Advance Network Intrusion Detection System Using Deep Learning Techniques

[1]Vali Sai Jitha Vaishnavi, [2]Sattaram Sai Praneeth Reddy,[3]Anumala Venu Prasad Rohan, [4]K. Beena [1,2,3]UG Student, [4]Assistant Professor [1,2,3,4]CSE- Artificial Intelligence and Machine Learning [1,2,3,4]Sreenidhi Institute of Science and Technology, Hyderabad, Telangana.

***Abstract** :With the rapid increase in cyber threats, traditional intrusion detection systems (IDS) struggle to keep up with sophisticated attacks. This project aims to develop an Advanced Network Intrusion Detection System (NIDS) using Deep Learning techniques to detect and classify network intrusions effectively. The system processes real-time network traffic and classifies it as normal or malicious using deep learning models such as ML models. The dataset is preprocessed using feature engineering techniques like One-Hot Encoding and Min-Max Scaling to improve accuracy. The trained model is deployed in a Flask-based web application that continuously monitors network activity and alerts administrators about potential threats. Unlike traditional signature-based IDS, this system can detect zero-day attacks by learning patterns from previous intrusions. By comparing multiple deep learning architectures, we aim to achieve high accuracy, precision, and recall in intrusion detection. The proposed system enhances network security and helps organizations prevent unauthorized access and data breaches effectively.*

Keywords: Advance Network Intrusion Detection System (NIDS),One-Hot Coding ,Min-Max Scaling ,Flask Based Web Application

1. INTRODUCTION

1. Background

concerns the increasing sophistication and magnitude of cyber threats targeting contemporary computer systems and networks. With the expansion of digital frameworks, traditional Intrusion Detection Systems (IDS) have found it difficult to contend with the ever-growing ingenuity of assaults, including zero-day vulnerabilities, nuanced breaking methods, and more.This project utilizes tweets from actual customers to augment churn prediction with sentiment analysis.This project seeks to resolve these issues with the implementation of an Advanced Intrusion Detection System (AIDS) which utilizes machine learning techniques for effective response and threat detection. The method applies both supervised and unsupervised learning techniques to system and network traffic patterns to improve detection accuracy while reducing false positives.

2. Motivation

Deep Learning Techniques for Advanced Network Intrusion Detection Systems arise from the paramount concern of reinforcing cybersecurity systems due to the relentless evolution and sophistication of cyber-attacks. The traditional intrusion detection approach that uses only signature-based techniques or simple anomaly detection methods is unlikely to identify a zero-day attack and is notorious for generating an excessive amount of false positives that impede response time. Alongside the explosive expansion of digital infrastructure, dependency on networks and interconnections increases, creating a necessity for more sophisticated, real-time, adaptive solutions. This project using deep learning algorithms, specifically Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM), aims to enhance accuracy in threat detection, diminish false alerts, and provide real-time surveillance monitoring. Incorporating this system into an easy-to-navigate web application allows organizations to adopt proactive measures to secure their networks which, in turn, protects data integrity, confidentiality, and availability, especially in a volatile cyberspace.

3. Objectives

- **Use deep learning models** (like CNN, LSTM, etc.) to analyze and identify patterns in network traffic.
- **Detect both known and unknown (zero-day) attacks** by learning from past intrusion data.
- **Preprocess the data** using techniques like One-Hot Encoding and Min-Max Scaling to improve model accuracy.
- **Compare the performance of various machine learning and deep learning models** to choose the most effective one.
- **Deploy the system as a real-time web application** using Flask for continuous monitoring.
- **Reduce false positives to improve the reliability of the system.**
- **Integrate with existing network infrastructure** for real-time alerts and improved security response.

2. RELATED WORKS

1. Machine Learning for Anomaly Detection in Network Traffic
Overview: Employing supervised learning to identify network anomalies by means of efficient feature extraction.
Advantages: High accuracy; scalable.
Disadvantages: Imbalanced data issues.

2. Real-Time Detection of DDoS Using Deep Learning
Overview: Real-time intrusion detection based on CNN for high-speed networks.
Advantages: Real-time processing; low false positives.
Disadvantages: High computational requirements.

3. Comparative Analysis of Machine Learning Algorithms in Cybersecurity
Overview: Performance comparison of various ML algorithms for threat detection.
Pros: Thought-provoking comparison between classifiers.
Cons: Lacks focus on real-time deployment.

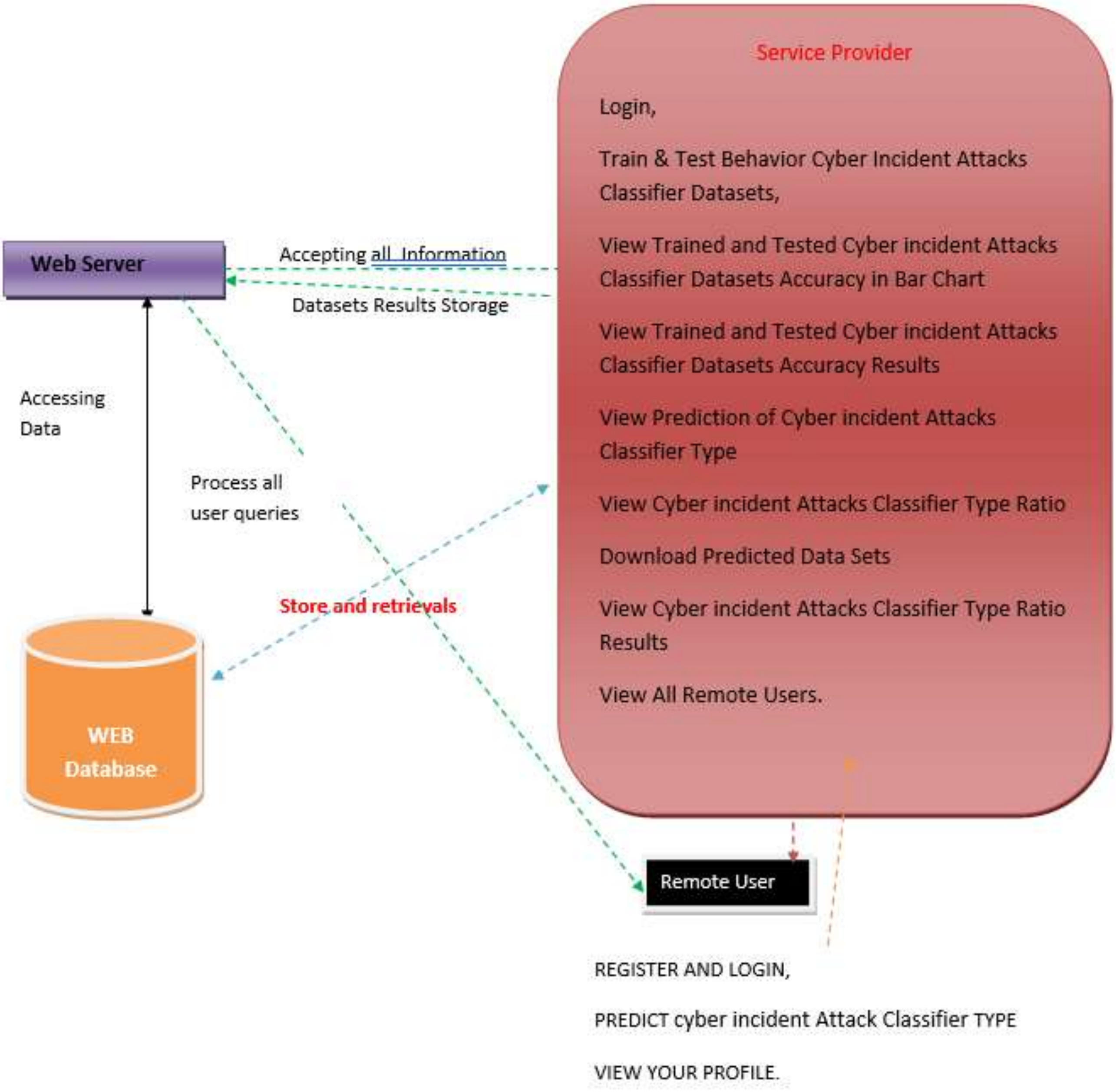
4. Artificial Intelligence in Cyber Threat Mitigation
Overview: Use of AI to recognize and react to threats in anticipation.
Pros: Predictive and adaptive features.
Cons: Data quality-dependent very heavily.

5. Behavioral-Based Intrusion Detection Methods
Overview: Traffic pattern analysis to find complex DDoS behavior.
Pros: Ineffective against clever attacks.
Cons: Vulnerable to false alarms in dynamic systems.

6. Feature Engineering to Improve Detection Precision
Overview: The significance of careful feature selection in ML-based IDS.
Pros: Higher model efficiency and detection accuracy.
Cons: Time-consuming and technically intensive.

7. Ensemble Learning Methods for Intrusion Detection
Overview: Application of ensemble classifiers (e.g., Random Forest, AdaBoost) for efficient detection.
Pros: Improved accuracy and resilience.

Cons: Computationally costly and complicated.


Service Provider
Login,
Train & Test Behavior Cyber Incident Attacks Classifier Datasets,
View Trained and Tested Cyber incident Attacks Classifier Datasets Accuracy in Bar Chart
View Trained and Tested Cyber incident Attacks Classifier Datasets Accuracy Results
View Prediction of Cyber incident Attacks Classifier Type
View Cyber incident Attacks Classifier Type Ratio
Download Predicted Data Sets
View Cyber incident Attacks Classifier Type Ratio Results
View All Remote Users.
Web Server
Accepting all Information
Datasets Results Storage
Accessing Data
Process all user queries
Store and retrievals
WEB Database
Remote User
REGISTER AND LOGIN,
PREDICT cyber incident Attack Classifier TYPE
VIEW YOUR PROFILE.

8. Adversarial Attacks on ML-Based Phishing Detection
Overview: Research on how adversarial examples can evade ML phishing detectors.
Pros: Identifies weaknesses in current models.
Disadvantages: Hard to defend without sophisticated adaptive methods.

3. SYSTEM ARCHITECTURE

The three-tier software architecture (a three-layer architecture) emerged in the 1990s to overcome the limitations of the two-tier architecture.The third tier (middle tier server) is between the user interface (client) and the data management (server) components. This middle tier provides process management where business logic and rules are executed and can accommodate hundreds of users (as compared to only 100 users with the two tier architecture) by providing functions such as queuing, application execution, and database staging.

4. PROPOSED SYSTEM

The technology developed in this project is an Artificial Intelligence (AI) based network intrusion detection system (NIDS). It monitors and protects computer networks using deep learning. Unlike baseline systems that depend on pre-defined attack signatures, this system utilizes historical data for real-time detection of both known and unknown (zero-day) attacks. The system captures network traffic and uses several machine learning models for classification, including Logistic Regression, Random Forest, CNN, and LSTM. Before data is fed into the model, it undergoes cleansing and preprocessing to maximize the model's effectiveness.

- The system preprocesses network traffic using feature selection and encoding methods.
- CNN is used to extract spatial patterns, while LSTM detects sequential attack behaviors.
- AdaBoost Classifier Logistic Regression CNN MLP classfier enhances classification by combining Random Forest with CNN for improved accuracy.
- The model is trained on datasets like NSL-KDD and CICIDS2017 for robust learning.
- A Flask-based web interface allows real-time intrusion monitoring and alerts.
- The system reduces false positives and detects zero-day threats efficiently.
- It integrates with firewall systems to block detected intrusions.
- This system significantly improves accuracy, scalability, and adaptability in intrusion detection.

5. METHODOLOGY

The methodology applied in this project includes several phases from dataset preparation to model deployment and prediction. It combines classical ML models and deep learning architectures for the detection of cyber attacks, specifically DDoS and evasion attacks.

Step 1: Dataset Collection
Source: A custom dataset was created or curated with characteristics like tweet data and labels to denote if data is forensic-related or not.
Objective: Simulate real-life cyber incident scenarios appropriate for training classification models.

Step 2: Data Preprocessing
Feature Extraction: Extract features and save them in variables (x_train, y_train).
Scaling: Standard Scaler is used to standardize data in order to normalize feature ranges.
Label Encoding: If there are categorical variables, they are encoded.

Step 3: Dataset Splitting
The dataset is split in several phases:
Initial split: 80% for training, 20% pre-training.
Additional split: Pre-training set split into pre-train and pre-test (80/20).
Final training split: Training set is divided into:
80% training
20% validation
Later further split into train/test sets for fine-tuning and testing.

Step 4: Model Initialization and Training
Algorithms Used:
Logistic Regression
K-Nearest Neighbors (KNN)
Random Forest
Support Vector Machine (SVM)

AdaBoost Classifier
Multilayer Perceptron (MLP)
Convolutional Neural Networks (CNN)
LSTM (Long Short-Term Memory)

Training:
All the models are trained over the training set.
Models are cross-validated using cross-validation (k-fold).
Hyperparameters are optimized to get the best model settings.
Best performing models are saved using Pickle for later prediction.

Step 5: Model Evaluation
Evaluation Metrics:
Accuracy
Precision
Recall
F1-Score
Confusion Matrix
ROC-AUC
Model Comparison:
Models are compared in terms of their mean absolute errors and detection accuracy.
Top-performing models are chosen for deployment.

Step 6: Model Deployment
Web Interface:
A Flask-based or Django-based app has a real-time prediction and monitoring interface.
Trained models are imported from Pickle files to predict on new or live data.
Functionality Includes:
Real-time cyber incident classification
Accuracy charts and attack type distribution
User management and authorization
Downloadable prediction datasets

Step 7: Real-Time Prediction
Input: New network traffic or log file.

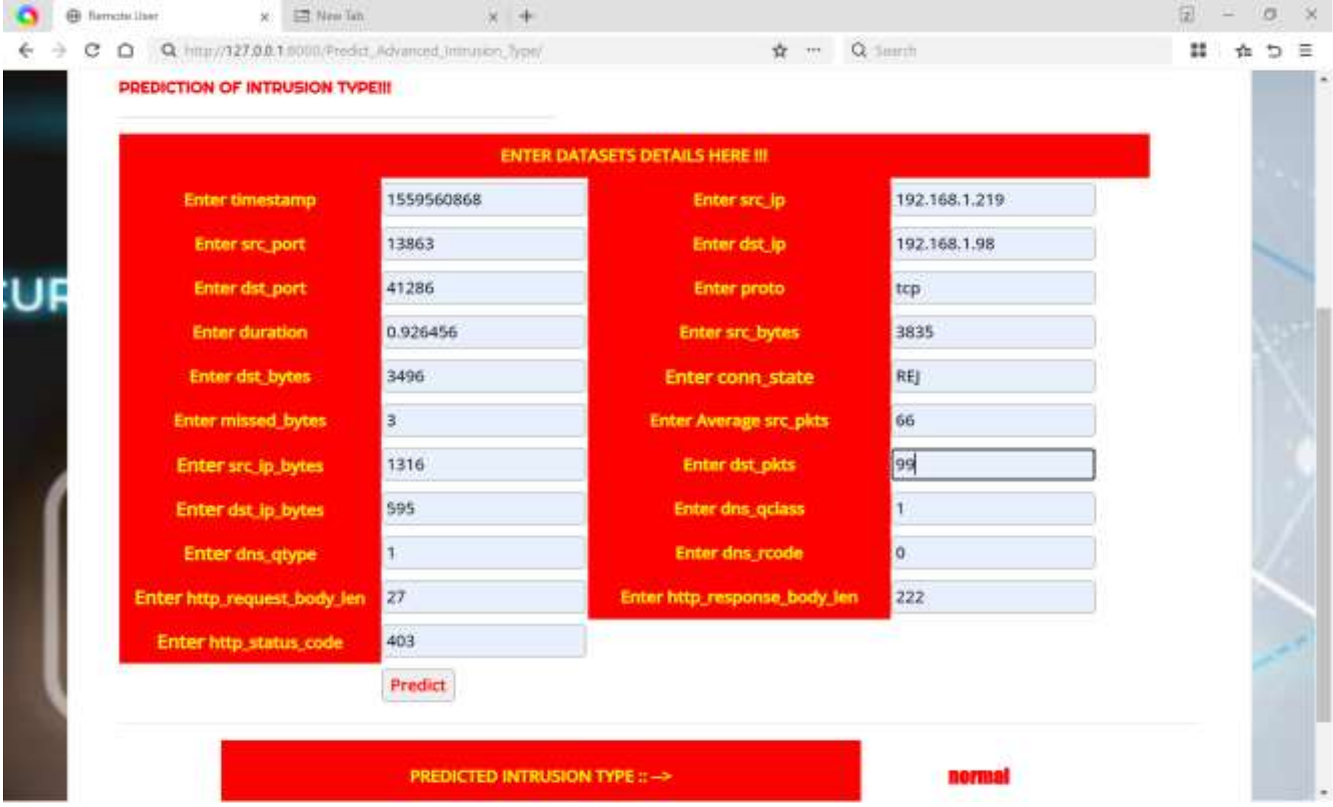
Processing:
Same pipeline as in training is used to preprocess data.
Model classifies the input and makes a prediction of whether the activity is malicious or not.

Output:
Prediction is shown in the web interface.
Alert or classification outcome is logged.

Step 8: Visualization & Reporting
Graphical Outputs:
Bar charts and pie charts comparing accuracy
Attack type ratios and model performance dashboards
Libraries Used:
Matplotlib, Seaborn for plotting
Pandas and NumPy for data processing

6. EXPERIMENTAL RESULTS

Output-1



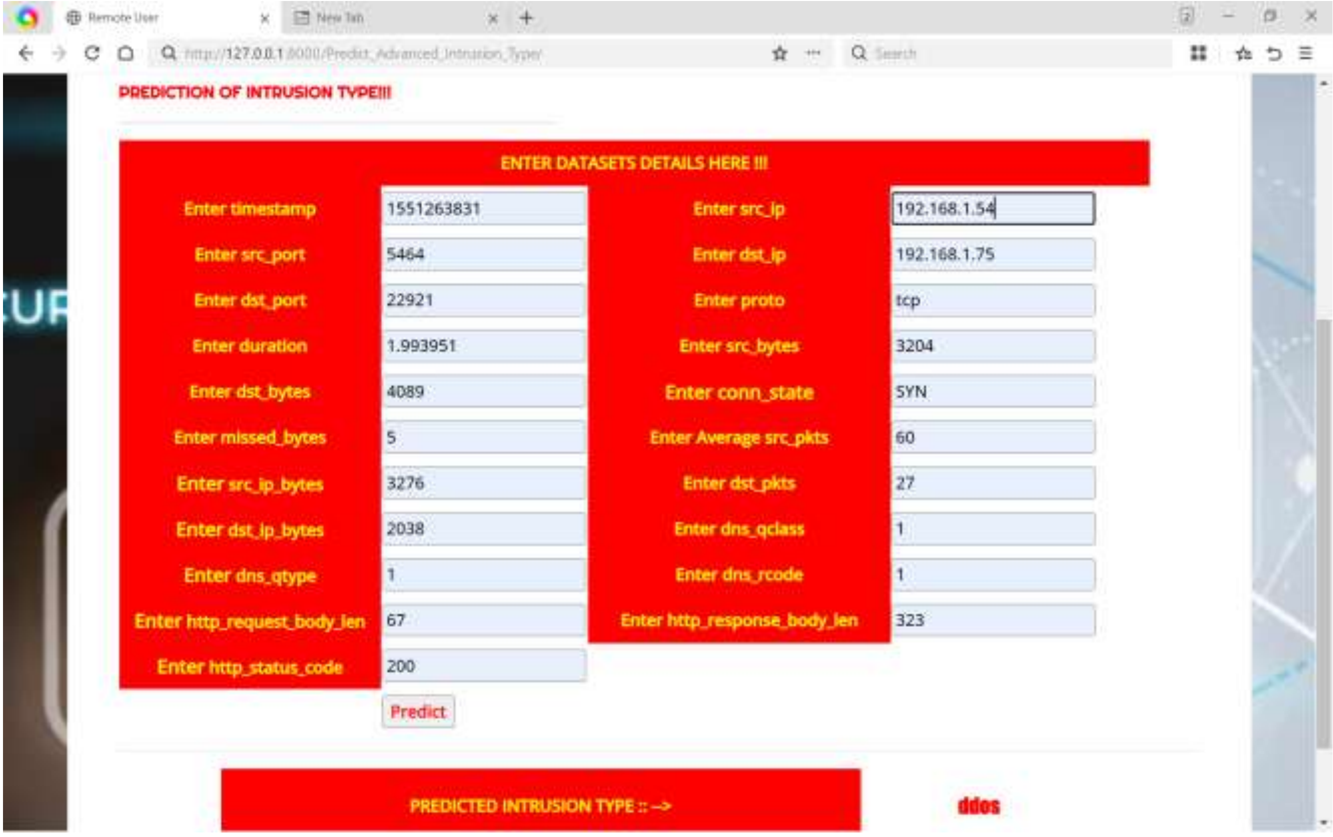
PREDICTION OF INTRUSION TYPE!!!

ENTER DATASETS DETAILS HERE !!!

Enter timestamp	1559560868	Enter src_ip	192.168.1.219
Enter src_port	13863	Enter dst_ip	192.168.1.98
Enter dst_port	41286	Enter proto	tcp
Enter duration	0.926456	Enter src_bytes	3835
Enter dst_bytes	3496	Enter conn_state	REJ
Enter missed_bytes	3	Enter Average src_pkts	66
Enter src_ip_bytes	1316	Enter dst_pkts	99
Enter dst_ip_bytes	595	Enter dns_qclass	1
Enter dns_qtype	1	Enter dns_rcode	0
Enter http_request_body_len	27	Enter http_response_body_len	222
Enter http_status_code	403		

Predict

PREDICTED INTRUSION TYPE :: --> normal



PREDICTION OF INTRUSION TYPE!!!

ENTER DATASETS DETAILS HERE !!!

Enter timestamp	1551263831	Enter src_ip	192.168.1.54
Enter src_port	5464	Enter dst_ip	192.168.1.75
Enter dst_port	22921	Enter proto	tcp
Enter duration	1.993951	Enter src_bytes	3204
Enter dst_bytes	4089	Enter conn_state	SYN
Enter missed_bytes	5	Enter Average src_pkts	60
Enter src_ip_bytes	3276	Enter dst_pkts	27
Enter dst_ip_bytes	2038	Enter dns_qclass	1
Enter dns_qtype	1	Enter dns_rcode	1
Enter http_request_body_len	67	Enter http_response_body_len	323
Enter http_status_code	200		

Predict

PREDICTED INTRUSION TYPE :: --> ddos

7. CONCLUSION

▶ This project demonstrates the effectiveness of machine learning and deep learning in addressing cybersecurity challenges. By providing real-time monitoring and DDoS detection, the tool offers a scalable solution for protecting Indian cyberspace. The comparative analysis ensures the selection of the most suitable model, enhancing overall system performance. With its user-friendly interface and automated alert mechanism, the tool bridges critical gaps in existing systems, paving the way for a more secure digital environment.

8. REFERENCE

[1] F. Song, Y. Lei, S. Chen, L. Fan, and Y. Liu, ‘‘Advanced cyber incident Attacks and mitigations on practical ML-based phishing website classifiers,’’ Int. J. Intell. Syst., vol. 36, no. 9, pp. 5210–5240, Sep. 2022.
[2] S. Anupam and A. K. Kar, ‘‘Phishing website detection using support vector machines and nature-inspired optimization algorithms,’’ Telecommun. Syst., vol. 76, no. 1, pp. 17–32, Jan. 2023.
[3] Log Files - Book of Zeek [Online]. Available: https://docs.zeek.org/en/master/script-reference/log-files.html, Accessed on: Dec. 1, 2023
[4] Gustavsson, V. I. L. H. E. L. M. "Machine Learning For A Networkbased Intrusion Detection System." Examensarbete Elektronık Och Datorteknık, Grundnıvå, 15 Hp (2020).
[5] Svoboda, Jakub, Ibrahim Ghafir, and Vaclav Prenosil. "Network monitoring approaches: An overview." Int J Adv Comput Netw Secur 5.2 (2015): 88-93.
[6] Rodríguez, María, et al. "Evaluation of Machine Learning Techniques for Traffic Flow-Based Intrusion Detection." Sensors 22.23 (2022): 9326.
[7] Andrews, Daniel K., et al. "Comparing machine learning techniques for Zeek log analysis." 2019 IEEE MIT Undergraduate Research Technology Conference (URTC). IEEE, 2021.
[8] Jimenez, Angel. USING MACHINE LEARNING FOR RASPBERRY PI NETWORK INTRUSION DETECTION. Diss. California State Polytechnic University, Pomona, 2022.
[9] Burr, Benjamin, et al. "On the detection of persistent attacks using alert graphs and event feature embeddings." NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2023.
[10] Rodríguez, María, et al. "Evaluation of Machine Learning Techniques for Traffic Flow-Based Intrusion Detection." Sensors 22.23 (2022): 9326