SDN-BASED INTELLIGENT INTRUSION DETECTION SYSTEM (IIDS) USING MACHINE LEARNING

Group ID: RP-24-25J-120

Research Logbook

Sriskandarajah J.P

IT21261978

BSc (Hons) Degree in Information Technology Specialized in Cyber Security

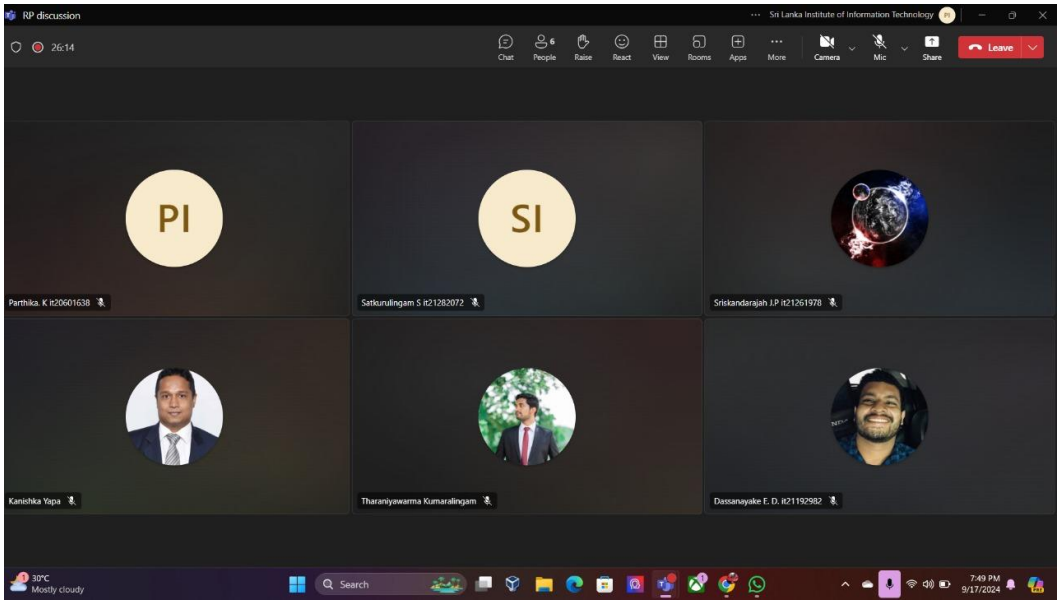
Sri Lanka Institute of Information Technology Sri Lanka

June 2025

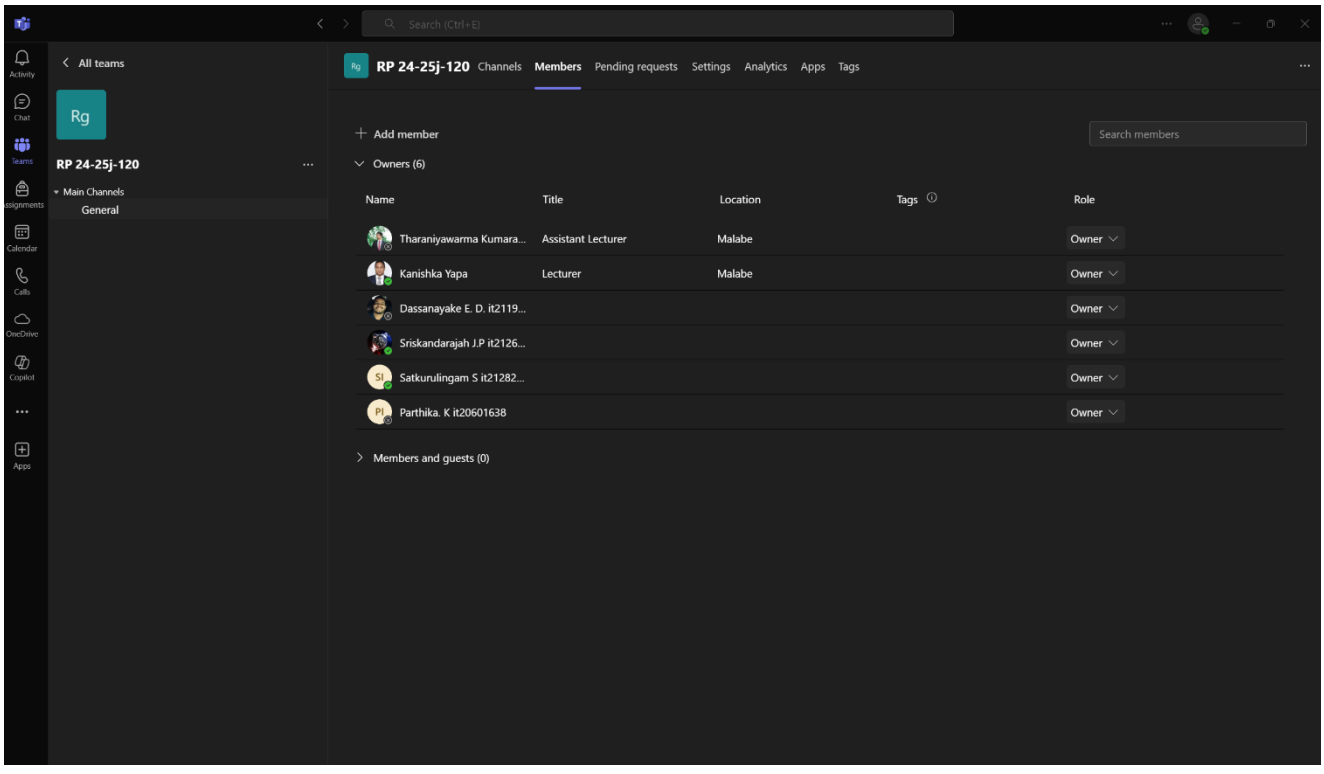
Tasks

► Meeting with the supervisor to discuss the project topic for the first time.

- Physically meet the supervisor.
- Discuss the research project topic area.
- Get the supervisor's ideas about the research topic via Microsoft Teams.

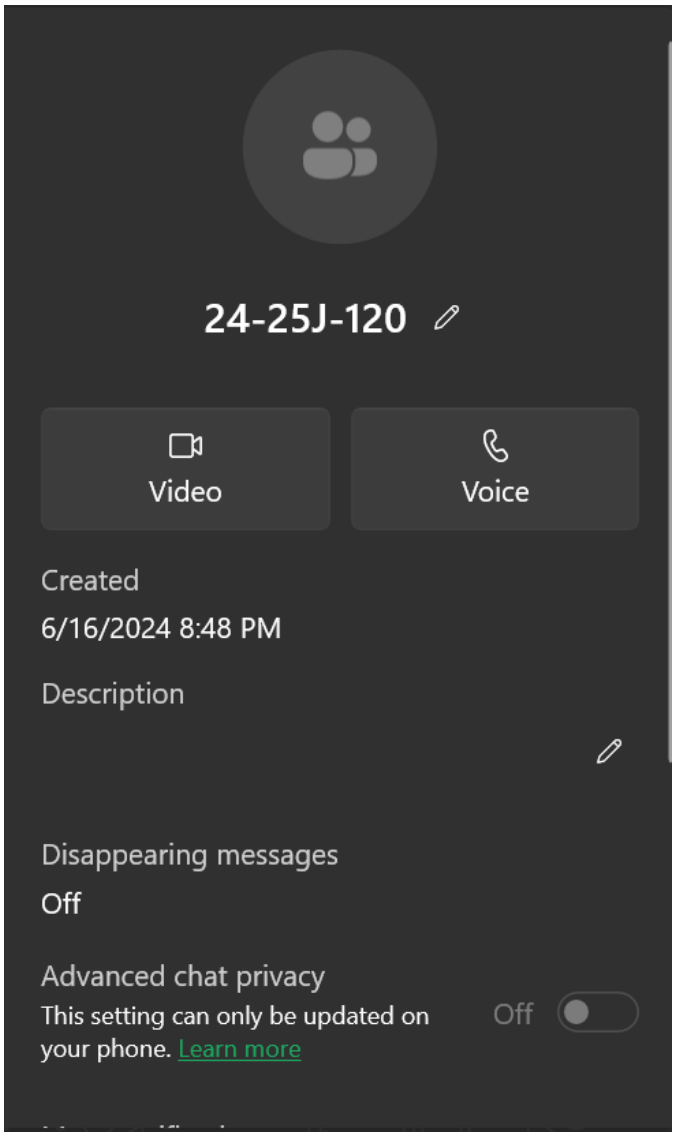


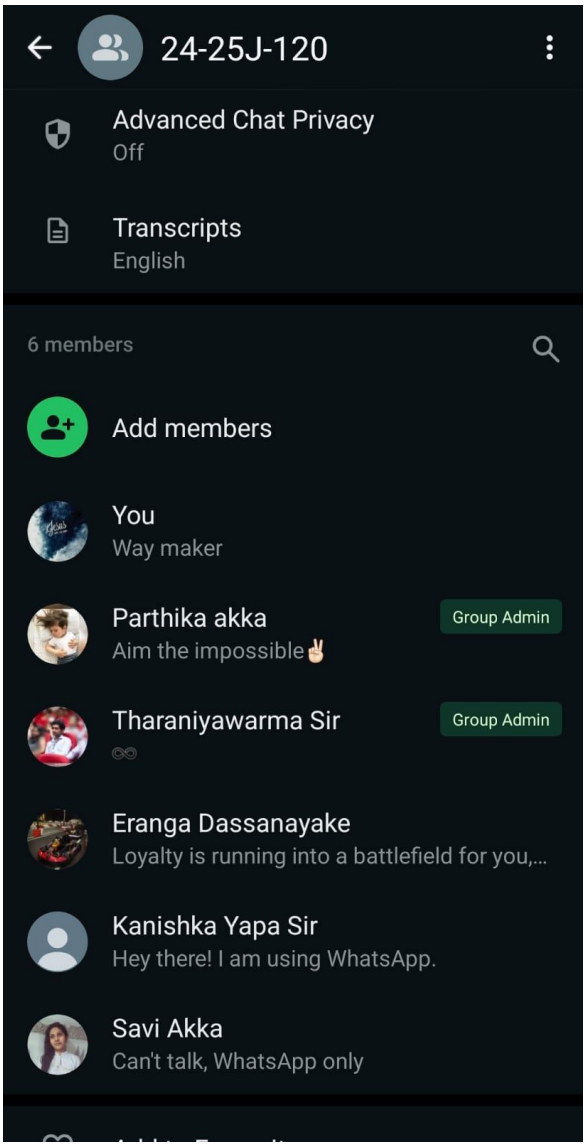
► Created separate MS Teams channels for Conversation.



► **Created the Research Team WhatsApp Group.**

- Discuss the research topic with team members.
- Discuss the research problem.
- Get the solution ideas with brainstorming sessions.
- Identify the main solutions.
- Assign tasks and conversation highlights.





► **Completed Task and Conversation highlights.**

- Creating the proposal document at supervisor request.
- Doing a literature review upon supervisor request.

The development of the SDN-based Intelligent Intrusion Detection System (IIDS) requires specialized expertise and knowledge in several domains: Software-Defined Networking (SDN): Understanding the principles, architecture, and operation of SDN, including SDN controllers, protocols, and network virtualization. Cybersecurity: In-depth knowledge of cybersecurity principles, threat landscapes, attack vectors, and defense mechanisms, particularly in network security. Machine Learning: Proficiency in developing and applying machine learning models for anomaly detection and pattern recognition. Network Traffic Analysis: Expertise in capturing, preprocessing, and analyzing network traffic data to identify features relevant to intrusion detection. System Integration and Development: Skills in integrating various modules and components, ensuring seamless communication and compatibility between the SDN controller, intrusion detection engine, and policy enforcement module. Performance Optimization: Ability to optimize algorithms and systems for real-time processing, high performance, and scalability.

Data Requirements- Network Traffic Data: Extensive datasets of network traffic, both normal and malicious, for training and testing machine learning models. Cybersecurity Threat Intelligence: Access to threat intelligence feeds and databases to update and refine detection models and security policies. System Logs and Event Data: Logs from network devices, servers, and security appliances to provide comprehensive visibility into network activities and potential threats. Simulated Attack Scenarios: Synthetic data representing different types of cyberattacks for testing the system's detection and response capabilities.

The proposed solution involves the development of an SDN-based Intelligent Intrusion Detection System (IIDS) that integrates machine learning for advanced threat detection and adaptive security policy enforcement. SDN-based Cybersecurity Information Collection and Management involves designing and developing a system for centralized and dynamic collection of cybersecurity-related information using SDN. It will integrate with network devices, sensors, and logs to enable real-time data collection and processing. SDN-based Cybersecurity Action and Response focuses on developing mechanisms to translate SDN outputs into actionable security responses, such as blocking malicious traffic or reconfiguring network paths. It aims to automate response strategies based on predefined security policies. Machine Learning-based Intrusion Detection Engine utilizes advanced machine learning algorithms to detect anomalies and potential threats in real-time. It involves data collection, preprocessing, model development, and integration with the SDN controller for continuous monitoring and analysis of network traffic. Adaptive Security Policy Enforcement Module develops a dynamic policy enforcement engine that adjusts security policies in real-time based on detected threats and network conditions. It ensures seamless integration with the SDN controller and the intrusion detection engine to provide a integrated and responsive security framework.

► **Proposed Machine Learning based Intrusion Detection Engine System**.

Sriskandarajah J.P	Develop Machine Learning based Intrusion Detection Engine	**1. Data collection and preprocessing** Gathering extensive network traffic data from reliable sources, Preprocessing the data includes cleaning it to remove noise, handling missing values, and normalizing features to ensure consistency and ensures that the dataset is in an optimal format for training machine learning models, enhancing their performance and reliability. **2. Develop machine learning model** Selecting the appropriate algorithms for learning, trained using the preprocessed dataset, where	Real time intrusion detection in SDN environment By integrating machine learning models directly with the SDN controller, the system can dynamically and continuously analyze network traffic, detect anomalies, and respond to threats in real-time. This approach significantly improves traditional intrusion detection systems by offering enhanced visibility, centralized control, and rapid adaptability to new and
		they learn to distinguish between normal and malicious network traffic based on the extracted features, develop models that not only achieve high accuracy but also generalize well to unseen data, thereby effectively identifying intrusions in real-world network environments **3. Integrate the trained model with SDN controller** use the machine learning models to continuously monitor and inspect traffic for anomalies and potential threats. This real-time integration ensures that the intrusion detection system can promptly identify and respond to security incidents, leveraging the agility and centralized control of the SDN architecture to enhance overall network security. **4. Evaluate and optimize the model** testing the models on separate validation and test	evolving threats. The integration of SDN's centralized management capabilities with sophisticated machine learning algorithms provides a robust and efficient solution for maintaining network security through continuous monitoring and adaptive policy enforcement.
		datasets to assess their accuracy, precision, recall, and overall effectiveness in detecting intrusions, Performance metrics are analyzed to identify any areas for improvement. Optimization techniques are applied to enhance the models' performance. Additionally, the system is subjected to various attack scenarios to test its robustness and reliability. The goal is to achieve a high-performing, efficient, and scalable intrusion detection engine that can adapt to evolving threats and maintain robust network security in real-time.	

► Completed Task and Conversation Highlights.

- Determining the components for each member and discussing with the Supervisor.
- Fine tuning the scope for each component.
- Discussing the proposed components with co-supervisor.
- Find the Related research paper for individual SDN Component.
- Get a full idea of each research paper.
- Mark down the not covering SDN areas in these research papers.
- Identify the novelty parts of each individual component.
- Creating the Topic Assignment Form (TAF)
- Getting the approval from the Supervisor.

Implementation of SDN-based IDS to protect Virtualization Server against HTTP DoS attacks

Saifudin Usman
Dept. of Information and Computer
Politeknik Elektronika Negeri Surabaya
Surabaya, Indonesia
saifudinu@politap.ac.id

Idris Winarno
Dept. of Information and Computer
Politeknik Elektronika Negeri Surabaya
Surabaya, Indonesia
idris@pens.ac.id

Amang Sudarsono
Dept. of Information and Computer
Politeknik Elektronika Negeri Surabaya
Surabaya, Indonesia
amang@pens.ac.id

Abstract— Virtualization and Software-defined Networking (SDN) are emerging technologies that play a major role in cloud computing. Cloud computing provides efficient utilization, high performance, and resource availability on demand. However, virtualization environments are vulnerable to various types of intrusion attacks that involve installing malicious software and denial of services (DoS) attacks. Utilizing SDN technology, makes the idea of SDN-based security applications attractive in the fight against DoS attacks. Network intrusion detection system (IDS) which is used to perform network traffic analysis as a detection system implemented on SDN networks to protect virtualization servers from HTTP DoS attacks. The experimental results show that SDN-based IDS is able to detect and mitigate HTTP DoS attacks effectively.

Keywords—SDN, IDS, HTTP DoS, Virtualization, Cloud.

I. Introduction

The rapid growth of telecommunications equipment, devices connected to the internet, cloud-based services, and server virtualization are some of the trends affecting the development of Software-defined Networking (SDN). Virtualization and SDN play a major part in cloud computing by making effective use of the available hardware, provides high performance, and on-demand availability of resources. It increases the availability of the system and many were developed to resilient servers [1].

Other side of this rapid growth not only represents technological advances but also opportunities for attackers to take advantage of the rapid infrastructure to attack a large number of network resources and information assets. One of the most common types of attacks we encounter is Distributed Denial of Service (DDoS). DDoS attacks aim to make a server, service or infrastructure unavailable. These designs used to detect attacks [2]. This paper focuses the on enabling and testing SDN-based IDS to protect virtualization servers against DDoS attacks. We use SDN-based IDS built on bare metal virtualization using Proxmox VE [4], with Ryu [5]as an SDN controller running on linux container and Snort [6] as IDS running on virtual machine.

II. Related Work

Winarno et. al [1] in his research entitled "Simulating Resilient Server using Software-Defined Networking", conducted simulation resilient server from perspective of network devices using SDN. Using shell script to detect number of TCP-based connection from the same IP address by utilizing iptables, and the tool used to simulate attacks is slowloris.

Hsiao-Chung Lin, et al [2] in his research entitled "Implementation of an SDN-based Security Defense Mechanism Against DDoS Attacks", conducted implementation of SDN-based network security defense against DDoS attacks. The defense system is built based on the mechanism of gathering traffic information on each Openflow Switch by using the sFlow-RT toolset such as sFlow Agent and sFlow Collector which work as agents and information collectors. All traffic information that has been collected by sFlow Collector is then sent to IDS and Controller for further analysis of abnormal traffic. System testing is done with DDoS simulation using hping3 to generate ICMP flood traffic to attack a host in the network connected directly to the switch, indicating that the system can identify and drop the ICMP flood packet.

Pedro Manso, et al [3] in his research entitled "SDN Based Intrusion Detection System for Early Detection and Mitigation of DDoS Attacks", conducted application of SDN

RESEARCH ARTICLE

DeMi: A Solution to Detect and Mitigate DoS Attacks in SDN

LUBNA FAYEZ ELIYAN[1] AND ROBERTO DI PIETRO[2], (Fellow, IEEE)
[1]College of Science and Engineering, ICT Division, Hamad Bin Khalifa University, Doha, Qatar
[2]CEMSE Division, RC3 Center, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia
Corresponding author: Lubna Fayez Eliyan (leliyan@hbku.edu.qa)

This work was supported by the Award Thematic Research Grant Program from Hamad Bin Khalifa University (HBKU), Office of the Vice President for Research, Doha, Qatar, under Grant VPR-TG01-009.

ABSTRACT Software-defined networking (SDN) is becoming more and more popular due to its key features of scalability and flexibility, simplifying network management and enabling innovations in the network architecture and protocols. In SDNs, the most crucial part is the controller, tasked with managing the entire network and configuring routes. Given its critical role, a failure or problem occurring at the controller may degrade and even collapse the entire SDN. A typical threat controllers are subject to is a Denial of Service (DoS) attack. To cope with the above-introduced threat, in this paper we propose a lightweight DoS attack detection and mitigation method (DeMi) as well as a heavy-load management module. The proposed solution for detection leverages a sample entropy approach coupled with an adaptive dynamic threshold considering an exponentially weighted moving average (EWMA); the mitigation approach is based on proof of work (PoW) combined with flow rule installations; and, the heavy-load management method implements a scheduling approach at the SDN controller. Results are staggering: for instance, when DeMi is deployed, in an attack scenario the number of exchanged control packets is roughly similar to the attack-free scenario—without DeMi, the number of control packets in the network is 2,7 times more than what experienced in an attack-free setting. As per the number of re-transmitted packets, again, DeMi is able to achieve a re-transmission rate similar to an attack-free scenario—without DeMi the of packets that need to be re-transmitted is roughly 3,7 times the number of packets re-transmission occurring in an attack-free scenario. Moreover, DeMi does not block legitimate traffic, contrary to other solutions in the literature. The novelty of the approach, the demonstrated complete end-to-end solution, and the quality of the achieved experimental results, other than being interesting on their own, do pave the way for further research in this field.

INDEX TERMS SDN, DoS, DDoS, security, detection, mitigation, load balancing, proof-of-work.

I. INTRODUCTION

The internet has revolutionized the development of communication and computer technologies. Cisco predicted that, ever-changing landscape of users, resources, and services. Such an infrastructure has several new demands, such as scalability, security, flexibility, and reliability [3]. Traditional

imperative of embedding machine learning elements.

4) Scalability and Performance: Machine learning algorithms showcased scalability, seamlessly accommodating escalating network traffic loads while maintaining unaltered precision. The side-by-side evaluation against established methodologies underscored the prowess of machine learning in orchestrating sprawling network environments. The enrichment of network functionality was achieved through automated resource allocation and optimization, even in intricate and heterogeneous traffic contexts.

5) Discussion of Implications: The outcomes of the conducted experiments illuminate the potential of amalgamating intelligent traffic pattern analysis and optimization, facilitated by machine learning, within the backdrop of SDN-empowered network infrastructures. The methodology not only bolsters network security and resource harnessing but also empowers network administrators with anticipatory decision-making competencies. These outcomes possess far-reaching implications for network governance, underscoring the notion that the amalgamation of cutting-edge technologies adeptly meets the challenges posed by contemporary network infrastructures.

V. Conclusion

Traffic pattern analysis is a beneficial factor in maintaining secure traffic and optimal performance endurance. Our proposed work explores diverse machine learning algorithms for training and testing, integrating them with an intuitive interface for user-friendly interaction. Experiments assessed system performance, classification accuracy, and prediction time. Our study began by acknowledging challenges in modern network management: voluminous data, complex applications, and shifting traffic patterns. Incorporating machine learning into Software-Defined Networking emerged as a way to decode and optimize patterns on the fly, resulting in a more responsive and efficient network. Our contributions emerged via unsupervised and supervised learning, revealing insights into network behaviors and congestion prediction. Dynamic optimization strategies, combining machine learning and SDN, underpinned resource allocation, bandwidth management, and routing for seamless data transfer. These findings have tangible implications, as our intelligent SDN system enhances efficiency and user experiences. Looking ahead, security-focused machine learning, advanced traffic pattern analysis, and multi-objective optimization hold promise for network enhancements. This work highlights the fusion of machine learning and SDN, shaping connected networks that are intelligent and tailored to users. Our journey towards an optimized network future has begun.

References

[1] Feng Wang, Heyu Wang, Baohua Lei, and Wenting Ma. A research on high-performance sdn controller. In *2014 International Conference on Cloud Computing and Big Data*, pages 168–174, 2014.

[2] Won-Ju Eom, Yeong-Jun Song, Chang-Hoon Park, Jeong-Keun Kim, Geon-Hwan Kim, and You-Ze Cho. Network traffic classification using ensemble learning in software-defined networks. In *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 089–092, 2021.

[3] AysÂe Rumeysa Mohammed, Shady A. Mohammed, and Shervin Shirmohammadi. Machine learning and deep learning based traffic classification and prediction in software defined networking. In *2019 IEEE International Symposium on Measurements Networking (MN)*, pages 1–6, 2019.

[4] Jungmin Kwon, Daeun Jung, and Hyunggon Park. Traffic data classification using machine learning algorithms in sdn networks. In *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1031–1033, 2020.

[5] Afeez Ajani Afuwape, Ying Xu, Joseph Henry Anajemba, and Gautam Srivastava. Performance evaluation of secured network traffic classification using a machine learning approach. *Computer Standards Interfaces*, 78:103545, 2021.

[6] Muhammad Aslam, Dengpan Ye, Aqil Tariq, Muhammad Asad, Muhammad Hanif, David Ndzi, Samia Allaoua Chelloug, Mohamed Abd Elaziz, Mohammed AA Al-Qaness, and Syeda Fizzah Jilani. Adaptive machine learning based distributed denial-of-services attacks detection and mitigation system for sdn-enabled iot. *Sensors*, 22(7):2697, 2022.

[7] Ons Aouedi, Kandaraj Piamrat, and Benoît Parrein. Intelligent traffic management in next-generation networks. *Future internet*, 14(2):44, 2022.

[8] Rashid Amin, Elisa Rojas, Aqsa Aqdus, Sadia Ramzan, David Casillas-Perez, and Jose M Arco. A survey on machine learning techniques for routing optimization in sdn. *IEEE Access*, 9:104582–104611, 2021.

[9] K Tamil Selvi and R Thamilselvan. An intelligent traffic prediction framework for 5g network using sdn and fusion learning. *Peer-to-Peer Networking and Applications*, 15(1):751–767, 2022.

[10] Osama Mohammed and Jalil Kianfar. A machine learning approach to short-term traffic flow prediction: A case study of interstate 64 in missouri. In *2018 IEEE International Smart Cities Conference (ISC2)*, pages 1–7, 2018.

[11] P Cheskidov, V Varkentin, and A Shults. A model for classifying network traffic using reinforcment learning. In *2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, pages 1–5. IEEE, 2020.

[12] Alfréd Csikós, Themistoklis Charalambous, Hamed Farhadi, Balázs Kulcsár, and Henk Wymeersch. Network traffic flow optimization under performance constraints. *Transportation Research Part C: Emerging Technologies*, 83:120–133, 2017.

[13] Konstantina Fotiadou, Terpsichori-Helen Velivassaki, Artemis Voulkidis, Dimitrios Skias, Sofia Tsekeridou, and Theodore Zahariadis. Network traffic anomaly detection via deep learning. *Information*, 12(5):215, 2021.

[14] Ankur Nahar and Debasis Das. Secr: Sdn-enabled spectral clustering-based optimized routing using deep learning in vanet environment. In *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, pages 1–9. IEEE, 2020.

SLIIT UNI
THE KNOWLEDGE UNIVERSITY

IT4010 – Research Project - 2024

Topic Assessment Form

Project ID : 24-25J-120

1. Topic (12 words max)

SDN-based Intelligent Intrusion Detection System (IIDS) using Machine Learning

2. Research group the project belongs to

Computing Infrastructure and Security (CIS)

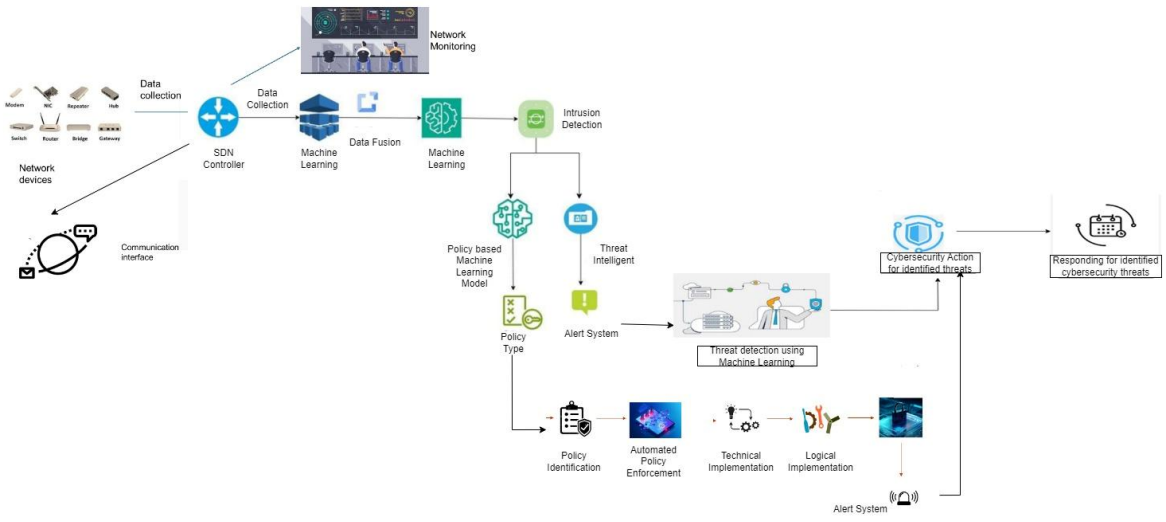
3. Research area the project belongs to

Cyber Security (CS)

4. If a continuation of a previous project:

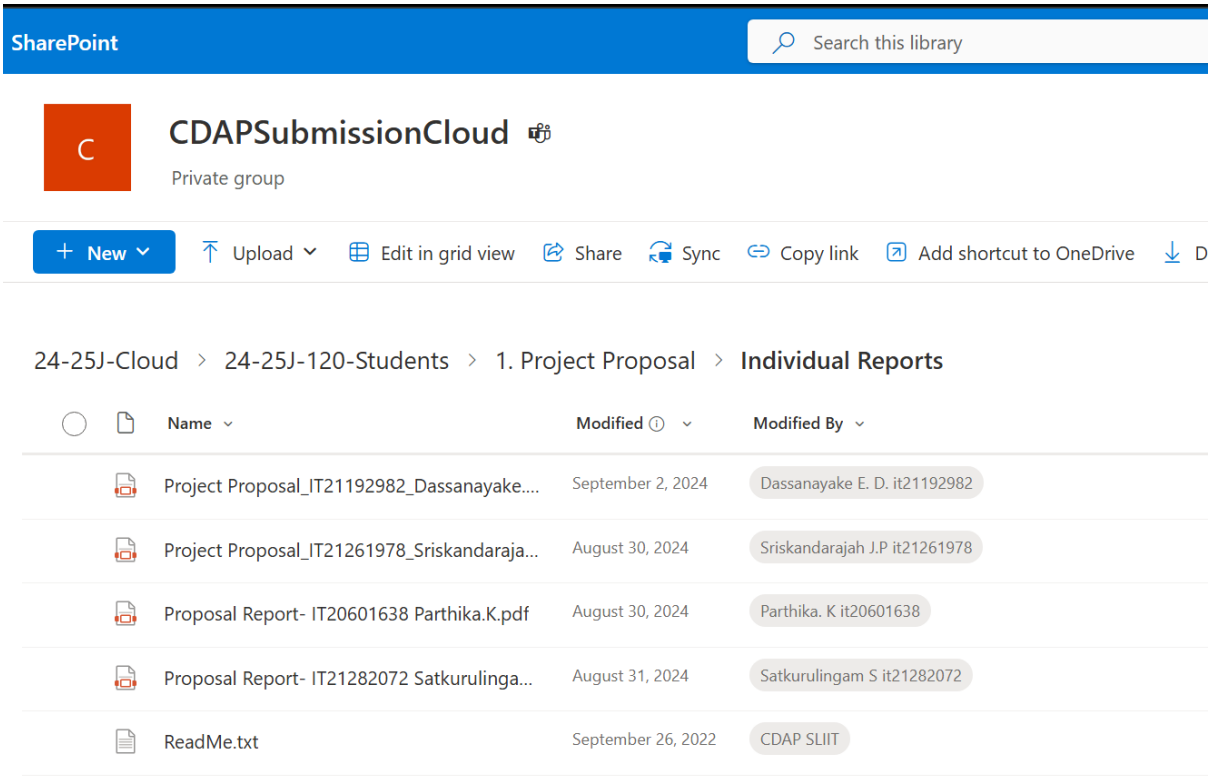
► Complete Task and Conversation Highlights

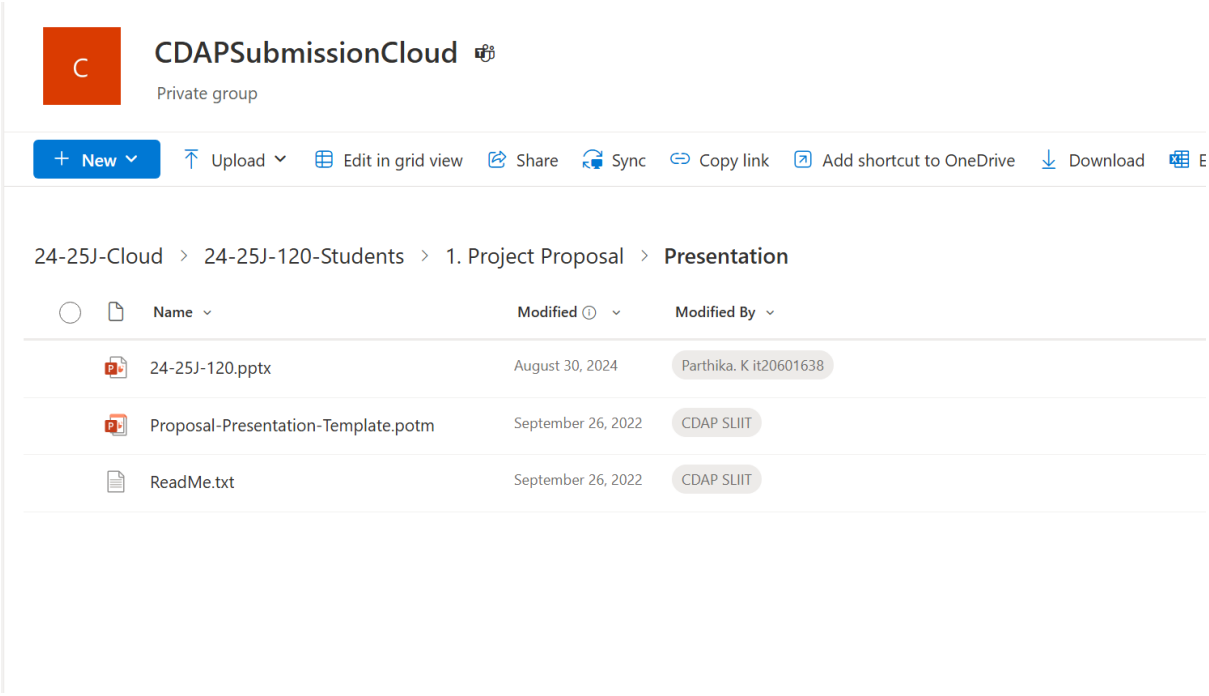
- Dividing the software components.
- Doing a thorough background investigation on each component.
- Creating the system architecture diagram of the proposed system.
- Discussing architecture with the Supervisor and Co-supervisor physically meeting.

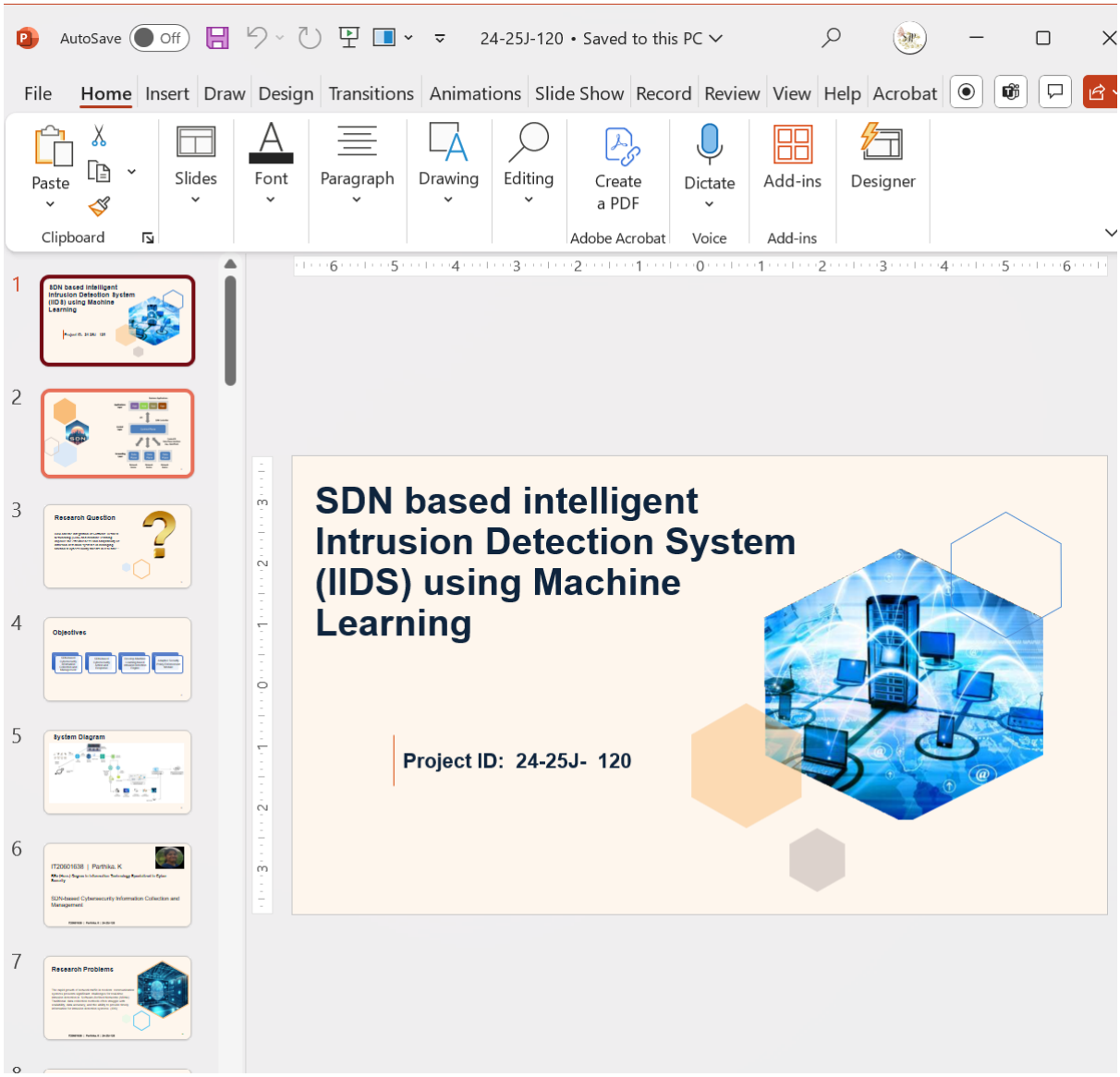


► **Complete Task and Conversation Highlights.**

- Finalizing the components and getting ready for the progress presentation.
- Discussion the project with the supervisor before the proposal presentation.

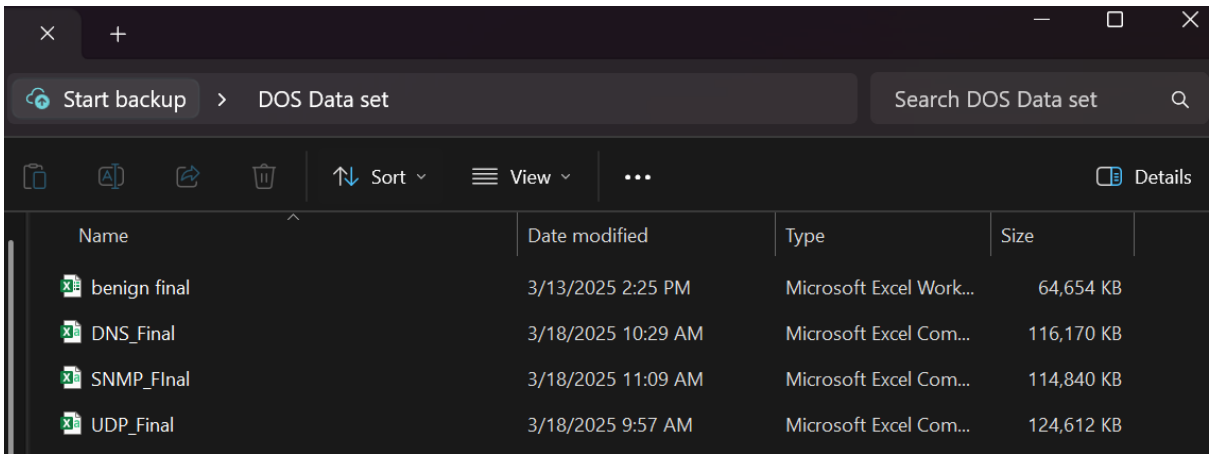






► Completed Task and Conversation Highlights

- Finding the sample dataset until SDN system develop.
- Discussing with the co-supervisor the potential model and its accuracy and which model we should proceed with for the prediction.



jupyter DNS Last Checkpoint: 2 months ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python 3 (ip

```
Requirement already satisfied: nest_asyncio in e:\anaconda\lib\site-packages (1.6.0)
Note: you may need to restart the kernel to use updated packages.
```

[17]:
```
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import classification_report, confusion_matrix,ConfusionMatrixDisplay, roc_curve, auc
from imblearn.over_sampling import SMOTE
import matplotlib.pyplot as plt
import pickle
import os
from fastapi import FastAPI
import nest_asyncio
import uvicorn
import warnings
warnings.filterwarnings("ignore")
```

[18]:
```
app = FastAPI()
```

[20]:
```
df = pd.read_csv("data/DNS_Final.csv")
```

[21]:
```
print(df.columns)
```

```
Index([' Destination IP', ' Source IP', ' Protocol', ' Destination Port',
       ' Fwd Packet Length Max', ' Fwd Packet Length Min',
       ' Fwd Packet Length Mean', ' Flow Packets/s', ' Flow IAT Std',
       ' Min Packet Length', ' Avg Fwd Segment Size', ' Average Packet Size',
       ' Packet Length Mean', 'Flow Bytes/s', ' Subflow Fwd Bytes',
       ' Max Packet Length', ' act_data_pkt_fwd',
       'Total Length of Fwd Packets', ' Bwd IAT Min', ' Inbound', ' Label'],
      dtype='object')
```

[25]:
```
df.columns = df.columns.str.strip()  # Remove spaces
```

[29]:
```
# Separate features and target
X = df.drop('Label', axis=1)
y = df['Label']

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

jupyter

File View Settings Help

Files Running

Select items to perform actions on them. New Upload

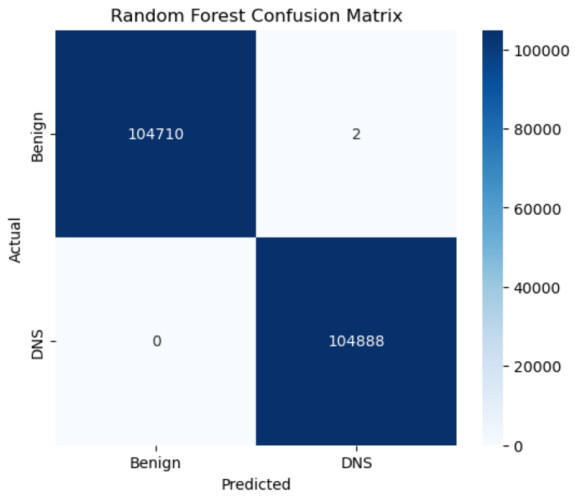
/ rp-sdn-ids / DoS IDS ML /

Name	Last Modified	File Size
data	2 months ago	
pkl_files	2 months ago	
tested	2 months ago	
DNS.ipynb	2 months ago	240 KB
SNMP.ipynb	2 months ago	260.1 KB
UDP.ipynb	2 months ago	256.3 KB
dns_attack_detector.pkl	2 months ago	56.7 KB
snmp_attack_detector.pkl	2 months ago	48.7 KB
udp_attack_detector.pkl	2 months ago	33.3 KB

[102]:
```python
def plot_confusion_matrix(y_true, y_pred, labels, title):
    cm = confusion_matrix(y_true, y_pred)

    plt.figure(figsize=(6,5))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=labels, yticklabels=labels)  # fmt='d' forces integers
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.title(title)
    plt.show()

# Call function
plot_confusion_matrix(y_test, rf_pred, ['Benign', 'DNS'], 'Random Forest Confusion Matrix')
```



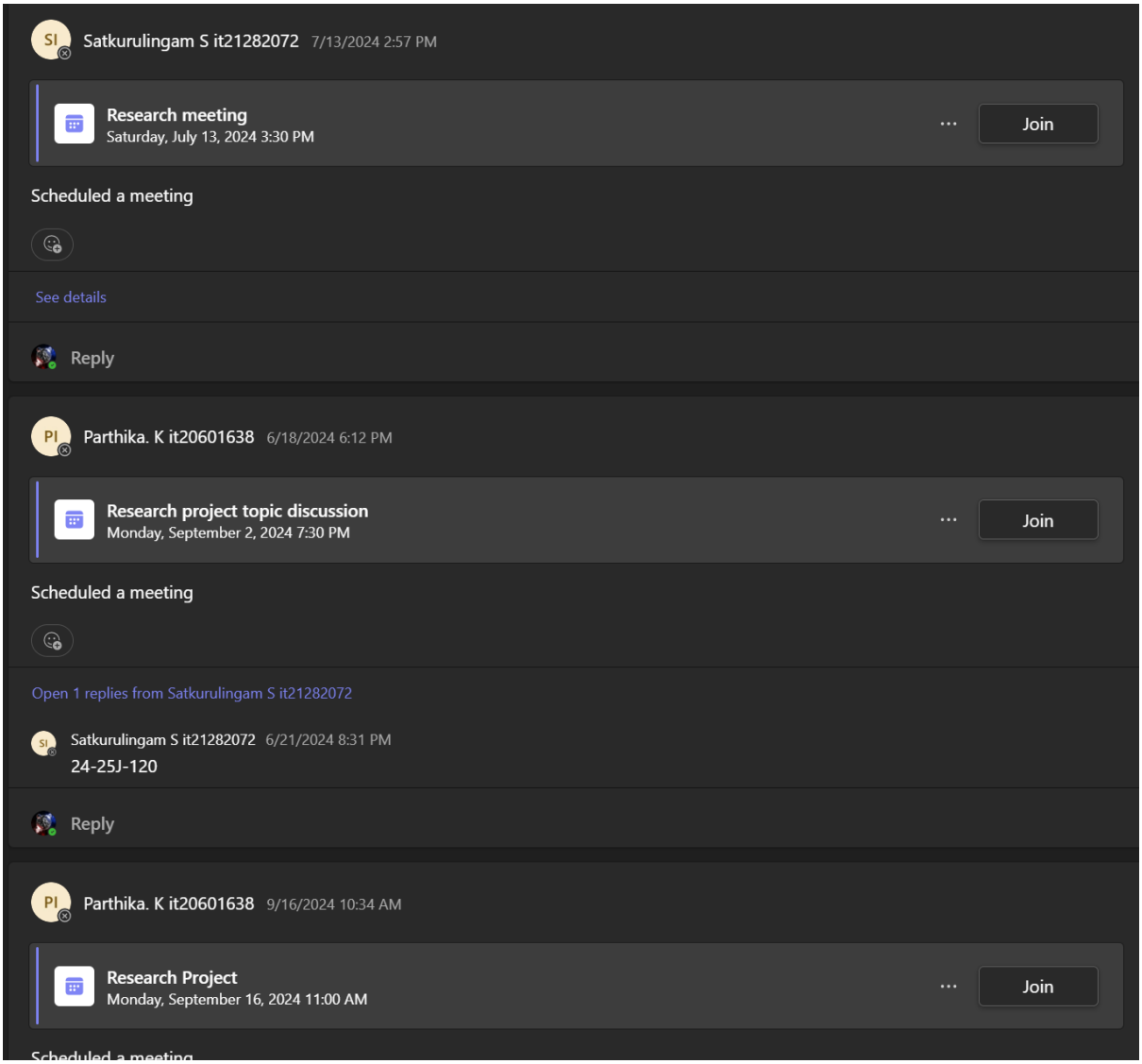
[122]:
```python
features = {
        'Source IP': '192.168.1.100',
        'Destination IP': '8.8.8.8',
        'Protocol': 17,
        'Destination Port': 53,
        'Fwd Packet Length Max': 60,
        'Fwd Packet Length Min': 60,
        'Fwd Packet Length Mean': 60,
        'Flow Packets/s': 10.0,
        'Flow IAT Std': 100.0,
        'Min Packet Length': 60,
        'Avg Fwd Segment Size': 60,
        'Average Packet Size': 64.0,
        'Packet Length Mean': 62.0,
        'Flow Bytes/s': 600.0,
        'Subflow Fwd Bytes': 120,
        'Max Packet Length': 60,
        'act_data_pkt_fwd': 2,
        'Total Length of Fwd Packets': 120,
        'Bwd IAT Min': 50.0,
        'Inbound': 0
    }
```

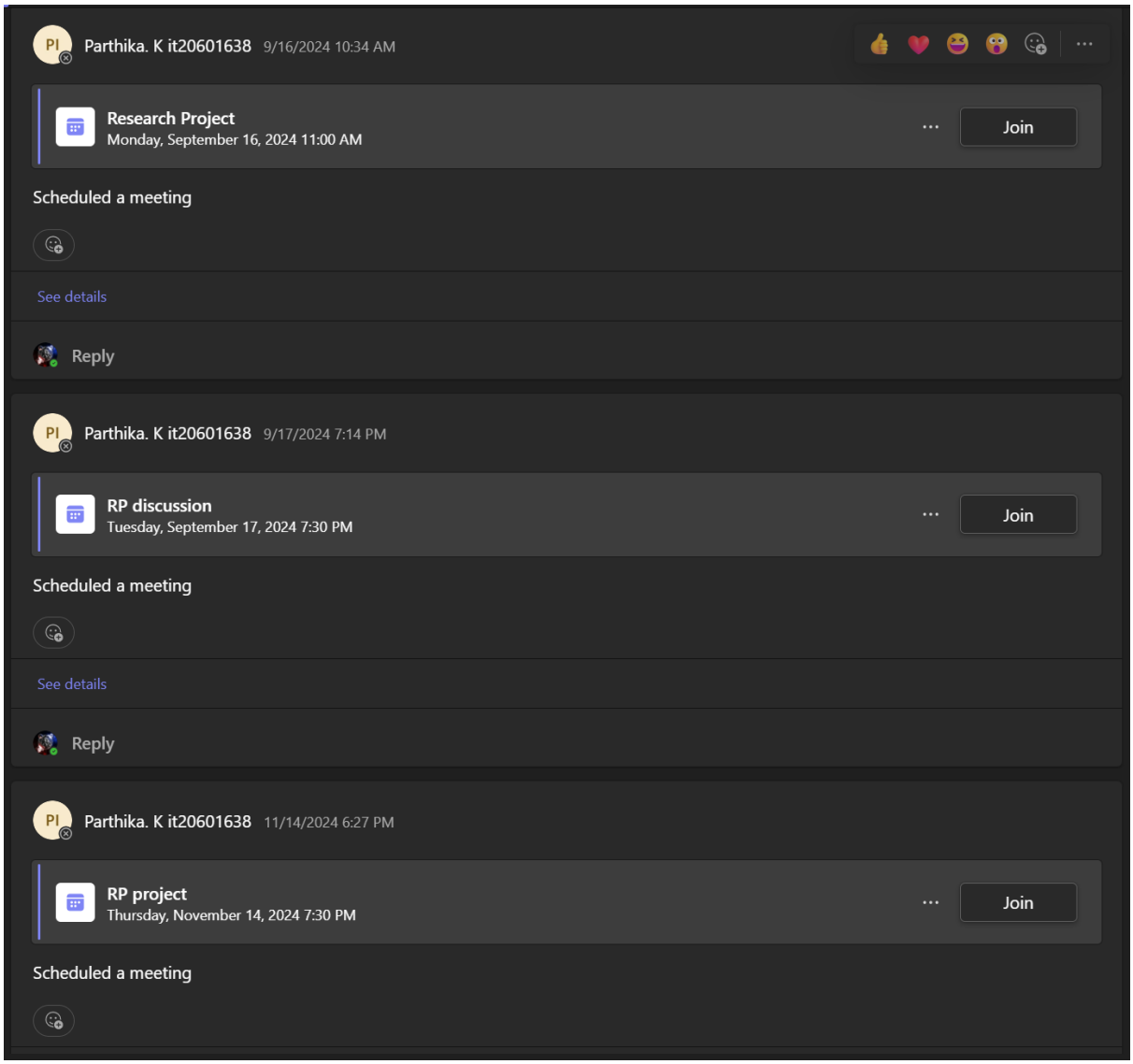
[124]:
```python
input_data = pd.DataFrame([features])
result = make_prediction(input_data, rf, scaler, label_encoders, columns_to_encode)
print("Prediction result:", result)
```

```
Prediction result: {'prediction': 0, 'probability': 0.05, 'is_attack': False}
```

► Complete Tasks and Conversation Highlights

- Meeting with the research team and deciding the implementation milestone on Microsoft Teams.




PI
Parthika. K it20601638 9/16/2024 10:34 AM
Research Project
Monday, September 16, 2024 11:00 AM
Join
Scheduled a meeting
See details
Reply
PI
Parthika. K it20601638 9/17/2024 7:14 PM
RP discussion
Tuesday, September 17, 2024 7:30 PM
Join
Scheduled a meeting
See details
Reply
PI
Parthika. K it20601638 11/14/2024 6:27 PM
RP project
Thursday, November 14, 2024 7:30 PM
Join
Scheduled a meeting

PI Parthika. K it20601638 12/3/2024 5:49 PM

RP
Wednesday, December 4, 2024 1:30 AM
Join

Scheduled a meeting

See details

Reply

PI Parthika. K it20601638 12/4/2024 8:21 PM

RP
Wednesday, December 4, 2024 8:30 PM
Join

Scheduled a meeting

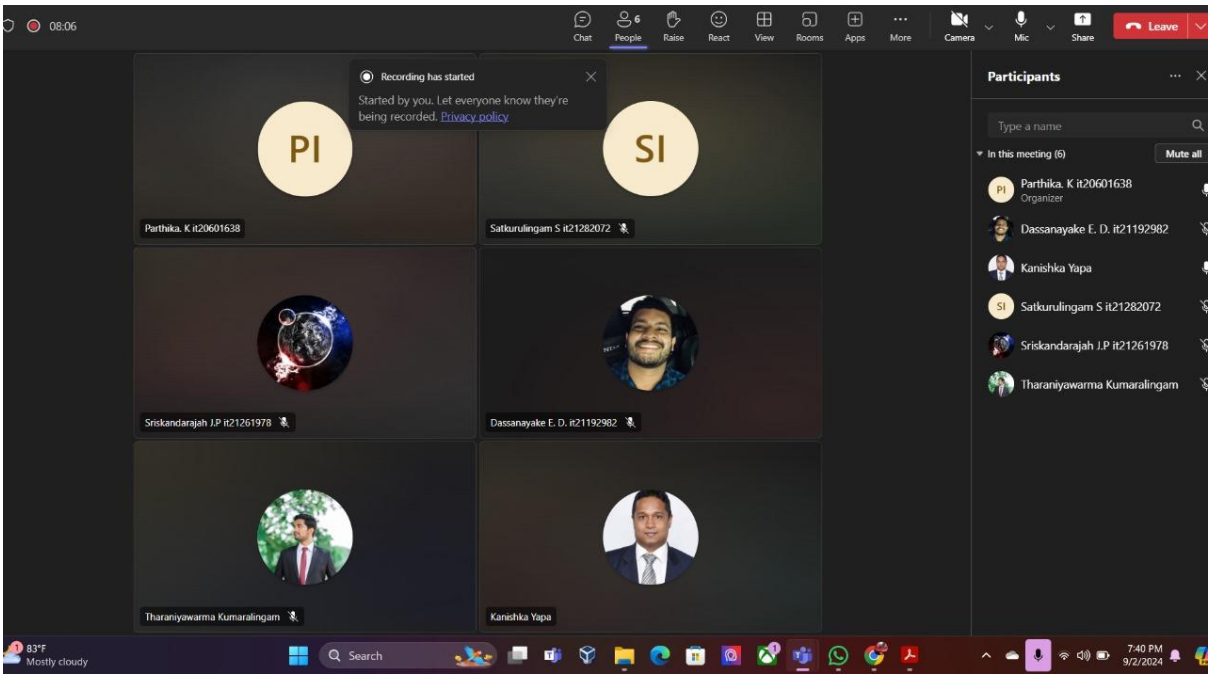
See details

Reply

PI Parthika. K it20601638 2/16 3:38 PM

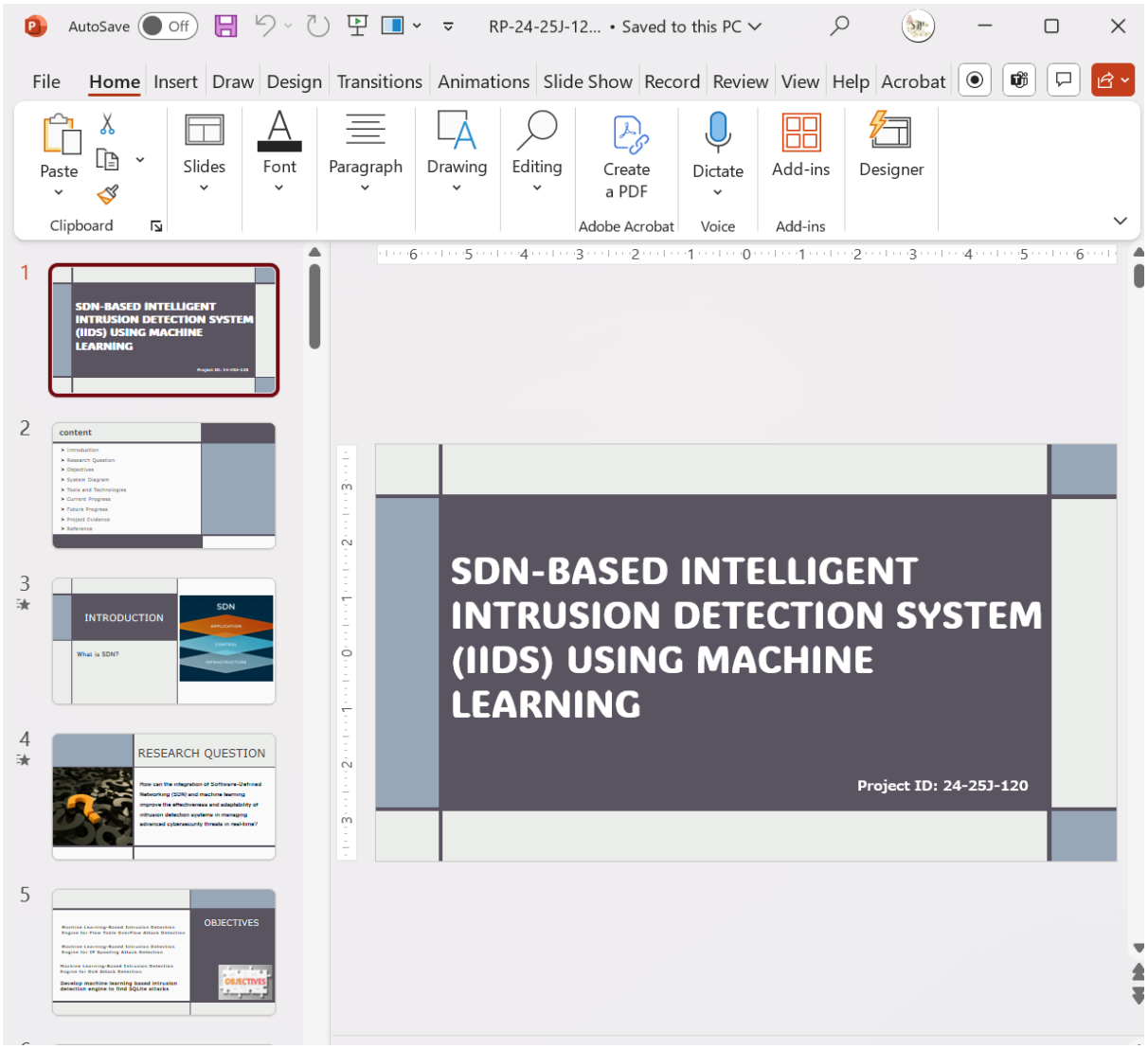
RP discussion
Tuesday, February 25, 2025 6:15 PM
Join

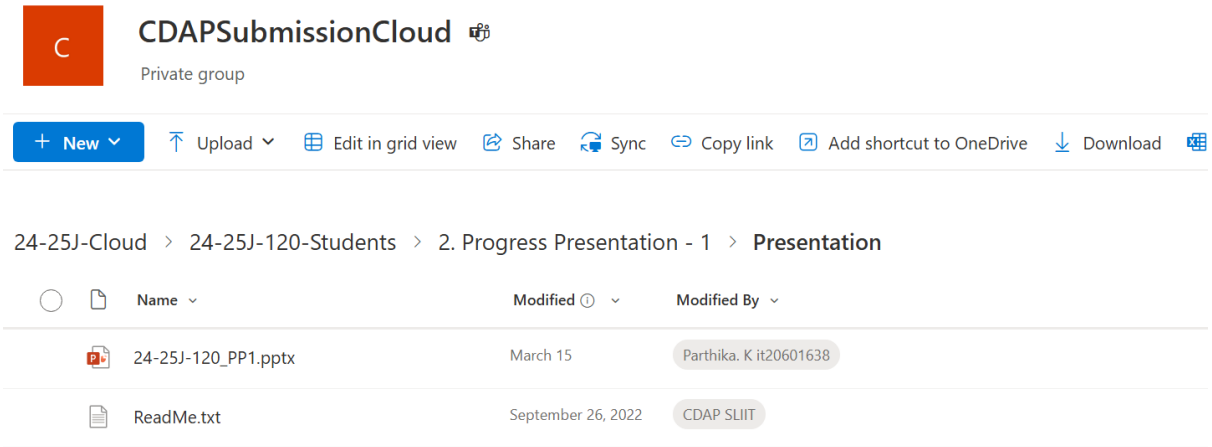
Scheduled a meeting



► Completed Tasks and Conversation Highlights

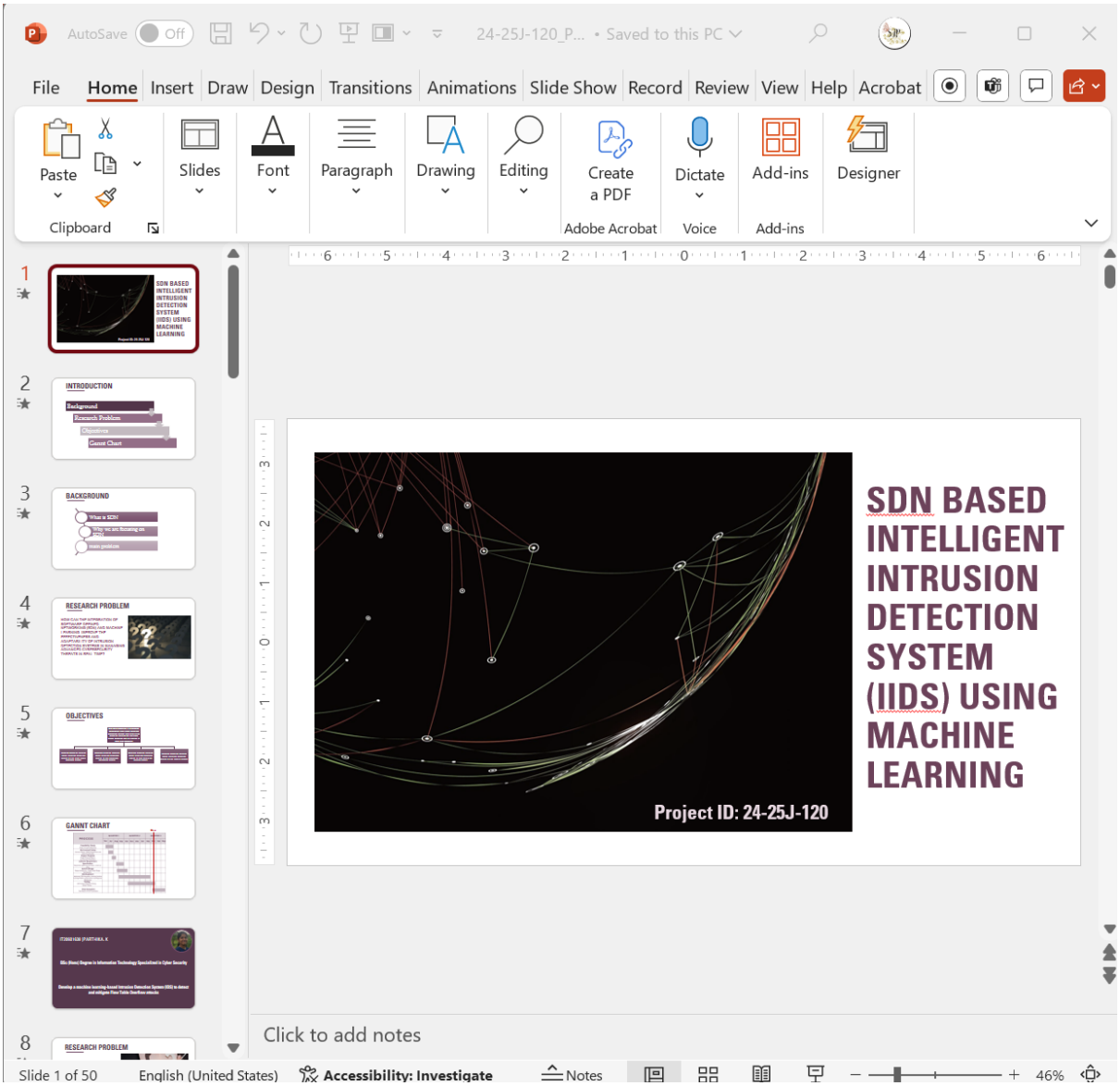
- Prepare for Progress Presentation 1 (PP1).
- Creating the presentation.
- Finalizing the Projects.
- Communication with the supervisor after finalizing the project.

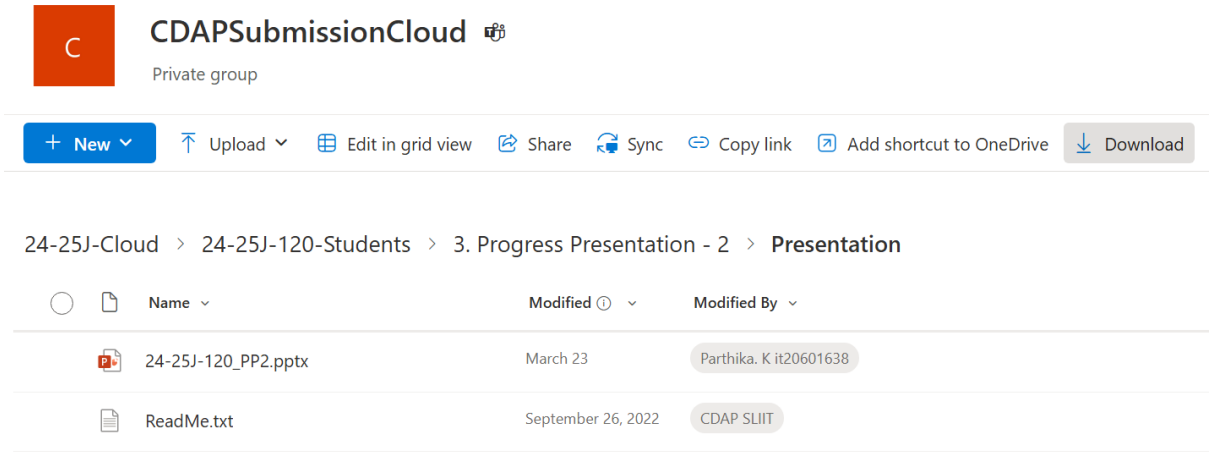




► Completed Task and Conversation Highlights

- Prepare for Progress Presentation 2 (PP2).
- Creating the presentation.
- Finalizing the Projects.
- Communication with the supervisor after finalizing the project.





► Completed Task and Conversation Highlights

- Started writing the research paper.
- Exploring the IEEE standards and word tools.
- Communicating with supervisor and getting the supervisor feedback.

SDN-based Intelligent Intrusion Detection System (IIDS) using Machine Learning

Parthika.K
Faculty of Computing
Cyber Security Specilization
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
kparthika@gmail.com

Satkurulingam.S
Faculty of Computing
Cyber Security Specilization
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
savithurisatkurulingam@gmail.com

Sriskandarajah J.P
Faculty of Computing
Cyber Security Specilization
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
srijoanna0@gmail.com

Dassanayake E.D
Faculty of Computing
Cyber Security Specilization
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
erangadassanayake15@gmail.com

Kanishka Prajeewa Yapa
Department of Computer Systems Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
kanishka.y@sliit.lk

Tharaniyawarma.K
Department of Computer Systems Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
tharaniyawarma.k@sliit.lk

***Abstract*— The increasing network complexity needs Software-Defined Networking (SDN) as a key solution to establish effective management systems through dynamic control mechanisms. SDN network infrastructure encounters four main security threats including Denial of Service (DoS), Flow Table Overflow, SQLite and Topology Poisoning attacks. This paper presents IIDS as an SDN-based intelligent intrusion detection system that operates with machine learning schemes to identify threats during real-time interactions. A dynamic system links the SDN controller with machine learning models for network traffic analysis to detect anomalies. The testing of the proposed method generates results for accuracy while also measuring precision and recall along with F1-score values. Starting from optimized attack detection the implemented technology is confirmed as an effective security framework for SDN networks because of its precise outcome.**

Keywords—cyber security, software defined networking, intrusion detection system, machine learning

I. Introduction

By implementing Software-Defined Networking (SDN) operators can execute dynamic management operations combined with automated system management tasks for their network infrastructure [1]. Multiple security threats can occur through SDN's central management architecture because it exposes itself to various cyber-attacks. Network resilience becomes unsustainable due to difficulties with implementing protection measures for new security threats within the current obtain live policy controls that improve network protection. Studies present evidence that SDN Technology integration with machine learning achieves successful threat detection solutions by using intelligent intrusion systems which improve security protection for network infrastructure.

II. Literature Review

Studies performed by technologists demonstrate IDS technology as a solution to enhance SDN security while dynamic network administration needs advanced threat management solutions [3]. Traditional IDS operations heavily depend on signature detection thus their ability to detect modern cyber threats remains low [4]. Machine learning within IDS technology speeds up performance and adjusts to new threats because it detects unknown attacks by analyzing patterns and statistical deviations [5]. The application of machine learning in IDS achieves superior performance speed as well as adaptability through its ability to detect unknown threats through anomaly detection and pattern recognition capabilities. Numerous researchers have focused on using multivariate statistical analysis to detect SDN attacks, as this approach enhances network security detection capabilities. By analyzing multiple variables simultaneously, this technique improves anomaly detection, identifying potential security threats more accurately. Researchers have explored various statistical models to enhance intrusion detection systems, making them more effective in mitigating security risks within SDN environments [6]. The evaluation of Flow Table

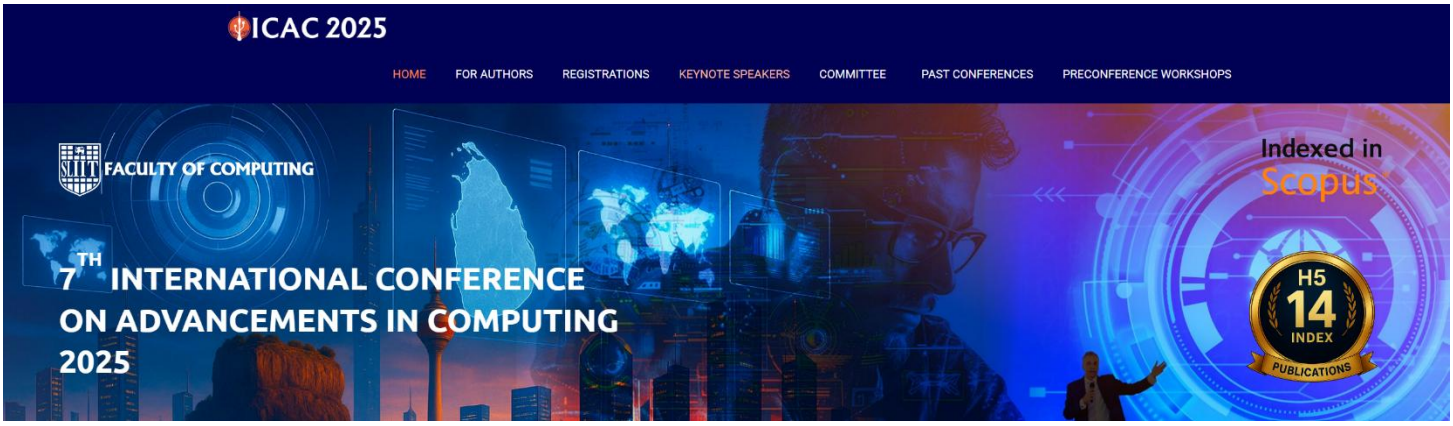
🌟 **Looking to Publish Your Research and Meet RP Deadlines?**

⛺ **ICAC 2025 is the Opportunity!**

If you're searching for a **reputed and impactful venue** to publish your research paper and meet your Research Project (RP) deadlines, **ICAC 2025** is ready to support you.

🗓️ **Date:** Wednesday, **21st May 2025**
🕖 **Time:** 7.30 PM
🎙️ **Speaker:** *Dr. Prasanna Sumathipala*

🔗 **Zoom Link:** https://zoom.us/j/94450599762?pwd=gxr3Zp8abKaS8kRN5mlA0san4sDRMZ.1



OBJECTIVES

Create a platform for networking and collaborative research and development activities among national and international researchers.

Promote advanced research culture among academic and cooperate communities around the globe.

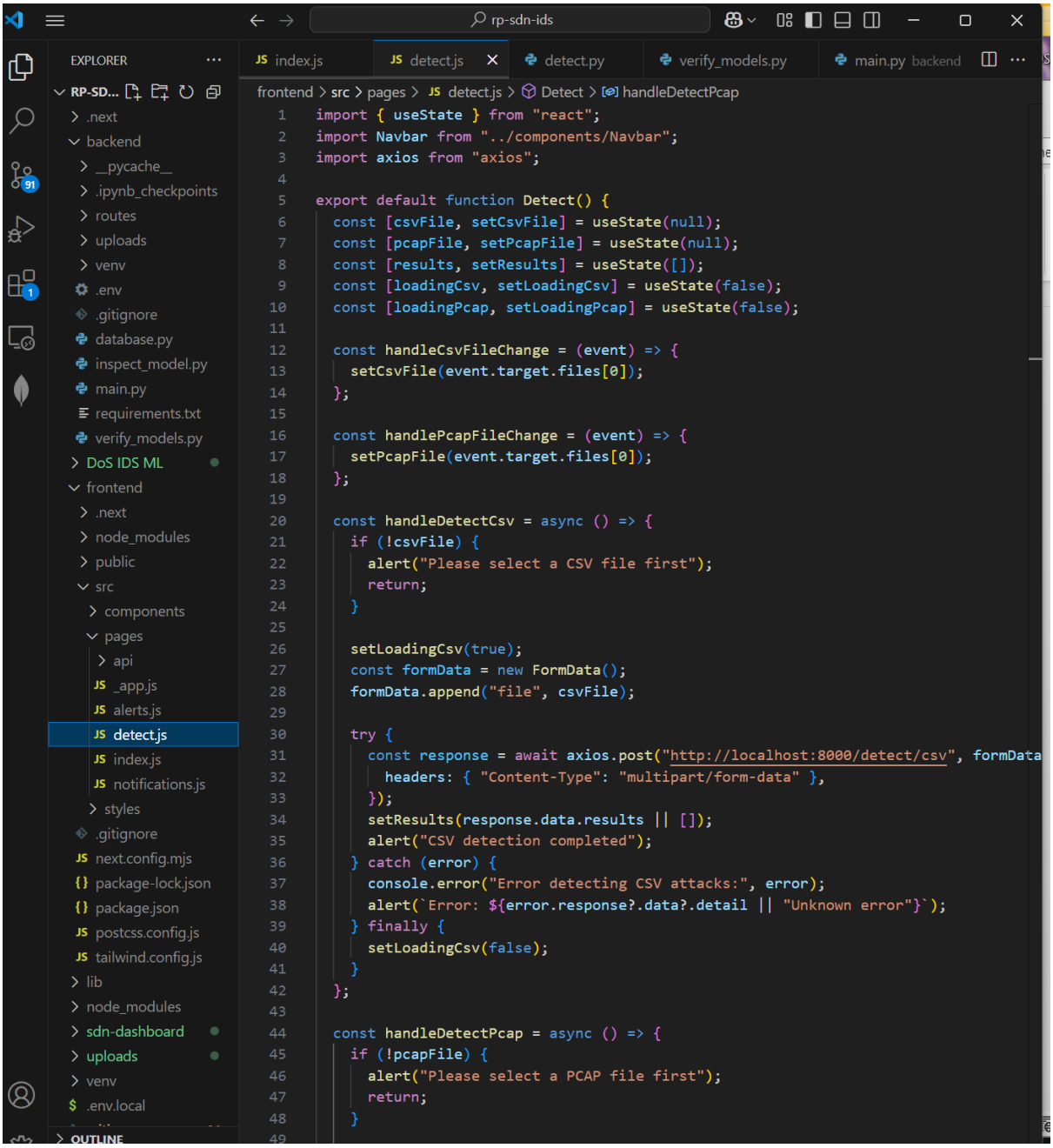
Highlight the achievements of local and international researchers in the field of computing.

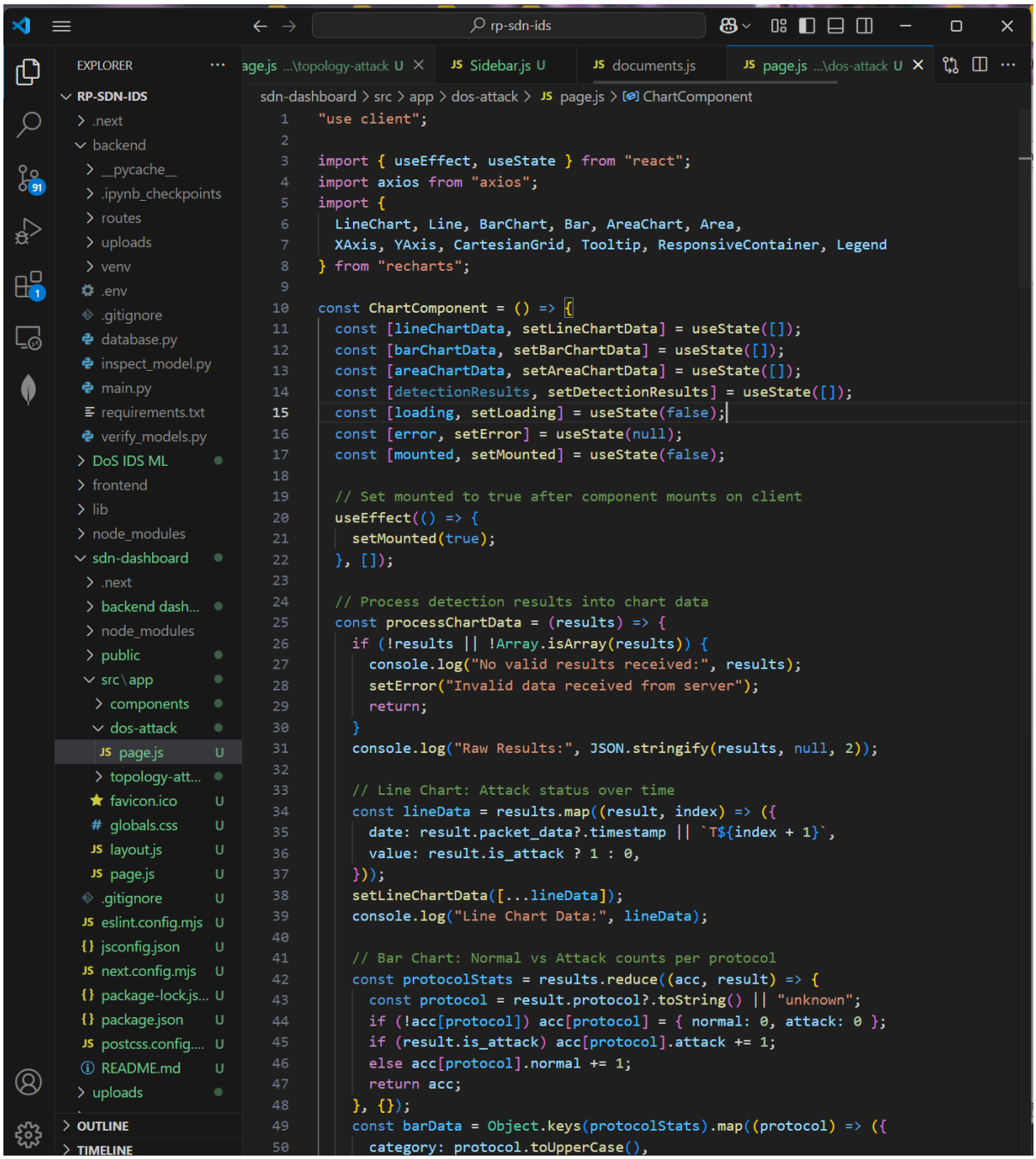
Foster groundbreaking innovations in computing by integrating Artificial Intelligence, Quantum Computing, and emerging technologies to solve complex challenges, drive cross-disciplinary advancements, and shape the future of industries worldwide.

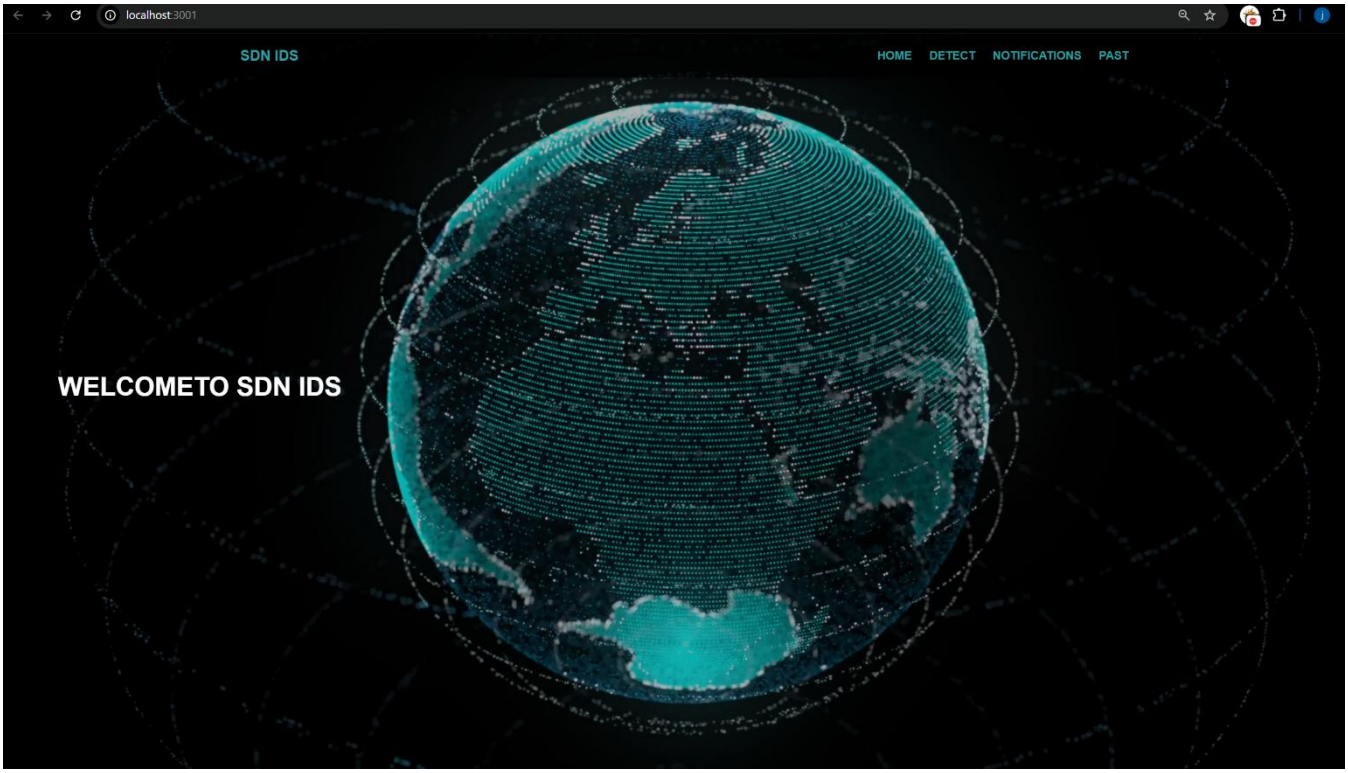
Paper Submission - Open	05th May 2025 Submit Now
Paper Submission - Close	20th August 2025
Acceptance Notification	20th October 2025
Camera Ready & Registration - Open	20th October 2025
Camera Ready & Registration - Close	10th November 2025
Conference	09th and 10th December 2025

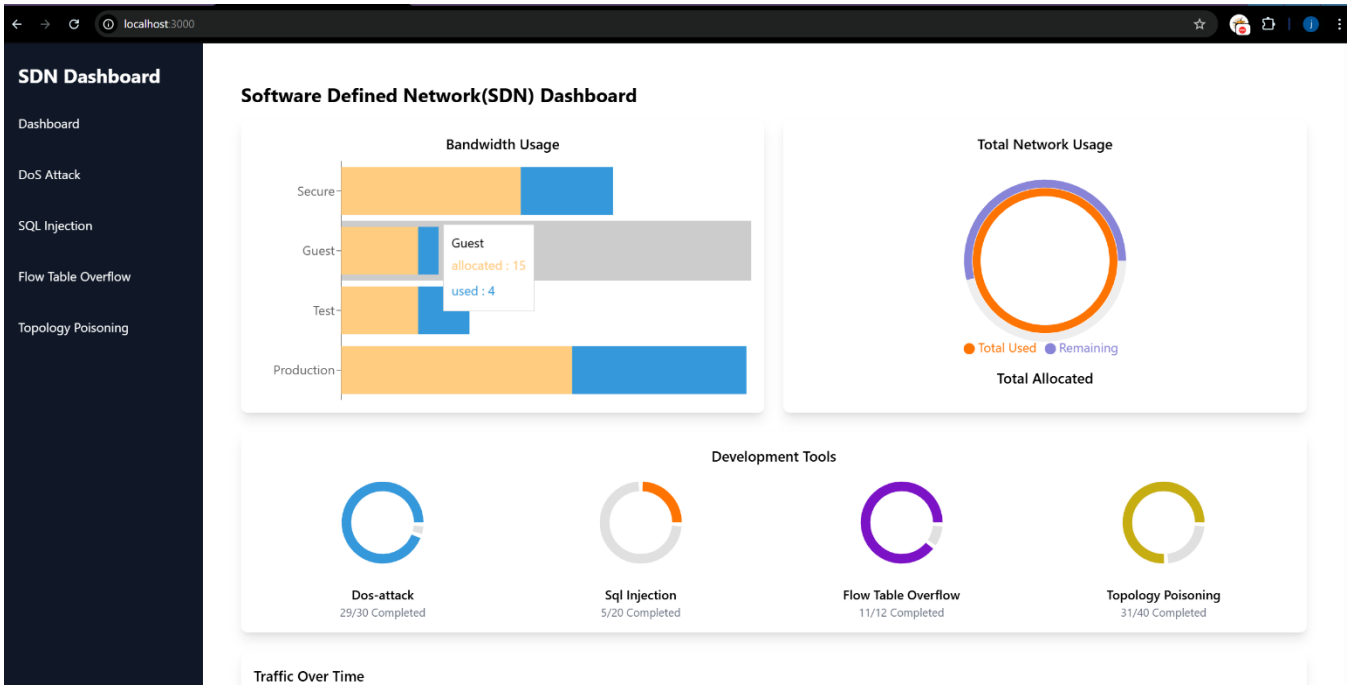
► Completed Tasks and Conversation Highlights

- Creating the front-end of the application.
- Integration of all the components.
- Discussing the supervisor's suggestions.




rp-sdn-ids
EXPLORER
RP-SDN-IDS
.next
backend
__pycache__
.ipynb_checkpoints
routes
uploads
venv
.env
.gitignore
database.py
inspect_model.py
main.py
requirements.txt
verify_models.py
DoS IDS ML
frontend
lib
node_modules
sdn-dashboard
.next
backend dash...
node_modules
public
src\app
components
dos-attack
page.js
topology-att...
favicon.ico
globals.css
layout.js
page.js
.gitignore
eslint.config.mjs
jsconfig.json
next.config.mjs
package-lock.js...
package.json
postcss.config....
README.md
uploads
OUTLINE
TIMELINE
age.js ...\topology-attack U
Sidebar.js U
documents.js
page.js ...\dos-attack U
sdn-dashboard > src > app > dos-attack > page.js > ChartComponent
"use client";
import { useEffect, useState } from "react";
import axios from "axios";
import {
LineChart, Line, BarChart, Bar, AreaChart, Area,
XAxis, YAxis, CartesianGrid, Tooltip, ResponsiveContainer, Legend
} from "recharts";
const ChartComponent = () => {
const [lineChartData, setLineChartData] = useState([]);
const [barChartData, setBarChartData] = useState([]);
const [areaChartData, setAreaChartData] = useState([]);
const [detectionResults, setDetectionResults] = useState([]);
const [loading, setLoading] = useState(false);
const [error, setError] = useState(null);
const [mounted, setMounted] = useState(false);
// Set mounted to true after component mounts on client
useEffect(() => {
setMounted(true);
}, []);
// Process detection results into chart data
const processChartData = (results) => {
if (!results || !Array.isArray(results)) {
console.log("No valid results received:", results);
setError("Invalid data received from server");
return;
}
console.log("Raw Results:", JSON.stringify(results, null, 2));
// Line Chart: Attack status over time
const lineData = results.map((result, index) => ({
date: result.packet_data?.timestamp || `T${index + 1}`,
value: result.is_attack ? 1 : 0,
}));
setLineChartData([...lineData]);
console.log("Line Chart Data:", lineData);
// Bar Chart: Normal vs Attack counts per protocol
const protocolStats = results.reduce((acc, result) => {
const protocol = result.protocol?.toString() || "unknown";
if (!acc[protocol]) acc[protocol] = { normal: 0, attack: 0 };
if (result.is_attack) acc[protocol].attack += 1;
else acc[protocol].normal += 1;
return acc;
}, {});
const barData = Object.keys(protocolStats).map((protocol) => ({
category: protocol.toUpperCase(),


localhost:3001
SDN IDS
HOME
DETECT
NOTIFICATIONS
PAST
WELCOMETO SDN IDS


localhost:3000
SDN Dashboard
Dashboard
DoS Attack
SQL Injection
Flow Table Overflow
Topology Poisoning
Software Defined Network(SDN) Dashboard
Bandwidth Usage
Secure
Guest
Test
Production
Guest
allocated : 15
used : 4
Total Network Usage
Total Used
Remaining
Total Allocated
Development Tools
Dos-attack
29/30 Completed
Sql Injection
5/20 Completed
Flow Table Overflow
11/12 Completed
Topology Poisoning
31/40 Completed
Traffic Over Time

► Completed Tasks and Conversation Highlights

- Complete Individual Thesis Reports.
- Creation Group Thesis Reports.

SDN-BASED INTELLIGENT INTRUSION DETECTION SYSTEM (IIDS) USING MACHINE LEARNING

Sriskandarajah J.P

IT21261978

BSc (Hons) Degree in Information Technology Specialized in Cyber Security

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

April 2025

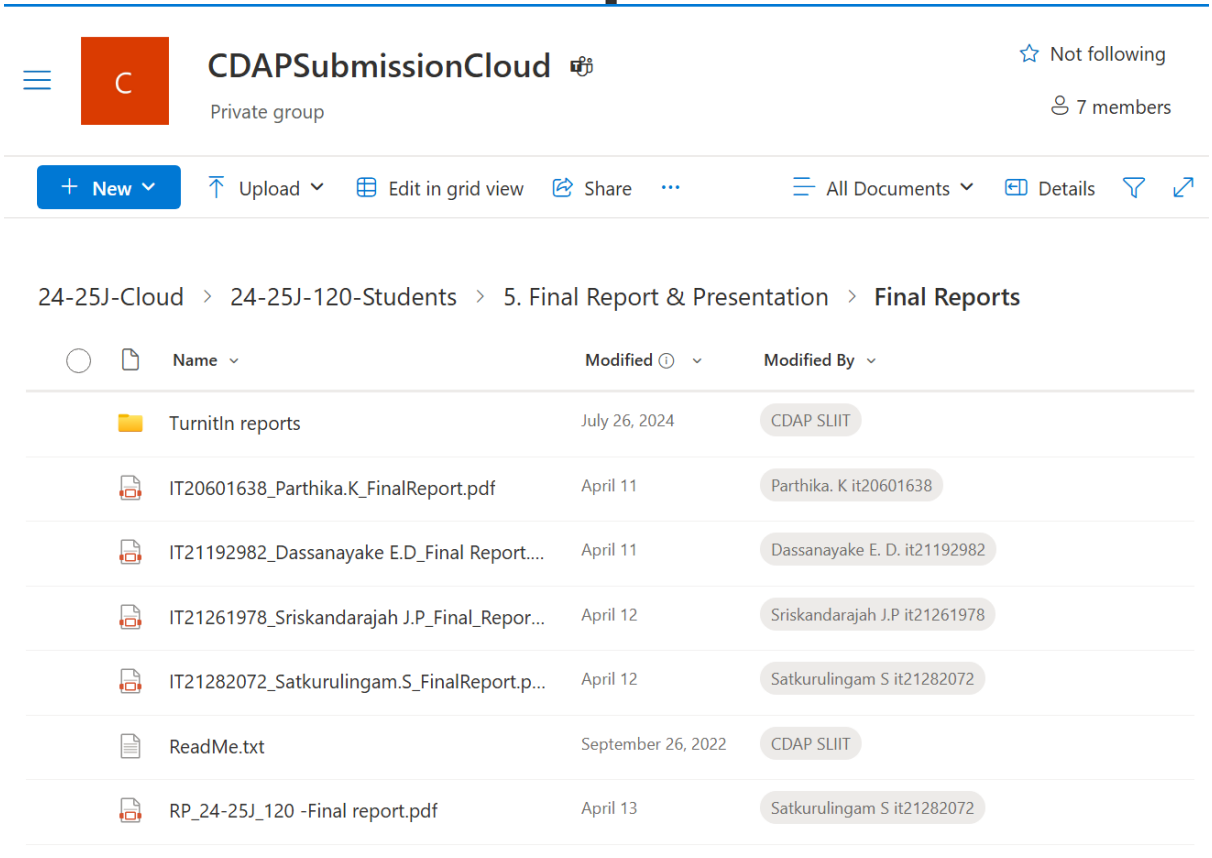
DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology, the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

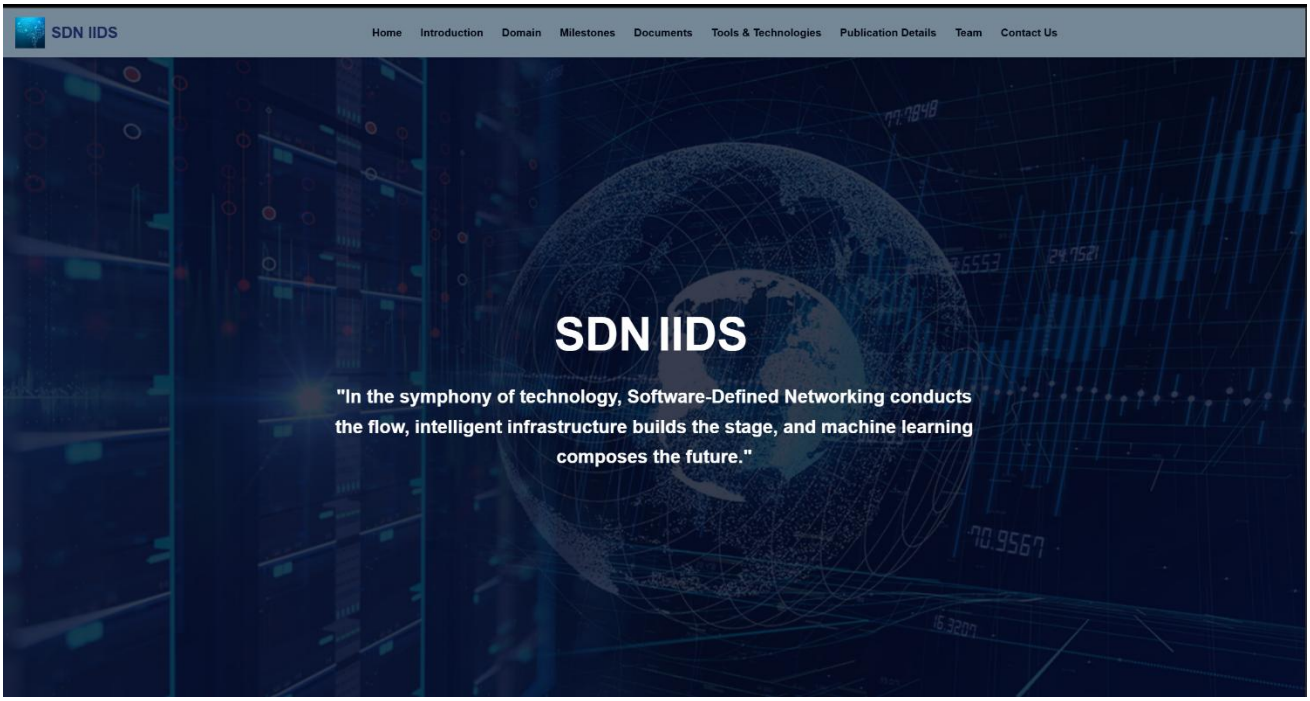
Signature: Date: 11.04.2025

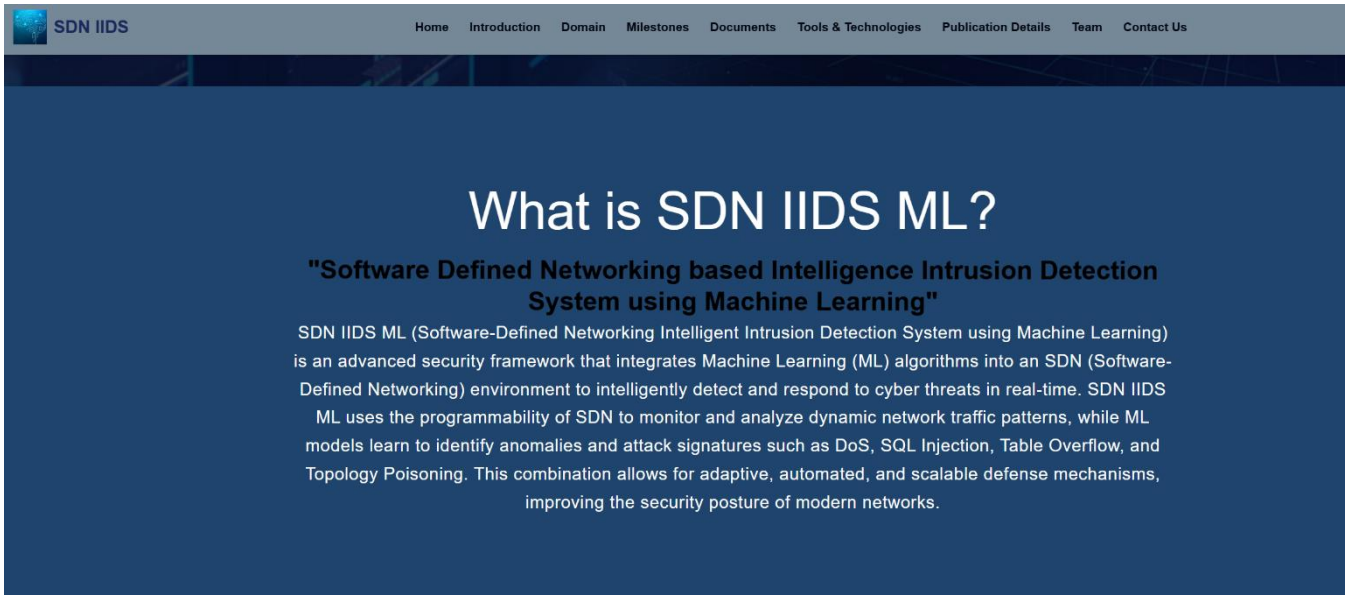
Signature of the supervisor: Date:



► **Completed Tasks and Conversation Highlights**

- Create a website for the solution.





Our Domain

Background | Research Gap | Research Problems | Research Objectives

Background

Our research presents an SDN-based Intelligent Intrusion Detection System (IIDS) utilizing machine learning for realtime attack detection and mitigation in SDN environments.

It uses machine learning to identify and mitigate attacks in real time. The system efficiently detects and stops a variety of cyberthreats, such as Denial of Service (DoS), Flow Table Overflow, SQLite, and Topology Poisoning attacks, by Integrating Machine Learning Model with the OpenDaylight SDN controller

Our Domain

Background | Research Gap | Research Problems | Research Objectives

Research Gap

Current research on SDN-based Intelligent Intrusion Detection Systems (IIDS) primarily focuses on limited attack types, often neglecting complex threats such as SQL injection, table overflow, and topology poisoning. Many existing systems also struggle with real-time detection due to high model latency and poor integration with SDN controllers, while the datasets used are often outdated or synthetic, failing to represent real-world SDN traffic patterns.

Furthermore, models typically overfit to specific attack scenarios, limiting their generalization to evolving threats. There's also a lack of comprehensive evaluation metrics, with most studies focusing solely on accuracy, ignoring critical aspects like false positive rates, resource usage, and network impact. Finally, scalability and deployment challenges remain underexplored, with few systems tested in large-scale, real-world environments. Our research aims to address these gaps by developing a robust, real-time IIDS that can detect a wide range of SDN-specific attacks while ensuring scalability and efficient integration.

Our Domain

Background | Research Gap | Research Problems | Research Objectives

Research Problems

Flow Table Overflow

Table overflow attacks in SDN target the limited flow table capacity of switches, overwhelming them with excessive flow entries. Existing detection methods either rely on static thresholds or reactive strategies that are ineffective under adaptive attack patterns. There is a critical need for proactive, intelligent detection models that can recognize subtle anomalies in flow dynamics and prevent table saturation without impacting legitimate traffic.

Topology Poisoning

Topology poisoning attacks exploit the dynamic nature of SDN by injecting false topology information, leading to incorrect routing decisions and network disruption. Existing detection approaches often rely on static rules or topology snapshots, which fail to adapt to rapidly changing network states. There is a pressing need for ML-based systems that can learn normal topology patterns, detect deviations in real time, and safeguard the network from such attacks.

Denial of Service

Current SDN-based IDS systems often focus on detecting common DoS attacks like UDP floods but fail to effectively identify protocol-specific threats such as SNMP and DNS amplification attacks in real time. The lack of protocol-aware models and comprehensive datasets limits the system's ability to distinguish between normal traffic and sophisticated DoS patterns, leading to high false positives and delayed mitigation in dynamic SDN environments.

SQL injection

SQL injection attacks in SDN environments are under-researched, as most studies focus on web applications. In SDN, malicious SQL queries can target northbound APIs or management systems, causing misconfigurations or unauthorized data access. The lack of tailored ML models for SQLi in SDN and the absence of real-time detection frameworks create a significant vulnerability, necessitating research into robust SQLi detection mechanisms for SDN control layers.

Our Domain

Background | Research Gap | Research Problems | Research Objectives

Research Objectives

Main Objective

Our main objective is to develop SDN-based Intelligent Intrusion Detection System (IIDS) utilizing machine learning for real-time attack detection and mitigation in SDN environments

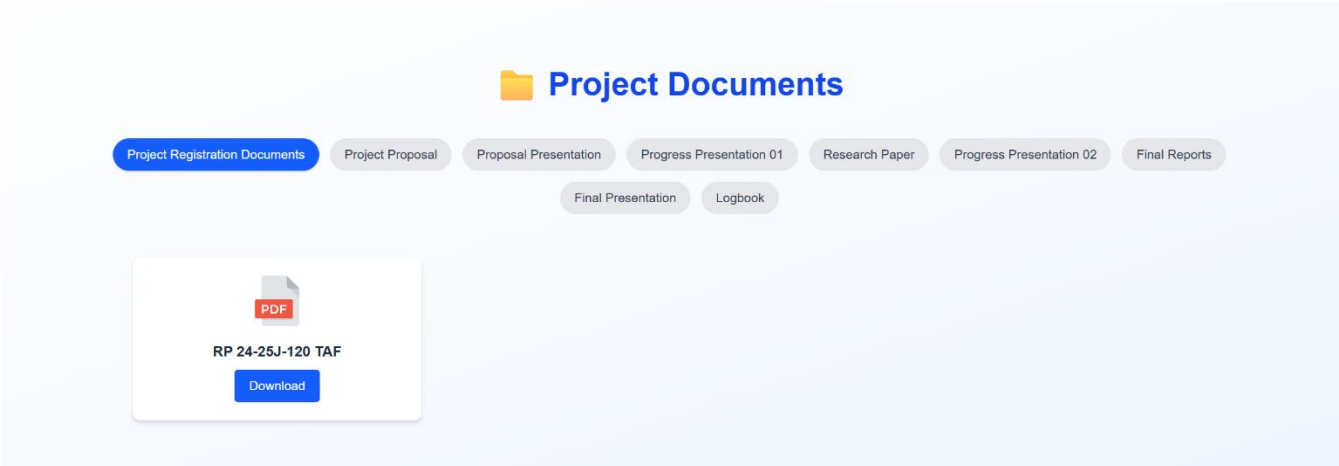
Specific Objectives

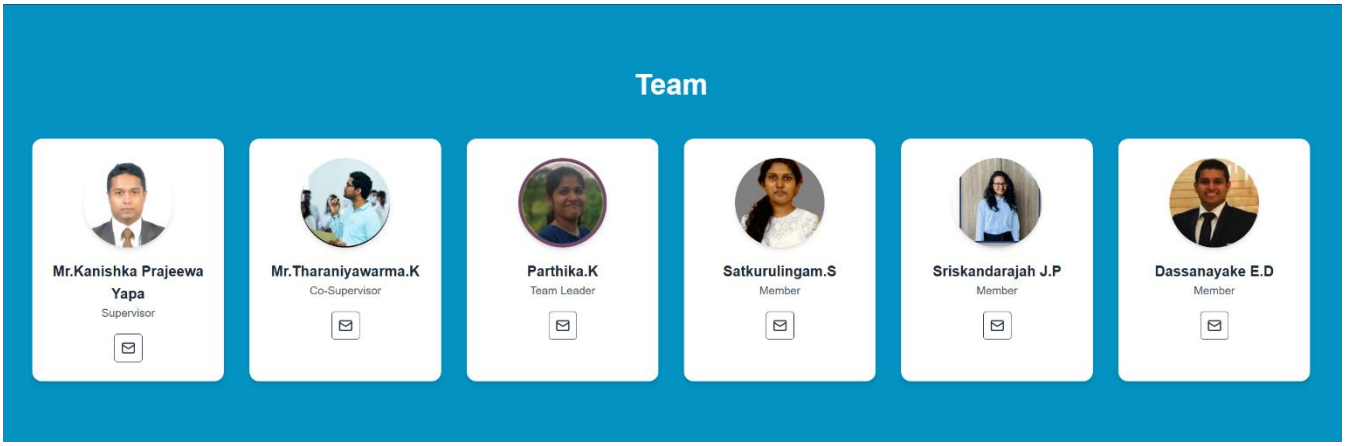
1. Develop machine learning based intrusion detection engine to find Flow Table Overflow attacks

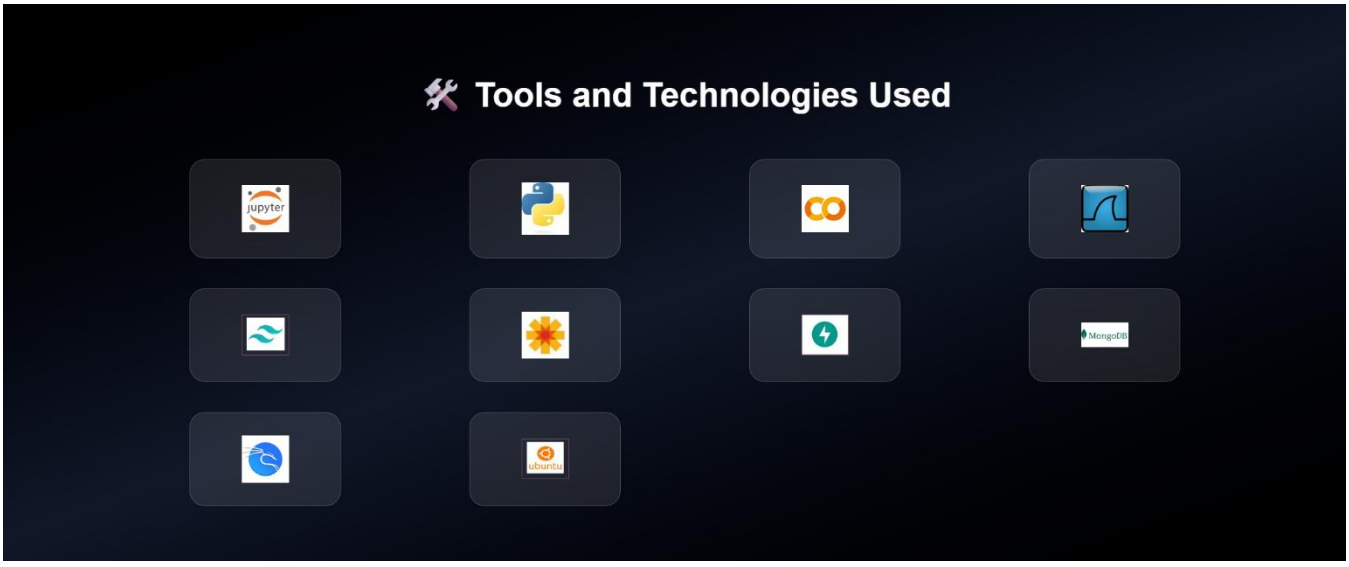
2. Develop machine learning based intrusion detection engine to find Topology Poisoning attacks

3. Develop machine learning based intrusion detection engine to find Denial of Service attacks

4. Develop machine learning based intrusion detection engine to find SQLite attacks

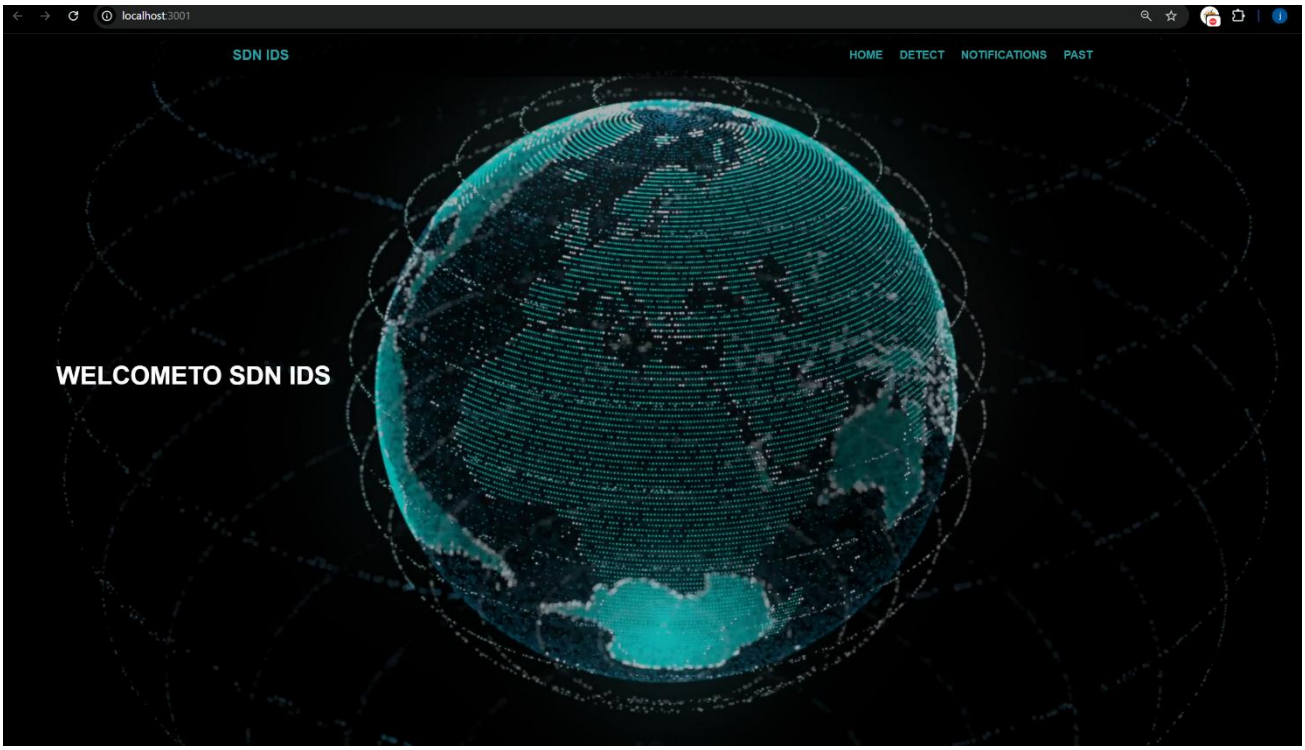

Project Documents
Project Registration Documents
Project Proposal
Proposal Presentation
Progress Presentation 01
Research Paper
Progress Presentation 02
Final Reports
Final Presentation
Logbook
PDF
RP 24-25J-120 TAF
Download

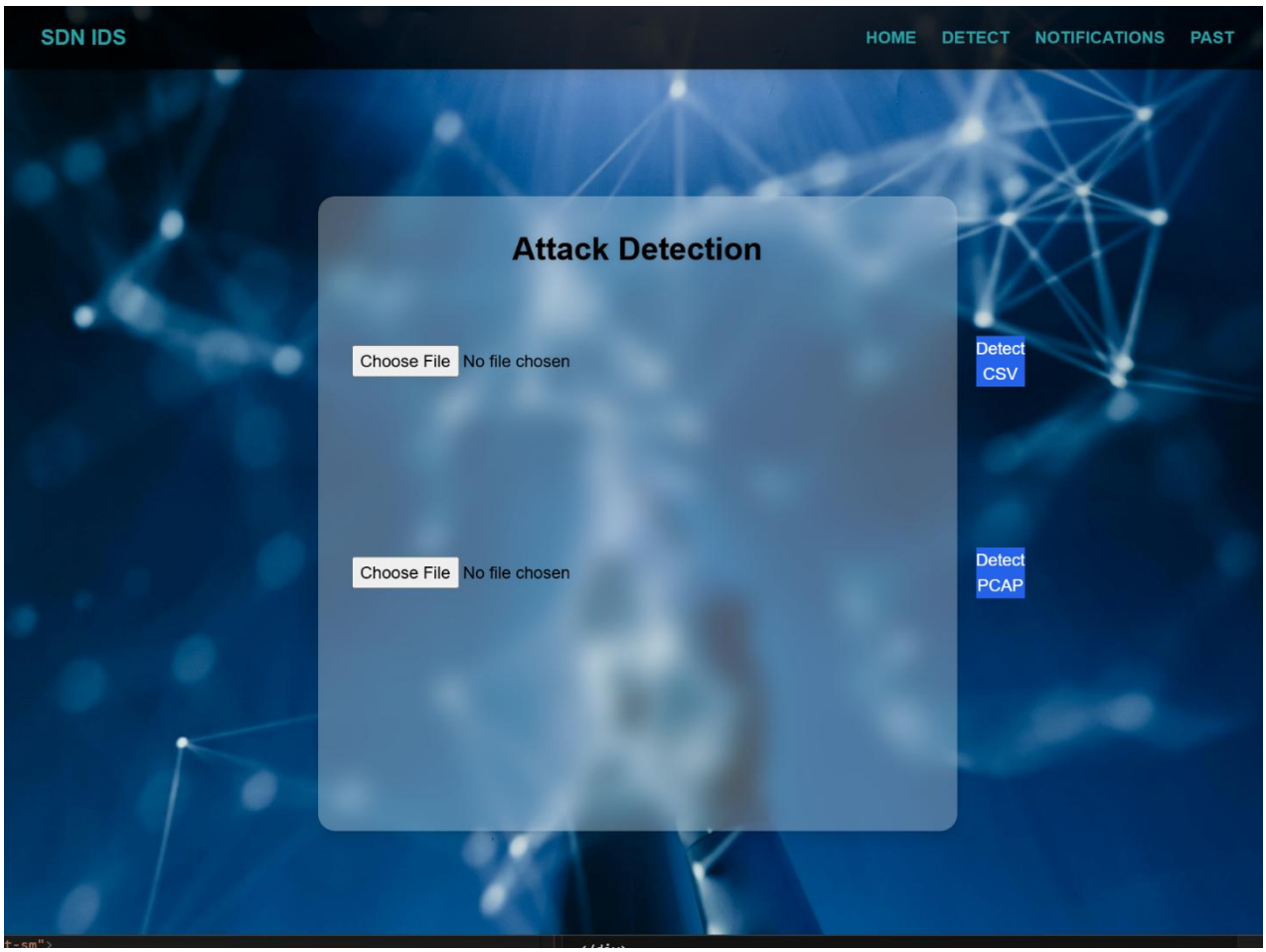

Team
Mr.Kanishka Prajeewa Yapa
Supervisor
Mr.Tharaniyawarma.K
Co-Supervisor
Parthika.K
Team Leader
Satkurulingam.S
Member
Sriskandarajah J.P
Member
Dassanayake E.D
Member

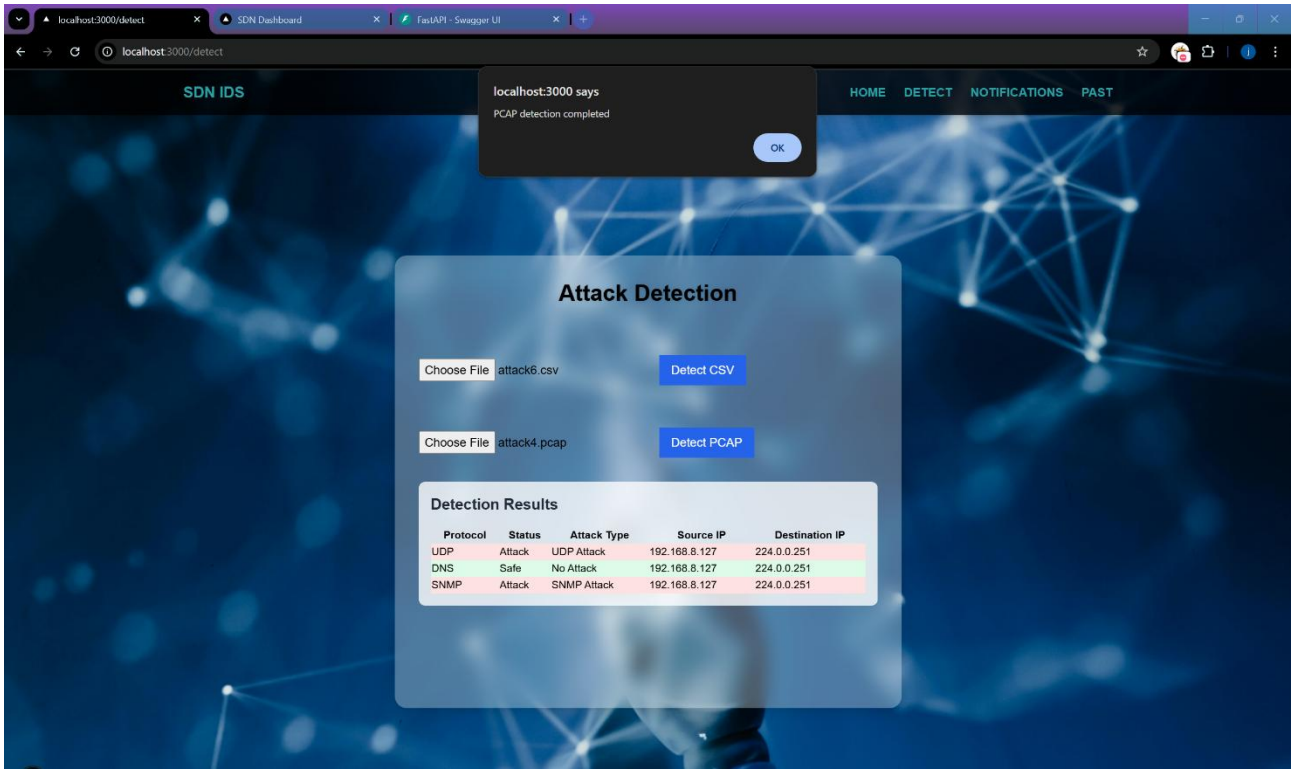

Tools and Technologies Used
jupyter
MongoDB
ubuntu

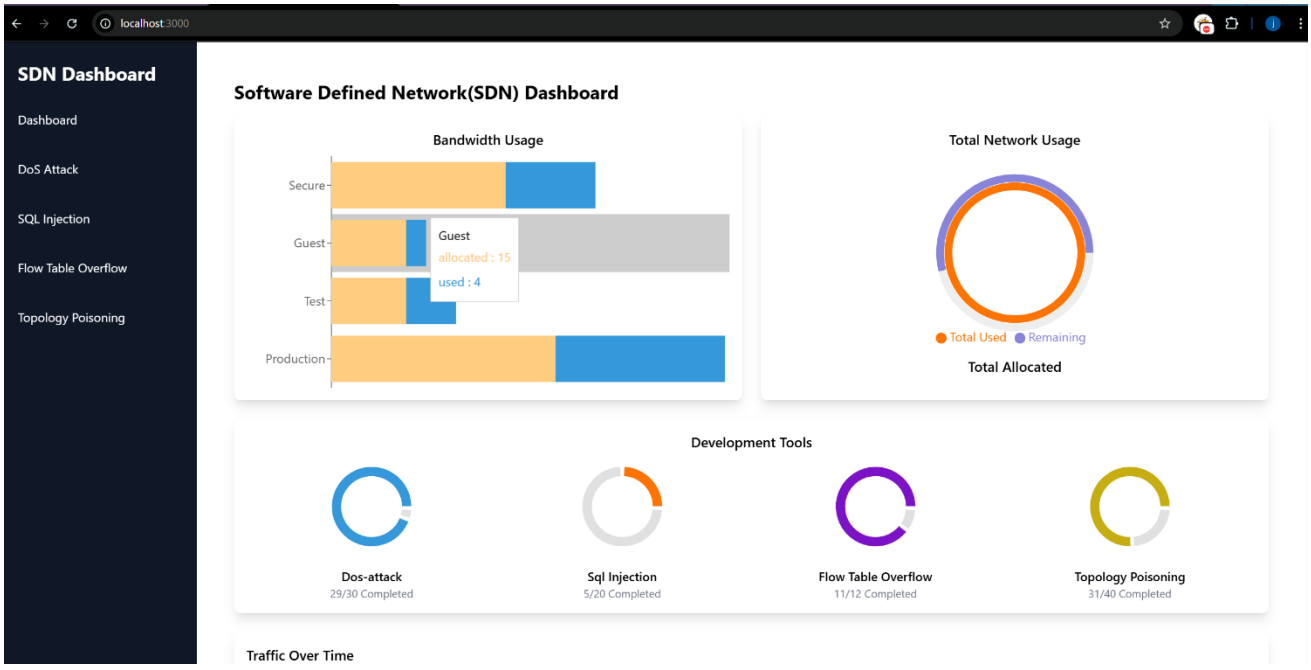
► **Completed Tasks and Conversation Highlights**

- Final Research Project Product





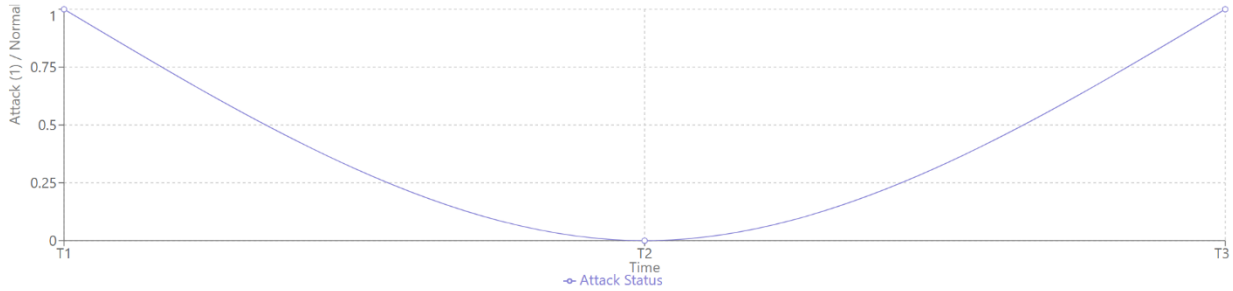

localhost:3000/detect
SDN Dashboard
FastAPI - Swagger UI
SDN IDS
HOME
DETECT
NOTIFICATIONS
PAST
localhost:3000 says
PCAP detection completed
OK
Attack Detection
Choose File
attack6.csv
Detect CSV
Choose File
attack4.pcap
Detect PCAP
Detection Results
Protocol
Status
Attack Type
Source IP
Destination IP
UDP
Attack
UDP Attack
192.168.8.127
224.0.0.251
DNS
Safe
No Attack
192.168.8.127
224.0.0.251
SNMP
Attack
SNMP Attack
192.168.8.127
224.0.0.251


localhost:3000
SDN Dashboard
Dashboard
DoS Attack
SQL Injection
Flow Table Overflow
Topology Poisoning
Software Defined Network(SDN) Dashboard
Bandwidth Usage
Secure
Guest
Test
Production
Guest
allocated : 15
used : 4
Total Network Usage
Total Used
Remaining
Total Allocated
Development Tools
Dos-attack
29/30 Completed
Sql Injection
5/20 Completed
Flow Table Overflow
11/12 Completed
Topology Poisoning
31/40 Completed
Traffic Over Time

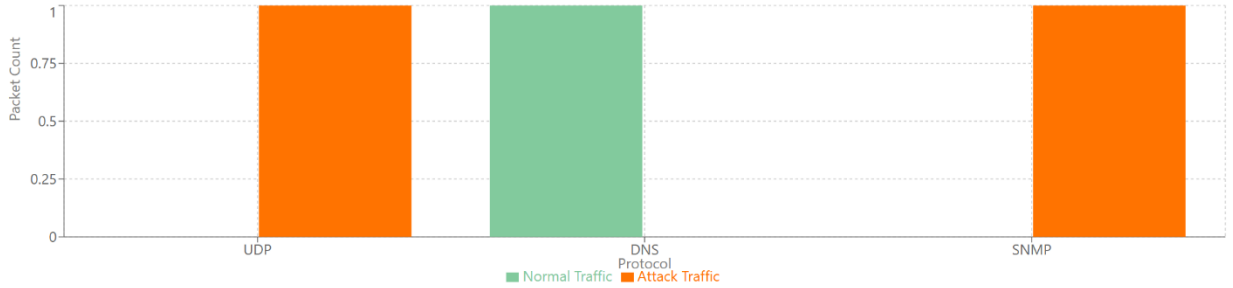
Upload PCAP to Detect Attacks

Choose File attack4.pcap

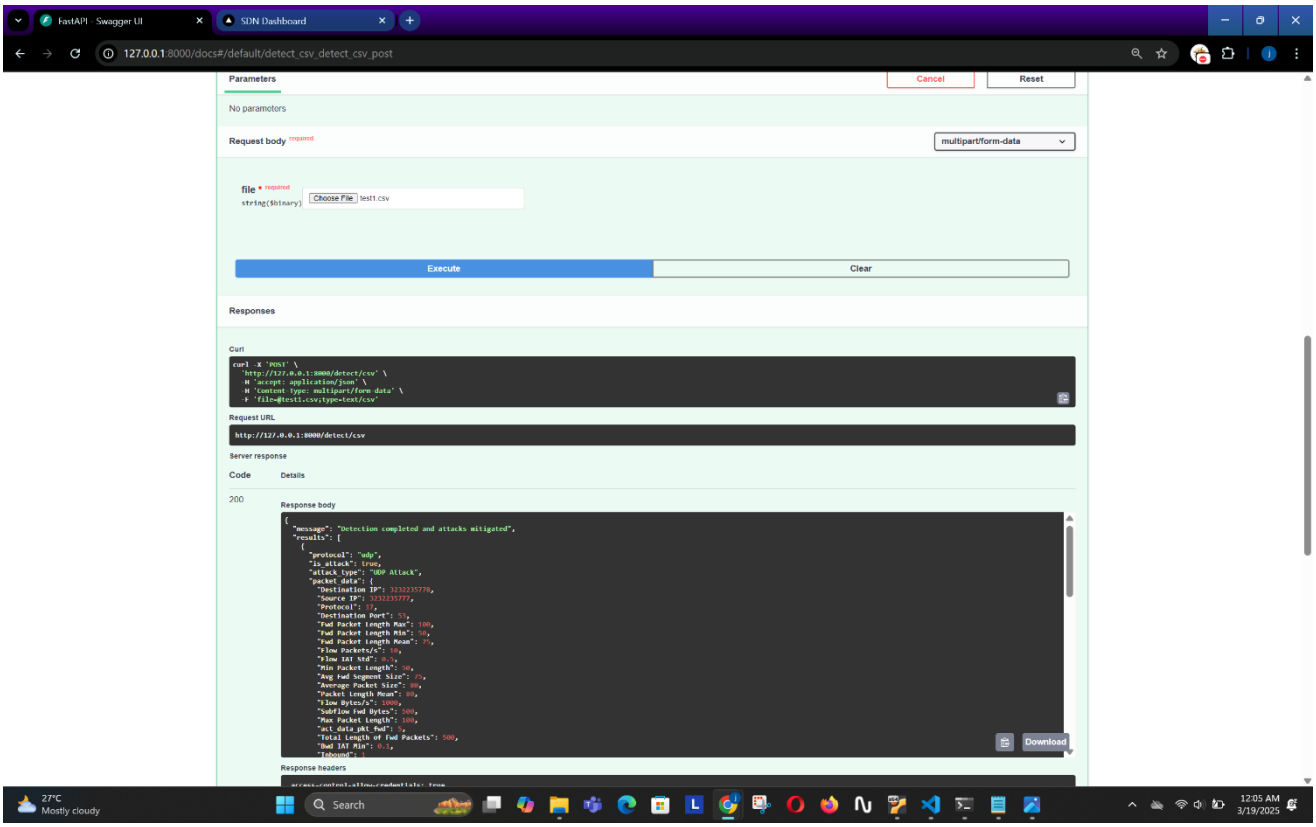
Packets Over Time

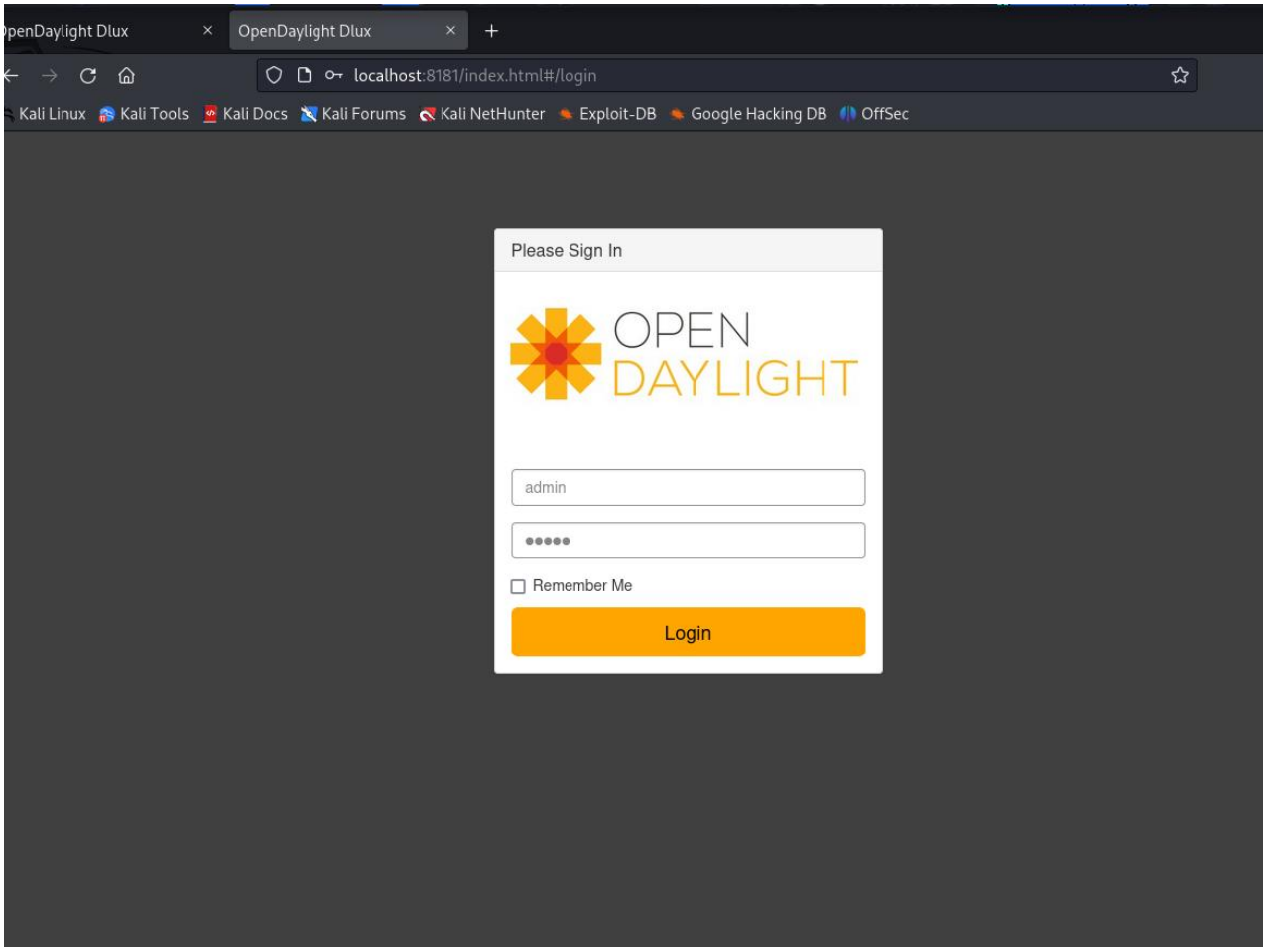


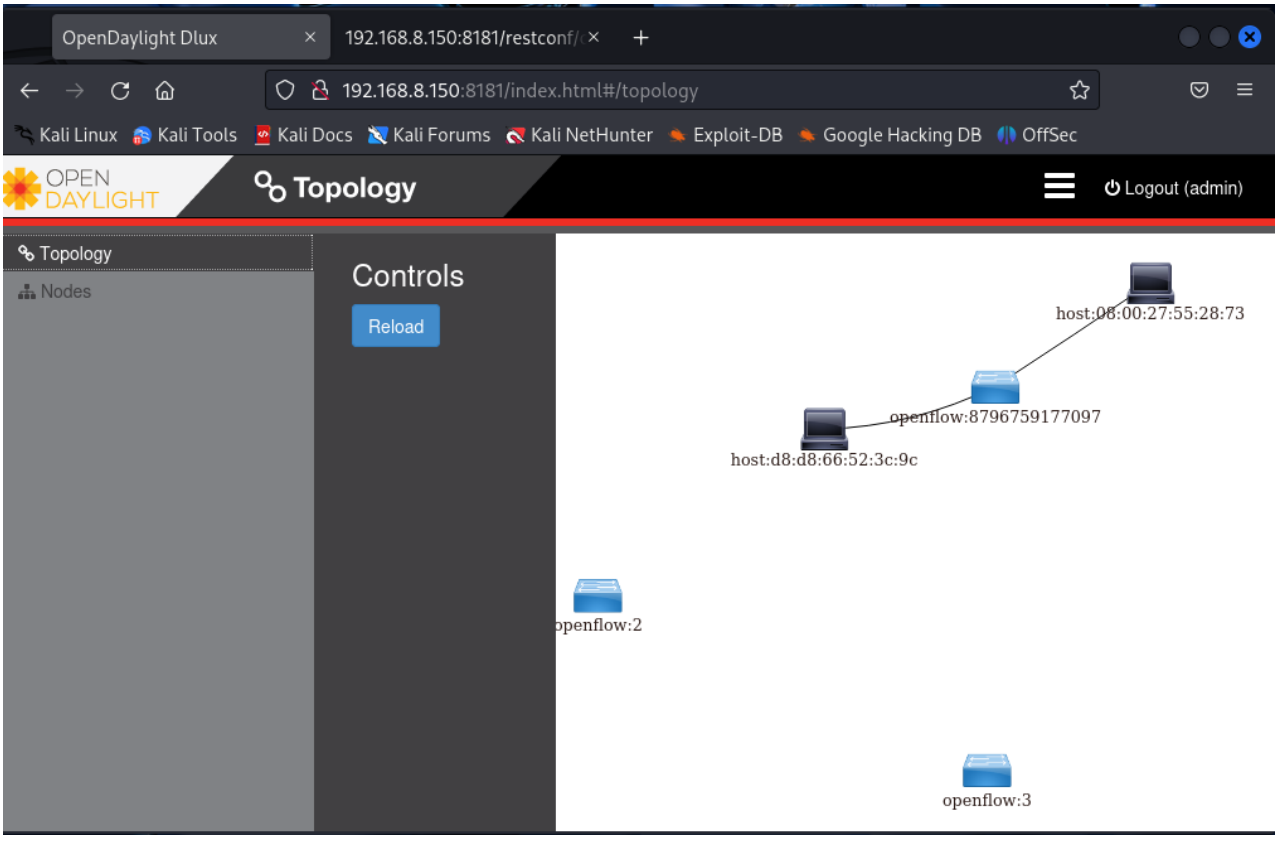
Normal vs Attack Traffic by Protocol

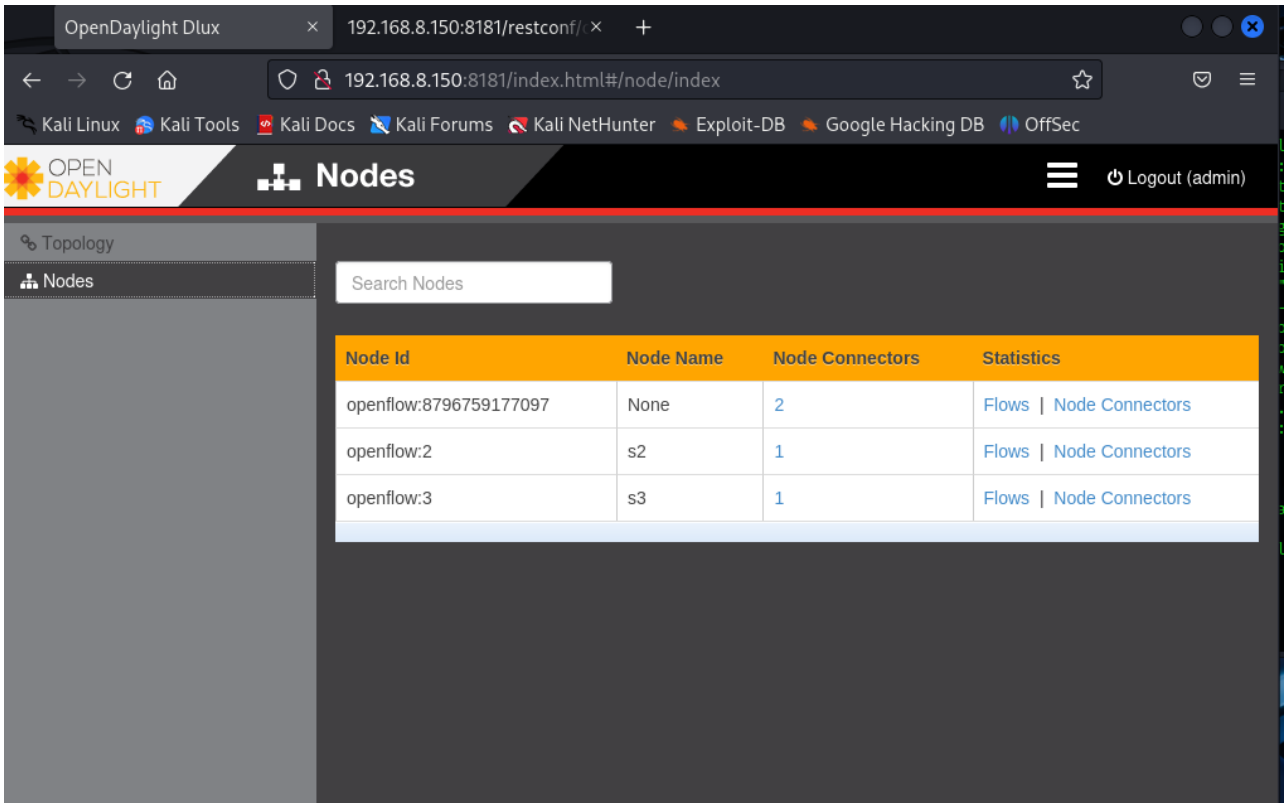


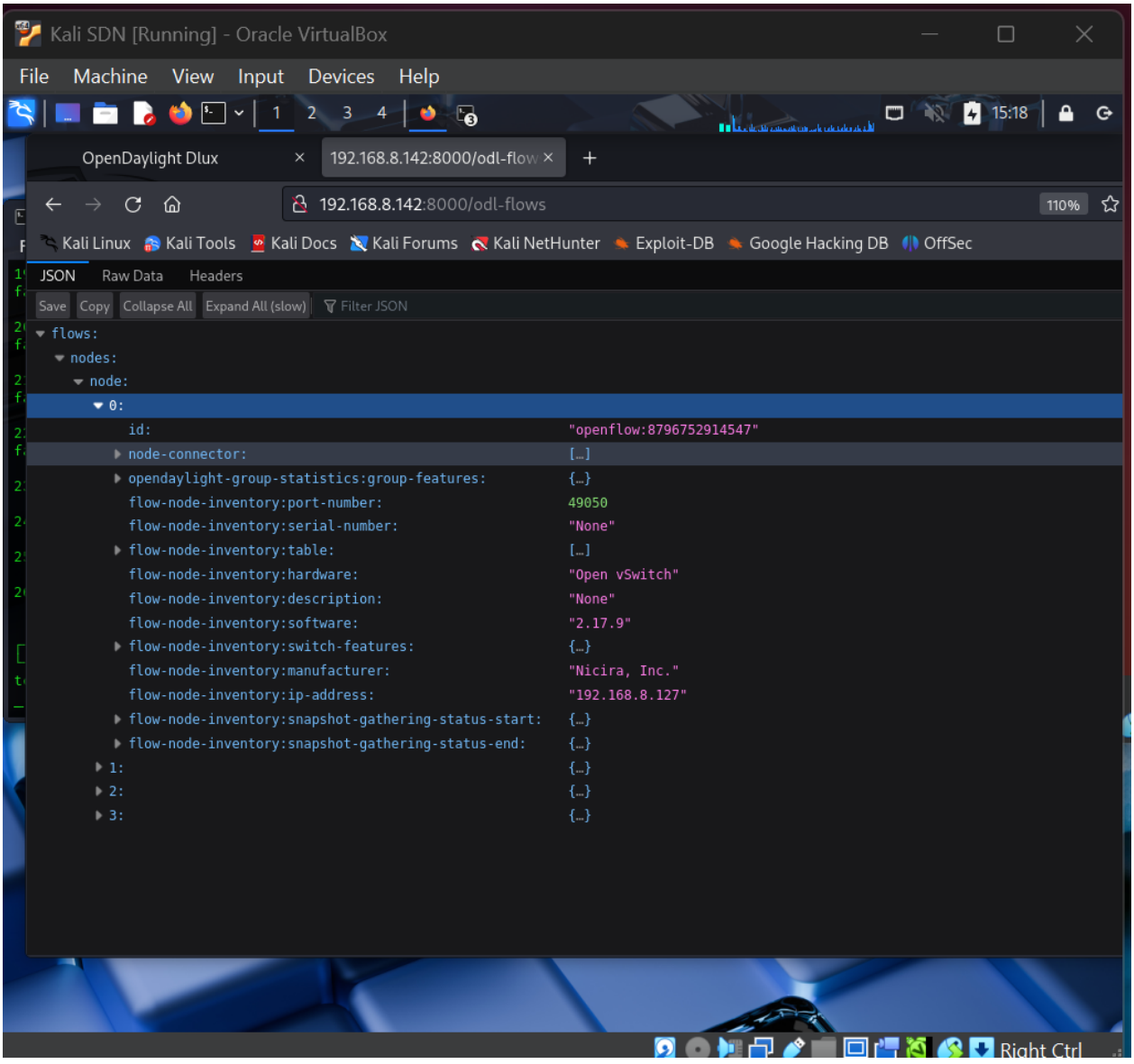
Traffic Volume with Attack Highlights

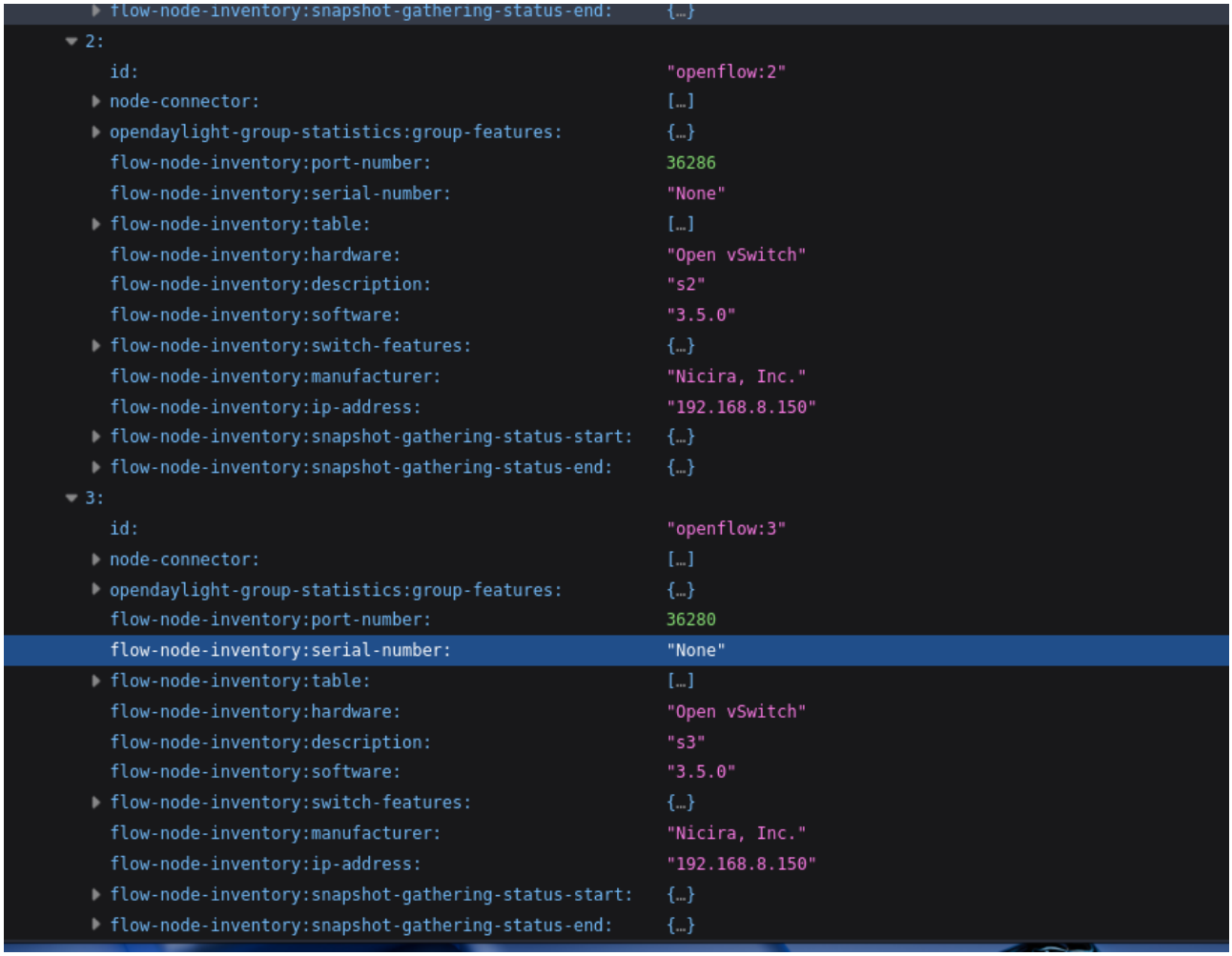



OpenDaylight Dlux
OpenDaylight Dlux
localhost:8181/index.html#/login
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Please Sign In
OPEN DAYLIGHT
admin
Remember Me
Login


OpenDaylight Dlux
192.168.8.150:8181/restconf/
192.168.8.150:8181/index.html#/topology
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
OPEN DAYLIGHT
Topology
Logout (admin)
Topology
Nodes
Controls
Reload
host:08:00:27:55:28:73
openflow:8796759177097
host:d8:d8:66:52:3c:9c
openflow:2
openflow:3

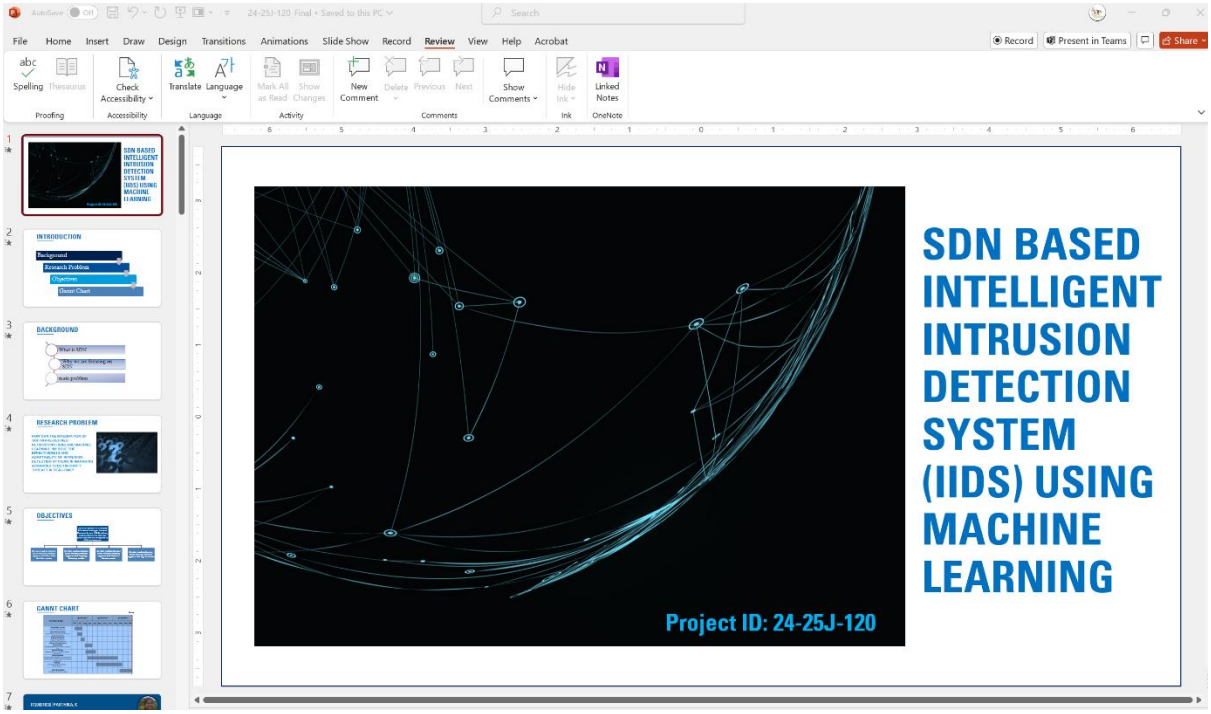

OpenDaylight Dlux
192.168.8.150:8181/restconf/
192.168.8.150:8181/index.html#/node/index
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
OPEN DAYLIGHT
Nodes
Logout (admin)
Topology
Nodes
Search Nodes
Node Id
Node Name
Node Connectors
Statistics
openflow:8796759177097
None
2
Flows | Node Connectors
openflow:2
s2
1
Flows | Node Connectors
openflow:3
s3
1
Flows | Node Connectors

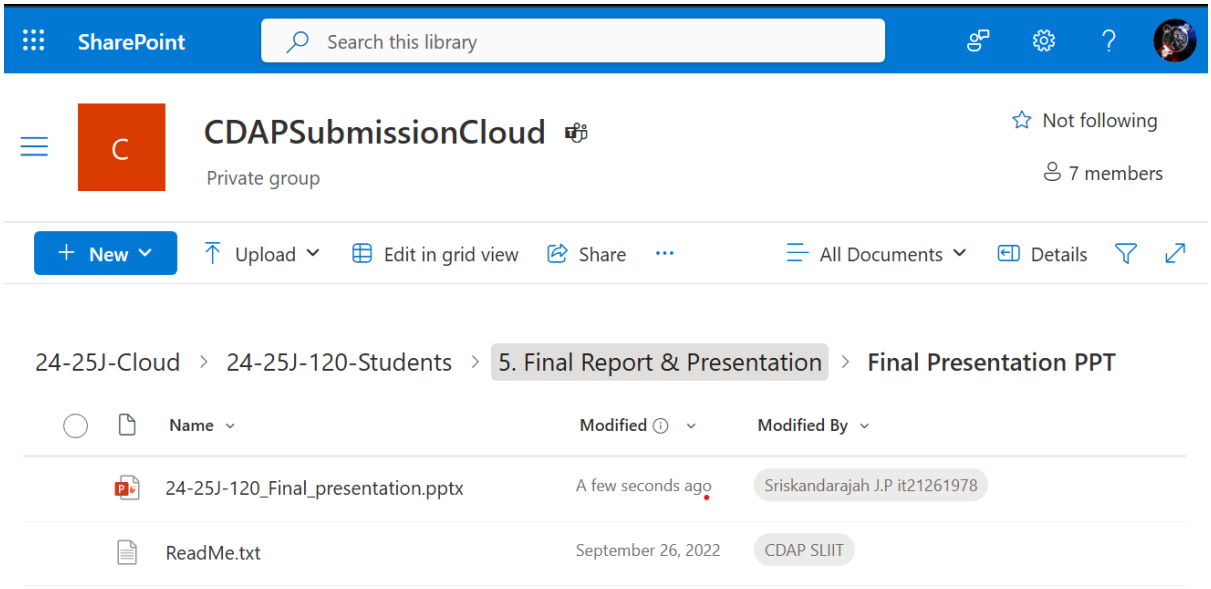

Kali SDN [Running] - Oracle VirtualBox
File
Machine
View
Input
Devices
Help
15:18
OpenDaylight Dlux
192.168.8.142:8000/odl-flow
192.168.8.142:8000/odl-flows
110%
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
JSON
Raw Data
Headers
Save
Copy
Collapse All
Expand All (slow)
Filter JSON
flows:
nodes:
node:
0:
id: "openflow:8796752914547"
node-connector: […]
opendaylight-group-statistics:group-features: {…}
flow-node-inventory:port-number: 49050
flow-node-inventory:serial-number: "None"
flow-node-inventory:table: […]
flow-node-inventory:hardware: "Open vSwitch"
flow-node-inventory:description: "None"
flow-node-inventory:software: "2.17.9"
flow-node-inventory:switch-features: {…}
flow-node-inventory:manufacturer: "Nicira, Inc."
flow-node-inventory:ip-address: "192.168.8.127"
flow-node-inventory:snapshot-gathering-status-start: {…}
flow-node-inventory:snapshot-gathering-status-end: {…}
1: {…}
2: {…}
3: {…}
Right Ctrl



► Completed Task and Conversation Highlights

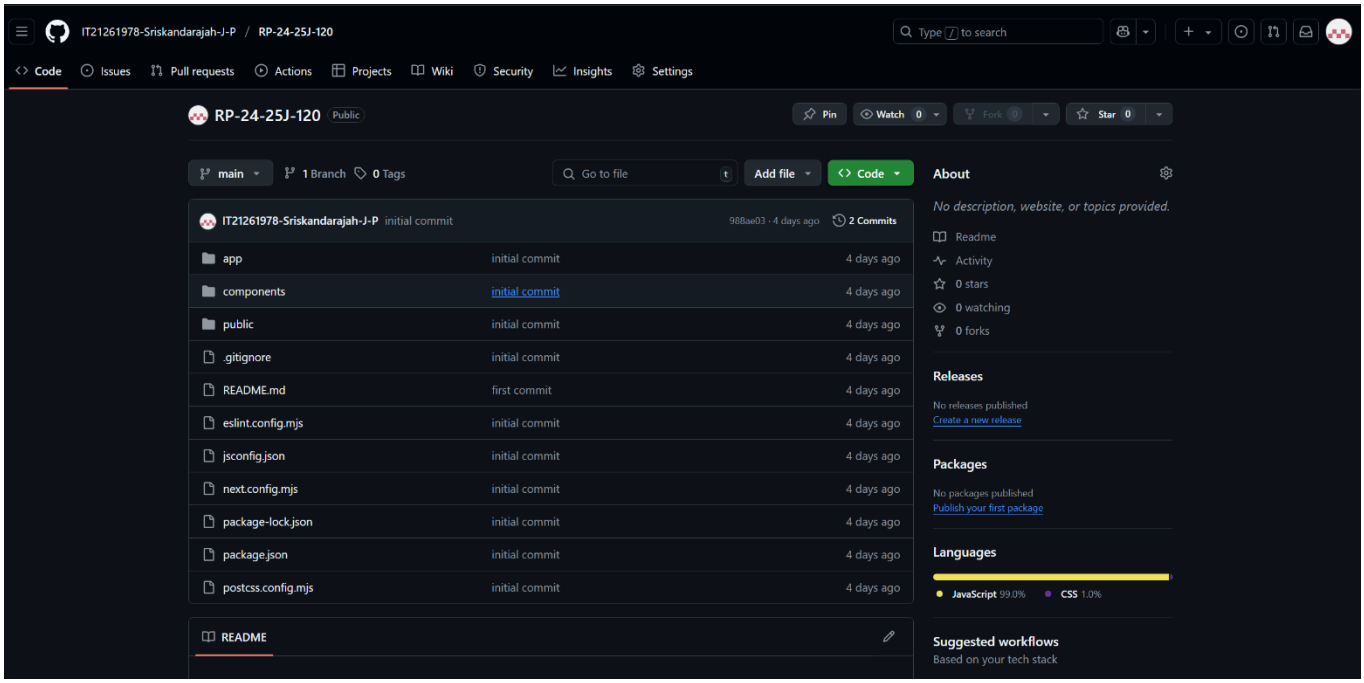
- Prepare for Final Presentation
- Creating the presentation.





► Completed Task and Conversation Highlights

- Commit and push the website codes in GitHub before deploying



► Completed Task and Conversation Highlights

- Deploy the website using vercel.