

Design and Development of a Lightweight Microcontroller-Assisted SDN Framework for Securing IoT Devices in Smart Homes

Hassan Abdul Ghaffar

Department of Computer Science, FAST National University of Computer and Emerging Sciences (FAST-NUCES), Islamabad, Pakistan

Muhammad Shoaib Hanif

Department of Computer Science, University of the Punjab, Lahore, Pakistan

Abstract

Background: Consumer IoT devices are increasingly deployed in smart homes but are rarely designed with security as a priority, leaving them exposed to unauthorized access and volumetric attacks. Existing SDN-based defenses depend on Open Flow-compliant gateways that most home routers lack, while authentication and encryption remain immature for resource-constrained devices. **Objective:** This study presents a universal, Open Flow-independent framework that restricts unauthorized access to smart-home IoT devices. **Methodology:** A microcontroller is positioned between the home gateway and the wider network, monitoring flows and forwarding trace files to a cloud-based SDN controller; the controller returns updated policies through a northbound interface, and a fractional-selectivity value flags and blocks volume-based attacks. The framework was implemented in CloudSimSDN across one-to-six-switch topologies and benchmarked against the low-cost flow-based solution (LCFS) of Sivanathan et al. **Key Results:** The framework achieved a lower average data-plane load (4.01 Mbps) than the LCFS baseline (9.65 Mbps), with reduced IoT traffic, camera-module load, CPU, RAM, and energy consumption across all configurations. **Main Contributions:** (i) an Open Flow-independent, microcontroller-augmented SDN architecture; (ii) a fractional-selectivity mechanism for flow-level attack detection; (iii) a CloudSimSDN evaluation benchmarked against an established baseline. **Conclusion:** A lightweight microcontroller paired with SDN can restrict unauthorized IoT access at lower resource cost than Open Flow-dependent alternatives, offering a practical route to scalable smart-home security.

Keywords: CloudSimSDN; Flow-Based Detection; Microcontroller; Smart Home Security; Software-Defined Networking

1. Introduction

The Internet of Things (IoT) connects computing, mechanical, and digital devices across agriculture, infrastructure, healthcare, and, most intimately, the household, where smart devices automate routine tasks and extend remote control over appliances [7], [11], [18]. This pervasiveness has not been matched by comparable maturity in device-level security. Roman et al. [24] and Sicari et al. [28] both observe that IoT security concerns extend beyond individual devices to the networks, platforms, and protocols that support them, implicating manufacturers as much as end users. Gartner estimates that more than a quarter of enterprise attacks will involve IoT components, while only a small share of security budgets addresses them [17]; [27] note that this gap is especially pronounced in the smart-home context.

The consequences are well documented: the 2016 Mirai botnet alone compromised over 2.5 million devices [28], and smart-home ecosystems remain vulnerable to Man-in-the-Middle attacks, data theft, and Denial-of-Service (DoS) campaigns. Conventional countermeasures-encryption, VPN tunnelling, network isolation-mitigate some risk but remain insufficient given the limited computation, memory, and energy budgets typical of consumer IoT hardware [26].

Software-Defined Networking (SDN) offers a promising response: by decoupling the control and data planes behind an open, programmable interface, it enables centralized, real-time enforcement of dynamic security policies [19]. Applied

to smart homes, SDN lets traffic be monitored, filtered, and redirected by centrally defined rules rather than fixed, manufacturer-hardcoded firmware.

1.1 Current State of Research and Existing Challenges

Most SDN-based IoT security solutions presuppose OpenFlow or OpenWrt compatibility [6], [8], [26], [31], [36], [38], since SDN-enabled switches must support OpenFlow to exchange state with a controller [3], [12], [25], [37]-rendering these approaches unusable on most commodity home routers. Authentication and encryption for low-power devices remain immature, and manufacturers' hardcoded credentials further weaken protection [16]. Flow-based detection is more promising but still needs integration with hardware-agnostic control infrastructure [22], [29], [30], [32]. Douha et al. [44] confirm that SDN-centric defenses remain among the most active directions in this space, while Gordon et al. [45] show that pairing SDN with lightweight ML classifiers achieves high-accuracy device identification and DDoS detection on commodity Open vSwitch infrastructure.

1.2 Research Gap and Motivation

Home gateways mediate an intimate but exposed relationship between users and their devices: the same connectivity that enables remote access also invites exploitation. Sivanathan et al. [31] demonstrate the value of monitoring traffic flow rather than packet content, but their approach depends on Open Flow-enabled routers absent from most homes. This, combined with the storage and power constraints of typical gateways, motivates a solution that is lightweight, scalable, customizable, and Open Flow-independent.

1.3 Research Objectives and Contributions

This research aims to design, implement, and evaluate a smart-home security framework that restricts unauthorized access to IoT devices by combining SDN technology with a microcontroller-based enforcement point. The specific objectives are to:

- Introduce a universal, easy-to-deploy security mechanism that does not require replacement of existing network hardware and that offers built-in scalability and customization.
- Restrict remote access to IoT devices according to centrally defined policies, and evaluate the resulting accuracy, anomaly-detection capability, and resource performance of the deployment.

The principal contributions of this work are: (i) an architecture that decouples smart-home traffic filtering from OpenFlow dependency by introducing a microcontroller as an intermediary enforcement point between the gateway and the SDN controller; (ii) a flow-level, fractional-selectivity-based mechanism for identifying and blocking volume-based attacks with minimal processing overhead; and (iii) a CloudSimSDN-based simulation and evaluation, benchmarked directly against the low-cost flow-based solution (LCFS) of Sivanathan et al. [31], across energy consumption, CPU usage, RAM consumption, and traffic-load metrics.

1.4 Paper Organization

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature on IoT security, Open Flow-based SDN deployments, microcontroller-based gateways, and the CloudSimSDN simulation environment, and presents a comparative summary of existing approaches. Section 3 formalizes the research gap. Section 4 details the proposed methodology, system architecture, algorithms, and simulation setup. Section 5 presents and discusses the results in comparison with the baseline model. Section 6 concludes the paper, and Section 7 outlines directions for future work.

2. Literature Review

This section surveys state-of-the-art approaches to securing IoT devices in smart-home and related environments, organized around four thematic areas: general IoT security, OpenFlow-based SDN deployments, microcontroller-based IoT gateways, and the CloudSimSDN simulation platform used for evaluation in this study.

2.1 IoT Security

Rizvi et al. [23] frame IoT security as a multi-faceted problem spanning device security, data security, and storage, noting that the mobility, heterogeneity, and scale of IoT deployments make holistic protection difficult to achieve. Hassan et al. [16] trace the evolution of IoT security research between 2016 and 2018, cataloguing unsecured communication channels, malicious network activity, and physically unsecured devices as persistent sources of vulnerability, and observing that authentication and encryption remain immature for resource-constrained hardware.

Sivaraman et al. [32] address the heterogeneity of device functionality and manufacturer practice by proposing network-level security enforced through an SDN-based Security Management Provider (SMP). Their three-part architecture, comprising an access-switch layer (Open vSwitch), a Floodlight controller, and a Ruby-on-Rails security orchestrator,

allows dynamic, context-aware rules that can quarantine misbehaving devices without requiring manufacturer intervention. Building on this direction, Sivanathan et al. [31] shift from packet-based to flow-based monitoring, introducing an Analysis Engine that interacts with the SDN controller through northbound APIs to selectively mirror suspicious flows for inspection, thereby limiting bandwidth and processing overhead. This flow-based methodology, validated on a home-network testbed using OpenWrt-compatible gateways over an 18-minute capture window, forms the direct baseline against which the present study is evaluated.

Sivanathan et al. [29], [30] extend this line of work through machine-learning classification of IoT devices based on statistical traffic attributes such as activity cycles, port usage, and cipher suites, achieving 95% accuracy in distinguishing IoT from non-IoT traffic across campus-scale deployments. Serror et al. [26] propose an in-network security model that specifies the minimum communication behaviour required for each device's intended functionality, filters traffic accordingly at a central router, and layers anomaly detection on top; a key limitation of this design is its assumption that all traffic traverses a single central router, leaving off-router channels such as Bluetooth outside its protective scope.

Several studies examine SDN-based gateways directly. Bull et al. [8] deploy a POX controller as an SDN-enabled gateway capable of actively monitoring and blocking TCP and ICMP flood attacks, though without support for integrating external dynamic-analysis tools. Bhunia and Gurusamy [6] propose SoftThings, a hierarchical Master/Cluster SDN controller architecture augmented with machine-learning-based anomaly detection, validated via Mininet simulation. Demetriou et al. [10] take a complementary, application-layer approach, using a smartphone-based monitor to inform SDN-driven access decisions enforced at the router. Wang et al. [35] classify smart-home devices into external-service, controller, and smart-device layers based on MAC address and HTTP request signatures to constrain unnecessary inter-device routing.

Smaller-scale integrations of SDN with physical sensor networks are reported by Zaied et al. [40], who link an Arduino/Raspberry Pi-based Wireless Sensor Network to a Mininet/OpenDaylight SDN topology, and by Al-Kassem et al. [3], who deploy a customized SD-NFV controller for 6LoWPAN M2M sensor gateways. Both studies confirm the feasibility of pairing lightweight embedded hardware with SDN control, though at limited scale and duration. At a broader level, Chaabouni et al. [9], Neshenko et al. [21], and Alladi et al. [4] provide surveys and taxonomies of IoT threats, vulnerabilities, and defence techniques, consistently identifying storage, power, and computational constraints as structural barriers to conventional security mechanisms, and calling for lightweight, network-level alternatives.

2.2 Open Flow-Based SDN Deployments

A parallel strand of research investigates the performance and applicability of Open Flow as the enabling protocol for SDN-based IoT security. Wrona et al. [36] present a low-cost Open Flow testbed for validating cross-layer, data-centric security policies in military systems, concluding that physical testbeds are well suited to evaluating the temporal characteristics of security mechanisms. Xiong et al. [37] analyses OpenFlow-based SDN performance using a queuing model, reporting that publicly available OpenFlow controllers such as NOX and Beacon become ineffective at scale and that every switch in the network must maintain direct or indirect connectivity to the controller. Frank and Ghaderi [12] propose a Flow Policy Enforcer to secure home networks over OpenFlow-enabled access points, while Sanger et al. [25] benchmark packet-in/packet-out latency across five hardware switches, finding inconsistent support for OpenFlow feature sets even among compliant hardware. Baddeley et al. [5] address a related constraint specific to low-power multi-hop wireless networks, proposing a lightweight SDN framework with IPv6 and underlying-routing-protocol interoperability, together with optimization techniques that reduce control overhead, achieving lower latency and jitter while providing redundancy to priority flows; their work illustrates that OpenFlow-style centralized control can be adapted for constrained environments even where full protocol compliance remains impractical. Collectively, this body of work substantiates the concern, echoed throughout the IoT security literature, that Open Flow dependency constrains deploy ability on the commodity hardware typically found in smart homes. Mostovich et al. [20] trace many of these vulnerabilities to their root causes, attributing the majority of high-level SDN vulnerabilities to Open Flow protocol errors, misconfigurations, and insecure APIs concentrated at the control and data-plane layers, and recommend integrating dedicated Software-Defined-Security (SDsec) subsystems into Software-Defined-IoT deployments as a mitigation.

Beyond OpenFlow-specific deployments, Yassein and Aljawarneh [39] survey solutions combining SDN and IoT across multiple application domains, comparing candidate protocols and operating systems for SDN-based IoT and concluding that no single combination of technologies dominates across use cases; this comparative perspective reinforces the motivation for the domain-agnostic, OpenFlow-independent design pursued in the present study.

2.3 IoT Gateways Based on Microcontrollers

A microcontroller is a compact, single-chip integration of a processor core, memory, interrupt controller, and debugging unit, valued for its low cost, reduced power consumption, and adaptability across embedded applications [13]. Grygoruk and Legierski [14] implement an Arduino-based IoT gateway for home environments, collecting sensor data and interfacing with web services without a dedicated network adapter. While well suited to small numbers of concurrent users, the authors report both instability under higher load and susceptibility to Man-in-the-Middle attacks, underscoring the need for an accompanying network-level security mechanism—precisely the gap the present study addresses by pairing a microcontroller with SDN-based policy enforcement rather than relying on the microcontroller alone.

2.4 CloudSimSDN

Son et al. [33] introduce CloudSimSDN as an extension of the CloudSim framework, providing a lightweight, scalable environment for simulating SDN-enabled cloud data centres without the cost of physical infrastructure. The toolkit supports modelling of hosts, switches, and network-aware virtual-machine placement policies, and has since been extended by Son et al. [34] to simulate network function virtualization across edge and cloud environments. CloudSimSDN was selected for the present study over alternatives such as Mininet because it supports policy modelling and resource-management evaluation in addition to network-topology emulation, capabilities that a pure network emulator does not provide.

2.5 Comparative Summary of Existing Approaches

Table I consolidates the approaches reviewed above, highlighting the method, evaluation context, principal results, and limitations reported by each study, and situates the proposed framework relative to this body of work. The comparison shows that the great majority of existing solutions either presuppose OpenFlow/OpenWrt-compatible infrastructure or validate their approach only at small scale (a handful of sensors or a short capture window). The framework proposed in this paper is distinguished by its explicit removal of OpenFlow dependency, its use of a microcontroller as a lightweight intermediary rather than a full SDN-compliant switch, and its direct, multi-metric benchmarking against an established flow-based baseline [31].

Table I. Comparative Summary of Reviewed Literature

Author(s) & Year	Method / Approach	Dataset / Testbed	Key Results	Limitations
Sivaraman et al. [32], 2015	SDN-based Security Management Provider (SMP); three-part architecture (Open vSwitch, Floodlight, Ruby-on-Rails orchestrator)	5 smart-home devices; university data-centre controller + Amazon cloud SMP	Demonstrated third-party, network-level security-as-a-service model with dynamic, context-aware rules	Dependent on OpenFlow-compliant switches; relies on external cloud SMP, raising latency and trust concerns
Sivanathan et al. (Gharakheili et al.) [31], 2016	Flow-based (not packet-based) monitoring via an Analysis Engine interacting with SDN controller through northbound APIs	Campus/home testbed with OpenWrt gateway; ~18-minute traffic capture	Selective mirroring of flows reduced inspection overhead versus full packet inspection	Requires OpenFlow/OpenWrt-compatible gateway; limited to header-level inspection; short capture window
Sivanathan et al. [29], 2018	Machine-learning classification of IoT devices using statistical traffic attributes (activity cycles, ports,	28 IoT devices over 6 months	Reliable device identification from network behaviour	Focused on classification rather than active access control or attack

Author(s) & Year	Method / Approach	Dataset / Testbed	Key Results	Limitations
	cipher suites)			mitigation
Sivanathan et al. [30], 2017	Traffic classification using the same statistical-attribute approach applied to a smart-campus environment	21 IoT devices, smart-campus network	95% accuracy distinguishing IoT from non-IoT traffic	Campus-scale deployment; not evaluated for resource-constrained home gateways
Serror et al. [26], 2018	In-network security combining device-behaviour specification, traffic filtering, and anomaly detection	Conceptual/home-router-centric architecture	Restricts device communication to the minimum required functionality	Assumes all traffic passes through a single central router; cannot detect off-router (e.g., Bluetooth) attacks
Bull et al. [8], 2016	SDN-enabled POX controller acting as gateway for active traffic monitoring	Home-network testbed	Successfully blocked TCP and ICMP flood attacks	No support for integrating dynamic traffic-analysis tools with the controller
Bhunja & Gurusamy [6], 2017 (SoftThings)	Hierarchical Master/Cluster SDN controllers with machine-learning anomaly detection	Mininet-based simulation	Detected and mitigated distributed attacks at the network level	Multi-controller hierarchy adds architectural complexity; simulation-only validation
Zaied et al. [40], 2017	Integration of Wireless Sensor Network (Arduino/Raspberry Pi) with a Mininet/OpenDaylight SDN topology	2-sensor small-scale testbed; 10-minute capture	Demonstrated feasibility of linking a physical WSN with a virtual SDN topology	Very small scale (2 sensors, 10 minutes); needs an additional security application above the controller
Al-Kassem et al. [3], 2017	Customized SD-NFV controller for 6LoWPAN M2M sensor gateways using Arduino	6LoWPAN testbed with open hardware boards	Improved programmability and management of cloud-based 6LoWPAN gateways	Purpose-built for 6LoWPAN M2M; limited generalizability to heterogeneous smart-home devices
Wang et al. [35], 2019	Layered SDN framework classifying devices into external, controller, and smart-device layers by MAC/HTTP signatures	Home-network conceptual model	Reduced unnecessary inter-device routes and device-type complexity	Classification logic dependent on MAC address and HTTP request patterns, which can be spoofed
This study (proposed)	SDN controller augmented with a	CloudSimSDN simulation, I–6	Lower data-plane load (4.0I vs	Header/flow-level inspection only (no

Author(s) & Year	Method / Approach	Dataset / Testbed	Key Results	Limitations
framework)	microcontroller placed between gateway and network; flow-based fractional-selectivity filtering, independent of OpenFlow/OpenWrt	switches, compared against Sivanathan et al. [31] baseline (LCFS)	9.65 Mbps average), reduced CPU, RAM, and energy consumption relative to baseline	payload inspection); relies on availability and scalability of the microcontroller

3. Research Gap

Synthesizing the literature reviewed in Section 2, three interlocking gaps motivate this research.

First, OpenFlow dependency remains pervasive. The majority of network-level IoT security mechanisms—including the influential work of Sivanathan et al. [31] and the gateway-based solutions of Bull et al. [8] and Xu et al. [38] require OpenFlow-compliant switches and, frequently, OpenWrt-compatible firmware [6], [8], [26], [31], [36], [38]. Because traditional home routers and gateways rarely support these protocols natively, a substantial share of existing research is not directly deployable in the very environment the ordinary smart home—that motivates it.

Second, authentication and encryption mechanisms for resource-constrained devices remain immature. As Hassan et al. [16] and others note, low-power IoT hardware cannot reliably support the computational overhead of conventional authentication and encryption schemes, and manufacturer practices such as hardcoded credentials further erode whatever protection these mechanisms might offer. This leaves a structural gap for network-level, device-agnostic defences that do not depend on strengthening the device itself.

Third, although flow-based attack detection and mitigation has repeatedly demonstrated cost and resource efficiency relative to packet-level inspection [22], [29], existing flow-based implementations are still tied to OpenFlow-capable infrastructure or validated only at a small scale, as in the two-sensor, ten-minute evaluation of Zaied et al. [40]. No reviewed study augments an SDN controller with a general-purpose microcontroller specifically to remove OpenFlow dependency while retaining flow-level, fractional-selectivity-based attack detection, and benchmarks the result against an established baseline across energy, CPU, RAM, and traffic-load metrics simultaneously.

This research addresses the gap by positioning a microcontroller as a hardware-agnostic intermediary between the home gateway and a cloud-based SDN controller. The microcontroller performs flow-level monitoring and enforces the controller's policies without requiring OpenFlow support on the underlying gateway, while the SDN controller retains centralized, programmable control over the applicable security rules. This design is evaluated using CloudSimSDN and benchmarked directly against the LCFS baseline of Sivanathan et al. [31], providing a like-for-like comparison that is largely absent from the reviewed literature.

4. Methodology

This section describes the proposed architecture, its operational workflow, the algorithms governing traffic handling, and the simulation environment, data, and configuration used to evaluate the framework.

4.1 Overview and Design Rationale

The proposed framework builds on the flow-based, network-level monitoring philosophy established by Sivanathan et al. [31], while removing its dependency on OpenFlow and OpenWrt-compatible gateways. In place of an OpenFlow-compliant switch, the architecture introduces a microcontroller (e.g., Arduino or Raspberry Pi) as a hardware entity positioned between the internet-facing network, the home gateway, and a cloud-based SDN controller. All inbound and outbound traffic destined for or originating from IoT devices is routed through this microcontroller, which filters traffic against locally installed policies and forwards a periodically generated trace file, summarizing unknown or unclassified traffic, to the SDN controller. The controller, accessed through a GUI application or Policy Manager via a northbound API, analyses the trace file and issues an updated set of policies and exceptions back to the microcontroller over a southbound interface. The microcontroller applies these updated rules to subsequent traffic before forwarding permitted flows to the gateway and, ultimately, to the connected end devices. Figure 1 illustrates this architecture and the associated northbound/southbound interfaces.

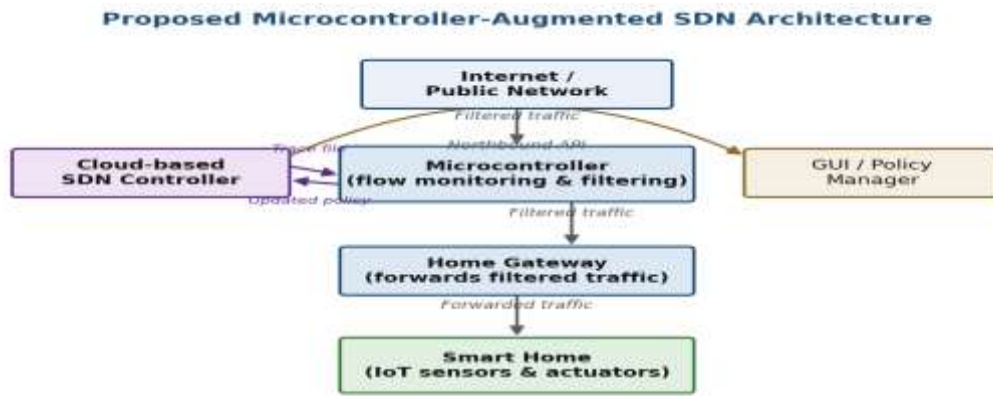


Figure 1. Proposed microcontroller-augmented SDN architecture for smart-home IoT access control.

4.2 System Architecture and Algorithmic Workflow

The workflow is governed by two symmetric procedures, corresponding to the two directions of traffic flow through the architecture. For traffic originating from the public network and destined for the smart home, the microcontroller initializes, monitors and captures incoming traffic, generates a trace file, and transmits it to the SDN controller; the controller updates its policy set and returns the updated policies to the microcontroller, which then evaluates the traffic's fractional-selectivity value. Traffic whose selectivity exceeds the threshold of 0.9 is blocked; all other traffic is forwarded by the microcontroller to the gateway and subsequently to the end devices. This logic is summarized in Algorithm 1 and illustrated in Figure 2.

Algorithm 1. Traffic Handling: Public Network → Smart Home

procedure TrafficFromNetwork

System Initialization

MonitorTraffic()

CaptureTraffic()

Generate TraceFile

Send TraceFile to SDNController

Update Policies = newPolicies

Send newPolicies to Microcontroller

if fractionalSelectivity > 0.9 then

Block Traffic

else

Microcontroller forwards Traffic to Gateway

Gateway forwards Traffic to End-Devices

The reverse direction, in which traffic originates within the smart home and is destined for the public network, follows an analogous procedure, with the gateway forwarding traffic to the microcontroller for monitoring, trace-file generation, and policy-based filtering prior to release onto the network, as summarized in Algorithm 2.

Algorithm 2. Traffic Handling: Smart Home → Public Network

```

procedure TrafficFromHome

```

```

    System Initialization

```

```

    Traffic from End-Devices

```

```

    Gateway sends Traffic to Microcontroller

```

```

    MonitorTraffic()

```

```

    CaptureTraffic()

```

```

    Generate TraceFile

```

```

    Send TraceFile to SDNController

```

```

    Update Policies = newPolicies

```

```

    Send newPolicies to Microcontroller

```

```

    if fractionalSelectivity > 0.9 then

```

```

        Block Traffic

```

```

    else

```

```

        Microcontroller forwards Traffic to Network
    end if
end procedure

```

The fractional-selectivity model used to distinguish legitimate from malicious traffic is drawn from the selectivity framework of Gupta et al. [15], which relates the number of output tuples to the number of input tuples processed by a module. In this implementation, fractional-selectivity values were configured between 0.9 and 1.0; a value exceeding 0.9 is treated as indicative of an attack and triggers blocking on the following interval, while a value of 1.0 signals traffic overflow. Attacks arriving from outside the home network and attacks originating from within the home network are both handled by this mechanism, with internally generated malicious traffic prevented from mixing with legitimate home-network traffic on the subsequent interval, as summarized graphically in Figure 2.

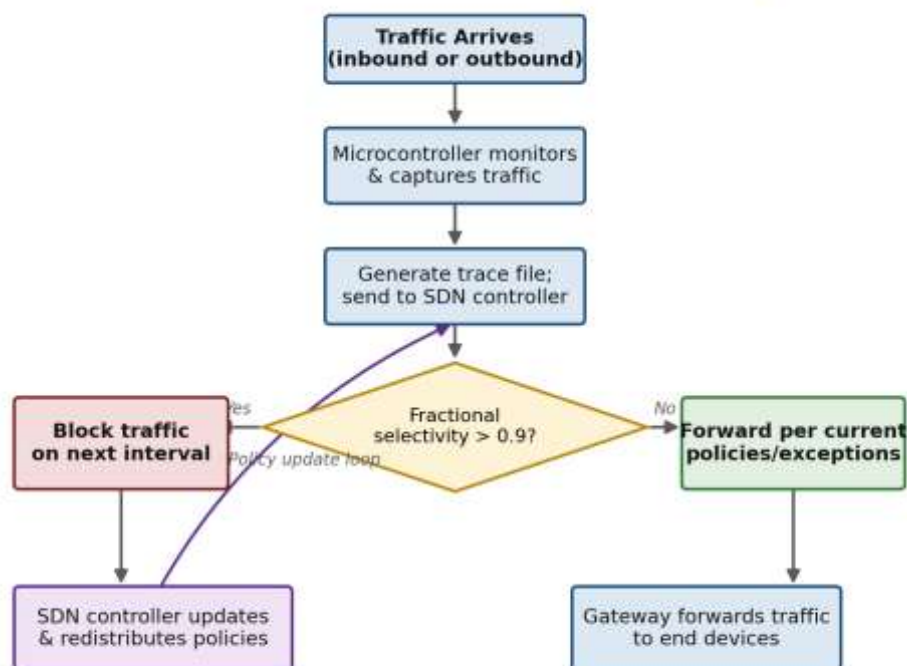
Flow-Level Fractional-Selectivity Decision Logic

Figure 2. Flow-level fractional-selectivity decision logic applied by the microcontroller.

4.3 Case Study Context

The evaluation scenario reflects a realistic home-network threat model. Although most home networks employ Network Address Translation (NAT) to shield internal devices from direct exposure, Universal Plug-and-Play (UPnP) port-forwarding features can still expose client-side devices to internet-based threats, and traffic originating from within a network is conventionally treated as trusted even though internal devices may themselves be compromised or exploited. A representative attack vector considered in this study involves malware on a smartphone using the Simple Service Discovery Protocol (SSDP) to enumerate IoT devices on the home network, followed by a UPnP port-forwarding request that exposes those devices to external attackers—a technique also associated with legitimate peer-to-peer applications such as Skype and online multiplayer games, which necessitates careful policy design to avoid false positives.

4.4 Data Collection and Preprocessing

Traffic was represented within CloudSimSDN using a CSV-format workload dataset, consistent with the simulator's Distributed Data Flow (DDF) processing model. Application-level dependencies were modelled using AppEdges and AppModules, the logical constructs through which CloudSim represents distributed application components: AppModules represent the mapped virtual-machine-hosted processing units, while AppEdges define the logical data flow and dependency relationships between them, each carrying a defined tuple type, direction, CPU length, and network bandwidth requirement. Sensor-originated traffic (e.g., from an ECG-type sensor module used as a representative IoT data source) was routed upward (Tuple.UP) through an Intrusion Detection and Prevention System (IDPS) module responsible for distinguishing IoT from non-IoT traffic and handling detected attacks, with corresponding state updates propagated downward (Tuple.DOWN) to a display actuator.

4.5 Experimental Setup and Implementation

The simulation was implemented in Java and JSON on CloudSimSDN version 4.0, an extension of the CloudSim toolkit [1], with source code obtained from its public GitHub repository [2], selected after a systematic comparison of candidate SDN simulators against criteria including network accessibility, resource provisioning, policy support, functionality, reliability, usability, efficiency, maintainability, and portability. Unlike Mininet, which emulates OpenFlow network topologies but offers limited support for resource-management policy modelling, CloudSimSDN natively supports virtualized hosts, data-centre network topologies, and the dynamic insertion of simulation elements, making it better suited to evaluating the combined SDN-controller/microcontroller architecture proposed here.

The host machine used for simulation ran Windows 10 Pro (64-bit) on an Intel Core i5-4200M CPU at 2.50 GHz with 8,192 MB of RAM. The simulated network topology comprised a cloud data centre, an SDN controller, and a microcontroller, instantiated via a dedicated topology-configuration routine; gateway devices and end-user (mobile/sensor) devices were instantiated through corresponding device-configuration routines, with the number of connected switches varied parametrically between one and six to evaluate scalability. Each configuration was executed both with and without the SDN controller enabled, allowing isolation of the controller's contribution to policy updating and attack handling. Five sensor-actuator pairs were used as the baseline device count, consistent with a representative small-scale smart-home deployment, with the framework designed to scale to larger configurations by substituting the microcontroller for a smartphone, CPU, or server where processing and memory requirements demand it.

4.6 Evaluation Metrics

The framework was evaluated across the following metrics, consistent with those reported for the baseline LCFS model [31], to enable direct comparison: (i) system execution time and system latency; (ii) energy consumption; (iii) CPU usage and RAM consumption; (iv) data-plane load; (v) IoT traffic-plane load; and (vi) camera-module load. Each metric was recorded for switch counts ranging from one to six, with the SDN controller enabled, and the resulting values were compared against the corresponding figures reported for the baseline model over its 18-minute (1,080-second) traffic-capture period.

5. Results and Discussion

5.1 System Execution Time, Latency, and Energy Consumption

Table 2 reports system execution time, system latency, and energy consumption recorded with the SDN controller enabled, for switch counts ranging from one to six. Execution time increased from 5,751 ms at a single switch to 23,252 ms at six switches, while system latency rose more moderately from 36.52 ms to 60.33 ms over the same range, with a visible increase between five and six switches. Energy consumption scaled from approximately 20.77 kWh-equivalent

units at one switch to 249.58 at six switches, an expected consequence of the additional processing and forwarding demanded by a larger topology.

Table 2. System Execution Time, Latency, and Energy Consumption (SDN Enabled)

No. of Switches	System Execution Time (ms)	System Latency (ms)	Energy Consumed (mWh)
1	5751	36.52	20765.75
2	6046	37.44	49143.00
3	8218	37.85	86187.25
4	13680	39.35	132129.75
5	16361	42.06	186856.00
6	23252	60.33	249577.50

5.2 CPU and RAM Utilization

Table 3 presents CPU usage (as a percentage) and RAM consumption (out of a total of 1,000 units) for the same switch configurations. CPU usage increased from approximately 1.98% at a single switch to 23.77% at six switches, while RAM consumption rose from approximately 2.13 to 10.21 units over the same range. Both metrics scale approximately linearly with the number of switches, consistent with the expectation that each additional switch introduces proportional processing and memory overhead at the microcontroller and SDN-controller layers.

Table 3. CPU Usage and RAM Consumption (SDN Enabled)

No. of Switches	CPU Usage (%)	RAM Consumption (of 1000)
1	1.978	2.129
2	4.680	3.746
3	8.208	5.363
4	12.584	6.980
5	17.796	8.597
6	23.769	10.214

5.3 Traffic-Plane Loads

Table 4 reports IoT traffic-plane and camera-module loads, in Kbps, for the proposed framework. IoT traffic load increased from 2.42 Kbps at one switch to 11.61 Kbps at six switches, while camera-module load rose from 1.84 Kbps to 8.84 Kbps over the same range.

Table 4. IoT Traffic and Camera Module Load (SDN Enabled)

No. of Switches	IoT Traffic Load (Kbps)	Camera Module Load (Kbps)
1	2.420	1.844
2	4.257	3.244
3	6.094	4.643
4	7.932	6.043
5	9.769	7.443
6	11.606	8.843

5.4 Comparison with the Baseline Model

The principal basis for evaluating the proposed framework is a direct comparison with the low-cost flow-based solution (LCFS) reported by Sivanathan et al. [31], whose testbed comprised real IoT devices, including a Samsung SmartThings hub, a D-Link DCS5300G IP security camera, a Belkin Wemo switch, and a Wemo motion detector, alongside a laptop and smartphone used to generate representative everyday traffic (live video streaming, social-media access) over an 18-minute capture window. In that baseline, the average data-plane load associated with IoT devices was reported as 9.65 Mbps. Under the proposed framework, the corresponding average data-plane load across the one-to-six-switch configurations was 4.01 Mbps a substantial reduction attributable to the more balanced distribution of processing between the microcontroller and the SDN controller, in contrast to the baseline architecture, which concentrates most of the load on the home gateway and its associated analysis engine.

This pattern extends to the IoT traffic-plane load: over the equivalent 18-minute window, the baseline model recorded a total IoT traffic-plane volume of 10,454 KB, while the proposed framework's average IoT traffic-plane load was 7.01 Kbps, again reflecting reduced load on the central gateway. The average camera-module load under the proposed framework was 5.34 Kbps, likewise lower than the corresponding LCFS values, which showed visible spikes attributable to the buffering behaviour of streamed video traffic in the baseline testbed. Because the baseline model concentrates flow-inspection responsibility in a gateway-resident analysis engine that must interact continuously with a cloud-hosted SDN controller, its load profile is inherently more gateway-intensive; the proposed framework's redistribution of this responsibility to a dedicated microcontroller appears to be the principal source of the observed efficiency gains.

5.5 Interpretation, Advantages, and Limitations

Taken together, the results support the central claim of this study: a microcontroller-augmented SDN architecture can restrict unauthorized access to smart-home IoT devices while consuming fewer computational and energy resources than an OpenFlow-dependent, gateway-centric alternative. The principal advantages evidenced by the simulation are a lower and more scalable data-plane load, reduced CPU and RAM utilization, and lower energy consumption across all tested switch configurations, together with successful blocking of volume-based attacks via the fractional-selectivity mechanism.

These advantages should be read alongside the framework's limitations. Because the mechanism operates on flow-level metadata rather than packet payloads, it is not designed to detect attacks that require content inspection, such as plain-text credential harvesting embedded within otherwise permitted flows. The approach is also dependent on the continued availability and scalability of the microcontroller itself; should the microcontroller become a bottleneck or fail, the filtering function it performs would need to fail over to an alternative device, such as the smartphone, CPU, or server substitution proposed for larger deployments. Finally, the appropriate configuration of policies and exceptions, and the choice between static and dynamic control policies, remain design decisions that will materially affect both detection accuracy and false-positive rates in a live deployment, and were not exhaustively explored within the scope of this simulation-based evaluation.

6. Conclusion

This paper has presented a microcontroller-augmented Software-Defined Networking framework for restricting unauthorized access to IoT devices in smart-home environments. Motivated by the observation that most existing

network-level defences depend on OpenFlow- or OpenWrt-compatible infrastructure that is rarely present in traditional home gateways, the proposed architecture positions a microcontroller between the home gateway and a cloud-based SDN controller, monitoring and filtering all inbound and outbound traffic according to policies issued through a northbound interface and updated dynamically in response to trace files generated by the microcontroller. A fractional-selectivity mechanism, applied at the flow level, was used to identify and block volume-based attacks, with a threshold of 0.9 distinguishing legitimate from malicious traffic.

Implemented and evaluated using CloudSimSDN across configurations of one to six switches, the framework was benchmarked directly against the low-cost flow-based solution proposed by Sivanathan et al. [31]. The results show that the proposed approach achieves lower average data-plane load, reduced IoT traffic-plane and camera-module loads, and lower CPU, RAM, and energy consumption than the baseline, while retaining the ability to detect and block both externally and internally originating volume-based attacks. These findings suggest that decoupling flow-based security enforcement from OpenFlow dependency, and distributing the associated processing load to a dedicated microcontroller, offers a practically deployable and resource-efficient pathway to securing smart-home IoT devices, with direct relevance to households and gateway manufacturers seeking scalable protection without wholesale hardware replacement.

7. Future Work

Several directions merit further investigation to extend and validate the proposed framework beyond its current simulation-based evaluation.

- Hardware-based testbed validation: Implementing the architecture on physical microcontrollers (e.g., Arduino, Raspberry Pi) and commodity home gateways would validate the simulation results under real network conditions and device heterogeneity.
- Larger and multi-site datasets: Extending traffic capture beyond the 18-minute window used for baseline comparison, and across multiple households, would strengthen confidence in the framework's generalizability and its false-positive/false-negative characteristics.
- Scalability for larger deployments: Replacing the microcontroller with a smartphone, CPU, or server for larger home networks with multiple gateways, as anticipated in the proposed design, warrants dedicated performance evaluation at that scale.
- Integration of dynamic traffic-analysis tools and machine learning: Augmenting the fractional-selectivity mechanism with machine-learning-based anomaly detection, as explored by related SDN-IoT security studies, could improve detection of attacks that evade purely volumetric thresholds.
- Support for content-aware inspection: Given the framework's current reliance on flow-level metadata, integrating complementary payload-inspection or Intrusion Detection and Prevention System (IDPS) modules would extend protection to plain-text and other content-dependent attacks.
- Real-time and automated policy management: Moving from manually configured static policies toward automated, context-aware policy generation (e.g., time- or occupancy-based rules, as explored by Sivaraman et al. [32]) would improve responsiveness to evolving threats.
- Industrial adoption and standardization: Engaging gateway manufacturers and standards bodies to evaluate the practicality of embedding microcontroller-augmented SDN enforcement directly into consumer hardware would be a necessary step toward broader real-world adoption.

More broadly, complementary surveys of network-level policy enforcement for IoT devices [41], SDN- and NFV-based security mechanisms for IoT systems [42], and machine-learning-driven routing optimization in SDN [43] point to adjacent directions—policy automation, virtualization, and learning-based control—that could be integrated with the architecture proposed here as this research programme matures.

References

- [1] "CloudSim: A Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services," 2020. [Online]. Available: <http://www.cloudbus.org/cloudsim/>
- [2] "CloudSimSDN," 2020. [Online]. Available: <https://github.com/Cloudslab/cloudsimsdn>
- [3] B. R. Al-Kaseem and H. S. Al-Raweshidy, "SD-NFV as an energy efficient approach for M2M networks using cloud-based 6LoWPAN testbed," *IEEE Internet of Things J.*, vol. 4, no. 5, pp. 1787–1797, 2017.
- [4] T. Alladi, V. Chamola, B. Sikdar, and K.-K. R. Choo, "Consumer IoT: Security vulnerability case studies and solutions," *IEEE Consum. Electron. Mag.*, vol. 9, no. 2, pp. 17–25, 2020.

- [5] M. Baddeley, R. Nejabati, G. Oikonomou, M. Sooriyabandara, and D. Simeonidou, "Evolving SDN for low-power IoT networks," in *Proc. 2018 4th IEEE Conf. Netw. Softwarization Workshops (NetSoft)*, 2018, pp. 71–79.
- [6] S. S. Bhunia and M. Gurusamy, "Dynamic attack detection and mitigation in IoT using SDN," in *Proc. 27th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, 2017, pp. 1–6.
- [7] P. Brous, M. Janssen, and P. Herder, "The dual effects of the Internet of Things (IoT): A systematic review of the benefits and risks of IoT adoption by organizations," *Int. J. Inf. Manage.*, vol. 51, p. 101952, 2020.
- [8] P. Bull, R. Austin, E. Popov, M. Sharma, and R. Watson, "Flow based security for IoT devices using an SDN gateway," in *Proc. 2016 IEEE 4th Int. Conf. Future Internet Things Cloud (FiCloud)*, 2016, pp. 157–163.
- [9] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 3, pp. 2671–2701, 2019.
- [10] S. Demetriou et al., "HanGuard: SDN-driven protection of smart home WiFi devices from malicious mobile apps," in *Proc. 10th ACM Conf. Secur. Priv. Wireless Mobile Netw.*, 2017, pp. 122–133.
- [11] F. Firouzi, B. Farahani, M. Weinberger, G. DePace, and F. S. Aliee, "IoT fundamentals: Definitions, architectures, challenges, and promises," in *Intelligent Internet of Things*, Springer, 2020, pp. 3–50.
- [12] M. Frank and M. Ghaderi, "Securing smart homes with OpenFlow," *Univ. of Calgary, Tech. Rep.*, 2019.
- [13] G. Gridling and B. Weiss, "Introduction to microcontrollers," *Vienna Univ. of Technology, Inst. of Computer Engineering, Embedded Computing Systems Group*, 2007.
- [14] A. Grygoruk and J. Legierski, "IoT gateway—implementation proposal based on Arduino board," in *Proc. 2016 Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, 2016, pp. 1011–1014.
- [15] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments," *Softw. Pract. Exp.*, vol. 47, no. 9, pp. 1275–1296, 2017.
- [16] W. H. Hassan et al., "Current research on Internet of Things (IoT) security: A survey," *Comput. Netw.*, vol. 148, pp. 283–294, 2019.
- [17] M. Hung, *Leading the IoT*. Gartner, 2017.
- [18] M. S. Jalali, J. P. Kaiser, M. Siegel, and S. Madnick, "The Internet of Things promises new benefits and risks: A systematic analysis of adoption dynamics of IoT products," *IEEE Secur. Priv.*, vol. 17, no. 2, pp. 39–48, 2019.
- [19] T. Mizrahi and Y. Moses, "Software defined networks: It's about time," in *Proc. IEEE INFOCOM 2016*, 2016, pp. 1–9.
- [20] D. Mostovich, P. Fabrikantov, A. Vladkyo, and M. Buinevich, "High-level vulnerabilities of software-defined networking in the context of telecommunication network evolution," in *Proc. 2017 IEEE Conf. Russian Young Res. Elect. Electron. Eng. (EIconRus)*, 2017, pp. 184–186.
- [21] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 3, pp. 2702–2733, 2019.
- [22] A. A. Pranata, T. S. Jun, and D. S. Kim, "Overhead reduction scheme for SDN-based data center networks," *Comput. Stand. Interfaces*, vol. 63, pp. 1–15, 2019.
- [23] S. Rizvi, A. Kurtz, J. Pfeffer, and M. Rizvi, "Securing the internet of things (IoT): A security taxonomy for IoT," in *Proc. 17th IEEE Int. Conf. Trust Secur. Priv. Comput. Commun. (TrustCom/BigDataSE)*, 2018, pp. 163–168.
- [24] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," *Comput. Netw.*, vol. 57, no. 10, pp. 2266–2279, 2013.
- [25] R. Sanger, B. Cowie, M. Luckie, and R. Nelson, "Characterising the limits of the openflow slow-path," in *Proc. 2018 IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, 2018, pp. 1–7.
- [26] M. Serror, M. Henze, S. Hack, M. Schuba, and K. Wehrle, "Towards in-network security for smart homes," in *Proc. 13th Int. Conf. Availability Reliab. Secur.*, 2018, pp. 1–8.
- [27] Z. Shouran, A. Ashari, and T. K. Priyambodo, "Internet of things (IoT) of smart home: Privacy and security," *Int. J. Comput. Appl.*, vol. 182, no. 39, pp. 3–8, 2019.

- [28] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Comput. Netw.*, vol. 76, pp. 146–164, 2015.
- [29] A. Sivanathan et al., "Classifying IoT devices in smart environments using network traffic characteristics," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1745–1759, 2018.
- [30] A. Sivanathan et al., "Characterizing and classifying IoT traffic in smart cities and campuses," in *Proc. 2017 IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2017, pp. 559–564.
- [31] A. Sivanathan, D. Sherratt, H. H. Gharakheili, V. Sivaraman, and A. Vishwanath, "Low-cost flow-based security solutions for smart-home IoT devices," in *Proc. 2016 IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, 2016, pp. 1–6.
- [32] V. Sivaraman, H. H. Gharakheili, A. Vishwanath, R. Boreli, and O. Mehani, "Network-level security and privacy control for smart-home IoT devices," in *Proc. 2015 IEEE 11th Int. Conf. Wireless Mobile Comput. Netw. Commun. (WiMob)*, 2015, pp. 163–167.
- [33] J. Son, A. V. Dastjerdi, R. N. Calheiros, X. Ji, Y. Yoon, and R. Buyya, "CloudSimSDN: Modeling and simulation of software-defined cloud data centers," in *Proc. 2015 15th IEEE/ACM Int. Symp. Cluster Cloud Grid Comput.*, 2015, pp. 475–484.
- [34] J. Son, T. He, and R. Buyya, "CloudSimSDN-NFV: Modeling and simulation of network function virtualization and service function chaining in edge computing environments," *Softw. Pract. Exp.*, vol. 49, no. 12, pp. 1748–1764, 2019.
- [35] S. Wang, K. M. Gomez, K. Sithamparamanathan, and P. Zanna, "Software defined network security framework for IoT based smart home and city applications," in *Proc. 2019 13th Int. Conf. Signal Process. Commun. Syst. (ICSPCS)*, 2019, pp. 1–8.
- [36] K. Wrona, M. Amanowicz, S. Szewczyk, and K. Gierłowski, "SDN testbed for validation of cross-layer data-centric security policies," in *Proc. 2017 Int. Conf. Military Commun. Inf. Syst. (ICMCIS)*, 2017, pp. 1–6.
- [37] B. Xiong, K. Yang, J. Zhao, W. Li, and K. Li, "Performance evaluation of OpenFlow-based software-defined networks based on queueing model," *Comput. Netw.*, vol. 102, pp. 172–185, 2016.
- [38] T. Xu, D. Gao, P. Dong, H. Zhang, C. H. Foh, and H.-C. Chao, "Defending against new-flow attack in SDN-based Internet of Things," *IEEE Access*, vol. 5, pp. 3431–3443, 2017.
- [39] M. B. Yassein, S. Aljawarneh, M. Al-Rousan, W. Mardini, and W. Al-Rashdan, "Combined software-defined network (SDN) and Internet of Things (IoT)," in *Proc. 2017 Int. Conf. Elect. Comput. Technol. Appl. (ICECTA)*, 2017, pp. 1–6.
- [40] A. N. Zaied, E. R. Mohamed, and M. N. Hemdan, "Wireless sensor networks and software defined networks integration," *Int. J. Digit. Content Technol. Appl.*, vol. 11, no. 1, 2017.
- [41] H. Habibi Gharakheili, A. Sivanathan, A. Hamza, and V. Sivaraman, "Network-level security for the Internet of Things: Opportunities and challenges," *IEEE Computer*, vol. 52, no. 8, pp. 58–62, 2019.
- [42] I. Farris, T. Taleb, Y. Khettab, and J. Song, "A survey on emerging SDN and NFV security mechanisms for IoT systems," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 1, pp. 812–837, 2019.
- [43] R. Amin, E. Rojas, A. Aqdu, S. Ramzan, D. Casillas-Perez, and J. M. Arco, "A survey on machine learning techniques for routing optimization in SDN," *IEEE Access*, vol. 9, pp. 104582–104611, 2021.
- [44] N. Y.-R. Douha, M. Bhuyan, S. Kashihara, D. Fall, Y. Taenaka, and Y. Kadobayashi, "A survey on blockchain, SDN and NFV for the smart-home security," *Internet of Things*, vol. 20, p. 100588, 2022.
- [45] H. Gordon, C. Batula, B. Tushir, B. Dezfouli, and Y. Liu, "Securing smart homes via software-defined networking and low-cost traffic classification," in *Proc. 2021 IEEE 45th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, 2021, doi: 10.1109/COMPSAC51774.2021.00143.