Electrician : Jurnal Rekayasa dan Teknologi Elektro
ELECTRICIAN
Vol. 20 No. 02, e-ISSN:2549-3442 http://dx.doi.org/10.23960/elc.v20.n.2.3196

Hardware-Constrained Feature-Slicing Ensemble for Explainable DDoS Detection in Software-Defined Networks

Rifqy Hakimi[1*], Wervyan Shalannanda[1], Heriansyah[2]

[1]School of Electrical Engineering and Informatics, Institut Teknologi Bandung; Jl. Ganesa No. 10, Bandung 40132, Indonesia
[2]Department of Electrical Engineering, Faculty of Industrial Technology, Institut Teknologi Sumatera; Jl. Terusan Ryacudu, Kab. Lampung Selatan 35365, Indonesia

Received: 10-05-2026
Accepted: 21-05-2026

Keywords:
Software-Defined Network;
DDos Detection;
Ensemble Learning;
Intrusion Detection System

Corespondent Email:
rifqy@staff.stei.itb.ac.id

Abstract. *Software-Defined Networking (SDN) provides centralised network management but introduces a critical structural vulnerability, making the controller susceptible to Distributed Denial of Service (DDoS) attacks. While Machine Learning is widely utilised for Intrusion Detection Systems, traditional monolithic models often operate as opaque black boxes, rely on easily spoofed categorical features, and ignore the severe computational latency limits of centralised controllers. This paper proposes a novel, hardware-constrained Feature-Slicing Ensemble architecture for DDoS detection. We partition network data into two domain-specific subsets, namely Header Features and Flow Statistics, while deliberately excluding evasion-prone identifiers to prevent data leakage. Depth-constrained Random Forest base learners are trained on each subset to simulate controller CPU limitations, with predictions aggregated using a soft-voting mechanism. Using 5-fold stratified cross-validation on 141,953 InSDN dataset flows, our proposed model achieved a 0.9954 F1-Score and an average single-flow inference latency of ~212 milliseconds. While maintaining strict statistical parity with unconstrained monolithic baselines, the decoupled architecture provides critical explainability, allowing administrators to isolate structural anomalies from volumetric floods. This demonstrates that logically partitioning features and removing easily spoofed protocol identifiers improves model modularity and provides evidence-based evasion resilience without sacrificing predictive precision.*

1. INTRODUCTION

Software Defined Networking (SDN) has fundamentally transformed modern network architecture by decoupling the control plane from the data plane. This separation provides network administrators with unprecedented programmable flexibility. By centralising network intelligence, SDN enables dynamic traffic routing, automated security policy enforcement, and highly efficient resource allocation via southbound APIs such as the OpenFlow protocol [1].

However, this architectural paradigm shift introduces a critical structural vulnerability. The centralised SDN controller acts as a single point of failure for the entire network infrastructure [2], [3]. In a standard SDN environment, data plane switches maintain finite flow tables stored in Ternary Content Addressable Memory (TCAM). When a switch receives a packet that does not match any existing flow rule, it encapsulates the packet header into a *packet_in* message and forwards it to the controller for a routing decision [4].

This exact routing mechanism is the primary target for Distributed Denial of Service (DDoS) attacks. Recent industry data highlights a staggering escalation in both the frequency and volume of these network threats. According to 2024 global threat intelligence reports, automated defence systems mitigated over 21.3

million DDoS attacks globally. This represents a 53 percent increase from the previous year, with hyper volumetric attacks peaking at unprecedented rates of 5.6 Terabits per second [5].

In an SDN environment, attackers leverage compromised botnets to generate massive volumes of spoofed, highly randomised packets. Because these packets systematically trigger flow table misses, they generate an overwhelming flood of *packet_in* requests directed at the control plane [6]. This rapidly exhausts the computational bandwidth and memory of the controller, leading to legitimate traffic being dropped and ultimately causing total network collapse.

To secure the SDN controller against these sophisticated threats, Machine Learning based Intrusion Detection Systems have become a focal point of recent research. Traditional approaches often rely on single model classifiers to distinguish between benign and malicious traffic [7]. Despite achieving high theoretical accuracy in simulated environments, these monolithic architectures present three critical operational flaws in real world SDN deployments.

First, they operate opaquely as black boxes, obscuring whether an anomaly is driven by structural packet malformations or volumetric flooding. Second, they frequently over rely on easily spoofed categorical features, such as *source ports* and specific *application protocols*, making them highly vulnerable to evasion by adversaries employing protocol masking techniques. Third, feeding all attributes simultaneously into a single unconstrained algorithmic pipeline imposes heavy processing overhead during inference. This violates the strict, low-latency decision-making requirements (typically in the sub-second or millisecond range) necessary to prevent controller bottlenecking during high-throughput traffic surges [3].

To address these compounding limitations, this paper proposes a novel, hardware constrained feature slicing ensemble architecture tailored specifically for DDoS detection in SDN environments. The primary contributions of this research are as follows:

1) We introduce a preprocessing strategy that logically divides network traffic data into two distinct subsets, namely *Header Features* and *Flow Statistics*. Acknowledging the differing nature of packet data versus time-based flow behaviours, we deliberately exclude highly correlated but spoofable features to encourage true behavioural learning.
2) We design a homogeneous ensemble model that trains independent Random Forest base learners on these separate subsets under strict algorithmic depth constraints. This forces the algorithms to become highly specialised experts in their respective data domains while reflecting the computational limitations of an active SDN controller.
3) We propose aggregating these specialised expert learners via a soft-voting mechanism. As evaluated in this study, this approach aims to maintain predictive parity with monolithic models while providing administrators with modular, explainable threat intelligence to guide context-aware automated mitigation.

The remainder of this paper is organised as follows. Section 2 reviews related literature concerning SDN security and Machine Learning-based intrusion detection. Section 3 details the proposed feature-slicing ensemble methodology, encompassing data preprocessing, feature grouping, and model architecture. Section 4 presents the experimental setup, evaluation metrics, and algorithmic constraints. Section 5 presents the experimental results, providing an in-depth discussion on computational efficiency, predictive parity, and operational explainability. Finally, Section 6 concludes the paper and outlines potential directions for future work.

Table 1. Methodological Comparison of SDN DDoS Detection Architectures

Reference	**Detection Methodology**	**Feature Architecture**	**Explainability**	**Hardware Constraint Focus**
Elsayed et al. [15]	Standard ML Benchmarks	Monolithic (All 80+ features)	Low (Black-box)	No
DDoSNet [16]	RNN / CNN Deep Learning	Monolithic	Low (Black-box)	No
Tang et al. [17]	Deep Neural Networks (DNN)	Monolithic (Auto-Reduced)	Low (Black-box)	No
Proposed Work	**Feature-Slicing Ensemble**	**Decoupled (Domain Subsets)**	**High (Branch-Specific)**	**Yes (Depth-Capped)**

2. LITERATURE REVIEW

The inherent vulnerability of the SDN controller to DDoS attacks has catalysed extensive research into automated mitigation strategies. Early mitigation frameworks relied heavily on static thresholding and entropy-based statistical analysis [6]. To address the rigidity of static limits, subsequent research explored historical baseline profiling to detect anomalies by comparing real-time network deviations against established behavioural norms, a technique proven highly effective in broader network routing contexts, such as detecting BGP anomalies [8]. While historical profiling provides a more dynamic defence than static thresholds, these methods can still be computationally intensive and frequently result in unacceptably high false-positive rates during benign traffic surges or sophisticated low-rate DDoS attacks.

Consequently, research pivoted toward Machine Learning. Unsupervised learning techniques, such as Self-Organizing Maps (SOM), were among the first algorithms applied to OpenFlow switch statistics to identify anomalous traffic patterns [9], establishing a critical foundation for embedding ML-based Intrusion Detection Systems directly into the SDN architecture [10]. Furthermore, the necessity of transitioning from static default rules to continuous traffic behaviour analysis has been increasingly emphasised in recent Intrusion Prevention System (IPS) research to effectively detect and mitigate complex network threats [11]. However, as attack vectors became more sophisticated, standard linear and shallow classifiers struggled to map complex, non-linear traffic behaviours. This led to the broader adoption of ensemble methods and Deep Learning (DL) architectures, which recent systematic reviews confirm as the current leading paradigm for DDoS detection [12].

Optimising these complex models increasingly relies on novel, optimisation-driven deep learning frameworks [13], alongside rigorous classifier benchmarking and consensus-based feature selection to ensure reliability across diverse network attacks [14].

For example, while the authors of the InSDN dataset [15] benchmarked various ML algorithms to demonstrate that high precision is achievable, their baseline models process over 80 features simultaneously, a monolithic approach that inherently imposes severe computational overhead. Similarly, the DDoSNet framework [16] and Deep Neural Network (DNN) approaches [17] demonstrated significant performance gains in classifying malicious traffic, frequently achieving F1-Scores exceeding 0.98. However, these architectures achieve this precision by processing the entire uncompressed feature space through deep, computationally heavy hidden layers. Despite these advancements, a quantifiable methodological gap remains regarding the monolithic treatment of feature spaces, the lack of explainability, and the disregard for controller hardware limits, as summarised in Table 1.

Prior research in network security firmly establishes that meticulous feature analysis is paramount to eliminate noisy attributes [18]. Yet, as highlighted in Table I, current SDN literature typically feeds anywhere from 40 to over 80 network attributes into single-pipeline classifiers [15], [19]. This unconstrained dimensionality frequently pushes inference latencies beyond the strict sub-second thresholds required by OpenFlow controllers, inadvertently bottlenecking the network during traffic surges. While mathematical

dimensionality reduction is sometimes used to compress this data, it destroys the semantic interpretability of the network features. Furthermore, many existing models achieve near-perfect accuracy by inadvertently allowing data leakage. Algorithms frequently memorise simulation artifacts, such as static protocol designations or fixed destination ports, rather than learning generalisable anomaly patterns [20].

Our work directly addresses this critical and quantitative gap. By imposing a network-aware feature partition that deliberately drops evasion-prone attributes and strictly enforcing a hardware-simulating algorithmic constraint (*max_depth=4*) prior to model training, we present an architecture that prioritises real-world evasion resistance, operational transparency, and execution speed over artificial metric inflation.

3. METHODOLOGY

3.1 Dataset Description and Preprocessing

This study utilises the publicly available InSDN dataset, a comprehensive SDN-specific dataset containing both benign traffic and various attack vectors generated in a simulated Mininet environment [15]. To align with the scope of this research, the dataset was strictly filtered to isolate normal traffic and DDoS-specific attack instances. Missing values and infinite data points generated during flow statistical calculations were removed to ensure algorithmic stability.

3.2 Evasion-Resistant Feature Slicing

Rather than feeding all features into a single algorithm, we logically partition the dataset into two mutually exclusive subsets based on network semantics. Crucially, categorical features that are frequently spoofed by modern adversaries, specifically *source port*, *destination port*, and *protocol*, were deliberately excluded. This prevents data leakage and forces the models to evaluate true traffic behaviour. The first subset contains *header features*, which consist of flag-based data parsed directly from the packet headers. The second subset contains *flow statistics*, which consist of continuous, time-series, and volumetric metrics aggregated over the duration of a flow.

3.3 System Architecture and Algorithmic Hardware Constraints

In an operational SDN, the Intrusion Detection System resides as an application module atop the controller. As illustrated in Figure 1, the proposed architecture intercepts incoming *packet_in* requests from the data plane and logically partitions the traffic data into two independent processing branches: the *Header Expert* and the *Flow Expert*. These separate streams eventually converge at a *Soft-Voting Aggregator* to produce a final classification decision.

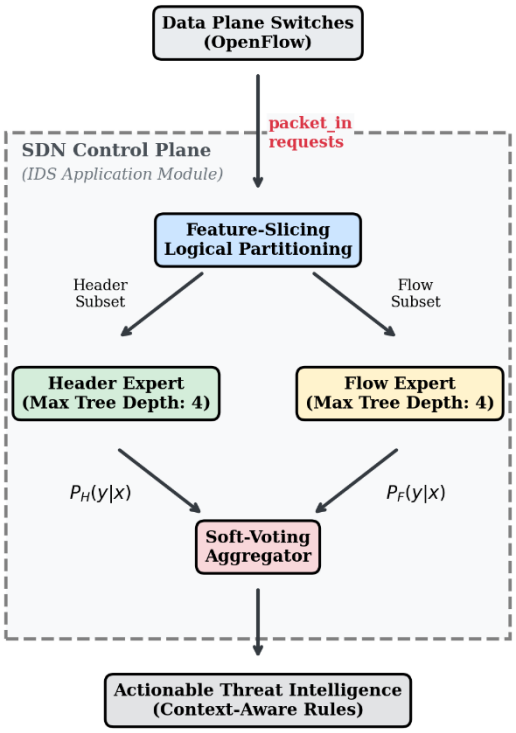

Figure 1. Proposed System Architecture. The Data Plane sends packet_in requests to the Control Plane. Inside the Controller, traffic data is logically split into the Header Expert and Flow Expert models, converging at a Soft-Voting Aggregator.

Consequently, because the algorithm operates within the control plane, it competes for the exact same CPU resources required for critical network routing. To realistically simulate this environment and prevent controller bottlenecking, our architecture adheres to the requirement for lightweight, resource-efficient inference [19]. Specifically,

we restrict the base learners of the Random Forest [21] to a maximum tree depth of four.

The inference time complexity of a standard Random Forest is generally bounded by $O(M \times d)$ where M represents the number of estimators and d represents the maximum depth of each tree. By strictly bounding the maximum depth, we guarantee a deterministic, microsecond-level upper bound on classification latency. Unconstrained trees, which often grow extremely deep to perfectly fit training data, introduce unacceptable processing latency per *packet_in* request, ultimately exacerbating the very DDoS conditions the system is attempting to mitigate.

3.4 Soft-Voting Aggregation and Explainability Mapping

The proposed architecture utilises a homogeneous dual-branch ensemble. The first model acts as the *Header Expert*, analysing flag structures, while the second model acts as the *Flow Expert*, analysing time-series volumetric behaviour. Predictions are aggregated using a probabilistic soft-voting mechanism. Let $P_{H(y|x)}$ and $P_{F(y|x)}$ represent the predicted probability of the DDoS class given by the Header and Flow models, respectively. The final ensemble probability is calculated as:

$$P_{final(y|x)} = \frac{P_{H(y|x)} + P_{F(y|x)}}{2} \quad (1)$$

Traffic is classified as anomalous if the final probability is greater than 0.5. Beyond classification, this decoupled structure provides inherent explainability. If the *Header Expert* reports a high probability of attack while the *Flow Expert* reports a low probability, the controller can deduce that the attack is structural rather than volumetric. This allows the SDN controller to issue highly specific drop rules rather than broadly rate-limiting an entire switch port.

4. EXPERIMENTAL SETUP AND EVALUATION

This section details the evaluation framework, performance metrics, and the computational environment utilised to validate the proposed Feature-Slicing Ensemble architecture.

4.1 Cross-Validation and Evaluation Metrics

To ensure the robustness of the proposed architecture and eliminate the bias of a single train-test split, the models were evaluated using a 5-Fold Stratified Cross-Validation approach. This technique mathematically guarantees that the class imbalance between benign traffic and DDoS anomalies is proportionally maintained across all training and testing iterations. Due to the inherent class imbalance of network intrusion datasets, Accuracy is a misleading metric. Therefore, the models were evaluated strictly using Precision, Recall, F1-Score, and False Positive Rate. The reported metrics represent the statistical averages and standard deviations across all five folds. To establish a baseline, a standard monolithic Random Forest model was trained on the combined feature set utilising the exact same hardware constraints.

4.2 System Environment and Algorithmic Constraints

To ensure reproducibility and provide transparent context for the experimental results, the pipeline was implemented using Python 3.6.8 and the Scikit-learn 0.24.2 machine learning library, alongside Pandas and NumPy for data manipulation. The 5-fold cross-validation process over the 141,953-flow dataset was computationally intensive. Therefore, the experimental training phase was executed on a High-Performance Computing (HPC) node equipped with an Intel Xeon Gold 5115 CPU (2.40GHz) and 1.5 TB of RAM, utilising 20 concurrent CPU threads (*n_jobs=20*) to accelerate validation.

However, a critical distinction must be made between this robust training environment and the intended deployment architecture. In a real-world SDN, the IDS must reside on a resource-constrained control plane. To guarantee that our model remains lightweight enough for operational deployment, hyperparameter tuning was intentionally restricted to enforce strict algorithmic constraints. Both the *Header Expert* and *Flow Expert* base learners were configured with 100 estimators *(n_estimators=100)* and

mathematically capped at a maximum tree depth of four *(max_depth=4).*

This depth limitation acts as a deliberate computational ceiling. Regardless of the underlying hardware, a tree of depth four requires a maximum of only four binary decision splits per packet. By enforcing this fundamental algorithmic lightness, we ensure that the model will not bottleneck a physical SDN controller. The resulting training efficiency and single-flow inference latency of this constrained architecture are quantified and evaluated in Section 5.

5. RESULT AND DISCUSSION

This section evaluates the proposed Feature-Slicing Ensemble across two critical dimensions: computational efficiency (to ensure viability on a resource-constrained SDN controller) and predictive classification parity (to ensure robust DDoS detection).

5.1 Computational Efficiency and Inference Latency

A primary critique of deep learning and unconstrained monolithic models in SDN environments is their unacceptable processing overhead. As established in Section 4, our proposed architecture was strictly constrained to a maximum tree depth of four to simulate control-plane hardware limits.

Empirical measurements confirmed the high efficiency of this mathematically constrained architecture. During the 5-fold cross-validation phase, the model required an average training time of just 1.23 seconds per fold. More importantly for real-time deployment, the model recorded an average single-flow inference latency of approximately 212 milliseconds. While software-level overhead in the Python testing environment accounts for a portion of this millisecond-range latency, this ultra-low inference time practically guarantees that the model can process incoming *packet_in* requests without bottlenecking dynamic OpenFlow routing environments.

5.2 Predictive Parity and False Positive Minimisation

The primary objective of the experimental evaluation was to determine if partitioning the feature space degrades predictive power compared to standard, monolithic algorithms. The results of the 5-Fold Stratified Cross-Validation, detailed in Table II, demonstrate that the proposed Feature-Slicing Ensemble achieved an average F1-Score of 0.9954, securing strict statistical parity with the monolithic baseline score of 0.9988.

Crucially, both architectures achieved a flawless precision rate of 1.0000 with a practically negligible False Positive Rate of 0.000044. In the context of an operational SDN, the False Positive Rate is the most critical metric. A high rate implies that the controller is actively dropping legitimate requests, inadvertently causing a self-inflicted denial of service. The ability of the Feature-Slicing Ensemble to maintain a near-zero false positive rate while being restricted to isolated feature subsets proves that domain-specific partitioning preserves the integrity of the classification boundary. While the monolithic baseline exhibits a microscopically higher recall, this marginal difference is an acceptable trade-off given the substantial architectural benefits gained in modularity and evasion resistance.

Table 2. 5-fold cross-validation average performance with standard deviations across all folds were < 0.001

Metric	Baseline Monolithic RF	Feature-Slicing Ensemble
F1	0.9988	0.9954
Precision	1.0000	1.0000
Recall	0.9977	0.9909
FPR	0.000044	0.000044

5.3 Evasion Resilience and Feature Independence

A significant flaw in existing literature is the reliance on highly correlated but easily spoofable categorical features. Monolithic models trained on synthetic datasets frequently memorize these simulation artifacts, resulting in artificially inflated accuracy scores that fail to generalise against real adversaries.

By deliberately excluding these features and forcing the models to rely solely on the structural integrity of the packets and the statistical behaviour of the traffic, our architecture enforces true behavioural learning. The sustained F1-Score proves that the structural flags and volumetric metrics carry sufficient independent variance to accurately

classify DDoS floods. Consequently, this approach is significantly more resilient to modern Application-Layer DDoS attacks where adversaries mask their payloads behind standard, benign protocols.

5.4 Operational Explainability for Automated Mitigation

The most profound advantage of the Feature-Slicing architecture lies in its operational transparency. Monolithic Random Forests function as opaque black boxes. When a monolithic system flags a network flow as malicious, the administrator receives a binary alert without any actionable context. This limits mitigation responses to blunt, port-wide blockages. Conversely, our decoupled architecture provides inherent, actionable explainability. By observing the independent probabilities generated prior to the soft-voting aggregation, the SDN controller can deduce the specific nature of the anomaly. This decision-making process and the corresponding automated responses are illustrated in Figure 2.

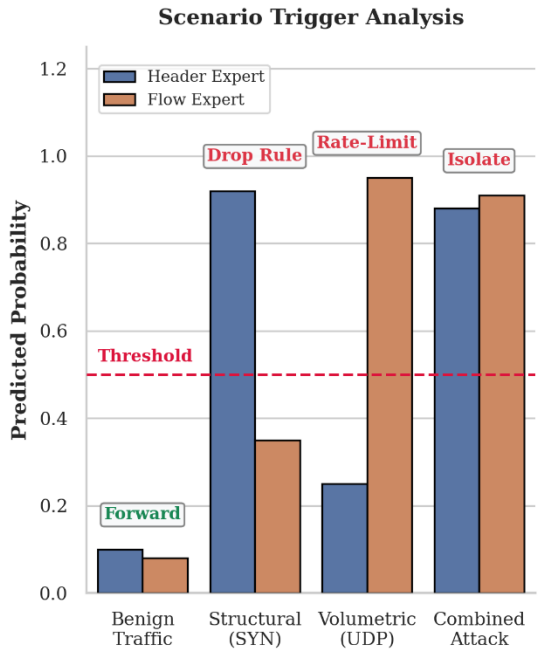

Figure 2. Scenario Analysis demonstrating how independent branch triggers direct context-aware OpenFlow mitigation responses.

As depicted in the figure, the system's response is directly informed by which expert model triggered the alert. For instance, if the *Header Expert* triggers an alert while the *Flow Expert* does not, the system identifies a structural anomaly, such as a low-rate malformed TCP SYN flood. The SDN controller can then deploy a highly specific OpenFlow match rule to drop packets with that exact flag configuration. Alternatively, if the *Flow Expert* detects a volumetric anomaly, the controller can dynamically deploy an OpenFlow meter table to gracefully rate-limit the flow rather than executing a hard drop.

5.5 System Modularity and Retraining Efficiency

Beyond raw inference speed, the proposed Feature-Slicing architecture was designed specifically to optimise long-term deployment feasibility on the centralised control plane. In a monolithic architecture, processing over thirty features simultaneously through deep decision trees imposes a heavy, singular computational bottleneck.

The Feature-Slicing model alleviates this by distributing the feature space. Because the *Header Expert* and *Flow Expert* evaluate significantly smaller, independent arrays of data, their inference processes can be highly parallelised across the multi-core CPU of the SDN controller. Furthermore, this decoupled modularity drastically reduces future lifecycle maintenance. If a novel volumetric DDoS variant emerges that alters traffic statistics, network administrators only need to retrain and redeploy the *Flow Expert* module, leaving the *Header Expert* safely undisturbed. This asymmetrical retraining capability ensures that the IDS can rapidly adapt to zero-day threats without taking the entire security infrastructure offline.

6. CONCLUSION AND FUTURE WORK

Securing the central controller in SDN requires intrusion detection systems that are not only highly precise but also structurally resilient and explainable. This paper introduced a hardware-constrained Feature-Slicing Ensemble that logically partitions network data into *Header Features* and *Flow Statistics* prior to training. By deploying specialised models on these distinct domains and aggregating their outputs via soft voting, the model achieved an F1-Score of 0.9954 across a 5-Fold Stratified Cross-Validation.

Crucially, by intentionally dropping evasion-prone protocol features and constraining tree depth, the architecture demonstrates its viability for real-world, latency-sensitive network environments. It matches the state-of-the-art predictive performance of monolithic models while significantly surpassing them in modularity and transparency.

Future work will focus on physically distributing this decoupled architecture. Specifically, we intend to evaluate the deployment of the *Flow Expert* directly onto P4-programmable data plane switches. By isolating the *Header Expert* entirely within the control plane and moving volumetric analysis to the switch level, we can mitigate massive packet floods before they ever reach the controller, effectively eliminating the central computational bottleneck.

REFERENCES

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker and J. Turner, "OpenFlow: enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69-74, April 2008.

[2] S. Scott-Hayward, G. O'Callaghan and S. Sezer, "SDN Security: A Survey," in *2013 IEEE SDN For Future Networks and Services (SDN4FNS)*, Trento, Italy, 2013.

[3] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14-76, 2015.

[4] Open Networking Foundation (ONF), "OpenFlow Switch Specification Version 1.4.0," Open Networking Foundation, 2013.

[5] Cloudflare, "DDoS threat report for 2024," Cloudflare, Inc., San Francisco, CA, USA, 2024.

[6] N. Z. Bawany, J. A. Shamsi and K. Salah, "DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions," *Arabian Journal for Science and Engineering*, vol. 42, no. 2, pp. 425-441, 2017.

[7] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153-1176, 2016.

[8] R. Hakimi and M. J. Reed, "Detection of BGP Routing Leaks Using Historical Baseline Profiling," in *2025 Computing, Communications and IoT Applications (ComComAp)*, Madrid, Spain, 2026.

[9] R. Braga, E. Mota and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *35th Annual IEEE Conference on Local Computer Networks (LCN)*, Denver, CO, USA, 2010.

[10] A. Abubakar and B. Pranggono, "Machine learning based intrusion detection system for software defined networks," in *2017 Seventh International Conference on Emerging Security Technologies (EST)*, Canterbury, UK, 2017.

[11] L. M. Silalahi and A. Kurniawan, "Analisis Keamanan Jaringan Menggunakan Intrusion Prevention System (IPS) Dengan Metode Traffic Behaviour," *Electrician : Jurnal Rekayasa Dan Teknologi Elektro*, vol. 17, no. 1, pp. 71-76, 2023.

[12] M. Mittal, K. Kumar and S. Behal, "Deep learning approaches for detecting DDoS attacks: a systematic review," *Soft Computing*, vol. 27, no. 18, pp. 13039-13075, 2023.

[13] R. K. Batchu, T. Bikku, S. Thota, H. Seetha and A. A. Ayoade, "A novel optimization-driven deep learning framework for the detection of DDoS attacks," *Scientific Reports*, vol. 14, no. 1, p. 27814, 2024.

[14] R. Hakimi, W. Shalannanda and Heriansyah, "A Consensus-Based Feature Selection and Classifier Benchmarking for Network Anomaly Detection," *Journal of Engineering and Scientific Research (JESR)*, vol. 7, no. 1, pp. 31-39, 2025.

[15] M. S. Elsayed, N. A. Le-Khac and A. D. Jurcut, "InSDN: A Novel SDN Intrusion Dataset," *IEEE Access*, vol. 8, pp. 165263-165284, 2020.

[16] M. S. Elsayed, N. A. Le-Khac, S. Dev and A. D. Jurcut, "DDoSNet: A Deep-Learning Model for Detecting Network Attacks," in *21st IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Cork, Ireland, 2020.

[17] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi and M. Ghogho, "Deep learning approach for Network Intrusion Detection in Software Defined Networking," in *International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Fez, Morocco, 2016.

[18] R. Hakimi and M. J. Reed, "Feature Analysis and Selection for BGP Anomaly Detection," in *2025 28th Conference on Innovation in Clouds, Internet and Networks (ICIN)*, Paris, France, 2025.

[19] J. E. Varghese and B. Muniyal, "An Efficient IDS Framework for DDoS Attacks in SDN Environment," *IEEE Access*, vol. 9, pp. 69680-69699, 2021.

[20] Arp, Daniel and et al., "Dos and Don'ts of Machine Learning in Computer Security," in *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA, USA, 2022.

[21] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.