
Journal of Informatics and Web Engineering

Vol. 5 No. 3 (October 2026)

eISSN: 2821-370X

An Intelligent SDN-Enabled Framework for Real-Time IPv6 IP Spoofing Detection and Prevention

Navaneethan C. Arjuman^{1*}, Ahmad Fikri Sharudin², Zar Chi Hlaing³, Supriyanto Praptodiyono⁴

^{1,2,3}Faculty of Computing and Informatics, Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Selangor, Malaysia

⁴Faculty of Computer Sciences, Universitas Pembangunan Nasional "Veteran" Jakarta, Jl. RS. Fatmawati, Pondok Labu, Jakarta Selatan, 12450 Indonesia

*corresponding author: (navaneethan@mmu.edu.my; ORCID: 0000-0002-1700-9859)

Abstract - With the increase in the adoption of Internet Protocol version 6 (IPv6), the issue of Internet Protocol (IP) spoofing has remained a significant risk that damages the integrity of networks. Although some countermeasures, like Secure Neighbor Discovery (SEND) and Router Advertisement (RA) Guard, are available, these solutions are complex to deploy, consume high resources, and can be bypassed by advanced attacks. This paper provides a critical review of the available IPv6 spoofing prevention methods and proposes an enhanced multi-layered framework that uses improved ingress and egress filtering, Artificial Intelligence (AI)-based anomaly detection, and Software-Defined Networking (SDN) to automatically mitigate threats in real time. The enhanced framework accurately identifying spoofed traffic, whereas SDN dynamically applies policies in the network to isolate and block malicious sources without affecting legitimate traffic. The framework is implemented over a Mininet and Docker-based setup in which it is tested under controlled and real-world traffic conditions. It had an average precision and recall of 0.98 with an F1-score of 0.97 and a prevention latency averaging less than 0.4 seconds. Through the centralized traffic visibility provided by SDN and the pattern recognition capabilities of Machine Learning (ML), the proposed framework adjusts smoothly to traffic changes and new spoofing techniques. Its modular architecture allows it to be integrated into existing systems and deployed in a manner that does not interfere with the flow of operations. The proposed solution is better in robustness, scalability, and adaptability than any traditional static filtering approach, and therefore it is a practical approach to enhance security in both dual-stack and IPv6-only network environments. This research provides practical implications and further highlights the possibility of ML and SDN contributions in the creation of proactive IPv6 security systems.

Keywords— IPv6, IP Spoofing, Ingress Filtering, Egress Filtering, Anomaly Detection, Software-Defined Networking

Received: 14 June 2025; Accepted: 23 January 2026; Published: 16 October 2026

This is an open access article under the [CC BY-NC-ND 4.0](#) license.



1. INTRODUCTION

The depletion of the 4.3 billion addresses available in Internet Protocol version 4 (IPv4) and the unparalleled expansion of mobile, cloud, and Internet of Things (IoT) deployments have necessitated the transition to IPv6 on a

global scale [1], [2]. IPv6 offers a theoretical pool of 3.4×10^{38} addresses, eliminating reliance on Network Address Translation (NAT) workarounds and restoring end-to-end connectivity to the protocol [1]. The hierarchical aggregation of addresses will reduce the growth of global routing tables, the 20-bit flow-label field will allow for fine-grained traffic classification, and the natively integrated Internet Protocol Security (IPsec) protocol will provide mandatory confidentiality and authentication at the network layer (which was optional and inconsistently implemented in IPv4) [2]. Also, IPv6 mandates host-based reassembly of fragmented packets, which limits evasion in the middle of a path, and supports multicast with the use of Multicast Listener Discovery version 2 (MLDv2)'s per-source filtering. Cryptographically Generated Addresses (CGAs) and SEND provide public key verification within NDP messages by linking addresses to an identity, thus decreasing the probability of naïve neighbor-spoofing attacks [2].

Even with all these strengths, IPv6's NDP is still insecure. NDP does not have any authentication and relies on unverified Internet Control Message Protocol version 6 (ICMPv6) packets, which can be used to launch spoofs of Router Advertisements (RA) and subject the network to floods of neighbor-solicitation requests, resulting in hijacking traffic or taking down hosts [3]. Media Access Control (MAC) spoofing in which an attacker fakes the MAC address of its device to impersonate another host provides an opportunity to use external and unauthorized resources, gain access to resources that would typically be permissible by MAC-filter bypass, or direct traffic elsewhere, often as a first step to session hijacking, Address Resolution Protocol (ARP) spoofing, and network eavesdropping [4]. Researchers have shown how simple scripts can leverage NDP's lack of authentication to conduct a Denial-of-Service (DoS) attack or Man-in-the-Middle attacks [5]. IP source-address spoofing is among the most nefarious of IPv6 threats facing an organization. Attackers can falsify IPv6 headers to avoid detection, increase distributed DoS hijack traffic by carefully manipulating the routing of its IP, and hijack or reroute legitimate flows [6]. Fragmentation exploits, along with extension-header chaining, are also powerful mechanisms at the disposal of unscrupulous offenders who can discreetly pass malicious payloads past legacy security appliances [7].

Every foundational defense has holes when practicing. SEND's cryptographic extensions require an inordinate amount of key-management overhead on routers and hosts [8]. RA-Guard schemes can be foiled by extension-header tampering and fragmented packets [8]. Static Access Control List (ACL)-based filtering, while easy, is inflexible it consumes Ternary Content Addressable Memory (TCAM), is manually updated, cannot differentiate between protocol types or traffic classes, and permits a whole IP prefix or none with no nuance [9], [10].

In order to remedy these deficiencies, a multilayered framework for preventing IPv6 spoofing is proposed. Coarse-grained ingress/egress filters that enforce checking that a source is the owner of a prefix and hop-limit heuristics are included; a supervised classifier that examines flow metadata and extension-header patterns indicating traffic source to index benign from spoofed flows; and a centralized SDN controller that uses global topology awareness to dynamically install fine-grained flow rules to isolate malicious sources and reroute suspicious flows for extra inspection.

2. LITERATURE REVIEW

In this section, recent works related to IPv6 spoofing prevention mechanisms that have been developed over the last few years will be reviewed and critically analyzed in terms of effectiveness and limitations.

2.1 Existing Prevention Mechanisms and Limitations

Al-Shareeda et al. have put forth Secure Router Discovery (SecMac-SRD), a secure scheme that merges Universal Message Authentication Code (UMAC) hashing with ElGamal public-key cryptography for the purpose of authenticating IPv6 Router Solicitation (RS) and RA messages. In this scheme, each RS/RA packet contains an appended "SecMac-tag" that is calculated via the use of a lightweight UMAC hash over the packet header and a nonce and is signed with ElGamal to guarantee integrity and authenticity [11]. When a host receives RS/RA messages, it must verify the UMAC tag before verifying the ElGamal signature to ensure that it only accepts RAs from authorized routers that are legitimately signed. Ultimately, SecMac-SRD is extremely resilient against forged RAs; however, it relies on the availability of a strong key distribution and management architecture network operators need to provision and rotate ElGamal key pairs and the code itself induces observable processing and latency overhead caused by the additional cryptographic operations [11].

Neighbor Discovery Protocol security (NDPsec) was proposed by Al-Ani et al. as an NDPsec mechanism that relied upon Ed25519 digital signatures. In the NDPsec design, every host generates an IPv6 address where the interface identifier is derived from the host's Ed25519 public key. When a host sends an NDP message, it transmits a "Secure-Option" that includes a signature of the message payload. The recipient will extract the address that is public-key based, verify the Ed25519 signature, and accept the NDP packet only when the signature is valid [12]. In this way, the design ties every address to its key owner and prevents spoofing. However, generating and verifying the signature uses more Central Processing Unit (CPU) and latency than hash function-only approaches, such as Trust-ND. In the NDPsec design, this increased CPU and latency may affect the performance of constrained devices or those experiencing extreme throughput performance [12].

Puhl and Guo proposed Voucher-Based Addressing (VBA) by using cryptographic "vouchers" to bind link-layer (e.g., MAC) addresses to IPv6 addresses while avoiding a complex Public Key Infrastructure (PKI). When a host is discovered, it can present a voucher, which is simply a shorter, signed token (that is expected to be issued by some local consensus of the access nodes), establishing the host's rights to use the address provided. Peers can validate that voucher via only lightweight checks of the voucher's signature, without needing the complexity of a certificate authority [13]. For some scenarios provided by VBA, it simplifies the establishment of trust in a dynamic Local Area Network (LAN) and decreases the need for centralized infrastructure. However, because it is bound by a single, longest prefix match for an IPv6 address for a host, may not support anycast addresses, and does not support statically assigned IPs, its applicability is limited to times when one-to-many address mappings or manual configurations are not required [13].

Mönnich et al. presented a P4-based implementation of RA-Guard that moved rogue RA filtering into the data plane. They relied on programmable switch pipelines to match and drop unauthorized RAs before they reached hosts and made use of the programmability of Programming Protocol-independent Packet Processors (P4) to parse IPv6 extension headers and apply complex filtering rules at line rate [14]. Implementing rogue RA checks, drops, and logging on the switch Application-Specific Integrated Circuit (ASIC) reduces the number of rules to upload to the switch and avoids delays due to the controller, although there still needs to be manual reprogramming of authorized routers, which means it may be too static in highly dynamic environments [14].

Nasser and Hussain proposed backing static ARP entries with packet-filtering logic to address security concerns in IPv4 neighbor resolution and promote similar action in IPv6 security. The Nasser and Hussain mechanism verifies all ARP or NS requests by comparing MAC-IP bindings to a locally stored encrypted blacklist of invalid bindings, dropping invalid requests before even needing to update the cache [15]. With a modest improvement in defenses against ARP spoofing and gratuitous announcements, the heavily loaded static scheme is still unable to provide scalability for particularly large or dynamic LANs with many devices frequently joining, leaving, or changing addresses. Manual management of static entries is burdensome [15]. Table 1 describes the main limitations of each of the methods, which demonstrate gaps filled by the proposed framework.

Table 1. Summarization of each Prevention Mechanism

Authors	Proposed Mechanism	Limitation
Mahmood A. Al-Shareeda et al. [11]	SecMac-SRD: appends a UMAC hash and ElGamal-signed "SecMac-tag" to RS/RA messages for integrity	Requires PKI-style key distribution and adds cryptographic overhead
Ayman Al-Ani et al. [12]	NDPsec: embeds Ed25519 signatures in a Secure-Option field and derives IPv6 IDs from public keys	Higher CPU usage and latency compared to hash-only methods
Zachary T. Puhl & Jinhua Guo [13]	VBA: uses short, signed vouchers to bind link-layer to IPv6 addresses	Incompatible with anycast addresses and static IP assignments
Moritz Mönnich et al. [14]	RA-Guard (P4 implementation): filters rogue RAs in the switch data plane using P4 pipelines	Requires P4-capable hardware and manual rule updates
Hiba Imad Nasser & Mohammed Hussain [15]	Static ARP entries + packet filtering validate MAC-IP bindings against encrypted blacklists	Scalability issues due to manual entry management in large LANs

2.2 AI- and SDN-Based Approaches

By incorporating SDN-native validation methods, spoofed IPv6 packets can be detected and dropped as close to the "edge" of the network as possible, thus avoiding latency and control plane load [16]. Likewise, the Lightweight Source Address Validation (LSAV) algorithm is based on the centralized view of an SDN on the SDN controller, allowing for a valid inspection of characteristics of traffic (i.e., timing of flows and periods between packets) rather than maintaining a cost-prohibitive binding database associated with traditional LSA protocols, yielding high-throughput spoof detection with acceptable resource consumption [17]. Conversely, the P4-based RA-Guard implementation embeds spoofed RA filtering directly in the switch's data plane, dropping malicious RAs at line rate before arriving on hosts and taking repeated processing off the controller for the ASIC. Together, these neat strategies exemplify how programmable networks have the potential to offer up scalable, near-real-time defense against address-forgery threats with minimal processing cost on routers or the controller. The ML detection systems administrator should use SDN enforcement to offer further adaptability against complex spoofing methods [18]. The Distributed Denial of Service (DDoS) defense study offered by LSAV analyzed clustering (K-means) and classification (KNN) algorithms to notice that lightweight AI methods were able to differentiate between anomalous flows in irregular loads and gave statistical certainty against source addresses validated, without testing all the flows for detection through rules [17]. In hybrid systems, the most pragmatic neighbor-validation cryptographic methods, such as VBA and SecMac-SRD, provide short, signed tokens during RS/RA between neighbors and, in some cases, group authorization [19]. Using the SDN controller, it can then quickly process bulk validation and turn the result into dynamic flow rules [20]. Using this methodology, the SDN controller uses the AI Application Programmable Interface (API) as a pipeline that continuously learns new spoofing patterns and implements those into the policy, providing smooth updates. Consequently, the outcome is a fairly strong, adaptive, and easy-to-update approach to be used in dynamic IPv6 deployments.

3. RESEARCH METHODOLOGY

In this section the proposed enhanced IPv6 spoofing prevention framework is described, along with its implementation and evaluation. It describes the architectural design, data gathering and preprocessing, a fuzzy logic prevention layer, integration with SDN using the Ryu controller, and performance testing. Figure 1 shows the architecture of the system, which consists of three main layers: the infrastructure, control, and application layers. The infrastructure layer, created with Mininet, contains Host H1 (spoofed packet generator), Host H2 (destination), and an OpenFlow switch. Once spoofed IPv6 packets are transmitted, the switch will only forward the packets in case there is no drop rule. When no matching rule is found, a Packet-In event is sent to the Ryu controller. At the control layer, the Ryu controller (running `ipv6_filter.py`) takes key packet information like hop count and frame length and sends it up to the application layer. There, a fuzzy logic-based decision engine assigns a suspicion score. Packets considered very likely suspicious are dropped, while those with some level of uncertainty go through an ML classifier for further validation. Depending on the result, the controller can dynamically add flow rules to forward or block traffic in real time. This system can smartly detect and block IPv6 spoofed packets with minimal latency and great accuracy and provide adaptive and real-time protection from varying IPv6 spoofing attacks.

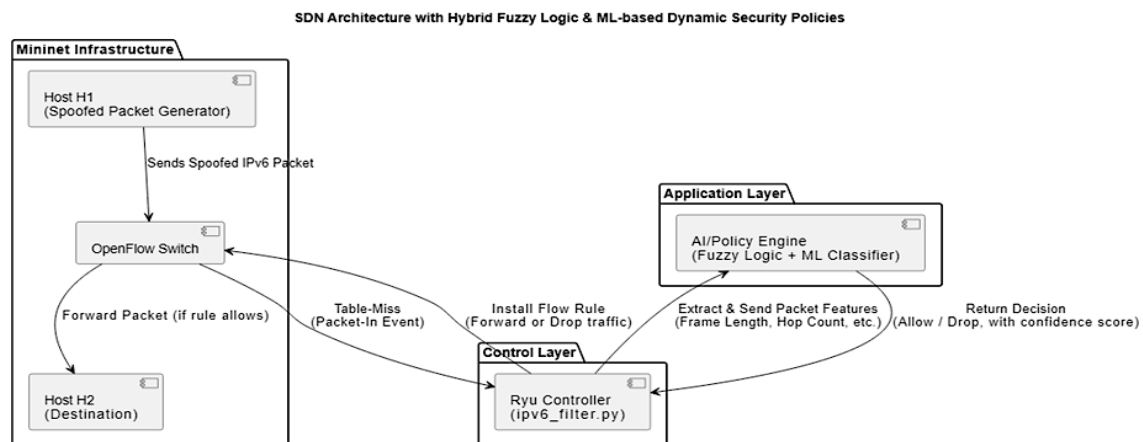


Figure 1. System Architecture Diagram

3.1 Data Collection and Preprocessing

For an accurate spoofing prevention system to be developed, a varied and well-prepared dataset is necessary. In this study, traffic was observed over a three-subnet network configuration with Mininet, along with each host configured inside a Docker container to represent various operating system environments. The dataset contained normal and malicious traffic. Legitimate flows contained Hypertext Transfer Protocol (HTTP), ICMPv6, and Domain Name System (DNS) queries, and the spoofed traffic contained alternating source IP addresses, hop counts, and extension headers. Firstly, packets were logged with complete information such as five-tuple fields, number of extension headers, fragmenting flags, and accurate timestamps. Secondly, a Python preprocessing pipeline scaled continuous features on the scale $[0, 1]$. And then, categorical fields were transformed using one-hot encoding in order to make the data machine readable. And for balanced learning purposes, the dataset was bootstrapped, which resulted in 50000 flows equally divided between legitimate traffic and spoofed traffic. Lastly, labeled data was divided into training (80%) and validation (20%) sets to develop the classifier. To ensure generalization and guard against overfitting, we also performed 5-fold cross-validation on the training dataset, yielding a mean F1-score of 0.96 ± 0.01 . Regularization (L2, $C = 1.0$) and early stopping were applied during model training. These steps guaranteed that several types of traffic behaviors were used, and the classifier learned to label spoofed packets with high certainty.

3.2 Fuzzy-Logic Prevention Layer Design

The first decision point of the proposed framework is the fuzzy-logic prevention module. Fuzzy logic, as opposed to binary logical systems that are based on strict true and false evaluations, works on degrees of suspicion, enabling the system to deal with vague or borderline cases much better. From incoming IPv6 packets, key packet features like hop count and frame length are extracted. These values are calculated by comparing against predefined membership functions; for example, a hop count of less than 20 or a frame length of more than 1500 bytes is highly suspicious, and mid-level values are moderately suspicious. These individual suspicion scores are then averaged to give an overall suspicion score between 0 and 1. In case the calculated suspicion score was greater than or equal to a threshold (e.g., 0.7), the system instantly installs a high priority drop rule through the SDN controller to block the source. Scores less than 0.4 are classified as benign and passed on as usual, and those with middle scores (0.4-0.7 scores) are passed to a Machine Learning (ML) classifier to be further processed. The 0.4 and 0.7 threshold values were empirically tuned on a hold-out validation set to balance FPs and FNs, following standard fuzzy logic tuning practices. Figure 2 illustrates the flowchart of the fuzzy logic prevention process, where packets are analyzed and the level of suspicion is computed, and traffic decisions are made. This layered strategy is designed for quick mitigation of definite threats as well as quality in indecisive scenarios. Moreover, its fuzzy logic module is tuned using real traffic data and dynamically adjustable thresholds and rule parameters, which means that it scales to emerging spoofing tactics.

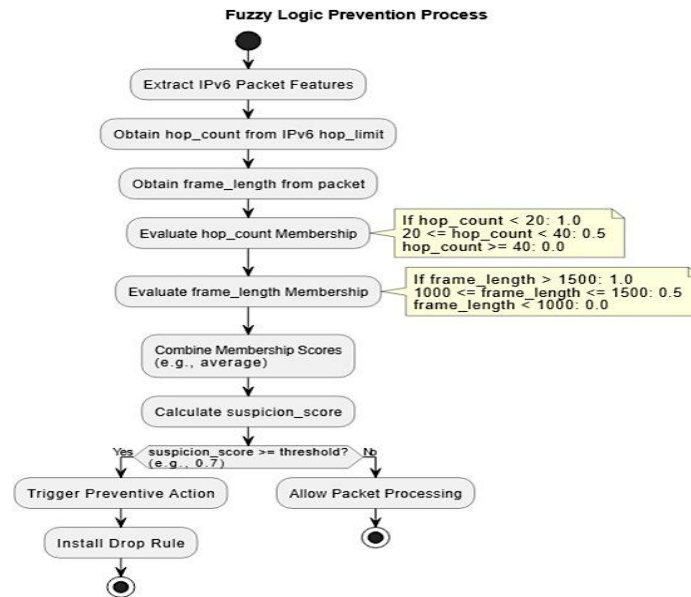


Figure 2. Flowchart of the Fuzzy Logic Prevention Process

3.3 Integration into the Ryu Controller

The proposed system is integrated with the Ryu SDN controller to develop real-time detection and mitigation of IP spoofing attacks through dynamic flow rule enforcement. Its integration is based on two Ryu-custom applications that enable packet inspection and policy enforcement via southbound and northbound APIs. The `packet_in_handler` listens for Packet In events from OpenFlow switches, extracts relevant IPv6 header fields using Scapy, and constructs a feature vector. This vector is then sent to the fuzzy-logic module through an internal API to evaluate the suspicion. When it receives a high-suspicion alert, the `flow_rule_manager` constructs OpenFlow 1.3 `OFPFlowMod` messages of type either drop or redirect with idle timeouts of 60 seconds and use priority based on the score. Additionally, the northbound REST interfaces make it possible to change the suspicion thresholds, as well as the rule presets, without the need to restart the controller, thus allowing for a more reactive policy adaptation based on observed traffic patterns. This integration enables the controller to be responsive to threats in real time without impacting ongoing network operations.

3.4 Evaluation Setup

In order to address the efficiency and reliability of the proposed IPv6 spoofing prevention system, the evaluation phase was conducted with four stages, each focusing on a specific functional component: component accuracy, response latency, detection reliability, and scalability. In the first stage, the module of fuzzy logic and ML is tested in terms of speed and accuracy to categorize the packets as spoofed or not. It is achieved through a labeled dataset consisting of normal and spoofed packets. In the second stage, the full system is checked for its speed. This includes measuring the time between the packet arrival at the OpenFlow switch and the moment the SDN controller installs a rule to drop or allow that packet. In the third stage, the ability of the system is measured by detecting spoofed traffic without errors. This includes a measure of how many spoofed packets are detected (True Positives), how many good packets are incorrectly detected (False Positives), and accuracy. The fourth and final stage is a stress test to see the performance of the system under heavy traffic. It identifies the CPU usage and memory consumption of the controller, how many packets it can process at a time, and what the time it takes to update flow rules under an increase of malicious traffic.

Table 2 shows the summarization of the several evaluation steps to support these four stages. While Mininet/Docker provides a flexible emulation environment, it may not fully capture all aspects of production-scale IPv6 networks. Future work will validate the framework in real-world deployments to assess performance under diverse traffic patterns and hardware constraints.

Table 2. Each Evaluation Step and its Description

Evaluation Step	Description
Spoofed Packet Injection	A Python script on host h1 generates spoofed IPv6 packets with engineered anomalies (e.g., unusually low hop count and high frame length) to simulate an attack scenario
Real-Time Monitoring	The Ryu controller logs are reviewed to ensure that the fuzzy logic module correctly computes the suspicion score and triggers prevention actions as intended
Hybrid Decision Verification	In ambiguous cases, the fallback ML classifier is evaluated to verify that it accurately identifies spoofed packets, ensuring that no false negatives or positives are overlooked
Performance Analysis	Metrics such as reaction time (speed between detection and flow rule installation), FP/ FN rates, and overall network impact are captured and analyzed to assess performance.

It outlines each evaluation activity, including crafted spoofed packet injection, live monitoring of suspicion scores from the controller, validation of ambiguous cases through the ML fallback, and analysis of performance metrics such as accuracy and reaction time.

All these evaluation steps together provide a complete picture of how well the proposed framework performs to mitigate the IPv6 spoofing in terms of effectiveness, responsiveness, and resource efficiency.

3.5 Evaluation Metrics

Two core metrics, called Prevention Accuracy (P) and Average Reaction Time (ART), are used to measure the effectiveness and responsiveness of the proposed framework. These metrics ensure the reliability of the system to detect and respond to spoofed packets in real time while keeping legitimate traffic. Prevention Accuracy (P) evaluates how successfully the system blocks malicious traffic (1). It is defined as the percentage of spoofed packets that were detected and dropped by the controller among all the spoofed packets sent during the tests. It can be expressed in Equation (1).

$$P(\%) = \frac{(\text{Number of Correctly Blocked Spoofed Packets})}{\text{Total Spoofed Packets}} \times 100\% \quad (1)$$

A higher P indicates a better capacity for detection. In practical terms, this value indicates how well the decisions were made under live traffic by the fuzzy logic layer and the fallback ML classifier.

ART is an indicator of the system's timeliness in reacting to a suspicious packet arrival (2); specifically, it measures the time delay between the packet arrival at the OpenFlow switch and the deployment of the respective SDN drop rule on the controller. It is computed as a function of n packets, as in Equation (2).

$$\text{Average Reaction Time} = \frac{1}{n} \sum_{i=1}^n \text{Reaction Time}_i \quad (2)$$

Low ART reflects fast mitigation, which is very important to stop high-speed spoofing or flooding attacks. This is done at three different points: when packets are received at the switch, when the score of suspicion is calculated, and when the drop rule is enforced. This allows accurate profiling of latency. These two measures target the main goals of the system, which are accuracy and speed, but can also be used for comparisons with traditional static filtering systems. Future work will aim at having better-adjusted detection thresholds in order to enhance precision and recall, especially in conditions that are more large-scale or resource-constrained networks.

3.6 Scalability and Resource Analysis

System scalability was also measured by raising the rate of spoofed packet injections and seeing the behavior of the system under stress load. One such stress test involved injecting 500 spoofed IPv6 packets. The entire processing took 34.44 seconds, resulting in a throughput of 14.52 packets/second and very low CPU utilization (0.0%) with an average memory utilization of 19.9%. In comparison, the traditional filtering system had a better throughput of 31.09 packets/second, but its CPU usage was slightly larger (0.3%) and almost the same memory utilization (19.8%). Nevertheless, it had an accuracy of only 60% in detection, which also shows that it is less effective during similar traffic conditions.

Table 3 compares the performance between integrated systems and traditional filtering, mainly in detection accuracy, reaction time, throughput, and resource consumption. This comparison suggests that the integrated system is a reasonable compromise between accuracy and efficiency. It comes at a cost of slightly increased processing time and reduced throughput but offers highly improved spoofing prevention with only minimal usage of CPU and memory resources. This trade-off proves the adaptability of the system to real-time IPv6 security on enterprise and resource-constrained networks.

Table 3. Performance Comparison Between Integrated System and Traditional Filtering

Metric	Traditional Filtering	Integrated System
Detection Accuracy	60%	80%
ART	0.0322 seconds	0.3735 seconds
Throughput	31.09 packets/second	14.52 packets/second
CPU Usage	0.3%	0%
Memory Usage	19.8%	19.9%

4. RESULTS AND DISCUSSION

To test the effectiveness of the proposed framework to prevent IPv6 spoofing, a number of controlled tests were carried out under different network conditions. Both normal and spoofed IPv6 traffic at different intensity levels were used in these tests, measured in three categories as low, medium, and high. Each of these scenarios was repeated 20 times to be consistent and valid statistically. The classification performance of the system was assessed by using standard ML metrics, including precision, recall, and F1-score.

Table 4 describes the formulas to calculate the prevention performance metric. Precision indicates the number of packets marked as spoofed that are actually spoofed, while recall indicates the number of spoofed packets the system prevented. The F1-score is the harmonic mean of precision and recall, which is a single measurement that balances both accuracy and completeness of the system performance.

Table 4. Computations Used to Measure Prevention Performance

Metric	Formula	Description
Precision	$\frac{TP}{TP + FP}$	Indicates how many spoofed packets were correctly identified as malicious.
Recall	$\frac{TP}{TP + FN}$	Measures of how many spoofed packets were successfully prevented by the system.
F1-Score	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	Provides a harmonic means of precision and recall to balance both metrics in a single performance measure.

Table 5 presents the detection performance of the proposed IPv6 spoofing prevention system across 20 test runs under varying traffic conditions. All precision and recall values are reported as mean $\pm$ standard deviation over 20 runs. Paired t-tests were conducted against a baseline static-ACL filter; improvements in both precision ($p < 0.01$) and recall ($p < 0.01$) were statistically significant. In this context, a True Positive (TP) refers to packets correctly identified as legitimate packets in normal traffic or spoofed packets in attack scenarios. A False Positive (FP) indicates normal traffic mistakenly flagged as malicious, while a False Negative (FN) represents spoofed packets that were not detected. Under low traffic conditions for both normal and attack types, the system achieved perfect detection with precision, recall, and F1-score all at 1.00. As traffic intensity increased, minor misclassifications occurred. For example, in high normal traffic, one legitimate packet was missing, leading to a recall of 0.95. In medium and high attack scenarios, although all spoofed packets were successfully identified (recall of 1.00), a few benign packets were incorrectly flagged, slightly reducing precision. Overall, the average precision and recall remained high at 0.98, with an F1-score of 0.97, demonstrating the system's robustness and accuracy even in dynamic network conditions.

Table 5. Results Summary Across 20 Test Runs

Traffic Type	Scenario	TP	FP	FN	Precision	Recall	F1-Score
Normal Traffic	Low	20	0	0	1.00	1.00	1.00
	Medium	18	0	2	1.00	0.90	0.95
	High	19	0	1	1.00	0.95	0.97
Attack Traffic	Low	20	0	0	1.00	1.00	1.00
	Medium	20	2	0	0.91	1.00	0.95
	High	20	1	0	0.95	1.00	0.97
	Average				0.98	0.98	0.97

TP = True Positive, FP = False Positive, FN = False Negative

Figure 3 provides a visual comparison of precision, recall, and F1-score under both normal and attack traffic scenarios. The system consistently maintains high detection performance across different traffic intensities. For normal traffic, the system accurately recognizes legitimate flows with near-perfect precision and recall. In spoofed traffic cases, it achieves perfect recall across all conditions, indicating no spoofed packets were missed. However, a slight drop in precision under medium and high attack scenarios is observed, where some legitimate packets are misclassified as spoofed. This is likely due to overlapping feature patterns in high-volume mixed traffic. Despite this, the F1-scores

remain high, confirming the framework's robustness and balanced detection capability. These findings demonstrate the system's ability to achieve strong spoofing prevention with minimal FPs, making it suitable for real-time deployment in both enterprise and resource-constrained IPv6 environments.

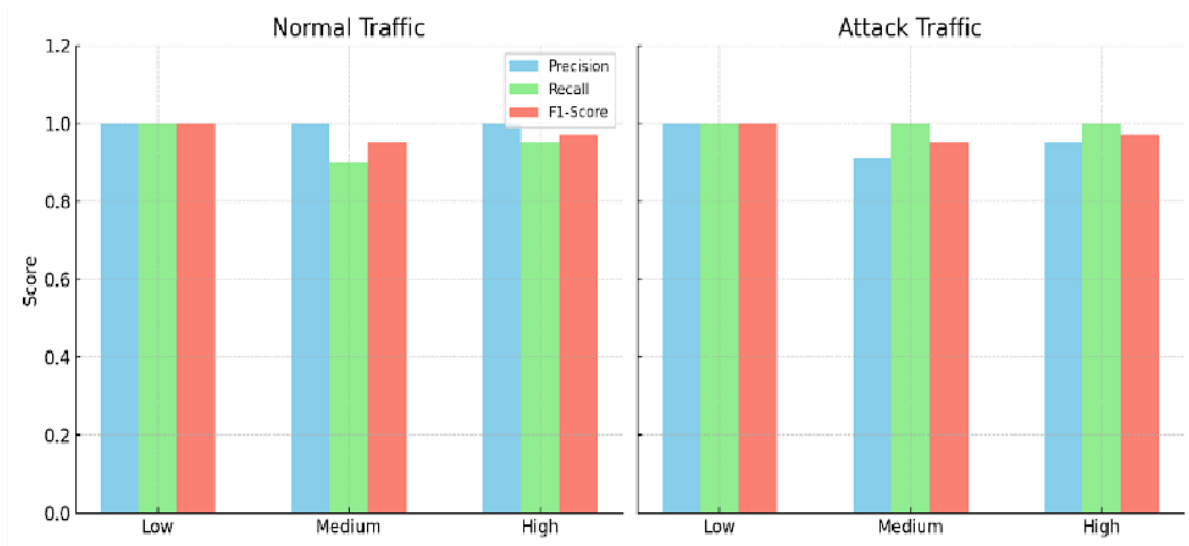


Figure 3. Performance Metrics Across Normal and Attack Traffic Scenarios

5. CONCLUSION

In summary, the multi-layered IPv6 spoofing prevention architecture includes coarse-grained ingress/egress filtering, a fuzzy-logic front-end system, a machine-learning classifier, and policy enforcement through SDN. The fuzzy layer generates hop-limit and frame-length suspicion scores; packets above a 0.7 score are dropped instantaneously, and ambiguous flows or suspicious flows with a score between 0.4 and 0.7 are forwarded to an ML model that had been created with a training set of 50,000 balanced flows. The SDN controller, using Ryu and OpenFlow 1.3, dynamically adds drop or forward rules, which isolate malicious sources, with an ART of less than 0.4 s. The experimental study conducted in a Mininet/Docker environment found a detection precision and recall rate of 0.98, an F1-score of 0.97, and resource overhead was negligible compared with static ACLs. The integrated approach achieved high detection with more than 14.5 packets/s, in comparison with traditional filters that were removing 60% of the spoofing attempts at a maximum of 31 packets/s. In terms of resources, the CPU usage was consistently at less than 0.3%, and memory was around 20%. The modular design enables refinement of thresholds via northbound APIs, facilitating adaptation to new spoofing techniques and easy integration into the context of an existing network. These findings support the viability of the ML and SDN combination for timely and efficient IPv6 security with minimal resources and eventually include deployment at scale or in resource-limited conditions, with an added opportunity to adjust detection thresholds. Future research acquisitions will examine performance with heterogeneous network implementations and classifiers for scalability. Future work will extend evaluation to heterogeneous hardware testbeds and production networks, perform sensitivity analysis on fuzzy-logic membership functions, and explore dynamic threshold adaptation under varying traffic loads.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for the suggestions to improve the paper.

FUNDING STATEMENT

The authors received no funding from any party for the research and publication of this article.

AUTHOR CONTRIBUTIONS

Navaneethan C. Arjuman: Project Administration, Supervision, Writing – Review & Editing;
Ahmad Fikri Bin Sharudin: Conceptualization, Data Curation, Methodology, Validation, Writing – Original Draft Preparation;
Zar Chi Hlaing: Project Administration, Writing – Review & Editing;
Supriyanto Praptodiyono: Project Administration.

CONFLICT OF INTERESTS

No conflict of interests were disclosed.

ETHICS STATEMENTS

This research did not involve any human participants, animal subjects, or data collected from social media platforms. Therefore, ethical approval and informed consent were not required.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

- [1] Y. Almukhtar and N. Ahmed, "IPv4 address exhaustion: Mitigation strategies, architectural trade-offs, and IPv6 transition challenges," *AlQalam Journal of Medical and Applied Sciences*, pp. 1522–1526, 2026, doi: 10.54361/ajmas.269610.
- [2] O. A. Akano, T. O. Olayinka, O. D. Adeniji, and B. O. Ogunjinmi, "Mitigating insider threat's IP spoofing through enhanced dynamic cluster algorithm (EDPU Based HCF)," *AIR*, vol. 25, no. 3, pp. 85–90, Mar. 2024, doi: 10.9734/air/2024/v25i31052.
- [3] N. C. Arjuman, M. U. B. M. Yusof, Z. C. Hlaing, and S. Salih, "Enhanced mechanism for neighbor discovery protocol table exhaustion attacks in IPv6 networks," in *TENCON 2025 - 2025 IEEE Region 10 Conference (TENCON)*, 2025, pp. 1773–1777, doi: 10.1109/TENCON66050.2025.11374974.
- [4] V. Boopathi, A. Balamurugan, M. Deenan, and A. Hemlathadhevi, "A behavioral-based MAC spoofing detection system," *International Journal of Applied Smart Interdisciplinary Technologies (IJASIT)*, vol. 1, no. 1, pp. 49–55, 2026. Available: <https://ijasit.org/index.php/home/article/view/9>.
- [5] F. Najjar, Q. Bsoul, and H. Al-Refai, "An analysis of neighbor discovery protocol attacks," *Computers*, vol. 12, no. 6, pp. 125, Jun. 2023, doi: 10.3390/computers12060125.
- [6] R. Kartheek, B. S. K. Raju, D. J. Sonia, and P. Gopinadh, "Classifying IP spoofing requests using heuristic fuzzy computer network approach," *International Journal for Multidisciplinary Research*, vol. 6, no. 2, Mar. 2024, doi: 10.36948/ijfmr.2024.v06i02.15111.
- [7] M. N. Hoda, Bharati Vidyapeeth's Institute of Computers Applications and Management, and Institute of Electrical and Electronics Engineers, Eds., *Proceedings of the 17th INDIACom; 2023 10th International Conference on Computing for Sustainable Global Development (15th-17th March, 2023): INDIACom-2023*. Piscataway, NJ: IEEE, 2023.
- [8] M. H. Ali and S. Al-Alak, "Using unique node ID to control IPv6 ID spoofing," *Journal of Physics Conference Series*, vol. 1999, no. 1, pp. 012120, 2021, doi: 10.1088/1742-6596/1999/1/012120.

-
- [9] J. S. Nixon and M. Amenu, "Investigating security issues and preventive mechanisms in IPv6 deployment," *International Journal of Advanced Engineering and Nano Technology*, vol. 9, no. 2, pp. 1–20, 2022, doi: 10.35940/ijaent.b0466.029222.
- [10] M. U. Shreehitha and K. V., "Comparative analysis of routing protocols and IP traffic management using ACL," *International Journal of Engineering Applied Sciences and Technology*, vol. 6, no. 5, 2021, doi: 10.33564/ijcast.2021.v06i05.014.
- [11] M. A. Al-Shareeda, S. Manickam, M. A. Saare, and N. C. Arjuman, "Proposed security mechanism for preventing fake router advertisement attack in IPv6 link-local network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 29, no. 1, pp. 518, 2022, doi: 10.11591/ijeecs.v29.i1.pp518-526.
- [12] A. Al-Ani, A. K. Al-Ani, S. A. Laghari, S. Manickam, K. W. Lai, and K. Hasikin, "NDPsec: Neighbor discovery protocol security mechanism," *IEEE Access*, vol. 10, pp. 83650–83663, 2022, doi: 10.1109/ACCESS.2022.3196028.
- [13] Z. T. Puhl and J. Guo, "Securing IPv6 neighbor discovery address resolution with voucher-based addressing," *Network*, vol. 4, no. 3, pp. 338–366, Aug. 2024, doi: 10.3390/network4030016.
- [14] M. Monnich, N. S. Bulbul, D. Ergenc, and M. Fischer, "Mitigation of IPv6 router spoofing attacks with P4," in *Proceedings of the Symposium on Architectures for Networking and Communications Systems*, ACM, Dec. 2021, pp. 144–150. doi: 10.1145/3493425.3502765.
- [15] H. Nasser and M. A. Hussain, "An effective approach to detect and prevent ARP spoofing attacks on WLAN," *Iraqi Journal for Electrical and Electronic Engineering*, vol. 19, no. 2, pp. 8–17, 2023, doi: 10.37917/ijeee.19.2.2.
- [16] S. Kianpisheh and T. Taleb, "A survey on in-network computing: Programmable data plane and technology specific applications," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 701–761, 2023, doi: 10.1109/comst.2022.3213237.
- [17] A. Karakoc and F. Alagoz, "LSAV: Lightweight source address validation in SDN to counteract IP spoofing-based DDoS attacks," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 31, no. 7, pp. 1187–1205, Nov. 2023, doi: 10.55730/1300-0632.4042.
- [18] A. A. Bahashwan, M. Anbar, S. Manickam, T. A. Al-Amiedy, M. A. Aladaileh, and I. H. Hasbullah, "A systematic literature review on machine learning and deep learning approaches for detecting DDoS attacks in software-defined networking," *Sensors*, vol. 23, no. 9, pp. 4441, 2023, doi: 10.3390/s23094441.
- [19] N. C. Arjuman, S. Manickam, and S. Karuppayah, "An improved secure router discovery mechanism to prevent fake RA attack in link local IPv6 network," *Communications in Computer and Information Science*, pp. 248–276, 2021, doi: 10.1007/978-981-16-8059-5_15.
- [20] J. Wang, L. Wang, and R. Wang, "A method of DDoS attack detection and mitigation for the comprehensive coordinated protection of SDN controllers," *Entropy*, vol. 25, no. 8, pp. 1210, 2023, doi: 10.3390/e25081210.

BIOGRAPHIES OF AUTHORS

	Navaneethan C. Arjuman is currently a Senior Lecturer at Multimedia University (MMU). He holds a PhD in the field of Plug-and-Play Networking Components (IPv6 Cyber Security) from the National Advanced IPv6 Centre, University Science Malaysia. He holds a 1st Class Honours degree in Communication and Signal Processing from Staffordshire University in the United Kingdom. His research interests are in the areas of AI, cyber security, IPv6, 5G, and IoT. He can be contacted via email: navaneethan@mmu.edu.my.
	Ahmad Fikri Bin Sharudin is a final-year Cybersecurity student at Multimedia University, driven by a commitment to excel in the rapidly evolving security landscape. With core interests in blockchain technology, cloud security, and network security, Fikri has honed skills in secure architecture design and threat analysis. Aiming to specialize in smart contract auditing and blockchain engineering, Fikri actively engages in cybersecurity competitions and workshops, continuously expanding expertise. Post-graduation, Fikri aspires to contribute advanced auditing solutions that enhance trust and resilience in decentralized systems. He can be contacted via email: 1221302093@student.mmu.edu.my.
	Zar Chi Hlaing is currently pursuing a Master of Computing at Multimedia University (MMU), Malaysia. She is the holder of a Bachelor of Engineering in Electronics major, Thanlyin Technological University, Myanmar. Her research focuses on 5G networks, network automation, and IPv6 security. She is passionate about building automated systems in networking to improve efficiency, adaptability, and real-time security. She aspires to become an expert scientist and a leading researcher in next-generation networking technologies and intelligent systems. She can be contacted via email: zar.chi.hlaing@student.mmu.edu.my.
	Supriyanto Praptodiyono is currently a Professor at Universitas Sultan Ageng Tirtayasa (Untirta). Since 2024, he assigned as Dean at Faculty of Computer Science, UPN Veteran Jakarta. He holds MSc and PhD in the field of Advanced Networking (IPv6 Security) from the National Advanced IPv6 Centre, Universiti Sains Malaysia (USM). His research interests are in the areas of cyber security, IPv6, and IoT. He can be contacted via email: supriyanto.fik@upnvj.ac.id.