**AI-Driven Network Monitoring and DDoS Detection Using SDN and Programmable Data Planes**

***A Project Report***

*Submitted in partial fulfilment of the requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

In

**ELECTRONICS AND COMMUNICATION ENGINEERING**

*By*

**P. CHAITANYA CHANDU (21B91A04H8)**

**M. PRASANNA KUMAR (21B91A04F0)**

**M. SAHIL (21B91A04E9)**

**S. RISHI (21B91A04L0)**

*Under the esteemed guidance of*

**Mr. N. KISHORE CHANDRA DEV**

*MTech.*

Assistant Professor of ECE Department

***DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING***

**S.R.K.R. ENGINEERING COLLEGE (AUTONOMOUS)**

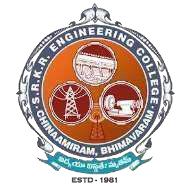
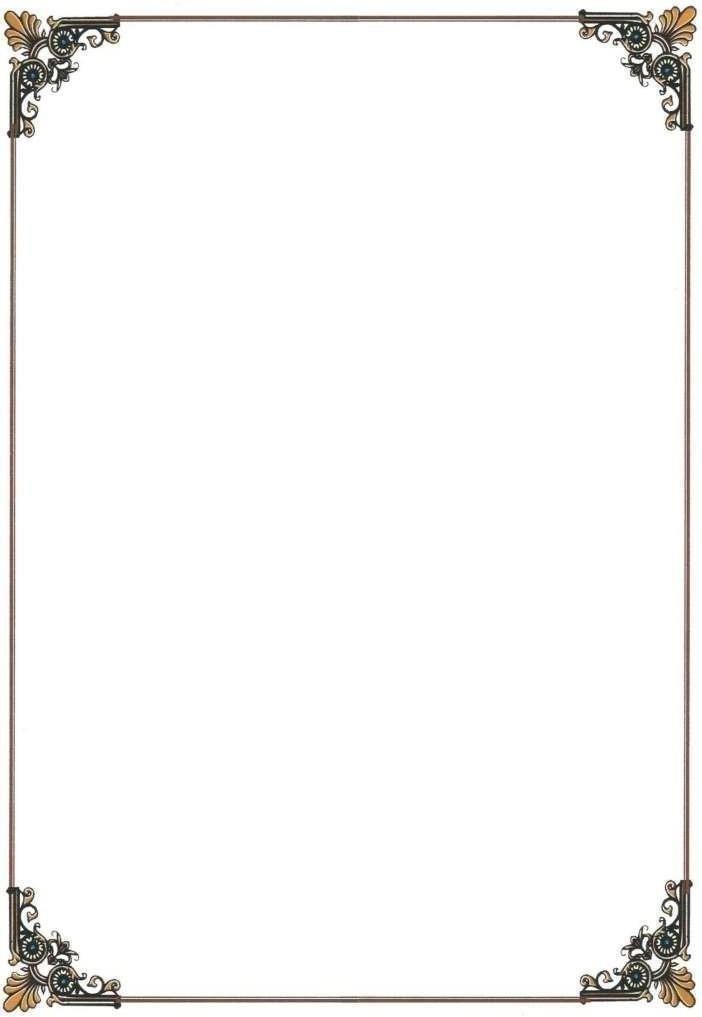
*(Affiliated to JNTU, KAKINADA)*

**(Recognized by A.I.C.T.E., Accredited by N.B.A & N.A.A.C with ‘A+’ Grade,New Delhi)**

**CHINNA AMIRAM, BHIMAVARAM**

**(2021-25)**

**S.R.K.R. ENGINEERING COLLEGE (AUTONOMOUS)**



*(Affiliated to JNTU, KAKINADA)*

**(Recognized by A.I.C.T.E., Accredited by N.B.A & N.A.A.C with ‘A+’ Grade, New**

**Delhi)**

**CHINNA AMIRAM, BHIMAVARAM-534204**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**BONAFIDE CERTIFICATE**

*This is to certify that this project work entitled* **“AI-DRIVEN NETWORK MONITORING AND DDOS DETECTION USING SDN AND PROGRAMMABLE DATA PLANES”** *is*

*the bonafide work submitted by* **Mr.P.CHAITANYA CHANDU (Regd.No.21B91A04H8), Mr.M.PRASANNA KUMAR (Regd.No.21B91A04F0), Mr.M.SAHIL (Regd.No.**

**21B91A04E9) Mr.S.RISHI (Regd.No.21B91A04L0)** *submitted in the partial fulfilment of the requirements for the award of the degree of* **BACHELOR OF TECHNOLOGY** *in* **ELECTRONICS AND COMMUNICATION ENGINEERING** *during the academic year* **2021-2025.**

**Guide Head of the Department:**

**Mr. N. KISHORE CHANDRA DEV Dr. N. UDAYA KUMAR**

**MTech MTech, Ph.D., MISTE, SMIEEE, FIEETE, FIE**

**Assistant Professor of ECE Department Professor & Head, Department of E.C.E**

## CERTIFICATE OF EXAMINATION

This is to certify that we have examined the report and here by accord our approval of it is a project carried out and presented in a manner required for its acceptance on a partial fulfilment for the award of the degree of BACHELOR OF TECHNOLOGY in ELECTRONICS AND COMMUNICATION ENGINEERING for which it had been submitted. This approval does not necessarily endorse or accepts every statement made, opinion expressed, or conclusions drawn as recorded in the project report, it only signifies the acceptance of the report for the purpose for which it is submitted.

**EXTERNAL EXAMINER INTERNAL EXAMINER**

i

## DECLARATION

This is to certify that the project entitled “**AI-Driven Network Monitoring and DDoS Detection Using SDN and Programmable Data Planes”** which is submitted by **P. Chaitanya Chandu (21B91A04H8), M. Prasanna Kumar (21B91A04F0), M. Sahil (21B91A04E9), S.**

**Rishi(21B91A0L0)** in partial fulfilment of the requirement for the award in degree in B.Tech in Electronics and Communication Engineering of S.R.K.R Engineering College[A], affiliated to JNTU KAKINADA. It comprises only our original work and due acknowledgement has been made in text to all other material used.

Date:

P. CHAITANYA CHANDU

M. PRASANNA KUMAR

M. SAHIL

S. RISHI

ii

## ACKNOWLEDGEMENTS

First and foremost, I would like to express my profound gratitude and sincere thanks to my esteemed guide Sri. N. KISHORE CHANDRA DEV, Assistant Professor, Department of ECE, for his excellent guidance and constant encouragement throughout the project work. His timely guidance and motivation helped me to complete the project work.

I would like to express my sincere thanks to **Dr. N. Udaya Kumar**, Professor, Head of the Department of ECE, S.R.K.R Engineering College, Bhimavaram for his encouragement, cooperation, and constant review during the project.

I would like to express my sincere thanks to **Dr. K V MURALI KRISHNAM RAJU**, Principal, S.R.K.R Engineering College, Bhimavaram for giving me this opportunity for the successful completion of my degree.

I would also like to thank other teaching and non-teaching staff for their assistance and help extended. I thank one and all that have contributed directly or indirectly to my project.

**PROJECT ASSOCIATES**

P. CHAITANYA CHANDU

M. PRASANNA KUMAR

M. SAHIL

S. RISHI

iii

## ABSTRACT

Machine Learning (ML) is increasingly utilized for accurate data classification in network monitoring and control. However, the computational complexity of ML algorithms makes them suitable for centralized control planes, while their reliance on large volumes of data from the data plane risks overwhelming control communication channels. To address this, we propose PRIVACY- PRESERVING AUTHENTICATION, an architecture that preprocesses and collects essential traffic data directly in the data plane using programmable switches compatible with NS2. This approach selectively mirrors data to the control plane for deeper analysis, leveraging NS2's support for dynamic data plane pipeline reconfiguration. As a result, it minimizes memory usage in the data plane, reduces data exchange volume by over 75%, and maintains high performance. The architecture's effectiveness is demonstrated through a volumetric DDoS detection use case, showcasing its ability to optimize network resource usage while ensuring secure and scalable monitoring.

iv

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE**  **NO** |
|  | **ACKNOWLEDGEMENTS** | iii |
|  | **ABSTRACT** | iv |
|  | **TABLE OF CONTENTS** | v-vi |
|  | **TABLE OF DIAGRAMS & SCREENSHOT** | vii |
|  | **TABLE OF SYMBOLS** | viii |
|  | **TABLE OF ABBREVIATIONS** | ix |
| **1.** | **INTRODUCTION** | 1-3 |
|  | 1.1. General Introduction | 2 |
|  | 1.2. Project Objectives | 2 |
|  | 1.3. Problem Statement | 3 |
|  | 1.4 Project Scope | 3 |
|  | 1.5 Algorithm | 3 |
| **2.** | **SYSTEM PROPOSAL** | 4-11 |
|  | 2.1. Existing System | 4 |
|  | 2.1.1 Disadvantages |  |
|  | 2.2. Proposed System | 4 |
|  | 2.2.1 Advantages |  |
|  | 2.3. Literature Survey | 5-11 |
| **3.** | **SYSTEM DIAGRAMS** | 12-17 |
| **4.** | **IMPLEMENTATION** | 18-22 |
|  | 4.1. Modules | 18 |
|  | 4.2. Modules Description | 18-22 |
| **5.** | **SYSTEM REQUIREMENTS** | 23-27 |
|  | 5.1. Hardware Requirements | 23 |
|  | 5.2. Software Requirements | 23 |
|  | 5.4. Testing of Products | 24-27 |

v

|  |  |  |
| --- | --- | --- |
| **6.** | **RESULTS** | 28-32 |
| **7.** | **CONCLUSION** | 33 |
| **8.** | **FUTURE ENHANCEMENT** | 34 |
| **9.** | **REFERENCES** | 35-36 |

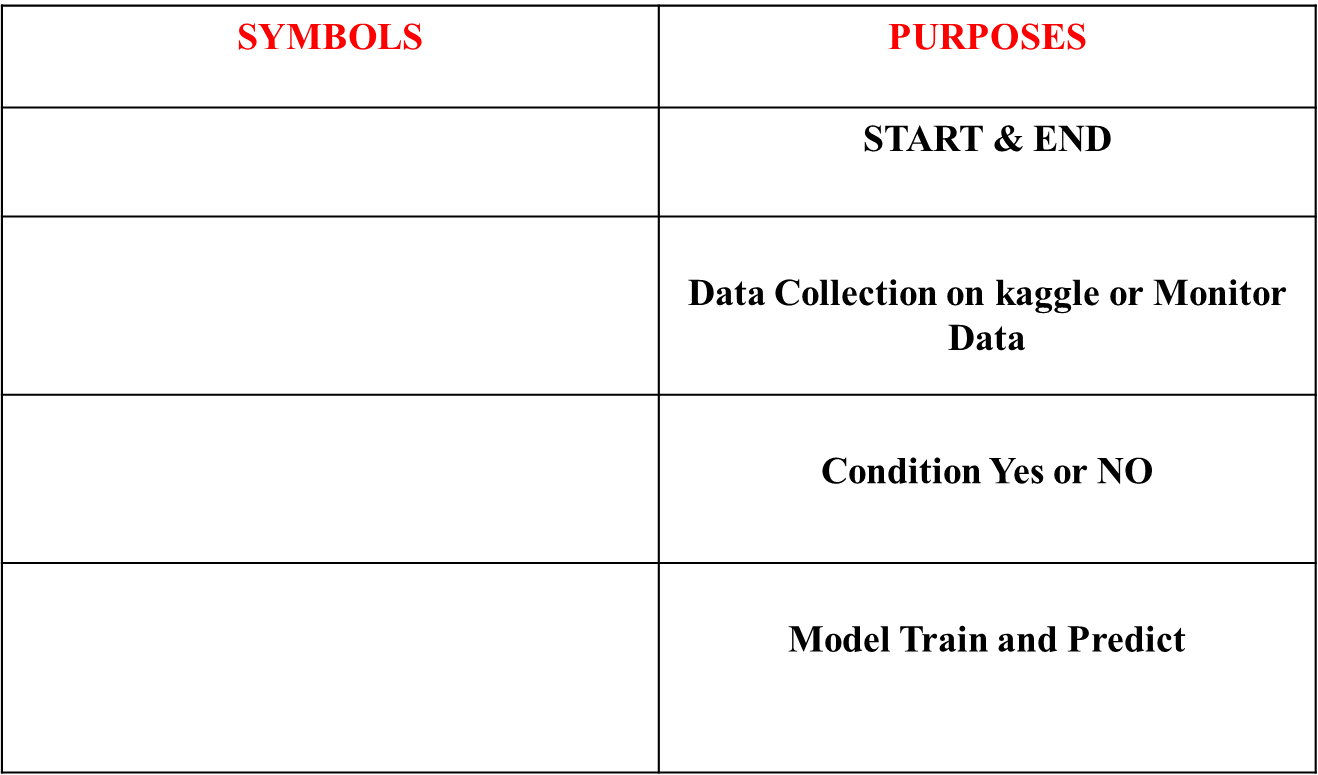
**TABLE OF DIAGRAM**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **DIAGRAM** | **PAGE**  **NO** |
|  | **SYSTEM DIAGRAMS** |  |
| **3.1** | Architecture Diagram | 12 |
| **3.2** | Flow Diagram | 13 |
| **3.3** | UML Diagram | 14 |
| **3.4** | ER Diagram | 15 |
| **3.5** | Sequence Diagram | 16 |
| **3.6** | Activity Diagram | 17 |

**TABLE OF SCREENSHOT**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **SCREENSHOT** | **PAGE NO** |
| **6.1** | Sumo Simulation | 28 |
| **6.2** | Data EDA plot | 29 |
| **6.3** | INTRUSION PLOT | 30 |
| **6.4** | CNN Intrusion Detection: Accuracy & Loss over Epochs | 31 |
| **6.5** | CNN Algorithm plot for Accuracy and Classification report | 32 |

**TABLE OF SYMBOLS**



**TABLE OF ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **KEYWORDS** | **ABBREVIATIONS** |
| 1 | DDoS | Distributed Denial of Service |
| 2 | NS2 | A programming language for protocol- independent packet processing |
| 3 | PRIVACY- PRESERVING AUTHENTICATION | Proposed architecture for data collection and analysis |
| 4 | NMS | Network Management System |
| 5 | SDN | Software-Defined Networking |
| 6 | API | Application Programming Interface |
| 7 | TCP | Transmission Control Protocol |
| 8 | UDP | User Datagram Protocol |
| 9 | VPN | Virtual Private Network |
| 10 | IDS | Intrusion Detection System |
| 11 | NAT | Network Address Translation |
| 12 |  |  |
| 13 | BGP | Border Gateway Protocol |
| 14 | IPS | Intrusion Prevention System |

**CHAPTER 1**

## INTRODUCTION

PRIVACY-PRESERVING AUTHENTICATION is a viable approach for implementing a closed-loop monitoring system that intercepts network behaviors and takes real-time actions. This architecture aims to minimize congestion in the control channel between the data and control planes, particularly during abnormal behaviors. We validate our solution through a volumetric Distributed Denial of Service (DDoS) attack detection, demonstrating how it can reduce the impact on the control channel while maintaining high detection rates. Following the Software Defined Networking (SDN) paradigm, our approach separates detection logic between a simple data plane and a more complex control plane strategy, striving for an optimal balance between monitoring performance, computational complexity, and control channel utilization.

Two key concepts relevant to our approach, particularly in the context of 5G, are (I) control and data plane separation (the CUPS approach) and (II) network programmability. CUPS enhances flexibility and scalability by decoupling logical problems from data forwarding issues, while programmability enables networks to respond effectively to unwanted situations. This closed-loop system allows the control plane to gather real-time information about the network's status and issue directives to the data plane to modify its behavior accordingly.

In summary, fitting ML models to the programmable data plane presents challenges, including the need for nontrivial operations to optimize code and memory consumption, as well as achieving high inference performance. Consequently, PRIVACY-PRESERVING AUTHENTICATION employs simple sketch-based strategies in the data plane for coarse- grained monitoring (CGM), while utilizing ML for in-depth analysis through fine-grained monitoring (FGM). Our proposed FGM DDoS strategy leverages Convolutional Neural Networks (CNN) to classify network traffic but incorporates data aggregation logic in the data plane. This approach alleviates control channel congestion by sending only extracted features from suspect traffic to the control plane via NS2 digests, thus optimizing resource usage during volumetric attacks.

## GENERAL INTRODUCTION

The increasing complexity and scale of modern networks have created a pressing need for effective and efficient network monitoring and control solutions. Machine learning (ML) algorithms have emerged as a promising approach, capable of analyzing large volumes of network data to detect anomalies, predict failures, and optimize performance. However, the centralized control plane architecture of contemporary networks presents challenges for deploying ML-based solutions. While these algorithms are well-suited for the control plane due to their computational complexity, they heavily depend on data collected from the distributed data plane. This reliance necessitates the transfer of substantial amounts of data to the control plane, which can congest the control communication channel and negatively impact overall system performance.

To address this issue, modern switches can perform line-rate data preprocessing and aggregation directly in the data plane. By reducing the volume of data that needs to be transferred to the control plane, this approach significantly alleviates communication overhead between the data and control planes. Consequently, it enables a more efficient deployment of ML-based network monitoring and control solutions, allowing for timely and accurate analysis without overwhelming the control channel. This optimization not only enhances the performance of the network but also supports the scalability required in today's complex networking environments.

## PROJECT OBJECTIVE

* + - Reduce the volume of data that needs to be transferred from the data plane to the control plane by performing preprocessing and aggregation in the data plane.
    - Optimize the interface between the data and control planes to minimize the impact on network performance.
    - Leverage the runtime reconfigurability of data plane programmable switches to enable dynamic adjustments to the data plane pipeline based on changing network conditions or monitoring requirements.
    - Improved Performance Analysis.

## PROBLEM STATEMENT

* + - In response to this challenge, we propose PRIVACY-PRESERVING AUTHENTICATION: an innovative architecture that collects relevant data within the data plane and mirrors it to the control plane for sophisticated analysis to reduce the interaction between data and control planes while maintaining high monitoring performance. To evaluate the efficacy of PRIVACY-PRESERVING AUTHENTICATION, we implemented it in the context of a volumetric Distributed

Denial of Service (DDoS) detection use case. Our results demonstrate that, compared to a purely control-plane-based solution, NS2RTHENO significantly reduces data plane memory usage and decreases the volume of exchanged data by over 75%, without degrading overall performance.

## PROJECT SCOPE

* + - The primary focus of this project is on the use case of network monitoring, with a specific emphasis on volumetric Distributed Denial of Service (DDoS) attack detection.
    - The project will explore how the PRIVACY-PRESERVING AUTHENTICATION architecture can be leveraged to enhance the performance and efficiency of DDoS detection compared to traditional, control-plane-based solutions.
    - While the DDoS detection use case will be the main focus, the project's design and implementation will strive to maintain a level of generality and flexibility to support a wider range of network monitoring tasks.

## ALGORITHM

1. Asymmetric Count-min Sketch Algorithm.
2. CNN Lightweight Deep Learning Solution for DDoS Attack Detection Algorithm.

# CHAPTER 2

**SYSTEM PROPOSAL**

## EXISTING SYSTEM

* ML-based security applications. Flow Lens collects features related to packets distribution at line speed and classifies flows directly in the switches, using their CPU. However, though highly flexible and reliable, Flow Lens cannot benefit from the network-wide view provided by a centralized SDN control plane. In addition, it does not envision any data plane pipeline reconfiguration at runtime, as supported by PRIVACY- PRESERVING AUTHENTICATION.
* Reconfiguring the data plane pipeline makes it possible to install specialized pipelines, instead of using a general-purpose one, and to optimize the amount of data exchanged between data and control planes.

## DISADVANTAGE

* + - * Optimizes data exchange between the data plane and control plane, enhancing overall less efficiency.
      * Lacks a network-wide view due to its decentralized nature, which restricts its ability to leverage insights from a centralized SDN control plane.
      * Does not support runtime reconfiguration of the data plane pipeline.
      * PRIVACY-PRESERVING AUTHENTICATION 's adaptability and optimization of the data plane-control plane interaction can lead to better overall performance and resource utilization.

## PROPOSED SYSTEM

Improved detection accuracy: The deep learning-based approach can effectively capture complex patterns in network traffic, leading to better detection performance. Reduced computational overhead: The lightweight design of CNN enables efficient deployment on resource-constrained devices, making it suitable for edge computing and IoT environments. Unified handling of different attack types: CNN's unified approach

simplifies the detection process and makes it more adaptable to various DDoS attack scenarios. Interpretability: The explanatory capabilities of CNN enhance the understanding of the detection process, aiding security analysts in their decision-making. The CNN algorithm has been evaluated on real-world network traffic datasets and has shown promising results in terms of detection accuracy, computational efficiency, and adaptability, making it a valuable tool for DDoS attack mitigation in modern network security environments.

## ADVANTAGE

* + - * Improved Detection Accuracy: Utilizes deep learning to effectively capture complex patterns in network traffic, resulting in enhanced detection performance for various attack types.
      * Reduced Computational Overhead: Designed to be lightweight, enabling efficient deployment on resource-constrained devices, which is ideal for edge computing and IoT environments.
      * Unified Handling of Different Attack Types: Offers a unified approach to detect various DDoS attack scenarios, simplifying the detection process and increasing adaptability.
      * CNN stands out as a valuable tool for DDoS attack mitigation in modern network security environments, combining high performance with resource efficiency and interpretability.

## LITERATURE SURVEY

* + 1. **LITERATURE SURVEY 1**

**Title**: Newton: Intent-Driven Network Traffic Monitoring

**Year**: 30 November 2022

**Author**: Zhaowei Xi; Yu Zhou; Dai Zhang; Kai Gao; Chen Sun; Jiamin Cao; Yangyang Wang

### Technologies and Algorithm Used:

Network monitoring systems are intended to fulfil operators' intentions and serve as critical instruments in modern networks. As network bandwidth and scale continue to rise, network monitors must provide on-demand network monitoring for ever-increasing traffic volumes. However, present monitoring systems are either unable to serve flexible intents on demand or generate considerable overheads. In this research, we offer Newton, an intent-driven traffic monitor that can express operators' intents through traffic monitoring queries while also deploying dynamic and scalable network-wide inquiries. Newton allows operators to flexibly customize and alter queries while maintaining network workflow. Furthermore, Newton proposes systematic optimizations at the device and network levels to reduce resource consumption when delivering queries. Newton combines resources across switches to Traffic Network.

### Advantages:

* Performance time and accuracy
* Flexibility: Newton allows operators to customize and modify monitoring queries dynamically without interrupting the network workflow. This provides a high degree of flexibility in adapting to changing network conditions and monitoring requirements.
* Resource Efficiency: Newton proposes systematic optimizations at the device level and network-wide level to reduce resource consumption while deploying queries.

### Disadvantages:

* Training model prediction on Time is high
* It is based on Low Accuracy
* Driven traffic monitoring, dynamic querydeployment, and systematic optimizations, may be complex and require significant engineering effort.
* Potential Performance Impact: The overhead associated with the dynamic deployment and optimization of monitoring queries may have some impact.

## LITERATURE SURVEY 2

**Title**: Flex Mesh: Flexibly Chaining Network Functions on Programmable Data Planes at Runtime

**Year**: 22-26 June 2023

**Author**: Yu Zhou; Jun Bi; Cheng Zhang; Mingwei Xu; Jinaping Wu

### Technologies and Algorithm Used:

Programmable data planes (PDP) enable operators to implement various functions (e.g. routing and access control) on high-performance switches and define the chains of these functions with a switch profile. However, with the number of deployed functions increasing, the switch profile faces growing complexity during development and inflexibility to chain functions at runtime. This paper presents Flex Mesh, an integrated platform which aims to introduce flexibility and simplicity to PDP while being compatible with existing programmable devices. Flex Mesh designs (1) a set of chaining primitives, so operators can easily describe the function chain for each flow without facing the complexityof customizing the switch profile during development.

### Advantages:

* Performance time and accuracy.
* Results indicate that with minor performance overheads, Flex Mesh can be an efficient development-assistance tool for operators, as well as an automated platform to chain NFs flexibly while keeping conformance to complex policies.

### Disadvantages:

* Training model prediction on Time is high
* It is based on Low Accuracy
* Switch profile faces growing complexity during development and inflexibility to chain functions at runtime.

## LITERATURE SURVEY 3

**Title**: Hyper Tester: high-performance network testing driven by programmable switches

**Year**: 03 December 2023

**Author**: Yangyang Wang

### Technologies and Algorithm Used:

Modern network research and operations are inseparable from network testers to evaluate performance limits of proofs-of-concept, troubleshoot failures, etc. Existing network testers

suffer from either constrained flexibility or a low performance-cost ratio. In this paper, we propose a new network tester, Hyper Tester. The core of Hyper Tester is to leverage new- generation programmable switches for generating and capturing test traffic with high performance, low cost, and remarkable flexibility. We design a series of efficient mechanisms, including template-based packet generation, false-positive-free counter-based queries, and stateless connections to realize various network testing tasks upon switches with limited programmability and resources.

### Advantages:

* Performance time and accuracy
* Results indict evaluations on the hardware testbed show that Hyper Tester supports line- rate packet generation (400Gbps in the testbed) with highly accurate rate control, while Hyper Tester can save $40150 per Tps and 9225W per Tbps when compared with the software network testers.

### Disadvantages:

* Training model prediction on Time is high
* It is based on Low Accuracy
* Switch profile faces growing complexity during development and inflexibility to chain functions at runtime.

## LITERATURE SURVEY 4

**Title**: Detecting Distributed Denial of Service Attacks Using Data Mining Techniques

**Year**: 01 September 2023

**Author**: Khalid Manaa

### Technologies and Algorithm Used:

In this study, we DDoS (Distributed Denial of Service) attack has affected many IoT networks in recent past that has resulted in huge losses. We have proposed deep learning models and evaluated those using latest CICIDS2017 datasets for DDoS attack detection which has provided highest accuracy as 97.16% also proposed models are compared with machine

learning algorithms. Users and organizations find it continuously challenging to deal with distributed denial of service (DDoS) attacks. . The security engineer works to keep a service available at all times by dealing with intruder attacks. The intrusion-detection system (IDS) is one of the solutions to detecting and classifying any anomalous behavior. The IDS system should always be updated with the latest intruder attack deterrents to preserve the confidentiality, integrity and availability of the service. In this paper, a new dataset is collected because there were no common data sets that contain modern DDoS attacks in different network layers, such as (SIDDoS, HTTP Flood).

### Advantages:

* Performance time and accuracy.
* The proposed solution can successfully detect network intrusions and DDOS communication with high precision.
* More Reliable.

### Disadvantages:

* Training model prediction on Time is high.
* It is based on Low Accuracy.
* It is less in efficiency and not give perfect result.
* This finding is disadvantageous to the organization experiencing such attack.
* The difficulty in identifying all articles that are related to this study.

## LITERATURE SURVEY 5

**Title**: A Multiple-Layer Representation Learning Model for Network-Based Attack Detection

**Year**: 01 Jan 2023

**Author**: Khalid Manaa

### Technologies and Algorithm Used:

Accurate detection of network-based attacks is crucial to prevent security breaches of information systems. The recent application of deep learning approaches for network intrusion detection has shown promising. However, the challenges remain on how to dealwith imbalance

data and small samples as well as reducing false alarm rate (FAR). To address these issues, this work has proposed a multiple-layer representation learning model for accurate end-to-end network intrusion detection by combining deep convolutional neural networks (DCNN) with gcForest. The contributions of this work lie in 1) a new data encoding scheme based on P- Zigzag to encode network traffic data into two-dimensional gray-scale images for representation learning without loss of original information; 2) The combination of gcForest and DCNN allows accurate detection on imbalanced data and small scale data with less hyperparameters comparing to most deep learning models, which increase computational efficiency.

### Advantages:

* Performance time and accuracy.
* method outperforms other single deep learning methods (i.e. Alex Net, VGG19, Google Net, InceptionV3, ResNet18) in terms of accuracy, detection rate and FAR, which successfully demonstrates its effectiveness in detecting fine-grained attacks and handling imbalanced datasets with high precision and low FAR.

### Disadvantages:

* Training model prediction on Time is high.
* It is based on Low Accuracy.
* It is less in efficiency and not give perfect result.
* This finding is disadvantageous to the organization experiencing such attack.
* The difficulty in identifying all articles that are related to this study.

## LITERATURE SURVEY 6

**Title**: A Vision for Runtime Programmable Networks

**Year**: 04 November 2022

**Author**: Matty Kadosh

### Technologies and Algorithm Used:

Our community has made significant progress in developing programmable network infrastructure, starting from the control plane and expanding to the data plane. As a latest trend, network devices are becoming runtime programmable while serving live traffic. This allows for reprogramming of individual device programs at fine-grained timescales to add or remove network functions. Many applications and services, however, need control over a combination of devices, including end host stacks, NICs, and switches, to accomplish their goals. We lay out our vision for runtime programmable networks, building upon device-level features to provide live, network-wide, runtime reprogramming.

### Advantages:

* Performance time and accuracy
* A whole-stack approach is needed with new programming models, compiler support, and network management abstractions. We outline a research agenda as a call to arms to the community.

### Disadvantages:

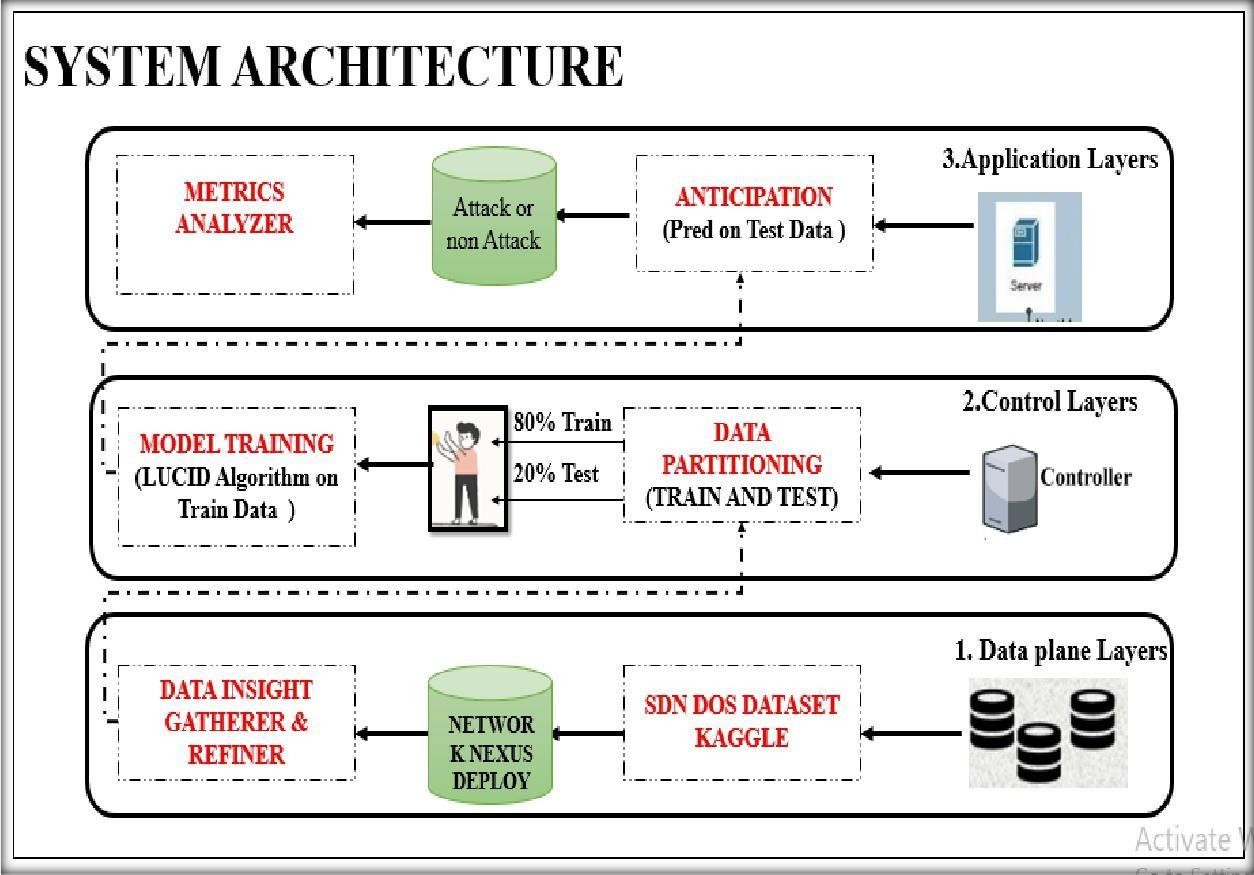
* Training model prediction on Time is high.
* It is based on Low Accuracy.
* The separation of the control and data planes, as well as the introduction of a centralized controller, can add complexity to the network architecture, requiring specialized skills and knowledge to configure and manage.



# CHAPTER 3

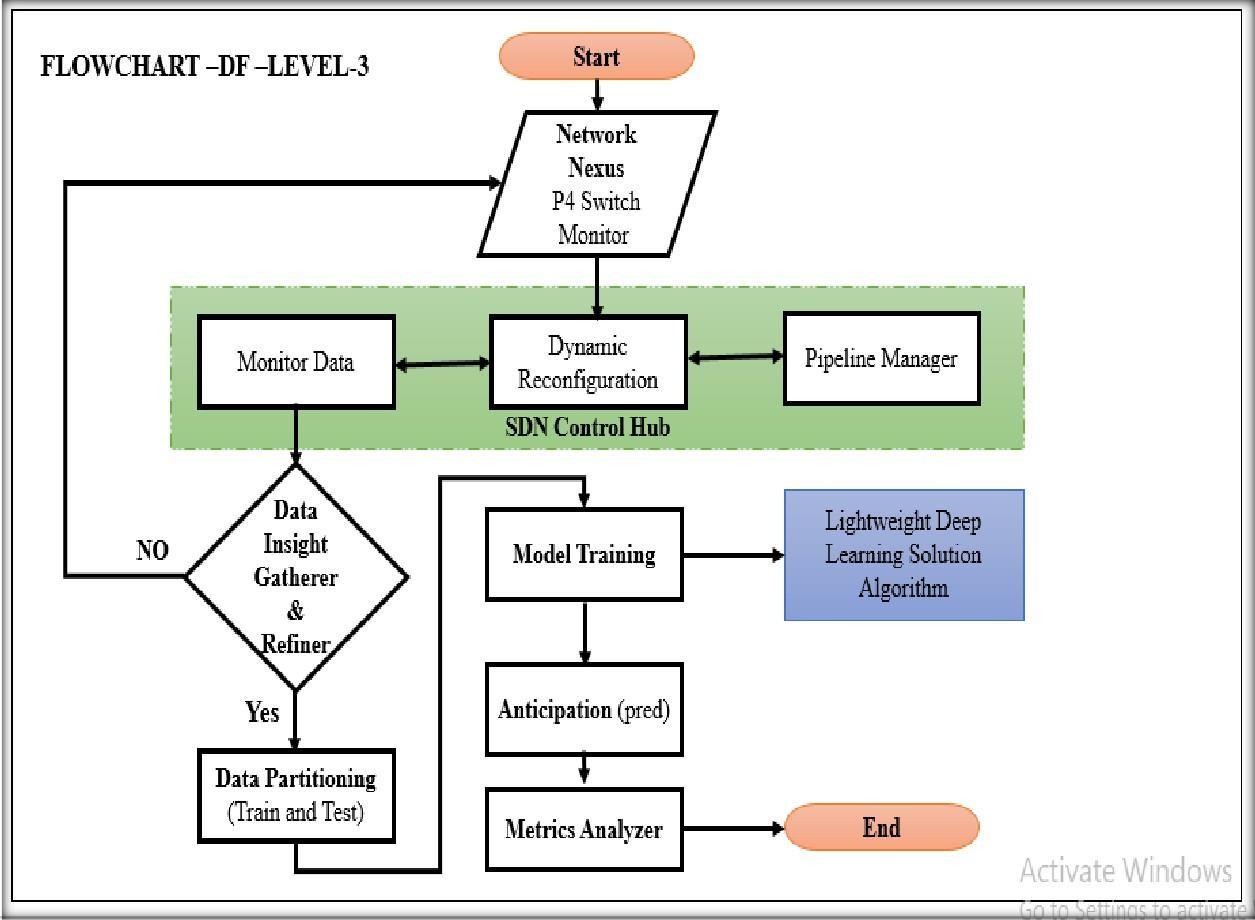
**SYSTEM DIAGRAM**

## ARCHITECTURE DIAGRAM



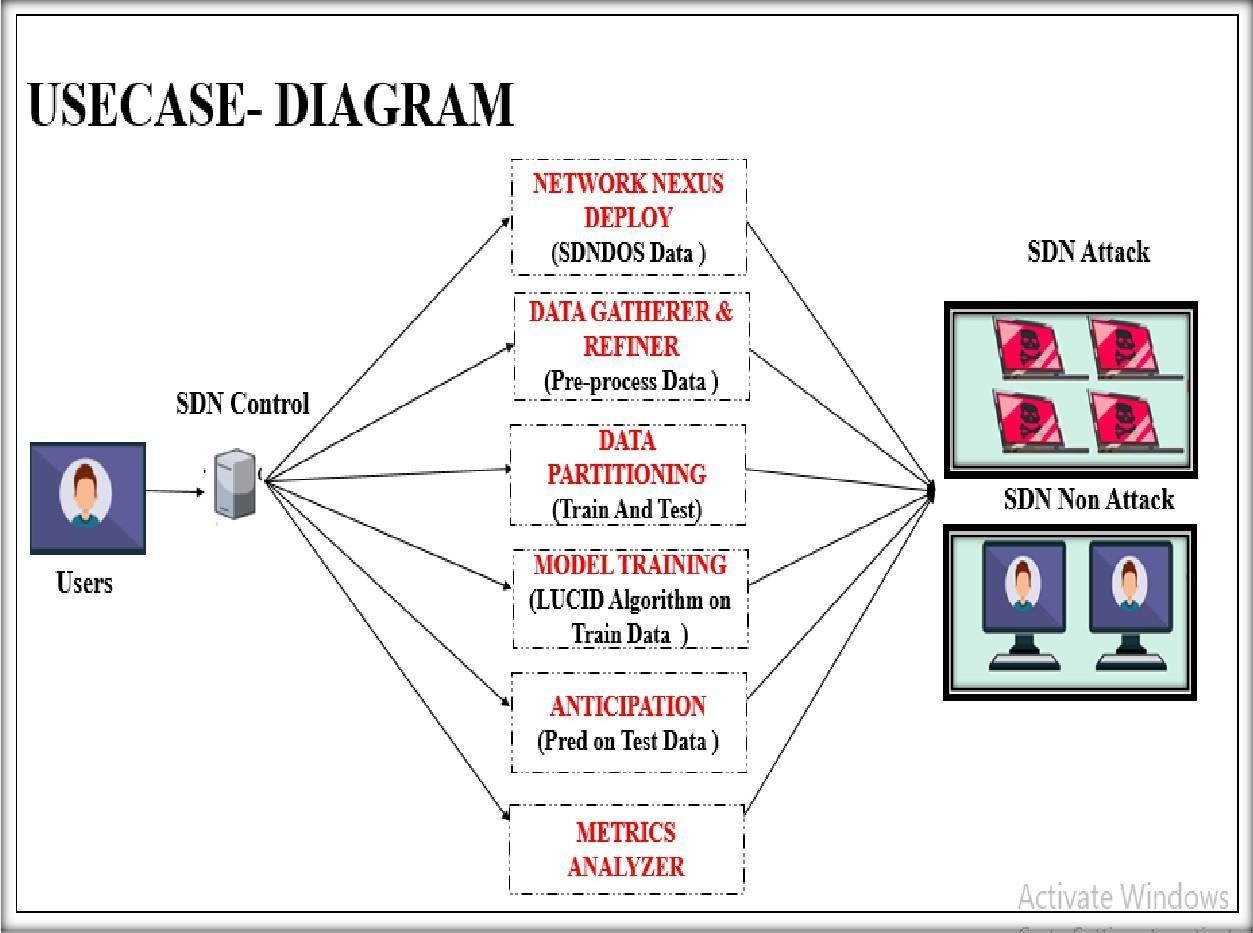
The illustrated system architecture presents a dynamic and intelligent framework for network monitoring and attack detection in SDN environments, integrating NS2-based switches with machine learning. At the data plane layer, SDN DoS datasets from Kaggle are deployed and refined through data insight mechanisms to extract meaningful features. This refined data is then passed to the control layer, where it is partitioned into training and testing subsets, with the LUCID algorithm applied for adaptive model training that accounts for concept drift in network traffic. The trained model moves to the application layer, where it performs realtime predictions on test data to classify network traffic as either normal or malicious. A metrics analyzer continuously evaluates the model’s performance. A key innovation of this system is the reconfigurable CNN-based NS2 switches, which enable adaptive monitoring and dynamic adjustment of switch pipelines based on traffic behavior, thereby optimizing detection accuracy and enhancing overall network performance.

## FLOW DIAGRAM



The presented flowchart outlines a dynamic Level-3 data flow for intelligent network traffic analysis and attack prediction using SDN-enabled infrastructure. The process begins with monitoring data from Nexus P4 switches, where incoming network packets are observed for anomalies. This data is sent to the SDN Control Hub, where dynamic reconfiguration is managed by a pipeline manager, allowing the switch pipelines to adapt in real time based on traffic conditions. Monitored data is then evaluated by a Data Insight Gatherer & Refiner module, which filters and refines the dataset. Upon validation, the refined data is partitioned into training and testing sets for further processing. A lightweight deep learning algorithm is applied during the model training phase to learn patterns in the training data, enabling accurate anticipation or prediction of attacks. The results are assessed through a metrics analyzer to ensure effectiveness. This entire adaptive loop ensures a responsive, intelligent system capable of evolving with changing network behaviors for optimized intrusion detection.

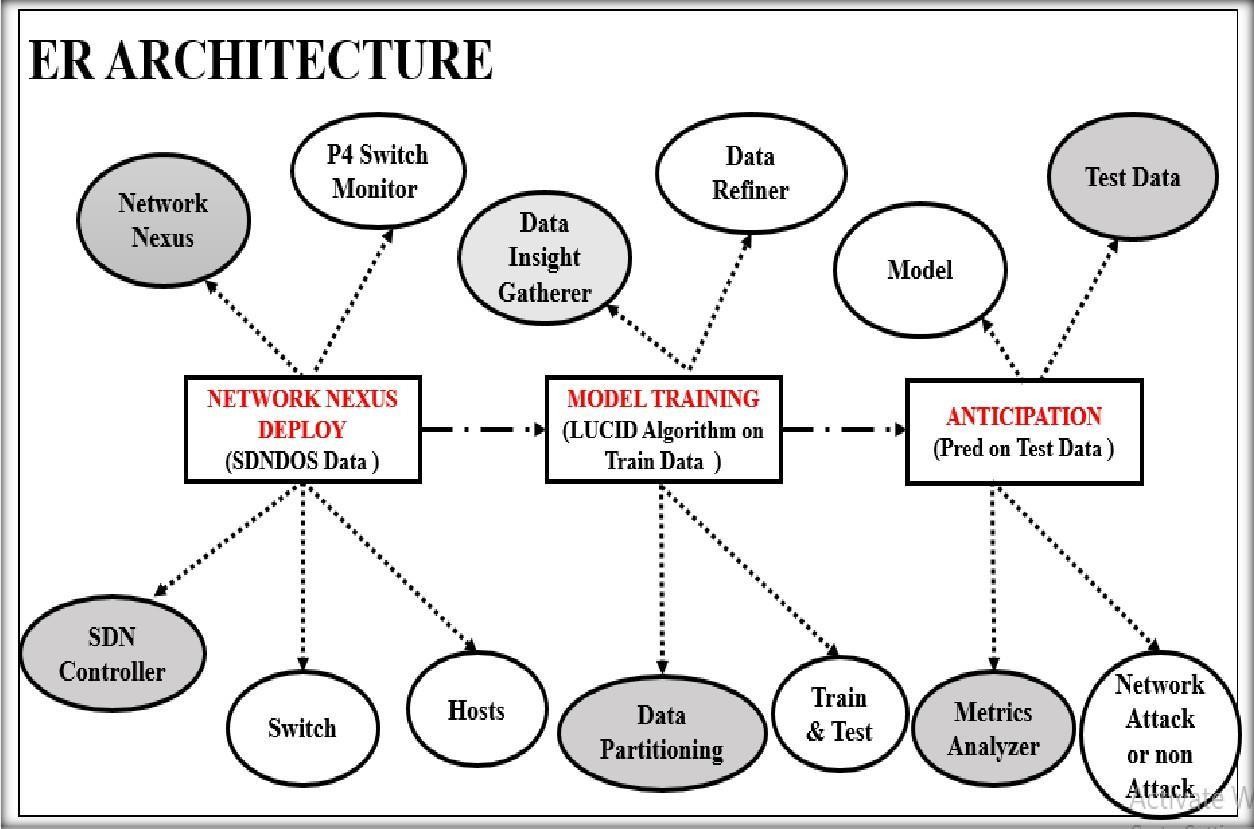
## USE CASE DIAGRAM



The use case diagram illustrates the interaction between the user—typically a network administrator— and the SDN control system in a secure network monitoring environment. The user initiates actions such as deploying the SDN-DOS dataset through the Network Nexus and initiates the data gathering and refining phase, which involves preprocessing the incoming traffic data. Once the data is refined, it is partitioned into training and testing sets, enabling the model training process to begin using the LUCID algorithm on the training data. The trained model is then used to anticipate or predict potential SDN- based attacks on the test data. The final output is analyzed through a metrics analyzer to evaluate the system's prediction performance. The outcomes are visually classified into either “SDN Attack” or “SDN Non-Attack” categories. Throughout this process, the user can view generated reports and configure pipeline settings manually, enhancing system responsiveness and adaptability for proactive

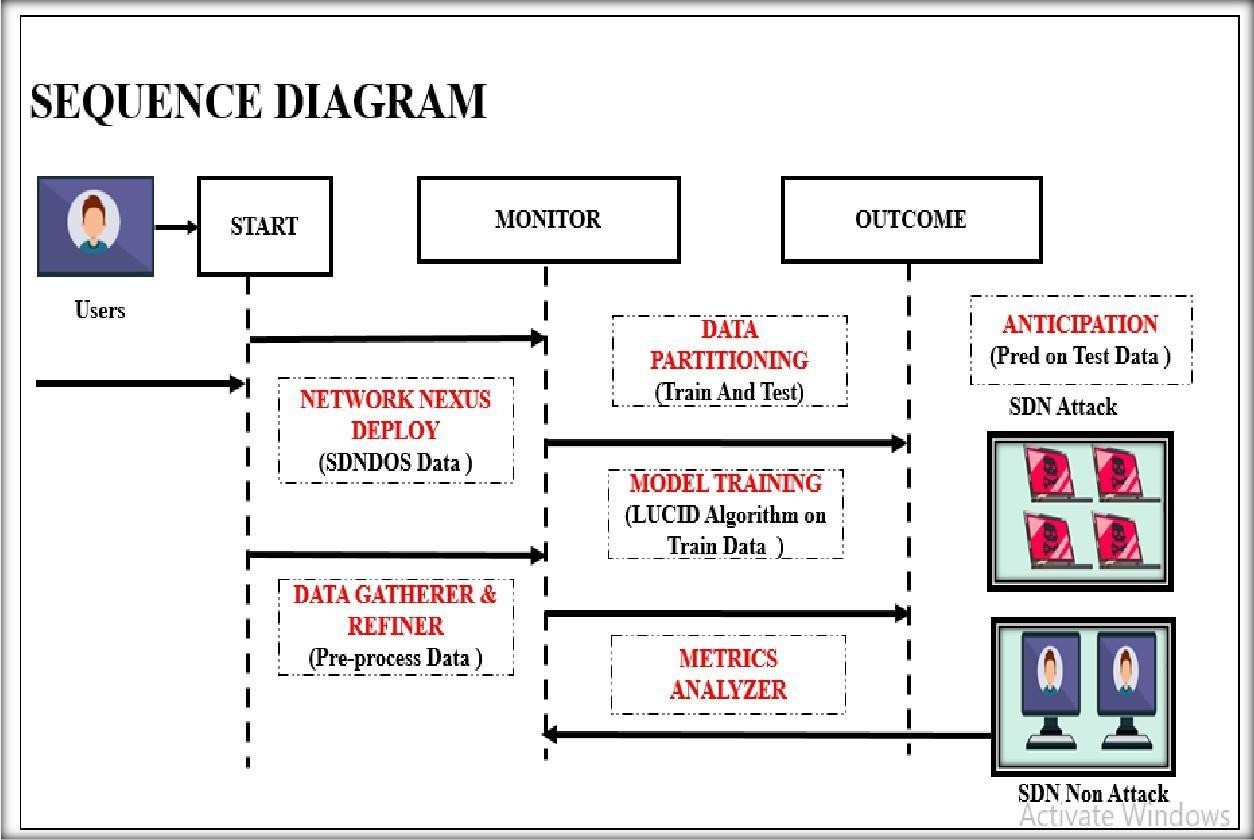
Network security management.

## SEQUENCE DIAGRAM



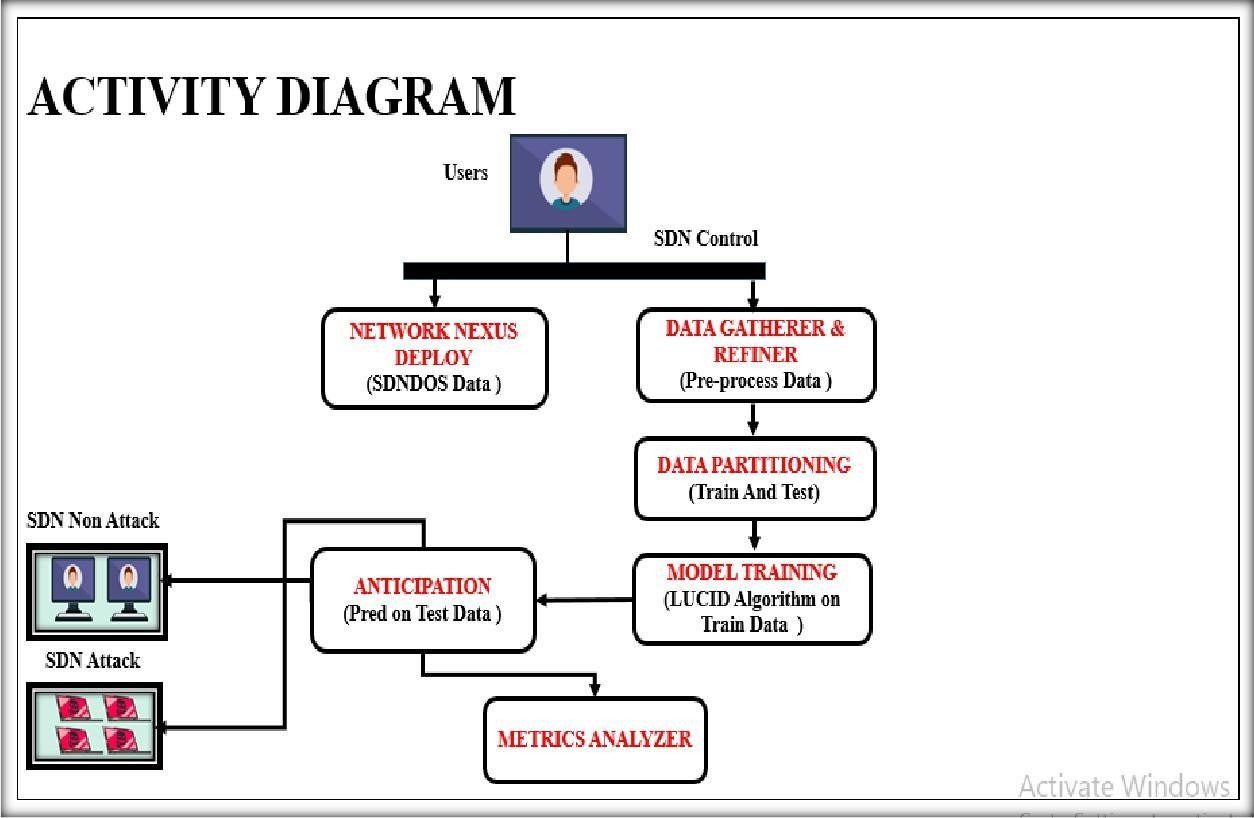
The ER architecture diagram provides a detailed overview of the entities and their relationships within the SDN-based monitoring and attack detection system. The process begins with the deployment of the SDN-DOS dataset through the Network Nexus, which includes integration with components like switches, hosts, and SDN controllers. Data gathered from the network is monitored using P4 switch monitors and passed through a series of processing stages including data insight gathering and refining. Once the data is preprocessed, it undergoes partitioning into training and testing sets, enabling the training phase using the LUCID algorithm. The trained model is then used in the anticipation phase to make predictions on the test data. These predictions are evaluated by a metrics analyzer, leading to the classification of the traffic as either a network attack or non-attack. The diagram captures how different system modules such as the controller, monitors, data partitioner, and analyzers are interrelated to build an efficient and intelligent monitoring architecture.

## SEQUENCE DIAGRAM



The sequence diagram illustrates the flow of interactions in the SDN-based monitoring system initiated by the user, typically a network administrator. The process starts with the deployment of the SNDNOS dataset via the Network Nexus, followed by data preprocessing using the Data Gatherer and Refiner. This refined data is then passed to the monitoring phase, which includes partitioning into training and testing sets and training a model using the LUCID algorithm. The trained model predicts outcomes on the test data in the anticipation phase. These predictions are analyzed by the Metrics Analyzer, resulting in either an SDN attack or non-attack classification. The diagram captures the step-by-step transition from user initiation to data processing, training, prediction, and outcome generation, forming a structured sequence of operations within the intelligent monitoring system..

## ACTIVITY DIAGRAM



This activity diagram outlines the workflow for dynamic pipeline reconfiguration in NS2 switches, emphasizing the role of deep learning in analyzing traffic data and enhancing monitoring efficiency User: The network administrator who initiates the monitoring process. Monitoring System: The central system managing monitoring and reconfiguration. NS2Switch: Programmable switches configured based on monitoring needs. Monitoring Module: Collects traffic data from the network. Report Generator: Creates reports based on analyzed data. Deep Learning Model: A deep learning model that analyzes traffic patterns for insights and detects anomalies.

# CHAPTER 4 IMPLEMENTATION

## MODULES

* + 1. Network Nexus
    2. SDN Control Hub
    3. Data Insight Gatherer and Refiner
    4. Data Partitioning(Train and Test)
    5. Model Training
    6. Anticipation (pred)
    7. Metrics Analyzer

## MODULES DESCRIPTION

* + 1. **NETWORK NEXUS**
       - Traffic Preprocessing Module captures and filters incoming data.
       - Data Distillation Module processes and distills this data to its essential components.
       - Runtime Reconfiguration Module ensures the data plane adapts to changing network conditions.
       - Control Plane Communication Module transfers distilled data to the control plane.
       - ControlPlane Analysis Module performs in-depth analysis.
       - Volumetric DDoS Detection Module specifically addresses DDoS threats.
       - Memory Management Module ensures efficient memory use throughout the process.

## SDN CONTROL HUB

* + - * The SDN controller is the brain of the SDN architecture, responsible for the centralized control and management of the network.
      * It sits in the control plane, separated from the data forwarding plane the SDN controller maintains a global view of the network, making it the central point for network intelligence and decision-making.
      * It has the ability to program the network behavior by configuring the forwarding devices (switches, routers, etc.) through southbound interfaces.
      * The SDN controller exposes northbound interfaces, typically using standard APIs (e.g., OpenFlow, NETCONF, and REST), allowing network applications and management systems to programmatically control and configure the network.

## DATA INSIGHT GATHERER AND REFINER

* + - * Data selection is the process of determining the appropriate data type and source, as well as suitable instruments to collect data.
      * Data selection precedes the actual practice of data collection and it is the process where data relevant to the analysis is decided and retrieved from the data collection.
      * In this project, the DDOS dataset is used for detecting Attack type prediction.

## DATA REFINER

* + - * The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.
      * Missing Data:
      * This situation arises when some data is missing in the data. It can be handled in various ways.
      * Ignore the tuples:
      * This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.
      * Fill the Missing values:
      * Encoding Categorical data: That categorical data is defined as variables with a finite set of label values. That most machine learning algorithms require numerical input and output variables. That an integer and one hot encoding is used to convert categorical data to integer data.

## DATA PARTITIONING(TRAIN AND TEST)

* Data splitting is the act of partitioning available data into two portions, usually for cross-validator purposes.
* One Portion of the data is used to develop a predictive model and the other to evaluate the model's performance.
* Separating data into training and testing sets is an important part of evaluating data mining models.
* Typically, when you separate a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing.
* To train any machine learning model irrespective of what type of dataset is being used you have to split the dataset into training data and testing data.

## MODEL TRAINING

* + - * Classification is the problem of identifying which of a set of categories a new observation belongs to, on the basis of a training set of data containing observations and whose categories membership is known

.

**Asymmetric Count-min Sketch Algorithm**

* Network traffic monitoring, where a few IP addresses may dominate the traffic.
* Database queryoptimization, where some queries are much more frequent than others.
* By employing the ACMS algorithm, these systems can achieve more accurate frequency estimations and better resource allocation, leading to improved performance and efficiency in handling skewed data distributions.

**CNN Lightweight Deep Learning Solution for DDoS Attack Detection Algorithm**

* CNN employs a lightweight deep learning model, which is designed to be efficiently executed on network devices, such as programmable switches or edge devices.
* The deep learning model is trained to detect DDoS attack patterns in network traffic data.
* CNN focuses on extracting relevant features from network traffic data, including packet-level characteristics (e.g., packet size, inter-arrival time) and flow-level statistics (e.g., flow duration, packet count).
* This allows the data plane to adapt and optimize the feature extraction and detection process based on the network conditions and observed traffic patterns.

### ANTICIPATION (pred)

* + - * CNN can be deployed in various network architectures, including traditional networks, software-defined networks (SDN), and edge computing environments.
      * The algorithm can be integrated with existing network monitoring and security solutions to enhance the overall DDoS attack detection and mitigation capabilities.
      * Predictive models can also be used to detect anomalies in network traffic or behavior, which can be indicative of potential security threats, such as DDoS attacks or network intrusions.
* By establishing baseline patterns of normal network activity, predictive models can deviate from the expected behavior, enabling early detection and response to potential threats.

## METRICS ANALYZER

The Final Result will get generated based on the overall classification and prediction. The performance of this proposed approach is evaluated using some measures like, **Accuracy**

* + - * Accuracy of classifier refers to the ability of classifier. It predicts the class label correctly and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.

𝐴𝐶 = (𝑇𝑃 + 𝑇𝑁)/(𝑇𝑃 + 𝑇𝑁 + 𝐹𝑃 + 𝐹𝑁)

### Precision

* + - * Precision is defined as the number of true positives divided by the number of true positives plus the number of false positives.

𝑃𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛 = 𝑇𝑃/(𝑇𝑃 + 𝐹𝑃)

### Recall

* + - * Recall is the number of correct results divided by the number of results that should have been returned. In binary classification, recall is called sensitivity. It can be viewed as the probability that a relevant document is retrieved by the query.

Recall= TP/(TP+FN)

## ROC

* + - * ROC curves are frequently used to show in a graphical way the connection/trade-off between clinical sensitivity and specificity for every possible cut-off for a test or a combination of tests. In addition, the area under the ROC curve gives an idea about the benefit of using the test(s) in question.

### Confusion matrix

* + - * A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier”) on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.



# CHAPTER 5 SYSTEM REQUIREMENTS

### Software Requirements

* + - * + O/S : Ubuntu
        + Language : Python
        + Front End : Sumo and Spyder

### Hardware Requirements

* System : Pentium IV 2.4 GHz
* Hard Disk : 200 GB
* Mouse : Logitech.
* Keyboard : 110 keys enhanced
* Ram : 4GB

### Testing of Product

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. The candidate system is subject to variety of tests-on-line response, Volume Street, recovery and security and usability test. A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational.

## UNIT TESTING

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as ‘module testing’. The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

## WHITE BOX TESTING

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases. Using the white box testing methods, we derived test cases that guarantee that all independent paths within a module have been exercised at least once.

## BLACK BOX TESTING

* Black box testing is done to find incorrect or missing function
* Interface error
* Errors in external database access
* Performance errors
* Initialization and termination errors

In ‘functional testing’, is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called ‘black box testing’. It tests the external behavior of the system. Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

### Training on the Application Software

After providing the necessary basic training on the computer awareness, the users will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design, type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. This training may be different across different user groups and across different levels of hierarchy.

### System Maintenance in Software Development

System Maintenance is a crucial phase of the software development life cycle that begins after the successful implementation of a system. Its primary goal is to ensure that the software continues to perform useful functions and remains adaptable to changing environments, such as social, technical, or operational changes. Maintenance helps enhance system functionality, user interface, and performance. It goes beyond just fixing bugs—it ensures the software evolves with user and environmental needs.

### Corrective Maintenance

Addresses issues discovered after deployment. Since not all errors are identified during testing, corrective maintenance involves diagnosing and fixing bugs reported by users during actual system use.

### Adaptive Maintenance

Modifies the software to adapt to changes in the external environment such as hardware, operating systems, or regulations. This ensures continued system compatibility and functionality.

### Perceptive (Perfective) Maintenance

Involves enhancements based on user feedback. These include adding new features, improving existing ones, or updating system performance. It accounts for the majority of maintenance efforts.

### Preventive Maintenance

Aims at improving the future maintainability and reliability of the software. This includes code optimization, documentation updates, and restructuring for better performance. Techniques like reverse engineering and re-engineering are commonly used.

### Types of Software Testing Ad-hoc testing

This type of software testing is very informal and unstructured and can be performed by any stakeholder with no reference to any test case or test design documents. The person performing Ad-hoc testing has a good understanding of the domain and workflows of the application to try to find defects and break the software. Ad-hoc testing is intended to find defects that were not found by existing test cases.

### Accessibility Testing

In accessibility testing, the aim of the testing is to determine if the contents of the website can be easily accessed by disable people. Various checks such as colour and contrast (for color blind people), font size for visually impaired, clear and concise text that is easy to read and understand.

### Agile Testing

Agile Testing is a type of software testing that accommodates agile software development approach and practices. In an Agile development environment, testing is an integral part of software development and is done along with coding. Agile testing allows incremental and iterative coding and testing.

### API Testing

API testing is a type of testing that is similar to unit testing. Each of the Software APIs are tested as per API specification. API testing is mostly done by testing team unless APIs to be tested or complex and needs extensive coding. API testing requires understanding both API functionality and possessing good coding skills.

### Boundary Value Testing (BVT)

Boundary Value Testing is a testing technique that is based on concept “error aggregates at boundaries”. In this testing technique, testing is done extensively to check for defects at boundary conditions. If a field accepts value 1 to 100 then testing is done for values 0, 1, 2, 99, 100 and 101.

### Fuzz Testing

Fuzz testing or fuzzing is a software testing technique that involves testing with unexpected or random inputs. Software is monitored for failures or error messages that are presented due to the input errors.

### Automated testing

This is a testing approach that makes use of testing tools and/or programming to run the test cases using software or custom developed test utilities. Most of the automated tools provided capture and playback facility, however there are tools that require writing extensive scripting or programming to automate test cases.

### All Pairs testing

Also known as Pair wise testing, is a black box testing approach and a testing method where in for each input is tested in pairs of inputs, which helps to test software works as expected with all possible input combinations.

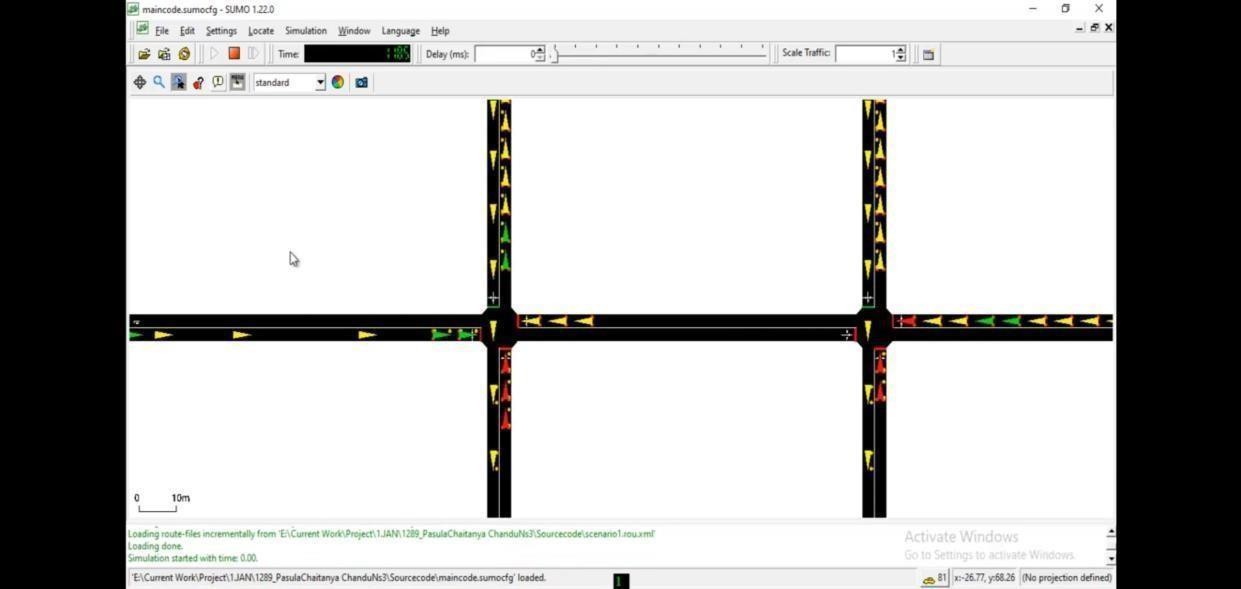
### Beta Testing

This is a formal type of software testing that is carried out by end customers before releasing or handing over software to end users. Successful completion of Beta testing means customer acceptance of the software.

# CHAPTER 6

# RESULTS

### SUMO Simulation

****

* + - **Simulation Tool:**

The figure shows the SUMO (Simulation of Urban Mobility) interface, used for simulating vehicular traffic at an intersection.

### Traffic Representation:

The vehicles are represented using colored arrows and blocks — green (free flow), yellow (moving), and red (waiting or congestion).

### Intersection Mapping:

The four-way intersection visually mimics the structure of SDN switches in a network, where vehicles represent data packets flowing through nodes.

### Analogous to Network Traffic:

This traffic scenario is used to simulate and analyze network conditions, representing normal flow (SDN Non-Attack) and congestion or blocking (SDN Attack).

### Data Source for ML Model:

The traffic patterns generated from SUMO feed into the Data Gatherer & Refiner, later used for model training using the LUCID algorithm.

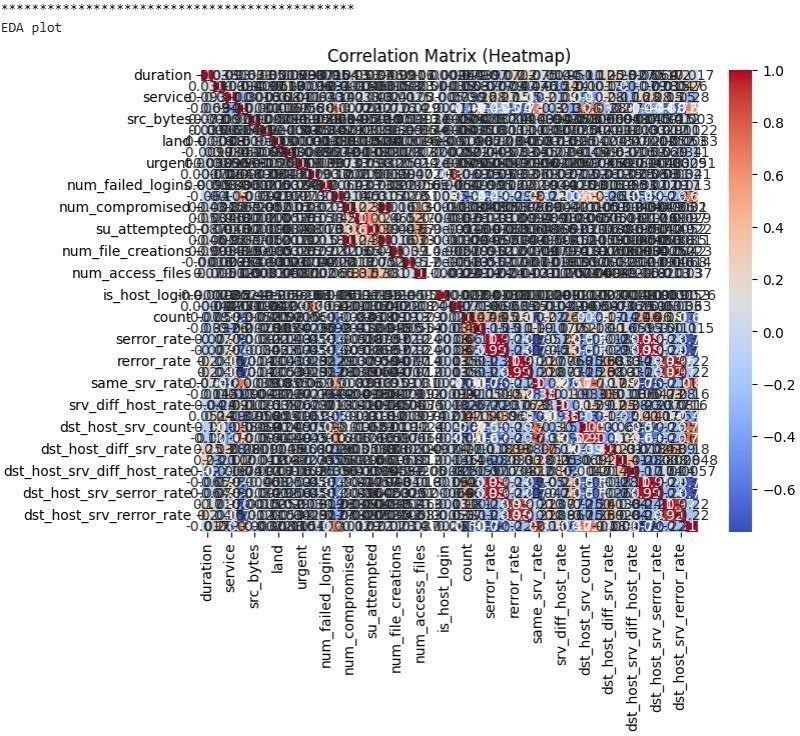
### DoS Attack Simulation:

Congested roads or blocked intersections in SUMO are symbolic of Denial of Service (DoS) attacks in SDN — enabling labeled data generation.

### Performance Evaluation:

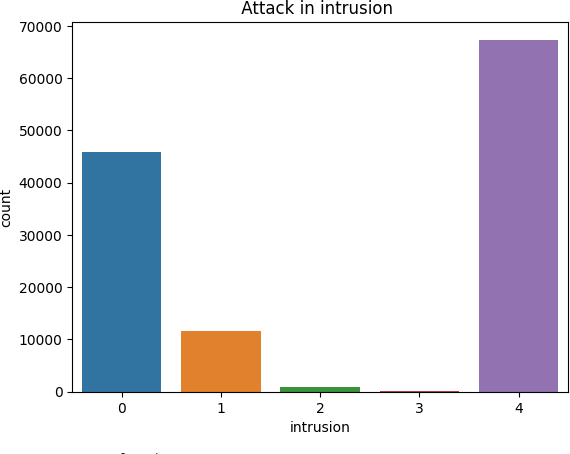
The simulation helps to visualize, validate, and evaluate the effectiveness of the attack detection mechanism by generating realistic traffic behavior.

### EDA Plot

****

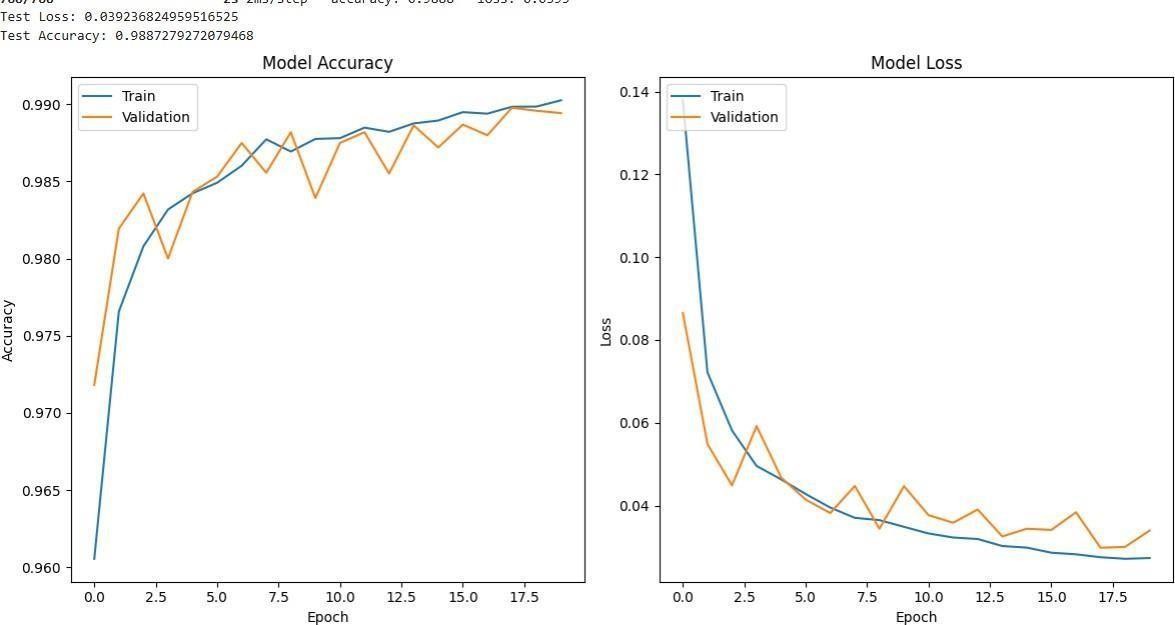
* + - The heatmap displays the correlation between different features used in the dataset for SDN-DOS detection, which helps understand relationships and dependencies among them.
    - Colors range from red (positive correlation) to blue (negative correlation), indicating how features influence each other—useful for identifying meaningful patterns.
    - Features like num\_compromised, num\_access\_files, and su\_attempted show strong inter- correlations, which may point to common behavior during potential attacks.
    - Some features have high correlation with others, suggesting redundancy. Removing such features can reduce dimensionality and improve model performance.
    - This correlation matrix is part of exploratory data analysis (EDA) and plays a crucial role in selecting relevant features before training the machine learning model.

### Intrusion Plot

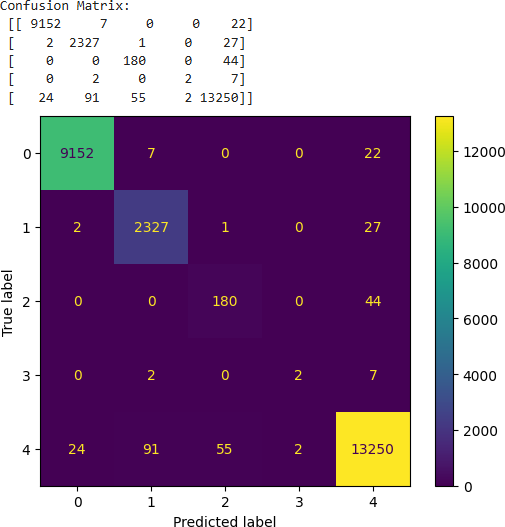
****

* + - This bar chart gives us a clear picture of how different types of intrusions are spread out in the dataset, with each type labeled from 0 to 4.
    - It’s easyto spot that intrusion type 4 is the most common, showing up nearly 70,000 times, followed by type 0 with about 45,000 cases.
    - On the other hand, types 1, 2, and especially type 3 appear way less often, which shows there’s a pretty big imbalance in the data.
    - That kind of imbalance can be tricky during model training—it might lead the model to favor the more common types unless we use methods like oversampling or tweaking the loss function.
    - Overall, this chart is super helpful for understanding how balanced (or not) the dataset is and helps us decide what kind of preprocessing or adjustments we might need before building our model.

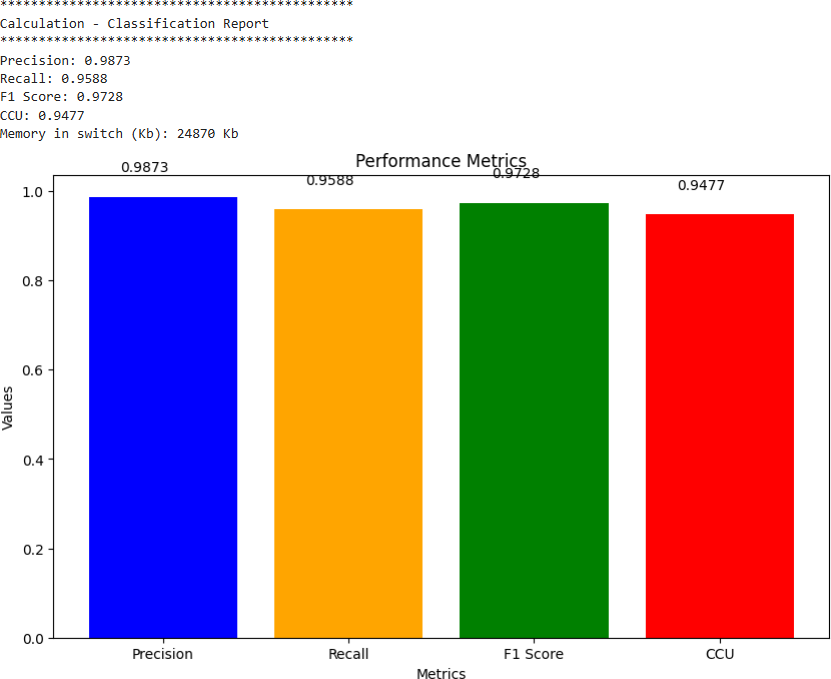
### CNN Intrusion Detection: Accuracy & Loss over Epochs

****

* + - Both training and validation accuracy improve steadily, reaching around 99%, indicating the model is learning effectively.
    - Validation accuracy closely follows training accuracy, suggesting good generalization and minimal overfitting.
    - Training and validation loss decrease consistently, reflecting improved prediction performance.
    - Loss curves remain smooth with no major spikes, showing stable training throughout the epochs.
    - Finally it’s show a test accuracy of 98.87% and test loss of 0.039, confirming strong overall model performance.



### CNN Algorithm plot for Accuracy and Classification report

****

* + - Precision: 0.9873

Indicates that 98.73% of predicted positive cases were actually positive.

* + - Recall: 0.9588

Shows that 95.88% of actual positive cases were correctly predicted.

* + - F1 Score: 0.9728

Harmonic mean of Precision and Recall. High value (97.28%) indicates strong model performance.

* + - CCU (Custom Computation Unit): 0.9477

Possibly a custom or domain-specific metric — might measure model efficiency or cost.

# CHAPTER 7 CONCLUSION

* + - * Minimizing data flow on the control channel for data-driven monitoring tasks is crucial in complicated networks. Introducing a new feature or service should not negatively impact the system.
      * PRIVACY-PRESERVING AUTHENTICATION encourages the use of lightweight and precise monitoring tasks to achieve these needs. It uses NS2-assisted real-time reconfiguration of programmable network devices, resulting in low overhead and traffic loss.
      * This could improve scalability, resilience, and the ability to handle monitoring tasks closer to the data sources, further reducing the control plane burden.

# CHAPTER 8

**FUTURE ENCHANCEMENT**

* + - * Further enhance the ability of PRIVACY-PRESERVING AUTHENTICATION to dynamically adjust the monitoring tasks and reconfigure the data plane pipeline based on changing network conditions and requirements.
      * Develop mechanisms to automatically detect shifts in network traffic patterns or the emergence of new threats and trigger the reconfiguration of monitoring tasks accordingly.
      * Incorporate AI/ML models for real-time anomaly detection to improve the system’s ability to predict and adapt to evolving threat landscapes, thereby optimizing the allocation and prioritization of monitoring resources without compromising user privacy.
      * Enable decentralized and collaborative threat intelligence sharing among network nodes using privacy-preserving techniques like secure multi-party computation or federated learning, allowing the system to respond to global threat trends without centralized data collection.

# CHAPTER 9 REFERENCES

1. Architecture Working Group. View on 5G Architecture. Tech. rep. Available on-line at https://5g-ppp.eu/wpcontent/uploads/2022/02/5G-PPP-5G-Architecture-WP-July2023.pdf. 5G PPP, 2023.
2. F. Callegati et al. “SDN for dynamic NFV deployment”. In: IEEE Communications Magazine 54.10 (2023), pp. 89–95. DOI: 10.1109/MCOM.2023.7588275.
3. Davide Borsatti et al. “Mission Critical Communications Support With 5G and Network Slicing”. In: IEEE Transactions on Network and Service Management 20.1 (2023), pp. 595– 607. DOI: 10.1109/TNSM.2022.3208657.
4. Andrea Melis et al. “P-SCOR: Integration of constraint programming orchestration and programmable data plane”. In: IEEE Transactions on Network and Service Management 18.1 (2023), pp. 402–414.
5. Menghao Zhang et al. “Control plane reflection attacks in SDNs: New attacks and countermeasures”. In: Research in Attacks, Intrusions, and Defenses: 21st International Symposium, RAID 2022, Heraklion, Crete, Greece, September 10-12, 2022, Proceedings 21.

Springer. 2022, pp. 161–183.

1. Graham Cormode and S. Muthukrishnan. “An improved data stream summary: the count- min sketch and its applications”. In: Journal of Algorithms 55.1 (2005), pp. 58–75. ISSN: 0196- 6774. DOI: <https://doi.org/10.1016/j.jalgor.2003.12.001>.
2. R. Doriguzzi-Corin et al. “CNN: A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection”. In: IEEE Transactions on Network and Service Management 17.2 (2023), pp. 876–889. DOI: 10.1109/TNSM.2023.2971776.
3. Athanasios Liatifis et al. “Advancing sdn from openflow to Ns2: A survey”. In: ACM Computing Surveys 55.9 (2023), pp. 1–37.
4. Damu Ding et al. “Design and Development of Network Monitoring Strategies in NS2- enabled Programmable Switches”. In: NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium. 2022.
5. Pat Bosshart et al. “NS2: Programming Protocol-Independent Packet Processors”. In:

44.3 (July 2022), pp. 87–95. ISSN: 0146-4833. DOI: 10.1145/2656877.2656890. URL:

https://doi. org/10.1145/2656877.2656890.

1. Lizhuang Tan et al. “In-band network telemetry: A survey”. In: Computer Networks 186 (2022), p. 107763.
2. Vimalkumar Jeyakumar et al. “Millions of little minions: Using packets for low latency network programming and visibility”. In: ACM SIGCOMM Computer Communication Review 44.4 (2022), pp. 3–14.
3. Yuliang Li et al. “HPCC: High precision congestion control”. In: Proceedings of the ACM Special Interest Group on Data Communication. 2023, pp. 44–58.
4. Naga Katta et al. “Clove: Congestion-aware load balancing at the virtual edge”. In: Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies. 2017, pp. 323–335.
5. Hui Han et al. “Applications of sketches in network traffic measurement: A survey”. In: Information Fusion 82 (2022), pp. 58–85.
6. Tooska Dargahi et al. “A survey on the security of stateful SDN data planes”. In: IEEE Communications Surveys & Tutorials 19.3 (2017), pp. 1701–1725.
7. Ran Ben-Basat et al. “Heavy hitters in streams and sliding windows”. In: IEEE INFOCOM 2023-The 35th Annual IEEE International Conference on Computer Communications. IEEE. 2023, pp. 1–9.
8. Ran Ben-Basat et al. “Efficient measurement on programmable switches using probabilistic recirculation”. In: 2022 IEEE 26th International Conference on Network Protocols (ICNP). IEEE. 2022, pp. 313–323.
9. Lu Tang, Qun Huang, and Patrick PC Lee. “A fast and compact invertible sketch for network-wide heavy flow detection”. In: IEEE/ACM Transactions on Networking 28.5 (2023),

pp. 2350–2363.

Tushar Swamy et al. “Taurus: a data plane architecture for per-packet ML”. In: Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems. 2022, pp. 1099–1114.