A Review Paper*, International Journal of Computer Science and Information Technology, 2019. doi: 10.5121/ijcsit.2019.11306.

[13] Yifan Tang, *Deep Stacking Network for Intrusion Detection*, in Sensors (Basel), 2021.

[14] Sarika Choudhary, *Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT*, Procedia Computer Science Volume 167, 2020.

[15] Mina Eshak Magdy, *A Comparative study of intrusion detection systems applied to NSL-KDD Dataset*, Egyptian International Journal of Engineering Sciences and Technology, doi: 10.21608/EIJEST.2022.137441.1156, 2022.

APPENDICES

All resources and source files can be found in this repository.

<https://github.com/Sruthi06-web/ThreatWatch>

PLAGIARISM REPORT

NIDS Report (1).pdf

ORIGINALITY REPORT

10 %	6 %	7 %	5 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dokumen.pub Internet Source	1 %
2	www.researchgate.net Internet Source	1 %
3	www.ninjaone.com Internet Source	<1 %
4	Submitted to Rivier University Student Paper	<1 %
5	Submitted to Charotar University of Science And Technology Student Paper	<1 %
6	www.catalyzex.com Internet Source	<1 %
7	Submitted to University of Warwick Student Paper	<1 %
8	www.arxiv-vanity.com Internet Source	<1 %
9	Nandhini P S, Bharani S, Harish M, Gomanishwaran S. "Feature Selection Using	<1 %

Firefly Algorithm for Classification of Attacks in Routing Protocol for Low Power Lossy Networks (LLNs) Based Internet of Things (IoT) Networks", 2023 4th International Conference on Smart Electronics and Communication (ICOSEC), 2023

Publication

10	Submitted to University of Hertfordshire <small>Student Paper</small>	<1 %
----	--	------

11	Tarak Nandy, Rafidah Md Noor, Raenu Kolandaisamy, Mohd Yamani Idna Idris, Sananda Bhattacharyya. "A review of security attacks and intrusion detection in the vehicular networks", Journal of King Saud University - Computer and Information Sciences, 2024	<1 %
----	--	------

Publication

12	hubvela.com <small>Internet Source</small>	<1 %
----	---	------

13	Submitted to Universiti Kebangsaan Malaysia <small>Student Paper</small>	<1 %
----	---	------

14	Zhong, Yinzheng. "Process Mining and Machine Learning for Intrusion Detection", The University of Liverpool (United Kingdom), 2023	<1 %
----	--	------

Publication

15	www.mdpi.com <small>Internet Source</small>	
----	--	--

		<1 %
16	"6th International Conference on Signal Processing and Information Communications", Springer Science and Business Media LLC, 2024 Publication	<1 %
17	robots.net Internet Source	<1 %
18	Submitted to American Public University System Student Paper	<1 %
19	Submitted to University of Pretoria Student Paper	<1 %
20	ajmrr:thelawbrigade.com Internet Source	<1 %
21	sist.sathyabama.ac.in Internet Source	<1 %
22	www.ijisae.org Internet Source	<1 %
23	www.prolekare.cz Internet Source	<1 %
24	N. Ragavendran, G. Aravind Swaminathan, Nithin, M. Mariammal, B. Diviyapriya, Euodial. "Retinal Vessel Classification and	<1 %

Segmentation Using Advanced Machine Learning Techniques", 2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), 2023

Publication

25

[e-century.us](https://www.e-century.us)

Internet Source

<1 %

26

Gerard Shu Fuhnwi, Matthew Revelle, Clemente Izurieta. "Improving Network Intrusion Detection Performance : An Empirical Evaluation Using Extreme Gradient Boosting (XGBoost) with Recursive Feature Elimination", 2024 IEEE 3rd International Conference on AI in Cybersecurity (ICAIC), 2024

Publication

<1 %

27

Taylor Bradley, Elie Alhajjar, Nathaniel D. Bastian. "Novelty Detection in Network Traffic: Using Survival Analysis for Feature Identification", 2023 IEEE International Conference on Assured Autonomy (ICAA), 2023

Publication

<1 %

28

researchonline.gcu.ac.uk

Internet Source

<1 %

29

www.astesj.com

Internet Source

<1 %

30	www.grafiati.com Internet Source	<1 %
31	Ajayi, Oluwagbemiga. "Developing Cross-Domain Intrusion Detection Systems", University of Maryland, Baltimore County, 2023 Publication	<1 %
32	Bayu Adhi Tama, Sunghoon Lim. "Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation", Computer Science Review, 2021 Publication	<1 %
33	Bidyapati Thiyam, Shouvik Dey. "Statistical methods for feature selection: unlocking the key to improved accuracy", International Journal of Computers and Applications, 2023 Publication	<1 %
34	Submitted to Concordia University Student Paper	<1 %
35	Emmanuel Hooper. "An Intelligent and Expert Mining Intrusion Detection and Response System", 2006 1st International Conference on Digital Information Management, 2007 Publication	<1 %
36	Stephen Kahara Wanjau, Geoffrey Mariga Wambugu, Aaron Mogeni Oirere, Geoffrey	<1 %

Muchiri Muketha. "Discriminative spatial-temporal feature learning for modeling network intrusion detection systems", Journal of Computer Security, 2023

Publication

37

j.mecs-press.net

Internet Source

<1 %

38

pages.cs.wisc.edu

Internet Source

<1 %

39

Bhavsar, Mansi Himanshu. "A Dynamic Architecture of an Anomaly Detection System in IoT Devices", North Carolina Agricultural and Technical State University, 2024

Publication

<1 %