INTERNSHIP REPORT

Machine Learning–Driven Traffic Prioritization in Software Defined Networking

Shreesh Srivastava
B. Tech CSE (AI/ML)
Graphic Era Hill University

Organization:
IIT Kharagpur – IEEE Computer Society SBC

Duration:
28 July 2025 – 25 November 2025

Mentor:
Kumarjit Ray

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **IIT Kharagpur IEEE Computer Society Student Branch Chapter** for providing me with the opportunity to work on this research-oriented internship project. Special thanks to my mentor **Kumarjit Ray** for his continuous guidance, technical support, and valuable insights throughout the duration of this internship.

I am also thankful to my university, **Graphic Era Hill University**, for encouraging research-driven learning and providing the foundational knowledge that helped me contribute to this project.

Finally, I extend my sincere appreciation to the authors of the research papers I studied, the SDN community, and the developers of Ryu and Mininet for the tools and documentation that made this project possible.

TABLE OF CONTENTS

INTRODUCTION

Software Defined Networking (SDN) is a modern networking paradigm that separates the control plane from the data plane. It allows centralized, programmable network management using SDN controllers such as **Ryu**, etc.

During this internship, I worked on designing a **Machine Learning–based dynamic network prioritization mechanism** without using traditional QoS queue management. The system learns from flow-level traffic statistics and automatically assigns priority to flows in real time.

This problem is highly relevant today because networks carry diverse traffic — video calls, IoT messages, gaming packets, backups, bulk downloads etc. Traditional QoS requires manual configuration and fixed rules, which cannot adapt to changing traffic patterns.

My research-driven internship aimed to address this challenge by applying **ML to predict traffic importance** based on throughput and flow parameters.

This report provides a detailed explanation of the system design, research background, implementation, dataset, results, and the overall learnings achieved during the internship.

OBJECTIVE OF THE INTERNSHIP

The main objective of this project was:

"To develop a machine learning–enabled SDN controller capable of predicting and prioritizing network traffic dynamically without using traditional QoS queues."

The sub-objectives included:

- Understanding SDN architecture and Ryu Controller
- Studying latest research on ML in networks and semantic communication
- Extracting OpenFlow 1.3 flow statistics
- Creating a clean, ML-ready dataset
- Developing features such as throughput, bytes per packet etc.
- Generating synthetic dataset of 2,500+ flows
- Training a Random Forest model for flow priority prediction
- Integrating ML into Ryu to perform real-time prioritization
- Validating using Mininet and iperf traffic flows

SOFTWARE DEFINED NETWORKING (SDN)

SDN fundamentally changes the way networks are designed and controlled. Traditionally, networking devices contain both:

1. **Control Plane** – routing, policy decisions
2. **Data Plane** – forwarding actual packets

In SDN:

- Control plane is moved to a central controller (software program)
- Devices (switches) become simple forwarding hardware
- Controller pushes rules (flows) using **OpenFlow**

Benefits of SDN

- Centralized control
- Programmability
- Dynamic traffic management
- Faster innovation
- Better security visibility

Why SDN is suitable for ML-based prioritization?

Because the controller has:

- A global view of the entire network
- Access to real-time statistics
- Ability to update flow rules instantly

This centralization makes SDN ideal for ML-driven automation.

LITERATURE REVIEW – RESEARCH PAPERS STUDIED

During the first phase of the internship, I studied three cutting-edge research papers related to intelligent networks, semantic communication, and generative vision of networks.

Below is a summary:

Paper 1: "Intelliscise Wireless Networks From Semantic Communications: A Survey, Research Issues, and Challenges"

Summary:
This paper explains the concept of **semantic communication**, where networks transmit meaning instead of raw bits. It highlights how ML techniques can reduce bandwidth usage by focusing on "important" information. It also discusses challenges like semantic metrics and cross-layer optimization.

Relevance to My Project:
Inspired the idea of prioritizing traffic based on "importance," not just packets — similar philosophy to semantic-aware networking.

Paper 2: "Large Language Models (LLMs) for Wireless Networks: An Overview from the Prompt Engineering Perspective"

Summary:
The paper explores how LLMs like GPT can assist network decision-making, automate configuration, generate routing rules, or interpret traffic patterns. It also introduces prompt engineering for adapting LLM behavior for network tasks.

Relevance:
Demonstrates how ML/LLMs can act as intelligent controllers — motivating the idea of integrating ML models inside SDN controllers.

Paper 3: “Wireless Multi-Agent Generative AI: From Connected Intelligence to Collective Intelligence”

Summary:
The paper discusses multi-agent generative AI systems cooperating in wireless networks to make intelligent decisions. It highlights distributed intelligence, real-time coordination, and adaptive optimization.

Relevance:
Introduced the concept of **adaptive learning** and real-time decision-making — which aligns with my ML-based SDN traffic prioritization.

SYSTEM ARCHITECTURE

The system consists of the following components:

1. Mininet Topology

- Single switch (OVS)
- Three hosts (h1, h2, h3)
- Controlled by Ryu via OpenFlow 1.3

2. Flow Poller Module (Ryu)

- Periodically polls OVS for flow stats
- Extracts fields:
  - packets
  - bytes
  - duration
  - eth/ip src/dst
  - ports
  - protocol
- Calculates **throughput (bps)**
- Saves to **CSV dataset**

3. ML Dataset Generator

- Cleaned and formatted 2,500 rows
- Assigns labels:
  - 1 → High priority
  - 0 → Normal priority
- Based on throughput threshold

4. ML Model

- RandomForestClassifier
- Trained on synthetic dataset
- Accuracy: **~100% on test data**

5. ML-Enabled Ryu Controller

- Loads trained model
- Extracts features from packet_in
- Predicts priority in real time
- Installs flow rule with priority:
    - 100 → High
    - 10 → Low

6. Traffic Generator (iperf)

Used to create:

- High-throughput TCP flows
- UDP flows
- Background network traffic
- Normal ping traffic

This complete pipeline ensures real-time ML-based network prioritization.

DATASET GENERATION & FEATURE ENGINEERING

Dataset Size:

2,500 rows synthetic dataset
+
Real flow data collected via Ryu during testing

Features Used

- n_packets
- n_bytes
- duration_sec
- throughput_bps
- ip_proto
- src_port
- dst_port
- bytes_per_pkt
- is_tcp
- is_udp

Target Label: priority_label

Defined as:

priority_label = 1 if throughput_bps > 1,000,000 else 0

This means:

- Any flow carrying more than 1 Mbps considered **high priority**
- Normal flows (ARP, ping, small traffic) → **low priority**

Dataset Balance Summary

- High priority: ~25.4%
- Low priority: ~74.6%

A reasonably good distribution for classification.

MACHINE LEARNING MODEL DEVELOPMENT

Algorithm Used:

Random Forest Classifier

Reason for selection:

- Handles non-linear data
- Robust to noise
- Good for small datasets
- Excellent accuracy on flow-based classification

Training Results

- Accuracy: **100% (train set)**
- Test Confusion Matrix:

```
[[373   0]
[0   127]]
```

- Precision: 1.0
- Recall: 1.0

Model File:

flow_priority_model.pkl

This file is loaded inside the Ryu controller.

IMPLEMENTATION IN RYU CONTROLLER

The controller performs:

1. Learning Switch Logic

Learns MAC → port mappings.

2. Extracts ML features from packet_in events

Example:

- msg.total_len
- ip_proto
- tcp_src, tcp_dst
- udp_src, udp_dst
- bytes_per_pkt

3. Sends features → ML model → prediction

pred = model.predict(features)

4. Installs Flow Rule Based on Prediction

- High priority → priority=100
- Low priority → priority=10

5. Forwards packets accordingly

High priority flows get placed earlier in flow table and forwarded faster.

EXPERIMENTAL SETUP (MININET)

Mininet Commands Used

sudo mn --topo single,3 --controller=remote --switch ovs

h1 iperf -s &
h2 iperf -c h1 -t 10

h3 ping h1 -c 20

iperf -u h1 h3

Types of Traffic Generated

- TCP high-throughput
- UDP variable-rate
- Ping
- Background ARP
- Multiple simultaneous flows

This produced diverse traffic for logging and ML inference.

RESULTS & OBSERVATIONS

1. High Throughput Flows Automatically Prioritized

Whenever **h2 → h1** iperf traffic ran:

- Ryu assigned **priority 100**
- Lower-latency routing
- Stable throughput ~67 Gbps (Mininet virtual limit)

2. Normal Traffic (ping/ARP) → priority 10

These were not prioritized.

3. Flow Table Confirmed ML-based Rules

Using:

sudo ovs-ofctl dump-flows s1

You could clearly see:

- High priority ML flows at the top
- Low priority flows below

4. No QoS queues were used

Meaning ML alone performed the prioritization.

WHY THIS IS DIFFERENT FROM TRADITIONAL QOS

Traditional QoS

- Uses fixed queues & manual configuration
- admin must set priority mapping
- cannot self-adapt
- cannot learn from network behavior

Our ML Approach

- Learns priority from data
- Automatically adapts to traffic
- No queues / HTB / policing
- Real-time prediction
- Controller-in-the-loop intelligence

This makes the proposed system much more flexible and modern.

CONCLUSION

The internship successfully demonstrated that:

- Machine Learning can replace manual QoS configuration
- Ryu SDN Controller can integrate ML models in real time
- Flow level statistics can predict traffic importance
- High-performance prioritization can be achieved dynamically
- A fully working SDN+ML testbed can be built using Ryu, Python, and Mininet

This establishes a foundation for future intelligent, semantic-aware networks.

FUTURE SCOPE

- Adding **semantic communication intelligence**
- Predicting traffic categories (video, voice, gaming, IoT)
- Using deep learning models
- Multi-agent reinforcement learning
- Expanding to multi-switch topologies
- Adding GUI dashboard for real-time monitoring

SKILLS GAINED

- Ryu SDN Controller
- OpenFlow 1.3 flow programming
- Mininet network simulation
- iperf traffic benchmarking
- Python networking
- Machine Learning model building
- Dataset creation and cleaning
- Research paper comprehension
- Real-world network automation

REFERENCES

[1] X. Liang, H. Ji, H. Zhang, and X. Li, "**Intelliscise Wireless Networks From Semantic Communications: A Survey, Research Issues, and Challenges**," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2654–2695, 2023.

[2] M. Shafiq, H. Li, G. Srivastava, and P. K. Atrey, "**Large Language Models (LLMs) for Wireless Networks: An Overview from the Prompt Engineering Perspective**," *IEEE Internet of Things Journal*, 2024.

[3] Y. Liu, Y. Zhou, H. Ji, and H. Zhang, "**Wireless Multi-Agent Generative AI: From Connected Intelligence to Collective Intelligence**," *IEEE Wireless Communications*, 2024.

[4] N. McKeown et al., "**OpenFlow: Enabling Innovation in Campus Networks**," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[5] Ryu SDN Framework, "**Ryu Controller Documentation**," Available: https://osrg.github.io/ryu/