National College of Ireland

An Efficient Hybrid Model For Scalable Distributed Denial-of-Service Detection in Software Defined Network

MSc Research Project
Msc CyberSecurity

Devanand Srinivasan
Student ID: 23232501

School of Computing
National College of Ireland

Supervisor: Dr Mosab Hamdan

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Devanand Srinivasan
Student ID:	23232501
Programme:	Msc CyberSecurity
Year:	2025
Module:	MSc Research Project
Supervisor:	Dr Mosab Hamdan
Submission Due Date:	11/12/2025
Project Title:	An Efficient Hybrid Model For Scalable Distributed Denial-of-Service Detection in Software Defined Network
Word Count:	7529
Page Count:	24

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Devanand Srinivasan
Date:	28th December 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

An Efficient Hybrid Model For Scalable Distributed Denial-of-Service Detection in Software Defined Network

Devanand Srinivasan
23232501

Abstract

Distributed Denial-of-Service attack detection in Software-Defined Networks is essential on a network-wide level to minimise service interruption and mitigate the number of victims. Current sketch-based detection techniques can be characterized as glued to the assumption of stable routing behaviour that may not hold under dynamic traffic conditions. This project involves a routing-oblivious sketching-based flow-based Distributed Denial-of-Service detection framework, based on a hybrid data-plane and control-plane architecture.In the proposed system, programmable switches perform lightweight data-plane monitoring using a Lemon-style sketch to aggregate per-destination traffic statistics over fixed time windows without keeping per-flow state. Periodically these sketches are combined in the control plane to get network-wide flow estimates. Flow-level features are also extracted and analysed with an adaptive Concept-Adapting Very Fast Decision Tree (algorithm) machine learning model to determine whether the traffic is benign or malicious when abnormal traffic behaviour is observed.The framework is deployed to a software switch-based Software-Defined Networks emulation environment and tested with benign and attack traffic based on the Canadian Institute for Cybersecurity dataset. Findings indicate that the hybrid flow-based detection is indeed effective in Software-Defined Networks concerning both proper detection and low false alarm rates with minimal control-plane overhead.

1 Introduction

DDoS attacks remain a significant issue to the current network structures, and they disrupt services in large scales, resulting in the loss of money and decreased accessibility to online services. As Software-Defined Networking (SDN) is progressively implemented in cloud and data-centre environments, DDoS attacks affect the SDN systems in a more severe way, with the SDN systems centralising network control and decision-making Batool et al. (2025);Janabi et al. (2024). Although SDN offers a higher level of programmability and global visibility, the same can be used by attackers to flood the control plane in high-rate flooding attacks.

There is a substantial literature that has aimed at identifying the DDoS attacks in SDN by control-plane-based monitoring and analysis. In these methods, the switches regularly send flow or port statistics to the controller, and machine learning (ML) or deep learning (DL) models are used to label the traffic as benign or maliciousAhuja et al.

(2021);Nadeem et al. (2022);Ataa et al. (2024).Whereas these approaches prove to be very accurate at detecting the object in the laboratory setting, they make significant communication overhead and computational load at the controller. On large-scale DDoS attacks, export of continuous telemetry to a centralised processor may lower scalability and make it harder to detect in timeAli et al. (2023);Golchin et al. (2023).

Recent developments in programmable data planes have facilitated the possibility of monitoring traffic on a network switch at a single provides sketch-based and probabilistic measurements. These methods permit line-rate accumulation of traffic data without per-flow condition, complimenting them to high-rate and massive networksSingh et al. (2025);Quinn et al. (2025). Specifically, Lemon shows that routing-oblivious sketch-based aggregation can make scalable, network-wide DDoS detection possible even without the use of stable routing behaviourWu et al. (2025)). Nonetheless, data-plane-only detection systems are usually based on an anomaly signalling method which is threshold-based and do not provide enough contextual information to perform more refined classification, thus misclassifying traffic when anomaly patterns fluctuate dynamically Zhou et al. (2023);Yan et al. (2024).

In order to deal with such constraints, hybrid and collaborative intrusion detection systems have been suggested that unite lightweight data-plane surveillance with more articulate control-plane surveying. CML-IDs are frameworks that use collaborative machine learning at the control plane to enhance detection accuracy through the use of shared knowledge across the networkGolchin et al. (2023). Nevertheless, these systems are still reliant on frequent control-plane processing and telemetry exchange, which can still be a bottleneck in conditions of sustained attackWang et al. (n.d.);Xia et al. (2025). These observations produce a research gap in the accomplishment of scalability and proper classification in SDN-based DDoS detection.

It is against this challenge that this research enquires the following question:

"Is it possible to bring together scalability of sketch-based monitoring and accuracy of machine learning-based classification with control-plane overhead constraints in SDN with a hybrid data-plane and control-plane DDoS detection method?"

The study objectives of this research are to answer this question by:

(i) apply sketch based, lightweight data monitoring of traffic over SDN data plane,

(ii) use selective machine learning-based classification at the control plane to make refined detection,

(iii) compare the performance of detection in terms of accuracy, precision, recall, F1-score, and confusion-matrix analysis and

(iv) benchmark the hybrid solution with an already developed base system.

This project introduces a hybrid model of data-plane and control-plane (DP-CP) of DDoS detection, which incorporates the use of Lemon-style sketch-based monitoring to realise the early signalling of anomalies, and implements a Concept-Adapting Very Fast Decision Tree (CVFDT) classifier inotroduced in the control plane using CML-IDS style of collaborative learning. This work has contributed to users in the sense that it actually implements and empirically tests this hybrid detection pipeline in a programmable SDN setting, and also shows, through epoch-based analysis, how selective control-plane escalation can enhance the reliability of detection and still be scalable.

The rest of the report is organized in the following way. Section 2 provides a review of related literature in SDN-based DDoS detection, programmable data planes, and hybrid intrusion detection systems. Section 3 outlines the methodology of the research and the design of the experiment. The system design and architecture are presented in section 4. Section 5 reports on the implementation and artefacts generated. Section 6 assesses the suggested framework and makes a comparison between it and baseline approaches. Finally, Section 7 brings the report to an end, and gives future work directions.

2 Related Work

2.1 Surveys and background

According to the recent surveys, SDN architectures focus visibility and decision-making at the controller, making it easier to detect, but also susceptible to overloading in case of a high-rate DDoS Batool et al. (2025). AI-centric survey work also further documents the transition from rule-based detection to ML/DL pipelines, alongside recurrent dataset and deployment gaps Apostu et al. (2025)). Systematisation studies on the programmable networking perspective indicate that P4 switches can support the in-network security capabilities (such as DDoS detection) through user-defined line-rate processing, but are limited in memory, pipeline stages, and model complexity Singh et al. (2025). Data-plane reviews also point out the fact that SDN security research has traditionally paid excessive attention to the control plane, and that data-plane security mechanisms are heterogeneous and disjointed Quinn et al. (2025). A systematic review of the research on ML/DL SDN DDoS detection finds that it is highly accurate in laboratory settings, though it has been found to have weaknesses in generalisation, reproducibility, and operational overheadsAli et al. (2023). To a larger degree, SDN IDS survey work contains an overview of typical SDN threat vectors, sources of features, and deployment, which solidifies the idea that real-world IDS should be scalable and accurate at the same time Janabi et al. (2024).

2.2 ML/DL SDN DDoS detection in control-plane.

The most common solution is to send flow/port statistics to the controller and categorize traffic based on ML/DL. As an example, Ahuja et al. (2021) suggest ML-based detection with engineered features and train them on SDN environments, where strong classification is demonstrated, but controller-side features processing and inference are necessary . The designs of deep learning IDS (e.g., CNN-LSTM or Transformer-style) are highly performing, yet, they raise the computational expenses and are highly reliant on the selection of features and training protocol Ataa et al. (2024). Equally,Nadeem et al. (2022) use ML pipelines and feature selection in detecting DDoS within SDN, where the results are also very predictive but periodically necessitate data exporting and control-plane processing.

This concept is expanded by collaborative control-plane learning through sharing models/knowledge with controllers. CML-IDS specifically addresses SDN IDS shortcomings by allowing collaborative machine learning for accurate detection effectiveness across the networkGolchin et al. (2023). Nevertheless, this type of system continues to rely on (i) long-term telemetry to the controller(s) and (ii) inference beyond the data plane both of which may act as bottlenecks during DDoS load.

2.3 Monitoring and sketching of programmable data planes

Programmable switches allow the ability to line-rate telemetry and early anomaly signalling without maintaining full per-flow state. Lemon suggests a routing-oblivious measurement and detection scheme with a special sketch, where network-wide DDoS can be detected scalably and with no routing assumptions Wu et al. (2025). Related work on intelligent data planes incorporates more logic within the switch pipeline: NetBeacon: In-network inference with multi-phase flow analysis for improved deployability and reduced controller-dependence Zhou et al. (2023). Brain-on-Switch (BoS) takes the next step on the direction of neural traffic analysis of data planes within the network at line-speed, which shows the feasibility and limits of NN-driven analysis at hardware-level constraints Yan et al. (2024).

Another line is the feature extraction and preprocessing in the data plane. P4DDLe provides programmable data plane packet-level analysis to use on intrusion detection workflow offloading some of the feature pipeline to the controllerDoriguzzi-Corin et al. (2023). The same hardware-assisted ML IDS that HALIDS discusses enables devices to make decisions or make decisions without involving the controller, trying to minimise the use of the controller, but retaining monitoring performance Brandino et al. (2024).

Victim identification and volumetric detection are also done using sketch designs. The BACON takes advantage of sketch-based in-network volumetric DDoS victim identification methods, enabling effective identifying the victims with limited resources**?**. Notably, more recent research also points to attacks on the detection structures themselves: INDDoS+ investigates weaknesses in DDoS detection data structures of programmable switches and offers measures to make them more resilient when using a limited hardware resource .

2.4 Hybrid, multi-stage and collaborative IDS.

Hybrid systems Hybrid systems use a combination of fast data-plane signalling with more in-depth control-plane analysis to lower the overall overhead. An obvious example would be to apply lightweight port/flow statistics to do coarse detection and additional expensive classification to only suspicious periods.Wang et al. (n.d.) explicitly takes a two stage SDN approach with coarse granularity detection (from switch statistics) and a deeper classifier stage, with the goal of reducing bandwidth and compute overhead in comparison with fully centralised approaches .Manso et al. (2021) introduce an SDN-based IDS to identify an attack and introduce mitigation measures triggered by the SDN controller, with the focus on timely identification and the mitigation measures based on operational constraints.

There are also collaborative and network-wide defence on programmable switches. CoDDoS is a hybrid of network-wide and device-local monitoring and relies on programmable switches and sketches to protect against various DDoS patterns, as well as combinations of volumetric and low-rate attacksXia et al. (2025).

Lastly, there are recent works on the subject of LLM-assisted classification of SDN. Swileh and Zhang (2025) have proposed a decentralised SDN framework in which lightweight port-level monitoring is coupled with "zero-training" LLM inference for classification and mitigation decisions. They also suggest converting flow data to a language-model-interpretable representation with BERT to detect unknown attacks, which in assessed environments are reported to have very high accuracy. These approaches are

promising in terms of adaptability, but raise practical questions in terms of inference cost, reproducibility and operational reliability under actual DDoS pressure.

2.5 Overview of the gaps and the motivation to do this work

Throughout the literature reviewed, there are three gaps that are consistent.

1. Control-plane ML/DL IDS offers good detection capabilities but poses the risk of even greater controller overhead and slower mitigation during large-scale floods Ahuja et al. (2021); Ataa et al. (2024);Golchin et al. (2023).

2. Programmable data plane approaches scale well and can be fast to react, but are limited by the resources of the switch and can be based on simplified logic, unless combined with further learning stagesWu et al. (2025);Zhou et al. (2023) ;Yan et al. (2024).

3. Hybrid/collaborative systems limit unnecessary control plane processing through selective escalation, but must still be carefully designed to avoid instability (through threshold tuning), guarantee robustness of measurement structures and adapt to evolving attack patternsWang et al. (n.d.);Xia et al. (2025);Ding et al. (2024).

These results inspire a multi-stage of SDN DDoS detection pipeline that integrates the routing-oblivious sketch-based signalling at the data plane (Lemon-style)Wu et al. (2025), with selective escalation and learning-based classification at the control planeGolchin et al. (2023), explicitly handling both the overhead and robustness issues of the prior art.

3 Methodology

This research uses an experimental approach to design, implement and evaluate a collaborative Distributed Denial-of-Service (DDoS) detection framework for Software-Defined Networking (SDN). The methodological design is based entirely on two basic works, Collaborative Machine Learning-based Intrusion Detection System (CML-IDS) and Lemon routing-oblivious network-wide DDoS detection system. The collective motivation of these works is the hybrid approach combining programmable data plane monitoring and control plane intelligenceGolchin et al. (2023);Wu et al. (2025).

The purpose of this methodology is to test the feasibility of using Lemon sketch-based monitoring in the data plane, and selective control plane machine learning as seen in CML-IDS, to achieve effective DDoS detection on the data plane and low control plane overhead. controlled experimental setting is used, in which the network traffic is repeated in a programmable SDN environment and the detection performance is measured during repeated epochs. The implementation of the system is done with P4-programmable BMv2 software switches, used for data-plane processing, and a Python-based SDN controller, responsible for the aggregation and analysis in the control plane.

The network topology is made of several P4 switches, connected via a divider switch, allowing traffic to be seen from several ingress points. This design is based on Lemon's routing-oblivious assumption where the path of packets is dynamic and unpredictable and the detection cannot rely on the previous knowledge of routing behavior Wu et al. (2025).

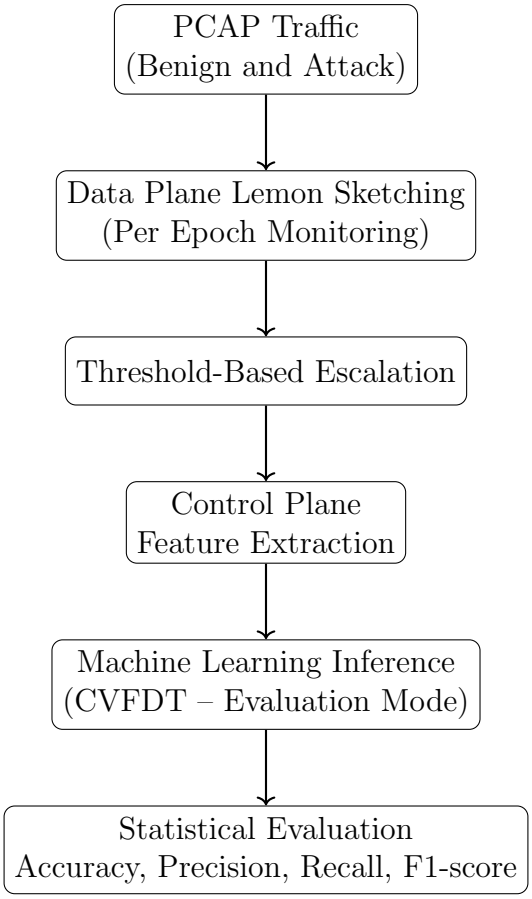

Figure 1: workflow diagram

Figure 1 shows an overview of the proposed workflow.In this workflow, PCAP traffic can be initially observed in the data plane in Lemon sketches on an epoch basis. Traffic that passes a threshold only is sent to the control plane and filtered to extract features and classify with the CVFDT model. Lastly, standard statistical metrics are used to assess the performance of detection, which include accuracy, precision, recall, and F1-score.

3.1 Data Plane Monitoring Using Lemon Sketch

At the data plane, packet monitoring is done purely in the P4 program with a multi-layer Lemon sketchWu et al. (2025). Incoming packets update bitmap-based sketch layers as well as a counter based on a hashed flow identifier. The sketch aggregates traffic statistics on a per destination IP basis, without storing per flow state, ensuring memory efficiency and line rate operation.

This design is a direct follow-up of the sketch construction by Lemon, which avoids the problem of packet over-counting and misallocation in the processing stage under dynamic routing conditionsWu et al. (2025). Detection at the data plane is done with a threshold-based mechanism, where destinations with a higher than predefined number of packets are flagged as a suspicious destination.

Only destinations exceeding this threshold are escalated to control plane and thus reduce unnecessary controller interaction.

When an alert is generated in the data plane, the control plane fetches the aggregated sketch data from all the switches and reconstructs a network-wide estimate of the traffic

sent to the suspicious destination. Additional features at a flow level such as estimated flow volume, changes over time and entropy based features are extracted for the further analysis.

A Concept-adapting Very Fast Decision Tree (CVFDT) model is used at the control plane, which is in line with the collaborative learning strategy proposed in CML-IDS Golchin et al. (2023). The model is trained offline and learned online with CIC-style traffic features and at run-time loaded in the controller.

the system supports online learning, online model updates are disabled when performing an evaluation to be sure that results are detecting capability rather than adaptive behaviour. This is similar to the evaluation strategy adopted in CML-IDS in that the training and test phases are not confused.

3.2 Control Plane Processing and Collaborative Learning

When an alert in the data plane occurs, the control plane obtains aggregated sketch data from all of the switches and reconstructs an estimate of the network-wide traffic headed towards the suspicious destination. Additional flow level features such as estimated traffic volume, temporal changes and entropy based features are extracted for further analysis.

A Concept-adapting Very Fast Decision Tree (CVFDT)Hulten et al. (2001) model is used at the control plane with. The classifier is first built offline with the baseline log from Control plane traffic features and installed inside the SDN controller.

In this work, online learning is also applied, that is, CVFDT model is able to update itself incrementally with newly observed traffic instances at runtime. This enables the adaptation of the detection system to changing traffic characteristics and the possibility of concept drift, which is one of the main reasons for using collaborative machine learning-based intrusion detection. During the experimental phase some updates of the online learning were selectively disabled when it was needed to ensure controlled and repeatable learning evaluation; the full online learning mechanism is implemented and supported by the system.

3.3 Evaluation & Metrics

Detection is assessed using epoch-based labelling strategy. Each epoch is labeled as attack or benign depending on whether or not traffic targeting the predefined victim IP is seen over a given time window. This way, even in epochs without alerts, it is possible to identify true positives, false positives, true negatives, and false negatives accurately.Evaluation Performed End-to-end , making it possible to directly compare it with two-stage detection pipeline in CML-IDS Golchin et al. (2023).

Detection performance is assessed by using conventional statistical measures. Accuracy, Precision, Recall and F1-score. Confusion matrices are generated at the epoch level in order to balance the evaluation on benign and attack periods.In addition to the accuracy of detection, control plane overhead is measured by the number of data plane to control plane escalation events per epoch. This metric is a direct manifestation of Lemon's design goal of minimizing the control plane's interaction while maintaining the accuracy of detecting these events Wu et al. (2025).

3.4 Compliance of Methodology with Base Line Systems

This methodology is explicit by combining strengths of the two base systems. While lemon has been scalable, routing-oblivious data-plane measurement, but refinement by learning, CML-IDSGolchin et al. (2023) can be described then with collaborative machine learning with higher involvement of the control plane. By combining the Lemon sketching techniqueWu et al. (2025) and imeplementing CVFDTHulten et al. (2001) model on the control plane we can reduce Controller Overhead, the introduced methodology assesses a balanced hybrid approach.

All experiments are reproducible; source code is available from a public GitHub repository [1]. Large packet capture datasets are placed on the outside and linked by the repository documentation.

4 Design Specification

The techniques and/or architecture and/or framework that underlie the implementation and the associated requirements are identified and presented in this section. If a new algorithm or model is proposed, a word based description of the algorithm/model functionality should be included.

4.1 System Architecture overview

This work presents a hybrid flow-based software-defined networking DDoS detection framework which incorporates both data-plane technique of sketch-based monitoring and control-plane technique of machine learning-based classification. The design is inspired by the routing-oblivious measurement principles proposed in Lemon and collaborative learning architecture in CML-IDS, and basically extend these with the flow-level feature integration and adaptive learning capability.

There is a two-stage detection pipeline in the framework:

1. **Data Plane (DP):** So programmable switches do light weight, line rate monitoring using a multi-layer, skip-fast-core, skip-layer, skip-layer Lemon-style sketch for abnormal traffic volume and characteristic identification without keeping the state of each flow.

2. **Control Plane (CP):** Aggregated output of the sketches and features extracted from the flow are analysed via an adaptive CVFDT (Hoeffding Adaptive Tree) model to classify as benign or malicious flow tr affic.

This separation guarantees early filtering of benign traffic at the data plane, and the computationally expensive learning is reserved so that the control plane is not too much burdened, which results in improved scalability.

The System is implemented on an SDN emulation environment with two leaf switches and one divider switch which is coordinated by a centralized SDN controller.

- **Leaf switches** implement the Lemon-style sketch logic to the data plane.

- **the controller** combines sketch states and conducts flow level analysis.

[1] GitHub repository: `https://github.com/Devanand007/hybrid-sdn-ddos`. Kaggle dataset: `https://www.kaggle.com/datasets/devanandsrinivasan/mwai-5k`

Traffic generators replay benign and, attack PCAP traces taken from the CIC datasetSharafaldin et al. (2018) and Slow rate DDos DatasetYungaicela-Naula et al. (2023). figure 2 shows overview of system Architecture

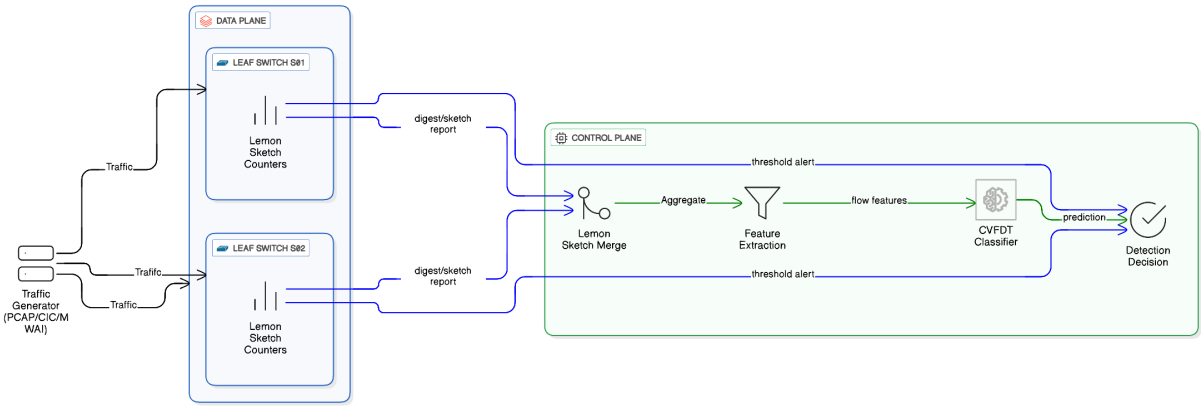

Figure 2: System architecture of the proposed hybrid DP–CP DDoS detection framework

figure 2 system is based on the hybrid data-plane and control-plane (DP-CP) platform of scalable DDoS identification. The data plane handles incoming traffic, with programmable leaf switches keeping Lemon sketch counters in order to conduct lightweight but line-rate monitoring without persisting any per-flow state. Digest/sketch reports are created when traffic surpasses preset thresholds or have a chronic abnormal operation and sent to the control plane.

The control plane combines and consolidates sketch information across a set of switches and allows viewing network behavior on a global scale. A feature extraction is then being carried out in order to obtain time and distribution-based features, which are subjected to a CVFDTHulten et al. (2001) classifier to be analyzed further. The end decision on detection is generated out of the classifier output, which provides accurate attacks but limits the processing of control plane to suspicious traffic only.

4.2 Data Plane Design

4.2.1 Lemon Sketch-Based Monitoring

Every programmable switch holds a multi-layer bitmap drawing, each layer of which is associated with a unique sampling granularity. CRC-based functions are used to hash incoming packets into mapping positions in the layers as a bitmap. Let :

- $\boldsymbol{m_l}$ consider as bitmap size of layer l.
- $\boldsymbol{V_l}$ consider as number of zero bit in layer l

Then Cardinality Estimation for each layer computed using logarithmic estimatar:

$$\hat{C}_l = -m_l \cdot \ln\left(\frac{V_l}{m_l}\right) \tag{1}$$

The last flow estimate is chosen according to saturation based on bitmaps, which is in line with the routing-oblivious estimation strategy of Lemon.

This design:

- Avoids per-flow state
- Checks against the possibility of counting twice the switching.
- Stands to survive the dynamic routing changes.

4.2.2 Heavy Hitter Identification

The data plane uses a filtering mechanism based on threshold to decrease the amount of data to be transmitted to the control plane:

$$C_f \geq \theta_{\text{DP}} \;\Rightarrow\; \text{Flow is escalated to CP} \tag{2}$$

Where:

- C_f is considered as estimated packet count for flow f.
- θ_{DP} considered as predefined Threshold for DP

Any flows above this threshold is reported to the control plane and this way, this decreases the digest traffic largely.

4.3 Control Plane Design

4.3.1 Feature Extraction

In case of an escalated flow, the controller obtains a concise collection of features based on sketch approximations and time behaviour:

- Estimated packet count
- The variation in approximated traffic volume (VF)Δ
- Traffic dispersion entropy.
- Lemon sketch estimate
- Exponentially smoothed packet counts variation (EMA).
- Capturing short term bursts of traffic.

The choice of these features is influenced by volumetric anomaly, burstiness, and skewness of distribution, which have been identified to be signs of DDoS attacks.

4.3.2 CVFDT Classification

The classification of traffic is done with a CVFDT (Hoeffding Adaptive Tree) model which is selected due to its capability to:

- Streaming data Incremental learning.
- Adapt to concept drift
- Operate with bounded memory

- The classifier gives a binary decision:

$$y \in \{0, 1\}, \text{ where } y = 0 \text{ is Benign and } y = 1 \text{ is Attack} \tag{3}$$

The system facilitates the online updating of models, which enables the classifier to change with changing attack patterns. Online learning is however, purposefully inhibited during evaluation in order to create a fair, reproducible and statistically valid experimental outcome. To enhance the ability to detect slow-rate and persistent DDoS attacks, the set of features was expanded with the temporal and persistence-sensitive features based on sketch counters. These are normalized traffic variation, burst ratios based on exponential moving averages (EMA) and persistence measures indicating sustained low-rate action in succeeding epochs. The classifier can use such features to differentiate between gradual patterns of attacks and temporary benign bursts.

Despite the system allowing online learning, the adaptive model updates are not allowed at the time of evaluation to guarantee fairness, reproducibility, and statistical validity. This will ensure that the performance reported is due to the efficiency of the proposed feature design and hybrid DP-CP detection logic and not model retraining effects.

Algorithm 1 epoch based DDos Detection in DP and CP

Require: Packet stream data eg PCAP P, threshold need tobe set τ_{DP}, epoch length can be fixed T

Ensure: Epoch level detection

Initialization

First step Initialing multi-layer Lemon bitmap sketches at each programmable switch

Load CVFDT Model at control plane

1: **for** each incoming packet $p \in P$ **do**
2: Compute the hash values from packet header fields
3: Update bitmap positions in lemon sketch layer
4: Increment counter
5: **end for**

End of Epoch T_s

6: **for** each destination marked as heavy hitter **do**
7: Estimate traffic volume $\hat{C}$
8: **if** $\hat{C} \geq \tau_{DP}$ **then**
9: sent this statistics to CP
10: **end if**
11: **end for**

CP Processing

Aggregate Lemon sketches from all switches

Extract features from dataplane side

Apply CVFDT classifier to classify the traffic

Generate CP decision

12: **if** online learning mode is enabled **then**
13: Update the CVFDT model
14: **end if**

Output

Record detection outcome store in CSV

In the algorithm 1, a hybrid flow-based DDoS detection strategy is introduced that runs at both the data plane and control plane. Data packet processing of incoming

packets in the data plane is performed at the line rate at multi-layer Lemon bitmap sketches where hash values of packet header values are used to modify sketch entries and counters. Traffic is accumulated within a fixed epoch and destinations that are identified to be heavy hitters are rated within an actual traffic volume that is approximated. The control plane is only escalated to destinations which are above the data-plane threshold.

At the control plane, the extracted flow-level features at the aggregated sketch information are categorized using a CVFDT model to identify whether the traffic is benign or malicious. With the feature turned on, online learning enables the model to fit the changing traffic pattern. Evaluation is done at the epoch level with detection results being stored.

The proposed design consists of combining routing-oblivious measurement of Lemon and CML-style collaborative intelligence to provide correct, scalable, and flow-aware DDoS detection that is applicable in contemporary SDN setting.

4.4 Network Topology

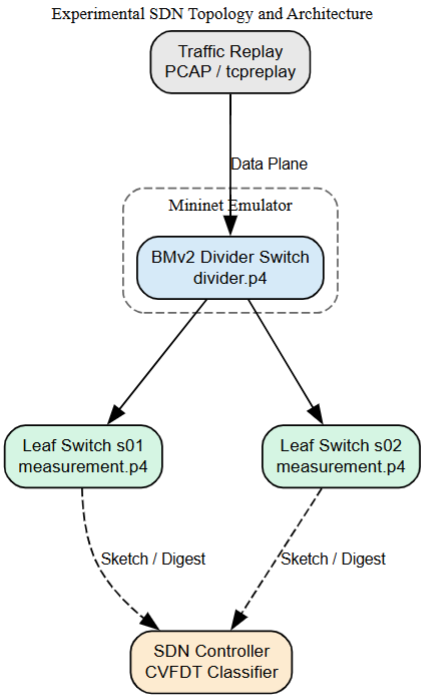

Figure 3: Network Topology

From Figure 3 Experimental setup involves an SDN topology modeled in Mininet whereby the traffic is replayed on PCAP files with the help of tcprelay into a BMv2 divider switch. Leaf switches executing Lemon sketch based measurementWu et al. (2025) in the data plane receive traffic allocated to them. The suspicious traffic summaries (sketches/digests) are sent selectively to the SDN controller where a CVFDT classifier is used to control-plane refine final DDoS detection.

5 Implementation

The section offers the completed implementation of the suggested collaborative intrusion detection framework, and it will be centered on the specific artefacts generated and the technologies available to implement the artefacts. The section contains a description of what was implemented and generated in accordance with the project specification instead of listing source code or giving instructions on how to operate it.

5.1 Implementation Environment/Platform:

The suggested solution will be implemented on a single computer workstation running OS Ubuntu Linux 22.0.04, an open-source operating system.

The entire system was developed and deployed in a Software-Defined Networking environment on the Amazon Web Services (AWS) as a cloud. A EC2 m7i.large instance was chosen to have an adequate computing power to support programmable data-plane and control-plane analytics and machine learning inference. The environment was configured to provide SDN emulation, traffic replay, and offline data analysis and it was running a Linux-based operating system.

Such a deployment option represents realistic cloud-based SDN cases, in which the control plane and the data plane are deployed on common infrastructure and have to work within resource constraints. To achieve reproducibility and controlled experimentation, all the necessary dependencies of programmable networking, packet processing, and data analysis were installed in the same virtual machine.

5.2 Technologies, Tools and Languages

The implementation incorporates the use of several technologies in the data plane and control plane. Main tools and languages used are summed up as given below:

- An implementation of programmable data-plane monitoring logic on BMv2 software switches was done with P4 16.
- The SDN control plane, sketch aggregation, feature extraction and machine learning inference were implemented in Python.
- P4 Based packet processing was supported by BMv2, which was a programmable software switch.
- P4Utils was topology management and access to registers through Thrift APIs.
- The emulation of a multi-switch SDN topology was done using Mininet.
- The packet capture files were replayed with Scapy and tcpreplay and traffic generated.
- Numrical processing and statistical computation was supported by NumPy and by standard Python libraries.
- Intermediate and final outputs were persisted in the form of CSV to be evaluated.

This toolchain allowed a modular and extensible implementation and still compatible with the current SDN research frameworks.

5.3 Implementation Outputs of DP and CP

Data plane This layer introduces a lightweight traffic monitoring mechanism which uses a Lemon-style sketching mechanism. Instead of doing explicit classification of attacks, the data plane generates small traffic summaries that can be used to do scalable network-wide monitoring.

The main products of the data plane would be:

- Multi-layer sketch bitmaps of probabilistic summative per-destination traffic activity.
- Traffic counters that do not use per-flow state to counter packet volumes, but count the packets in a count based manner.
- Having heavy-hitter identifiers of the destination addresses with traffic surpassing specific limits.
- Epoch aligned register values, which allow time-based aggregation and comparison between monitoring intervals.

Such outputs are held in programmable switch registers and are periodically accessed by the control plane. Notably, raw packets and detailed flow records are never exported and this guarantees low memory overhead and constant-time processing per packet. The control plane is an entity that converts the data-plane summaries into analytical artefacts. With help of Python-based control logic, the following outputs are achieved:

- Merged network-wide sketches, this is created by combining the data of many switches (bitmap images).
- Estimates of traffic volume, calculated by means of a set of the Lemon methodology based on the estimation techniques based on a bitmap.
- Derived traffic attributes, such as estimated magnitude of traffic, temporal variation of estimates, and entropy based indicators.
- Binary decision detection, that is, whether observed traffic behaviour is benign or malicious.

The extracted features are used to classify the traffic behaviour using a trained Concept-adapting Very Fast Decision Tree (CVFDT)Hulten et al. (2001) model. Even though the framework allows updating the models online.

5.4 Generated Artefacts

The implementation facilitates supporting variety of traffic sources in order to test the behaviour of detection in varying conditions. These include:CIC dataset packet captures, which are labelled benign and attack traffic.And slow Rate DDos attackDataset .Artificial attack traffic, created through custom scripts in order to recreate high-volume flooding conditions.Control replay files Replayed packet capture files, injected into the topology being emulated with controlled replay rates.

The system yields a number of artefacts that are persistent to the following sources of traffic important logs

- cvfdt metric CSV files with feature vectors which are the inputs of the machine learning model.
- EPOCH level summary files, where the detection decisions and escalation events were written.
- Traffic estimate logs stores as example lemon only log file, which record both raw and probabilistic estimates.

These artefacts allow the inspection of offline and reproducible assessment without necessitating replaying of the traffic.The major input to the evaluation process is the CSV data generated in the implementation phase. According to these recorded outputs, standard statistical measurements; accuracy, precision, recall, and F1-score, are calculated in the evaluation section. The design guarantees that all quantitative performance analysis is done independently and repeatable whereas the implementing stage only generates artefacts.

6 Result and Discussion

The tests are performed on three traffic conditions, namely Slow Rate DDos Oneattacker and one victim pcap file **?** repetition with 120 seconds, synthetic SYN flood traffic repetition with 120 seconds which are Synthetic pcap generator used for DDos and included in code artifact, and CIC-IDS-2017Sharafaldin et al. (2018) Friday dataset, which includes 60 seconds of benign traffic, followed by 60 seconds of attack traffic. The experiments in the case of each dataset are conducted independently in the data-plane (DP only), control-plane (CP only), and DP-CP configuration.In Result We Calculate metric Per epoch basis.

An epoch is a fixed period of time where the network traffic is aggregated and the network traffic is analysed as a single observation. Lemon sketch-based measurements are summary measurements of traffic behaviour at the data plane, and do not have any state per packet or flow. Once the aggregated statistics in an epoch reach a set threshold, the associated traffic is raised to the control plane to be analysed further.

The analysis has the same epoch-based control-plane reasoning as Lemon by Wu et al. (2025), in which each epoch is regarded as a unitary classification example and labeled as benign or attack according to the prevailing traffic behaviour. The performance metrics of detection, such as the accuracy, precision, recall, recall, F1-score, and the confusion matrix are thus calculated on an epoch-by-epoch basis. This method demonstrates the system capability to appropriately detect long attack durations instead of individual packets or flows in line with real-time DDoS detection needs where endurance to momentary traffic instabilities is necessary.

6.1 Dataset and enviroment

In order to test the proposed detection framework,benchmark attack datasets is used to simulate real-life network conditions. slow DDos attack which maybe look like Benign like behaviour using SDN-SlowRate-DDoS dataset trafficYungaicela-Naula et al. (2023), and attack scenarios are modelled using synthetic SYN flood traffic and CIC-IDS-2017Sharafaldin et al. (2018) Friday working hours DDoS traces. Each dataset is played through PCAP format and over the test topology.

SDN-SlowRate-DDoS dataset traffiMawi (2025) replicated over a period of 120 seconds.The test_2.pcap SYN flood traffic is created and replayed over a duration of 120 seconds to confirm the system based on voluntary volumetric attack conditions. Also, the CIC-IDS-2017Sharafaldin et al. (2018) Friday working hours data are transformed to PCAP format using a custom Python script and divided into 60 seconds of benign traffic (benign_from_flow) and 60 seconds of attack traffic (attack-from-flow) to allow making attack injection period clear.

The epoch length of 1 second is used to process the traffic with the detection decisions being generated at the epoch level. Benign traffic is replayed with a rate of 500 packets per second (pps) whereas the attack traffic is replayed with a rate of 1100 pps so that there is observable difference in volumetric characteristics. All experiments are subject to a specific data-plane threshold of 900 in order to ensure a consistent detection behaviour.

Each dataset is run separately under data-plane only (DP), control-plane only (CP) and hybrid DP -CP. All experiments will be run on an AWS EC2 m7i-flex.large instance (x86-64 architecture, 8 vCPUs, to a maximum network bandwidth of 12.5 Gbps) with 100 GB of storage, which has enough computational and network bandwidth to support real-time replay and analysis of packets.

6.1.1 SDN-SlowRate-DDoS Dataset Evaluation

The SDN-SlowRate-DDoS Yungaicela-Naula et al. (2023) is utilized to test the behaviour of the aimed detection framework in the conditions of realistic slow-rate attacks. The dataset obtains the labeled traffic patterns which describe the stealthy DDoS cases in Software-Defined Networking systems, and this allows a supervised evaluation of detection performance. This makes the proposed system analyseable in terms of its capacity to recognise malicious slow-rate flows, and accept legitimate background traffic under stable operation in the normal network conditions.

Table 1: SDN Slow Rate DDos Performance

Plane	**Precision**	**Recall**	**F1-score**	**Accuracy**
Data Plane (DP)	1.00	97.5	98.7	97.7
Control Plane (CP)	1.00	94.0	96.9	94.0

Table 2: SDN Slow-Rate DDos Confudion Matric

Plane	**TP**	**TN**	**FP**	**FN**
Data Plane (DP)	117	10	0	3
Control Plane (CP)	110	0	0	7

From Table 1 & 2The combined confusion matrix shows that the data plane (DP) correctly detects 117 slow-rate attack epochs while missing only 3, achieving a recall of 97.5% with zero false positives. This indicates that the persistence-based escalation mechanism is effective in identifying low-intensity attacks while maintaining stability under benign traffic conditions. The DP accuracy reaches 97.7%, confirming reliable global detection performance.

Out of 130 total epochs, the data plane escalates 117 attack epochs to the control plane, corresponding to approximately 90% of the observed attack activity, while filtering

benign traffic locally. This selective escalation significantly reduces control-plane workload. the data plane finds the majority of attack epochs (117 TP) correctly without any false positives which can be trusted to escalate to the control plane. This initial filtering minimizes the unwarranted processing of the control-plane, and the control plane narrows the ultimate ruling on the escalated traffic. Consequently, detection is also effective and concentrated on sustained attack behaviour.

At the control plane (CP), 110 of the 117 escalated attack epochs are correctly classified, resulting in a recall of 94.0% and perfect precision (1.000). No additional false positives are introduced by the machine-learning classifier, demonstrating its effectiveness in refining data-plane alerts. The remaining misclassifications reflect the inherent difficulty of distinguishing stealthy slow-rate attacks using lightweight features.

Overall, the results confirm that the hybrid DP–CP framework achieves high recall at the data plane and high precision at the control plane, enabling accurate slow-rate DDoS detection with limited escalation overhead.

6.1.2 Synthetic Dataset Evaluation

In order to test the detection ability when subjected to controlled attack conditions, a synthetic UDP flood data is created with a custom traffic generation script based on Scapy. The resultant traffic types to represent the attack behaviour of distributed attacks are: high-volume attack flows are generated against a fixed victim destination IP (192.168.1050), different source IP addresses and transport-layer ports.

Every synthetic flow is composed of various UDP packets that have randomly chosen payloads, source IP addresses, and source ports. There is an explicit control of the number of packets per flow and this gives an opportunity to carefully regulate the intensity of the attack. The number of packets in the total volume to be transmitted in the conducted experiments is fixed to 144,000 packets which is to be allocated to a configurable number of flows to make the attack scenarios repeatable and scalable.

the proposed detection framework is assessed under the condition of fully-controlled and repeatable conditions by the help of synthetically generated DDoS traffic. Artificial traffic allows the control of parameters of the attack, including packet rate, persistence period and victim selection exactly, which are challenging to factor out in real-life data sets. This enables systematic confirmation of the persistence-based escalation reasoning in the information plane and the temporal feature study in the regulation plane.

Table 3: Synthetic Dataset Performance

Plane	**Precision**	**Recall**	**F1-score**	**Accuracy**
Data Plane (DP)	1.00	97.4	98.7	97.6
Control Plane (CP)	1.00	1.00	1.00	100

Table 4: Synthetic Confusion Matrice

Plane	**TP**	**TN**	**FP**	**FN**
Data Plane (DP)	111	10	0	3
Control Plane (CP)	111	0	0	0

From Table 3 and 4 For the synthetically introduced DDoS traffic, the data plane recalls 97.4% of the attacks, and the 111 out of 114 attack epochs are successfully escalated

with 0 false positives and the overall accuracy of 97.6. The control plane labels 100 percent of the escalated attack epochs and this has perfect precision and recall of forwarded traffic. These findings show that the data plane is efficient in terms of balancing high recall and selective escalation and the control plane ensures correct refinement without creating extra false alarms. the data plane is able to identify nearly all the attack epochs (111 TP) with a false positive of zero with only a few attacks left out (3 FN). The control plane then appropriately categorizes all the escalated epochs with zero false positives and zero false negatives. Herein, the effectiveness of DP filtering in the context of the CP making fully accurate decision on this dataset is indicated.

6.2 CICID2017 Dataset Evaluation

To test the suggested framework in a benchmark situation of intrusion detection using marked benign and attack traffic, the CIC-IDS-2017Sharafaldin et al. (2018) Friday working hours dataset is applied in a framework test scenario. CIC-IDS-2017, in contrast to SDN Slow-rate Ddos DatasetYungaicela-Naula et al. (2023) and the synthetic dataset, gives a clear separation of normal and malicious traffic, which allows comprehensively determine the accuracy of the detection.

To perform this assessment, the dataset is transforming into PCAP format with a self-written Python script and is divided into roughly 60 seconds of benign traffic (benign_from_flow) and roughly 60 seconds of DDoS attack traffic (attack_from_flow). traffic is replayed at an epoch length of 1s, benign traffic of 500 packets per second, attack traffic of 1100 packets per second, and a fixed data-plane threshold of 900.

Table 5: CIC Dataset Performance

Plane	**Precision**	**Recall**	**F1-score**	**Accuracy**
Data Plane (DP)	91.5	1.00	95.6	96
Control Plane (CP)	93.1	1.00	96.4	93.2

Table 6: CIC Confusion Matrix

Plane	**TP**	**TN**	**FP**	**FN**
Data Plane (DP)	54	65	5	0
Control Plane (CP)	54	1	4	0

Based on Tables 5and 6 (CIC Dataset Performance and CIC Confusion matrix) both data plane (DP) and control plane (CP) have a recall of 1.00 which implies that there were no missed DDoS attacks (FN = 0). At the DP, 54 attack examples were positively identified, and 5 false positives were reported, leading to precision and accuracy of 91.5 and 96 percent respectively. These false positives can be attributed to the sketch-based, threshold-based monitoring, which is highly prejudiced towards early detection of anomalies. The suspicious traffic detected within the DP was only escalated to the CP, (59 alerts, 54 TP and 5 FP). The machine-learning classifier at the CP maintained all true positives and minimized false alarms, raising the precision to 93.1% and F1-score to 96.4%. In the case of this dataset, the data plane catches all the attack epochs with no false negatives, yet it has a limited number of false positives (5), which makes sure that one is not losing attacks when filtering at the very beginning. Control plane, once these decisions are refined by the control plane, uncertainty is minimized and the number of

false negatives is zero, hence resulting into the final classification being more accurate. This demonstrates that the DP takes more priority in recall as compared to the CP which enhances the overall decisions in this dataset.

6.3 Final Result and discussion

Table 7: Dataset and Traffic Summary

Dataset	**Total Duration**	**Total Packets**
CIC-IDS-2017 IDS	120.87	95642
SDN Slow-rate Ddos	125.65	138225
Custom-generated DDoS	119.60	131562

Table 7 Overall Traffic inject for each dataset

Table 8: End to End Detection performance

Dataset	**Precision**	**Recal**	**F1-score**	**Accuracy**
CIC-IDS-2017	93.1	100	96.4	93.2
SDN Slow-Rate-DDos	100	94.0	96.9	94.0
Synthetic DDoS	100	100	100	1

- **Synthetic DDoS**:From Table we can see 8The synthetic dataset attains an ideal precision, recall, and F1-score (100%), since the traffic is completely under control and there are distinct patterns of attack. This validates the correctness of the detection logic and the usefulness of CP refinement in an ideal ground-truth environment.
- **CIC-IDS-2017**:In the case of CIC-IDS-2017, the system has a perfect recall (100%) meaning that there is no attack that is overlooked, but the accuracy is high (93.1). The resultant F1-score of 96.4 percent demonstrates that there is a high sensitivity and false-positive rate cut-off on realistic benchmark traffic.
- **SDN Slow-Rate DDoS**: Precision is 100% in the SDN slow-rate dataset, indicating that all the attacks detected are accurate, whereas recall is slightly lower at 94.0% which shows that there are few low-rate attacks missed. Nevertheless, the F1-score of 96.9 remains a good result, which proves that it is resistant to stealthy slow-rate behaviors.

Table 9: Dp Escalation Statistics

Dataset	**TotalEpoches**	**DP-Flagged**	**Escalated to CP**	**Escalation Rate (%)**
CIC-IDS-2017	124	59	59	47.6
SDN Slow-Rate DDoS	130	117	117	90.0
Synthetic DDoS	124	111	111	89.5

Table 9 gives the statistics of the escalation of data plane to control plane. CIC-IDS-2017 has the smallest rate of escalation (47.6%), where more than a half of the traffic

is completely processed at the data plane and does not need any control plane. This shows that there is an efficient threshold-based Lemon sketch detector and that it works effectively to suppress benign or low-intensity traffic at line rate.SDN Slow-rate DDoS and Synthetic DDoS, in turn, have significantly greater escalation rates (90.0% and 89.5% respectively). The long-term volumetric anomalies and burst patterns in such datasets are more often data-plane violators, which need further investigation at the control plane.In general, CIC-IDS-2017 is the most effective, due to the fact that a bigger fraction of the traffic is consumed at the DP, which lowers the control-plane load, without compromising the end-to-end detection effectiveness.

As depicted in Figure 4, CIC-IDS-2017 has better recall (100 percent) compared to SDN slow-rate DDoS data (94 percent) that covers the entire range of attacks in benchmark traffic. Conversely, SDN slow-rate DDoS dataset is more precise (100% vs. 93.1%), which means that it has fewer false positives. Nevertheless, both datasets have a similar and high F1-score (96-97%), which indicates a consistent end-to-end detection.

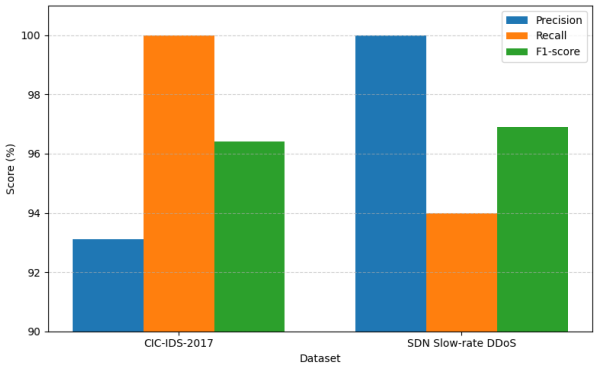

Figure 4: SDN slow rate DDos vs CIC2017

Besides that From Table 10, The proposed hybrid DP-CP framework performs better than CML-IDS on CIC-IDS-2017 as the F1-score is higher (96.4% vs. 93.4%). The proposed epoch-level granularity, in contrast to flow-level evaluation deployed in CML-IDS, records long-term traffic behaviour and eliminates sensitivity to short-term changes. This leads to the more stable predictions and higher consistency of classification between evaluation epochs.In addition, at the data plane, benign traffic is mainly handled and suspicious epochs are only sent to the control plane to enhance scalability without compromising precision. And also Comes with Limitations of our Work

1. Software-Based Evaluation:The hybrid DP-CP framework was tested with software-defined SDN switches and, thus, the findings should not represent all the performance barriers in the actual hardware implementation.

2. Data-Plane Thresholds: Static.The data-plane detection mechanism is based on known threshold values, and therefore, it might not automatically respond to significantly dynamic or changing traffic patterns.

3. CVFDT Adaptation Delay ,In equal measure, despite the fact that CVFDT allows online learning, sudden increases or decreases in traffic might need several epochs before the model fully adjusts, which might compromise the detection accuracy in the short-term.

Table 10: CML-IDS Vs Proposed Hybrid DP–CP Framework Using CICIDS2017

Metric	**CML-IDS (Base Paper)**	**Proposed DP–CP**
Detection Architecture	DP + CP	DP + CP
Data-Plane Technique	Decision Tree	Lemon Sketch-Based Monitoring
Control-Plane Model	NN and Xgb boost	CVFDT
Evaluation Granularity	Flow-level	Epoch-level
F1-score (%)	93.4	96.4
Number of Registers	1	23

7 Conclusion

The given research investigated the question whether a hybrid data-plane and control-plane (DP-CP) structure can deliver effective and scalable DDoS detection in Software-Defined Networking and reduce the control-plane overhead. The study examined the possibility of achieving a better result with lightweight sketch-based monitoring over the data plane and selective machine-learning-based monitoring over the control plane than with single-plane detection methods.

The hybrid DP-CP system was developed in a programmable SDN platform with P4 switches and a Python based controller. Research A sketch-based tracking in the data plane allowed managing early anomaly detection with no per-flow state, and selective information transfer to the control plane enabled a refined classification with a CVFDT model. The insight of the detection performance was looked at in terms of standard classification measures and confusion matrix analysis and compared to the CML-IDS baseline at the epoch level.

The findings show that the hybrid DP-CP method has a better average accuracy and F1-score compared to data-plane only detection, with less false classifications being resolved with control-plane refinement. The proposed framework, when compared to the baseline CML-IDS system, has similar or better detection results at reduced levels of continuous control-plane intervention through selective escalation.These results indicate that efficient DDoS detection in SDN does not involve continuous controller-side analysis and telemetry export. Rather, scalability can be traded with a controlled refining against detection accuracy by using lightweight in-network monitoring and selective learning-based refinement, which is why the method can be used in large scale and high-speed SDN applications.

The given research can be developed into a number of valuable directions in the future. It would be logical to then apply the framework on hardware programmable switches to test the performance and resource requirements in the real world. At the data plane, adaptive/learning-based threshold selection mechanisms may be proposed that enhance resilience to changing traffic patterns. The future research may also examine the concept of multi-controller or distributed control-plane cooperation to increase the scalability of SDN deployments in larger scale. Lastly, the implementation of automated mitigation measures, such as flow rate limiting or dynamic rerouting, would make the framework more of a detection-only system instead of a full-blown DDoS defence solution that is useful in practice.

Overall, this study shows that a hybrid data-plane/control-plane architecture of DDoS detection using existing Lemon sketch-based monitoring system and selective machine

learning can provide the optimal tradeoff between detection accuracy, scalability, and operational efficiency in Software-Defined Networking applications.

References

Ahuja, N., Singal, G., Mukhopadhyay, D. and Kumar, N. (2021). Automated ddos attack detection in software defined networking, *Journal of Network and Computer Applications* **187**: 103108.
URL: *https://doi.org/10.1016/j.jnca.2021.103108*

Ali, T. E., Chong, Y.-W. and Manickam, S. (2023). Machine learning techniques to detect ddos attacks in sdn: A systematic review, *Applied Sciences* **13**(5): 3183.
URL: *https://doi.org/10.3390/app13053183*

Apostu, A., Gheorghe, S., Hîji, A., Cleju, N., Pătraşcu, A., Rusu, C., Ionescu, R. and Irofti, P. (2025). Detecting and mitigating ddos attacks with ai: A survey, *arXiv preprint* .
URL: *https://doi.org/10.48550/arXiv.2503.17867*

Ataa, M. S., Sanad, E. E. and El-Khoribi, R. A. (2024). Intrusion detection in software-defined networks using deep learning, *Scientific Reports* **14**: 29159.
URL: *https://doi.org/10.1038/s41598-024-79001-1*

Batool, S., Aslam, M. and Akpokodje, E. (2025). A comprehensive review of ddos detection and mitigation in sdn environments, *Electronics* **14**(21): 4222.
URL: *https://doi.org/10.3390/electronics14214222*

Brandino, B., Grampin, E., Dietz, K., Wehner, N., Seufert, M., Hoßfeld, T. and Casas, P. (2024). Halids: a hardware-assisted machine learning ids for in-network monitoring, *2024 8th Network Traffic Measurement and Analysis Conference (TMA)*, pp. 1–4.
URL: *https://doi.org/10.23919/TMA62044.2024.10559083*

Ding, D., Kesgin, O. and Zilberman, N. (2024). INDDoS+: secure DDoS detection mechanism in programmable switches, *2024 IEEE 25th International Conference on High Performance Switching and Routing (HPSR 2024)*.
URL: *https://doi.org/10.1109/HPSR62440.2024.10635967*

Doriguzzi-Corin, R., Siracusano, G. and Sanvito, D. (2023). Network telemetry and security analytics in programmable data planes, *IEEE Communications Magazine* **61**(5): 48–54.

Golchin, P., Zhou, C., Agnihotri, P., Agnihotri, P., Hajizadeh, M., Kundel, R. and Steinmetz, R. (2023). Cml-ids: Enhancing intrusion detection in sdn through collaborative machine learning, *2023 19th International Conference on Network and Service Management (CNSM)*, pp. 1–9.
URL: *https://doi.org/10.23919/CNSM59352.2023.10327863*

Hulten, G., Spencer, L. and Domingos, P. (2001). Mining time-changing data streams, *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, Association for Computing Machinery, New

York, NY, USA, p. 97–106.
URL: *https://doi.org/10.1145/502512.502529*

Janabi, A. H., Kanakis, T. and Johnson, M. (2024). Survey: Intrusion detection system in software-defined networking, *IEEE Access* **12**: 164097–164120.
URL: *https://doi.org/10.1109/ACCESS.2024.3493384*

Manso, P., Moura, J. and Serrão, C. (2021). Sdn-based intrusion detection system for early detection and mitigation of ddos attacks, *arXiv preprint* .
URL: *https://doi.org/10.48550/arXiv.2104.07332*

Mawi (2025). Traffic trace info.
URL: *https://mawi.wide.ad.jp/mawi/samplepoint-F/2025/202501011400.html*

Nadeem, M. W., Goh, H. G., Ponnusamy, V. and Aun, Y. (2022). Ddos detection in sdn using machine learning techniques, *Computers, Materials & Continua* **71**(1): 771–789.

Quinn, T., Bouhafs, F. and den Hartog, F. (2025). Securing the sdn data plane in emerging technology domains, *Future Internet* **17**(11): 503.

Sharafaldin, I., Habibi Lashkari, A. and Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization, *Proceedings of the 4th International Conference on Information Systems Security and Privacy - ICISSP*, INSTICC, SciTePress, pp. 108–116.

Singh, G., Acharya, H. B. and Kwon, M. (2025). Programmable data planes for network security, *arXiv preprint* .

Swileh, M. N. and Zhang, S. (2025). Proactive ddos detection and mitigation in decentralised sdn, *arXiv preprint* .

Wang, K., Fu, Y., Duan, X. and Liu, T. (n.d.). Detection and mitigation of DDoS attacks based on multi-dimensional characteristics in SDN, **14**(1): 16421.
URL: *https://doi.org/10.1038/s41598-024-66907-z*

Wu, W., Li, Z., Liu, X., Wang, Z., Pan, H. and Xie, G. (2025). Lemon: Network-wide ddos detection with routing-oblivious per-flow measurement, *Proceedings of the USENIX Security Symposium*.
URL: *https://www.usenix.org/conference/usenixsecurity25/presentation/wu-wenhao*

Xia, J., Tian, L., Hu, Y. et al. (2025). Coddos: Detecting and mitigating diverse ddos attacks with programmable switches, *Computer Communications* **240**: 108215.

Yan, J., Xu, H., Liu, Z., Li, Q., Xu, K., Xu, M. and Wu, J. (2024). Brain-on-Switch: Towards advanced intelligent network data plane via NN-Driven traffic analysis at Line-Speed, *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, USENIX Association, Santa Clara, CA, pp. 419–440.
URL: *https://www.usenix.org/conference/nsdi24/presentation/yan*

Yungaicela-Naula, N. M., Vargas-Rosales, C., Pérez-Díaz, J. A., Jacob, E. and Martinez-Cagnazzo, C. (2023). Sdn-slowrate-ddos dataset.
URL: *https://dx.doi.org/10.21227/amrt-8y98*

Zhou, G., Liu, Z., Fu, C., Li, Q. and Xu, K. (2023). Netbeacon: In-network inference on programmable switches, *Proceedings of the USENIX Security Symposium*.