ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

UDC 621.398:004.71/.72 **doi: 10.32620/reks.2025.4.11**

Anatolii BANAR, Heorhii VOROBETS

Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine

AI-BASED ADAPTIVE MANAGEMENT OF LIMITED RESOURCES IN SDN-IOT ECOSYSTEMS

The subject *of the study is the integration of artificial intelligence (AI) methods into software-defined networks (SDN) for adaptive control of access to limited resources within the infrastructure of Internet of Things (IoT) ecosystems. The* ***goal*** *of this work is to develop a model and architectural solution for a hybrid Cloud-SDN-IoT framework with embedded AI components, enabling the intelligent allocation of network and computing resources, and to experimentally validate the improvement of the fair distribution of a limited IoT resource across different traffic patterns in an emulation environment. The main* ***tasks*** *of the research are: 1) to analyze modern approaches to energy-efficient resource management and security in SDN-IoT networks; 2) to create the architecture of a hybrid Cloud-SDN-IoT framework that combines centralized SDN network control with the flexibility of cloud infrastructure; 3) to develop an experimental methodology using machine learning components to improve resource allocation and reduce load imbalance among competing clients; 4) to evaluate the system's efficiency in relation to the stated objectives and the fair distribution of limited IoT resources by assessing the request distribution and the accuracy of detecting resource access violations. The paper proposes an improved three-layer SDN architecture model incorporating AI-based analytics: the IoT infrastructure layer, the SDN control layer, and the cloud application layer. The experimental part was implemented in a virtual Linux environment using Mininet and Ryu, where the trained AI model makes decisions about allocating the limited resource. The experimental* ***results*** *demonstrated that integrating the AI module into the SDN controller workflow increases the accuracy of detecting resource access violations, reduces load imbalance among clients, and improves the stability of real-time request distribution.* ***Conclusions****. The scientific novelty of the obtained results lies in the development of a reproducible hybrid Cloud-SDN-IoT architecture model that enables adaptive management of access to limited IoT node resources by combining centralized SDN control with AI-based predictive analytics. The AI-enabled control loop increased the average fairness accuracy of request distribution from 79.2% to 90.98%, an increase of 11.78 percentage points (14.87% relative), demonstrating improved proportional access to the limited IoT TokenServer API while preserving stable, real-time request regulation. The* ***practical significance*** *lies in the potential application of the proposed approach to optimize access to limited cloud services, APIs, energy resources, or IoT devices in smart city systems, healthcare, or industrial networks.* ***Further research*** *will focus on expanding the AI components with various machine learning models, forming new datasets, and conducting comparative evaluations of each model's effectiveness in dynamic SDN-IoT resource management and reproduction under real-world conditions.*

Keywords: *software-defined networks; Internet of Things; artificial intelligence; resource management; cloud infrastructure; SDN controller; machine learning.*

1. Introduction

The increasing data flows in information and communication systems (ICS) require greater resource controllability to optimize access to data, servers, communication devices, and other components [1, 2]. This necessitates more in-depth research into SDN networks, which are predominantly used in ICS. In particular, issues of scalability and controllability in heterogeneous SDN-IoT networks remain relevant, where limitations in energy, bandwidth, and computing power are combined with high security and quality of service (QoS) requirements. There is a need to define the practical challenges arising in IoT network scenarios and the role of SDN as an architecture with logically centralized control and programmable interfaces. It is appropriate to review the current state of the art in resource management, security, and integration with AI-based approaches for traffic analysis and anomaly detection. Further research is required to develop a model and architecture of a hybrid Cloud-SDN-IoT framework using artificial intelligence for adaptive control of access to limited resources and to conduct its experimental validation in real-time conditions.

1.1. Motivation

The use of IoT technologies has become so widespread that human interaction with smart devices now occurs in everyday life, often in various forms and fre-

quently unnoticed by users themselves. IoT plays a significant role in nearly all areas where such devices are deployed, such as smart homes, smart cities, energy systems, healthcare, and agriculture. At the same time, the number of devices whose operation depends directly on the network has increased, both for global interactions and for communication among devices in smart system scenarios. Consequently, confidentiality, security, and resource management in IoT networks will remain essential challenges for the foreseeable future.

Software-defined networks are a relatively new architecture whose main feature is the separation of control and data forwarding functions, aimed at simplifying and accelerating the management of large-scale networks. The architecture consists of three layers, each corresponding to a specific role. The control layer assumes responsibility for network management, removing this function from network devices as in traditional networks. The control layer may include one or more SDN controllers interconnected and interacting with other layers. The infrastructure layer includes network devices that directly process and forward data packets (such as switches or routers). This layer acts as an executor of commands received from the control layer, with nodes not making independent routing decisions. The application layer implements the interfaces and applications that interact with the controllers to provide network functions. Interaction with applications occurs through software interfaces (APIs), enabling management of quality of service, security control, monitoring, and load balancing without the need for physical intervention in network equipment.

SDN can manage resources across various network types, including IoT, cellular networks, internet service providers, and cloud services. To coordinate networks of different natures, integrating artificial intelligence is a necessary step [3, 4]. AI at the SDN controller level can significantly enhance efficiency by taking over automated decision-making and diagnostics. This enables the modeling of complex networking problems such as dynamic configuration, routing, and access management to limited resources.

Resource management in IoT networks faces challenges, the main ones being limited computing power, energy resources, and bandwidth. Figure 1 summarizes the key resource components of a typical autonomous IoT node, including the power source, communication module, data-processing module, and memory [5]. Most IoT devices operate autonomously, and their batteries' limited capacity complicates long-term operation, requiring algorithms to reduce energy consumption.

The functions of an IoT device include collecting, processing, storing, and transmitting data from the physical world to the virtual environment. Within a single automation stage, the number of IoT devices can be quite large. The heterogeneity of devices and communication technologies complicates integrating various elements into a unified network, thereby increasing the complexity of resource management. Furthermore, the growing volume of data generated by IoT devices [6] places additional strain on computing resources and data storage, affecting the overall reliability of the system.

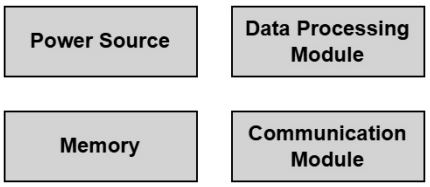


Fig. 1. Key resource components of an IoT node

Given the challenges outlined above, this work aims to systematize and analyze current approaches to integrating AI into SDN for adaptive resource management in IoT networks, and to develop a framework architecture that addresses the challenges of SDN-IoT networks, followed by its experimental deployment. This will help identify areas where problems and constraints persist, determine which methods are effective for optimizing network resources, and, at the same time, open new research avenues that will drive further progress in the field.

1.2. State of the art

- *Modern methods of IoT resource management.* Resource management mechanisms in IoT nodes are typically divided into the following areas: process management, memory management, energy management, and communication management. The communication needs of an IoT ecosystem are supported by its communication architecture. Due to the limited resources of each node, the communication protocols of sensor networks that enable such interaction as per [7] must be effective in terms of reliability, scalability, and energy efficiency. This is further complicated by device heterogeneity and by the fact that heterogeneous sensor networks may operate under widely varying transmission conditions [8]. Efficient communication at the device level directly affects the network's overall performance. Therefore, in such systems, communication should primarily be oriented toward energy efficiency, and only then toward achieving high bandwidth. It is also essential to consider the unique characteristics of device traffic and ensure appropriate quality of service management.

The processing and management needs of the large data flows generated by the IoT ecosystem can largely be met through cloud computing [9, 10]. Cloud compu-

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

ting provides a flexible and scalable infrastructure for storing and processing data generated by Internet of Things devices. Its application enables network state management from any location, allowing organizations to perform real-time data analysis, create tasks for the IoT ecosystem, integrate a wide range of tools (such as machine learning, data storage, and scaling), and make informed decisions based on the obtained information. However, processing IoT applications exclusively in the cloud is not always an optimal solution [11], particularly for tasks that require minimal latency. A promising approach involves using fog and edge computing to address the high bandwidth requirements for transmitting data from end devices. These technologies enable the processing of large volumes of data directly near their sources, reducing dependence on the cloud and minimizing delays.

- *The application of SDN architecture* in IoT provides global visibility into network state via programmable APIs, centralized logical control over physically distributed resources, and flexibility in implementing new network management methods. Owing to these properties, SDN serves as a solid foundation for building IoT ecosystem management frameworks. The integration of IoT and SDN [12], known as SDN-IoT (Fig. 2), creates a unified architecture that separates the control plane from the data plane and efficiently coordinates interactions among individual IoT nodes within the network.

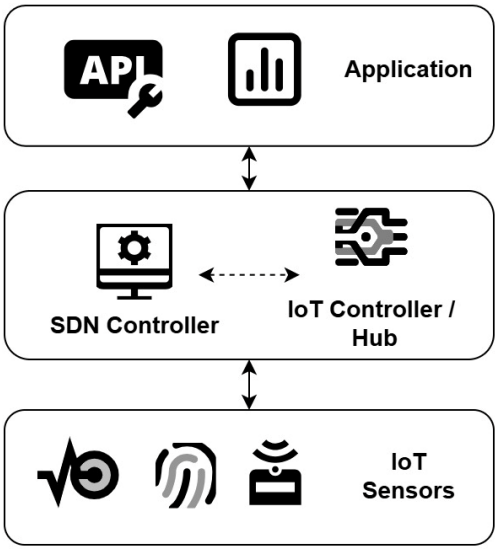


Fig. 2. Structure of the SDN-IoT framework

For the long-term operation of the SDN-IoT infrastructure, energy-efficient routing is essential, as it minimizes device power consumption, extends network lifetime, and ensures optimal utilization of limited resources. The use of intelligent algorithms for dynamic load balancing increases data transmission efficiency even in large and complex networks. In [13], the use of genetic algorithms (particle swarm optimization and artificial bee colony methods) was proposed to address these tasks. Architecturally, the SDN controller is deployed within a cloud infrastructure (Fig. 3). The cloud provides centralized management of routing, cluster formation, and load balancing across the network. In particular, genetic algorithms are executed on cloud infrastructure, as its computational capacity eliminates the need for local infrastructure for such tasks (or when it is not feasible).

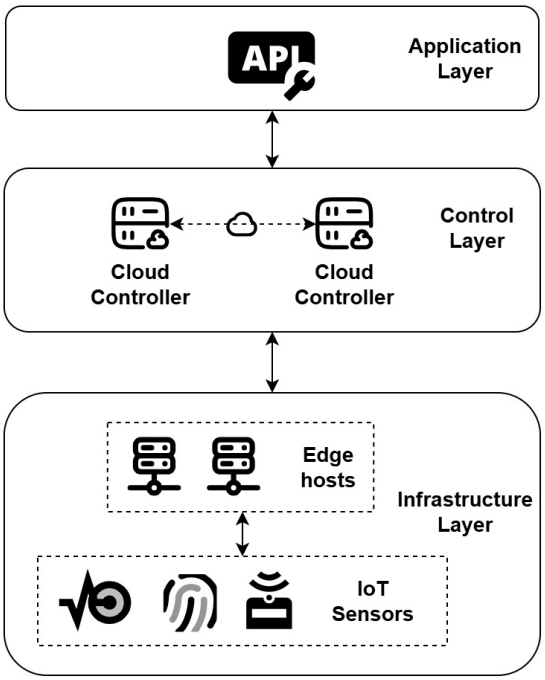


Fig. 3. Architecture of Cloud-based SDN for IoT

The cloud provides flexibility in network scaling, allowing resource allocation to increase or decrease depending on load. This helps reduce costs compared to maintaining local infrastructure, especially during periods of low network usage. It also simplifies network maintenance (and thus extends its lifespan) in cases where genetic algorithms are replaced with other methods or when the resource management approach is modified - changes that would be costly or even impossible if the SDN AI controller were implemented within the network itself.

The integration of blockchain into SDN for IoT is a promising approach, as it enhances security, privacy, and data integrity in IoT applications [14]. Blockchain provides a decentralized, immutable ledger, helping ensure that data cannot be tampered with when exchanged across SDN-orchestrated IoT networks. This feature is particularly relevant in IoT domains that require trustworthy, secure data exchange.

- *Integration of AI into SDN for IoT.* Security in

IoT is critically essential because individual nodes are vulnerable, leading to cyberattacks that can harm both individual users and organizations. Protecting IoT devices from such threats requires implementing reliable security measures, particularly intrusion detection systems (IDSs) [15]. The limited computing power, energy and memory resources of IoT devices make it difficult to implement complex and comprehensive security mechanisms on each device. Therefore, efficient lightweight security solutions are needed to ensure protection and adaptability at the network level.

Deep learning (DL) at the intrusion detection system level to analyze network traffic and detect anomalies is applied. The combination of deep learning models such as convolutional neural networks (CNNs) and long short-term memory (LSTMs) effectively handles the heterogeneous nature of IoT devices and enables classification of IoT traffic as benign or malicious, achieving high accuracy and efficiency in threat detection. LSTM is an advanced form of recurrent neural networks that allows modeling of temporal dependencies in data. LSTM networks consist of memory blocks that use three types of gates (input, output, and forget) to control the flow of information. This makes LSTM ideal for analyzing sequential data, such as network traffic, for anomaly detection [16]. In the context of IoT security, LSTM is used to analyze network traffic to detect anomalies and potential cyberattacks, owing to its ability to capture long-term dependencies in data. The system proposed in [17], called IDSIoT-SDL (Fig. 4), consists of three main components:

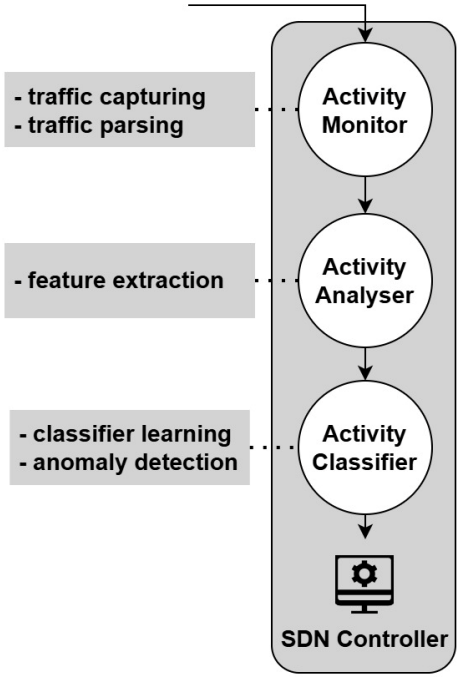


Fig. 4. Structure of IDSIoT-SDL

- Activity Monitor – responsible for collecting and analyzing network traffic. OpenFlow switches are used to gather data about network flows. Its functions include traffic capture, parsing, and analysis preparation;
- Activity Analyzer – uses statistical data to detect anomalies in the network. The LSTM algorithm helps identify potential attacks and classify threat types. Its main task is to determine traffic characteristics for model training;
- Classifier – responsible for classifying traffic as usual or malicious. When an attack is detected, an alert is generated, and the attack type is identified for subsequent response.

The approach proposed in [17] demonstrated effectiveness in protecting IoT networks by integrating SDN and deep learning, enabling adaptability to security challenges.

1.3. Objectives and tasks

Despite its advantages, the application of SDN architecture in IoT faces several challenges. Centralized control introduces the risk of a single point of failure, which affects network stability. The growing number of connected IoT devices places a significant load on SDN controllers, resulting in data processing delays and reduced overall network performance. Studies [18, 19] show that embedded controllers are effective in small networks, but as the scale increases, it becomes more appropriate to use external controllers to ensure adequate performance.

SDN architecture faces new security challenges, particularly concerning the protection of its own infrastructure. New risks have emerged related to potential attacks on the controller itself. A compromised SDN controller may allow an attacker to take control of the entire network. In contrast, a compromised router can only disrupt the proper functioning of the traffic passing through it [20]. Attacks on control systems can cause significant disruptions to industrial processes, negatively affecting product quality and potentially endangering human life. Therefore, it is necessary to develop a strategy that ensures the isolation and monitoring of suspicious devices, for example, through network segmentation and quarantining malicious IoT devices [21].

The use of the SDN approach in IoT, if not optimized, can increase device energy consumption by generating additional control traffic [22, 23], thereby reducing IoT devices` autonomy.

The Internet of Things is a key source of big data, as it relies on connecting many smart nodes to the network to transmit information about the environmental conditions they monitor. When integrated into a distributed environment, the SDN architecture can amplify its

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

internal limitations. Under current conditions, creating a single central controller [24, 25] to manage all subnetworks is a risky approach, since the Internet itself is inherently distributed. This highlights the need for a new perspective on the overall architecture.

Summarizing the above analysis, it can be noted that the development and integration of SDN into the Internet of Things environment are accompanied by a set of systemic challenges, the resolution of which determines the overall network efficiency. The key tasks for SDN in the context of IoT include effective *energy management, scalability, enhanced security, load balancing, and interoperability* among heterogeneous devices and technologies. Implementing these tasks requires the development of flexible, intelligent control mechanisms capable of adaptively responding to changes in network state and load.

A promising approach to addressing the above challenges is the use of artificial intelligence in SDN-IoT networks. The integration of AI assists in device authentication, enhances security, effectively mitigates attacks on SDN-IoT networks, and ensures operational continuity. Through capabilities such as real-time data analysis and network event prediction, AI enables SDN-IoT networks to achieve a new level of performance, delivering high QoS, scalability, and adaptability across interconnected systems.

The paper is structured as follows: Section 1 introduces the motivation for improving the management of limited resources in heterogeneous IoT environments and provides an overview of the SDN architecture, its integration with cloud and AI technologies, and the key challenges in ensuring scalability, energy efficiency, and security. Section 2 describes the materials and methods of research, detailing the design of the proposed hybrid Cloud-SDN-IoT framework, the interactions among architectural layers, and the experimental setup implemented within a software-defined testbed with an external SDN controller, control API, monitoring, and AI module for traffic optimization. Section 3 presents the algorithm and results of experimental testing of the proposed framework, including performance metrics, adaptive access control evaluation, and analysis of system stability. Section 4 provides a detailed discussion of the results, demonstrating the framework's ability to enhance network performance and security, ensure adaptive access control to constrained resources, and maintain service quality. Section 5 concludes the paper by summarizing the key findings, highlighting the advantages and limitations of the proposed approach, and outlining directions for future research toward large-scale implementation and real-time optimization of hybrid Cloud-SDN-IoT systems.

2. Materials and methods of research

Let us consider the basic theoretical approaches, methods, and technologies that enable the design of an improved Hybrid Cloud-SDN-IoT framework architecture and, accordingly, provide adaptive control of access to limited resources in Internet of Things networks and ecosystems. We will analyze and justify the extended functionality of the new framework architecture, describe the selection and interaction of the hardware–software resources and components, and the artificial intelligence algorithms required for the experimental implementation of the framework and the emulation environment. The proposed methodology is structured in accordance with the stated research objectives and allows the experiment to be reproduced.

2.1. Proposed Methods and Design of the Hybrid Cloud-SDN-IoT Framework

The improved and updated architecture of the hybrid Cloud-SDN-IoT framework for adaptive management and access control to limited resources in IoT networks (Fig. 5) is built as a three-layer structure based on the cloud application layer, the control layer, and the infrastructure layer. Unlike typical SDN architectures, it employs the MQTT protocol and blockchain technologies to enable cross-layer feedback, enhance system security, and support deep learning, among other functions.

The functionality and implementation approaches for the individual layers of the proposed architecture may be adjusted based on the problem-oriented tasks addressed by a particular ecosystem. However, the general approach to implementing the framework can be based on the following principles.

1) Infrastructure Layer. As shown in Fig. 5, this layer serves as the foundation of the framework architecture. It comprises multiple distributed IoT environments (e.g., a room in a smart hospital, a bank branch, a part of a smart manufacturing system, or a smart home). Each environment consists of IoT nodes connected to their respective IoT controllers.

- The *IoT Controller* plays a key role in enabling device interaction with the network, supporting the MQTT (Message Queue Telemetry Transport) protocol for communication and utilizing blockchain for secure authentication and authorization within the MQTT broker. For further routing and integration with the SDN network, several IoT controllers are connected to an OpenFlow SDN switch that manages traffic and allocates network resources.

The infrastructure layer interacts with the control layer via the SDN Southbound Interface, enabling centralized network management. In addition, the IoT

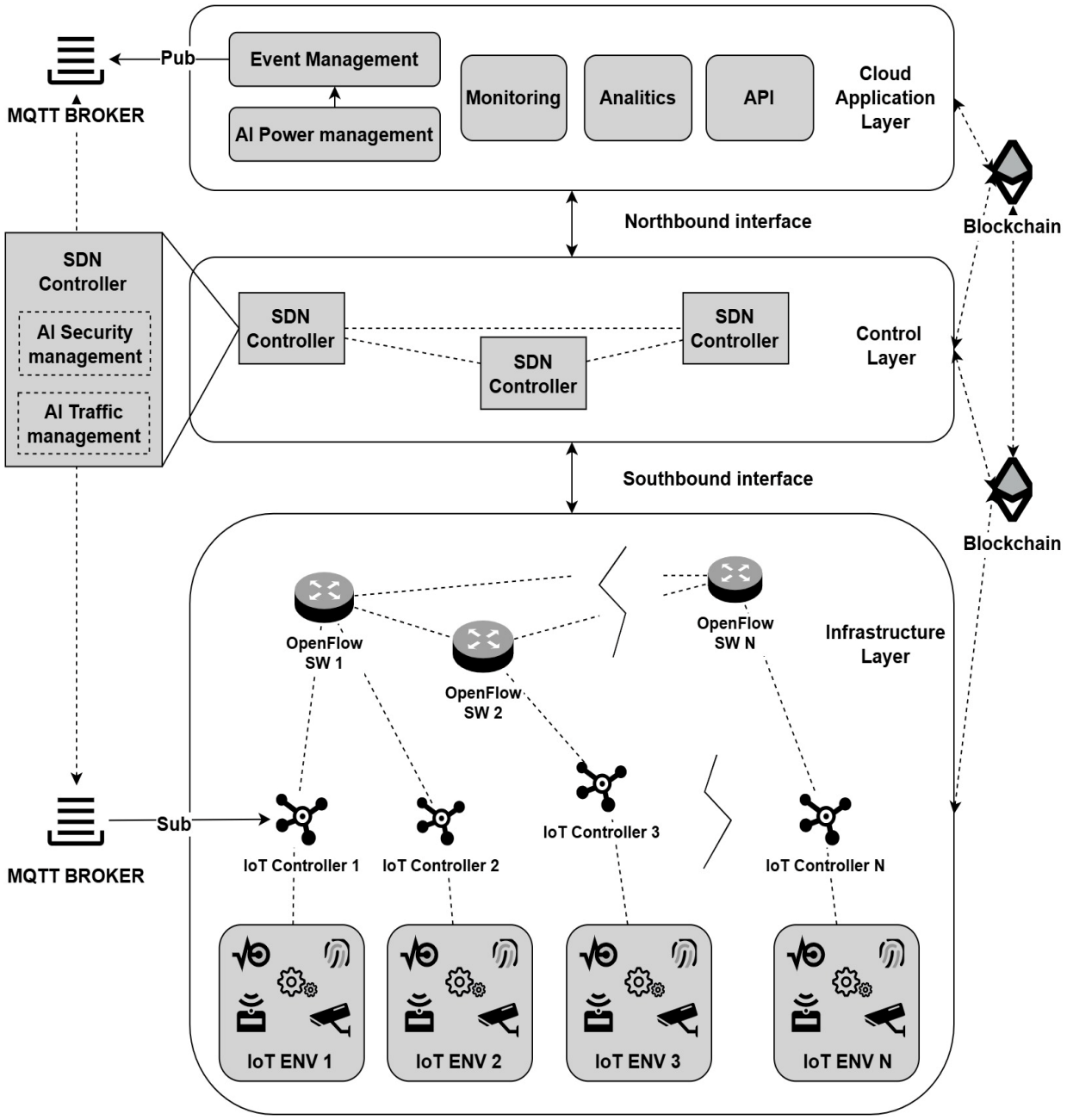


Fig. 5. Architecture of the hybrid Cloud-SDN-IoT framework

controller can publish information about connected devices to the application layer via the MQTT protocol and is itself subscribed to events for dynamic management of IoT device power consumption. This enables automatic control of the power supply for individual IoT nodes to optimize energy use or security management.

2) Control Layer provides centralized management of network traffic and security within IoT environments. This layer includes SDN controllers that perform intelligent network control, as well as two embedded AI modules:

- *AI Security Management Module* - employs deep learning methods, including convolutional neural networks and long short-term memory. The AI model is trained on historical network traffic data obtained from IoT controllers, OpenFlow SDN switches, and blockchain records. Based on this, it classifies traffic as safe or malicious, detecting anomalous patterns that may indicate DDoS attacks, unauthorized connections, or device configuration changes. When a threat is detected, the SDN controller blocks suspicious traffic or adjusts routing, notifying the Cloud Application Layer of the danger.

- *AI Traffic Management Module* - applies supervised learning to balance traffic within the SDN-IoT environment. The AI model is trained on data related to the load on OpenFlow SDN switches,

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

historical traffic metrics from resource-constrained IoT nodes, and QoS requirements. Based on the predicted changes in network load, the SDN controller optimizes routing to prevent delays and congestion, while dynamically adjusting bandwidth allocation for mission-critical IoT devices. To minimize the risks of unauthorized access and data tampering, SDN controllers interact with the blockchain infrastructure. They record transactions related to network policy changes and store key configurations, ensuring protection against compromise by enabling rollback to the last known operational state. Blockchain is also used to authorize the SDN controller when interacting with the cloud layer, providing secure authentication during the exchange of control commands.

Interaction with the infrastructure layer occurs through the Southbound API, where the controller receives traffic statistics, IoT device connection and disconnection events, and routing requests. In response, the controller provides updated routing rules to the OpenFlow SDN switches, ensuring flexible traffic management. Interaction with the application layer occurs through the Northbound API, where the SDN controller sends analytical data and threat notifications while receiving new network management policies from cloud-based applications.

3) Cloud Application Layer is the top level of the architecture, providing monitoring, analytics, high-level network management by the administrator, and energy consumption optimization. It performs essential functions related to integrating data from lower layers, processing this information, and making decisions for dynamic management of the IoT infrastructure. The cloud architecture was chosen for its flexibility, as it allows easy updates or replacement of the AI model without requiring modifications to the entire infrastructure, unlike direct implementation. It also provides scalability, enabling the addition of new analytical modules or increased computational capacity without requiring additional local hardware. Moreover, it allows the deployment of new algorithms for energy optimization, management, or IoT ecosystem protection without interrupting its operation, ensuring network stability. The components of this layer include:

- The *monitoring* and *analytics module* provides visualization of the current network state, resource usage statistics, and traffic anomaly detection. Interactive analytical dashboards for operators enable strategic decision-making.
- The SDN *network API* is used to receive data from the Control Layer and to transmit configuration commands. It ensures feedback with SDN controllers for adaptive routing.
- The *AI Energy Management module* uses a deep reinforcement learning (DRL) algorithm to optimize energy consumption across IoT devices. The control process is implemented through an event management mechanism, specifically by generating commands based on telemetry data about the state of IoT devices, battery charge levels, and changes in the IoT network topology. These commands are published to an MQTT queue, to which IoT controllers subscribed to specific event types respond by performing corresponding actions, such as turning off a device or switching it to a power-saving mode.

Interaction with the Control Layer occurs through the Northbound Interface, which provides data on the current traffic state, the status of IoT controllers, and overall network load. At this level, blockchain is used to authorize requests to the MQTT broker, control access to the event queue, and verify energy management commands to prevent request forgery.

2.2. Experimental Setup

To verify the functionality of the proposed framework, a controlled experimental environment was created and deployed (Fig. 6), integrating network emulation, an embedded AI module for adaptive access control to limited resources, an event broker for IoT resource management, and cloud-based monitoring and management tools (metrics, dashboards, APIs).

The environment was deployed on a Linux-based platform, which provided flexibility in organizing the operation of individual open-source components. The choice of Mininet (Fig. 7) and Ryu (Fig. 8) was motivated by their widespread use in research and experimental studies [26], reproducibility across independent conditions, open-source availability, and extensibility via custom Python modules (AI module, traffic generation, statistics collection, and limited-resource simulation).

Prometheus was used to collect network metrics, and Grafana (Fig. 9) was employed for real-time visualization of client activity, API request success and failure statistics, and IoT device status. Manual network operation control was implemented via a REST API using Postman (Fig. 10).

To emulate the IoT data plane, an MQTT broker and an IoT device hub were configured on one of the Mininet hosts (h7) and the control plane. Additionally, an IoT token-server API with a limited number of requests per minute was connected to host h1 as a constrained resource. Power management was implemented using control queues with on/off/sleep/awake commands. The controller or AI module can temporarily switch devices into power-saving states during overloads or when access quotas to the limited resource are depleted and receive confirmation via telemetry feedback. To simulate user

behavior in the network, two groups of clients were used: h2-h4 (regular clients) generated requests to the limited resource at uniform intensity, while h5-h6 (aggressive clients) generated excessive requests, emulating overload scenarios beyond acceptable limits.

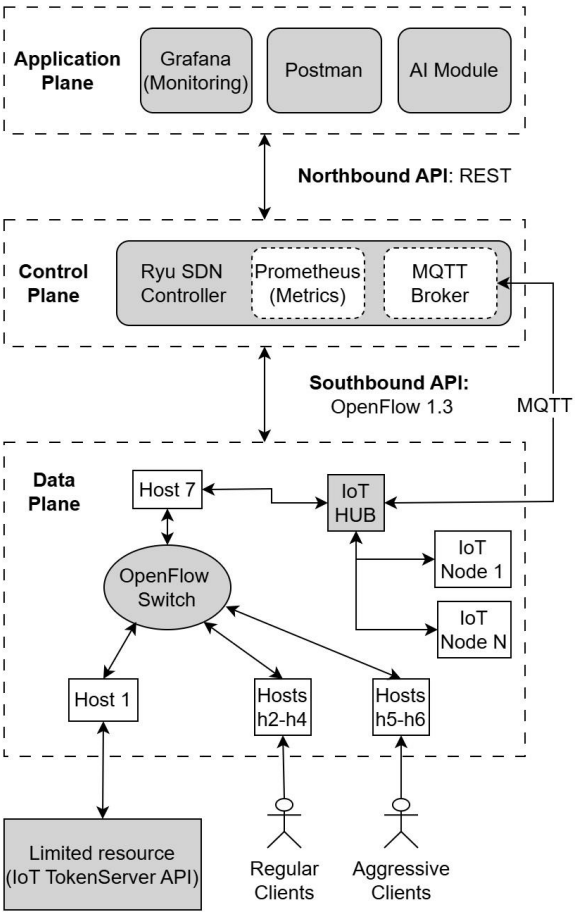


Fig. 6. Structure of the simulated experimental environment

The AI module was developed in Python using scikit-learn and integrated into the control plane. The AI model was trained to classify traffic and manage access of individual nodes to the limited IoT resource based on statistical data collected by the Ryu controller. This demonstrates how machine learning methods can extend the capabilities of the SDN controller through predictive decision-making and ensure fair (proportional) access to the limited resource before it becomes depleted. A complete description of the tools used is presented in Table 1.

This integrated configuration enabled the modeling of a realistic hybrid AI-SDN-IoT ecosystem, in which access to limited resources is managed adaptively through artificial intelligence algorithms. It also provided flexibility for experiment replication, result visualization, and expansion of the test environment through additional IoT or cloud service emulations.

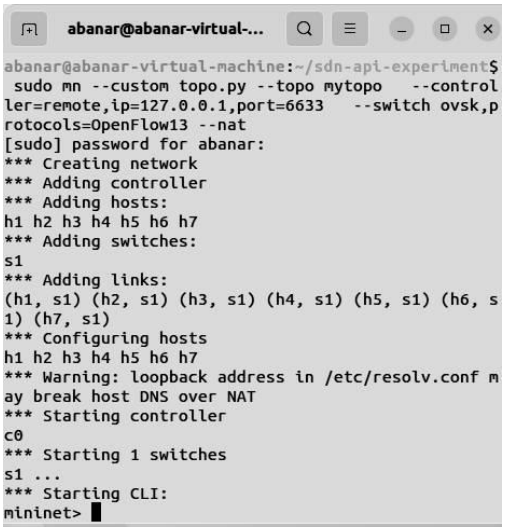


Fig. 7. Mininet terminal of the environment

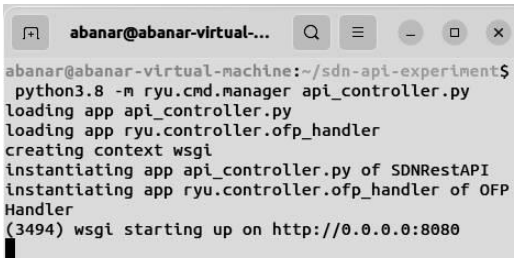


Fig. 8. Ryu terminal of the environment

Table 1

Platform and basic tools

Platform component	Tools used
OS and environment	Ubuntu 24.04 LTS (x86-64), Python 3.8
SDN network emulator	Mininet with a custom topology (Open vSwitch switch, hosts h1…h7, external SDN controller)
SDN controller	Ryu, OpenFlow 1.3 protocol, extended controller application with a northbound REST interface
Event broker	Eclipse Mosquitto for managing the IoT hub and publishing telemetry/command messages (MQTT) for IoT device control
Monitoring	Prometheus (metric collection) + Grafana (data visualization)
Control tools	Postman for interacting with the Application Plane API, Python traffic generation scripts (normal and aggredive modes)
AI module	Python + scikit-learn (Supervised Learning, Logistic Regression) as an Application Plane service that analyzes flows/events and triggers access policies for the limited resource
Limited resource and users	h1: IoT TokenServer API with an access quota (req/min). h2-h4: normal clients (Python module, steady uniform traffic to h1). h5-h6: burst clients (Python module, traffic spikes to h1). h7: IoT hub with connected devices

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

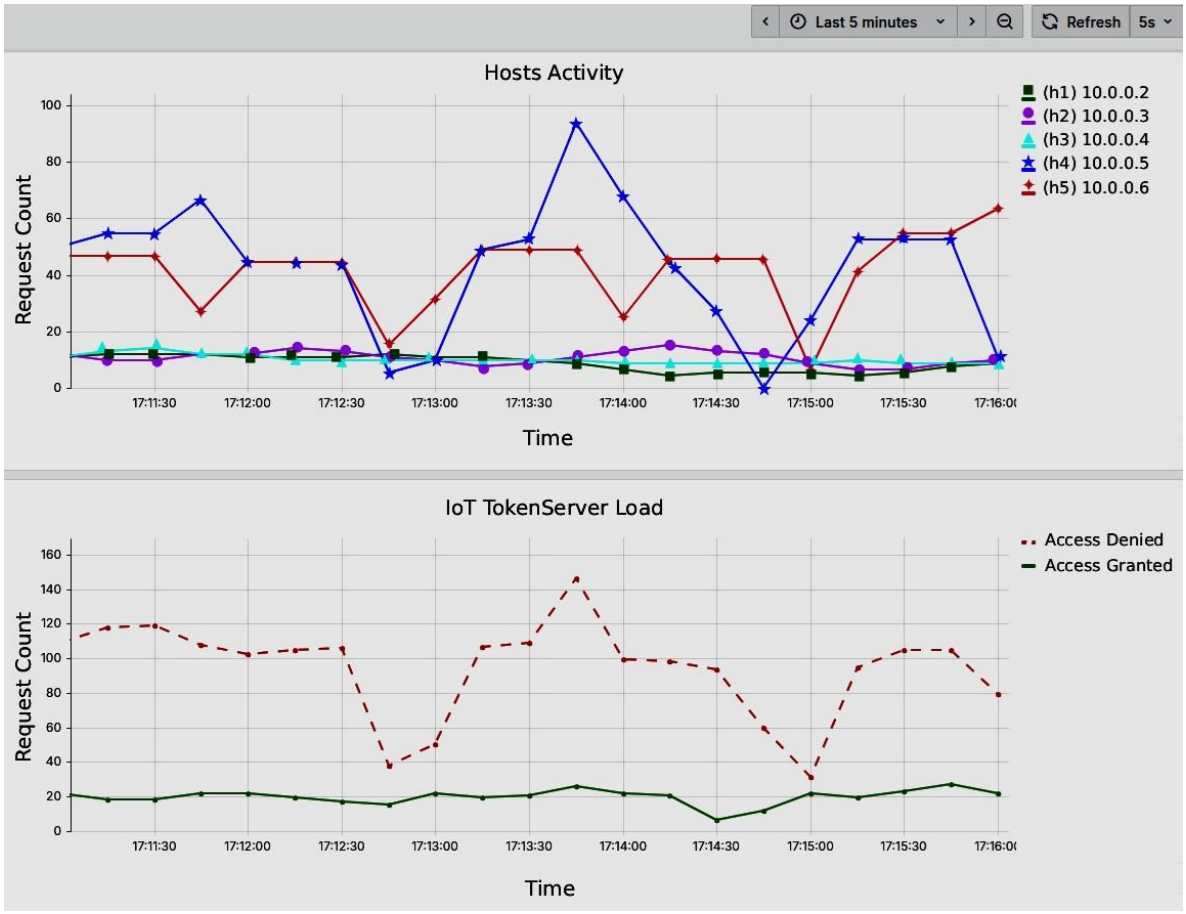


Fig. 9. Interactive network monitoring charts in Grafana

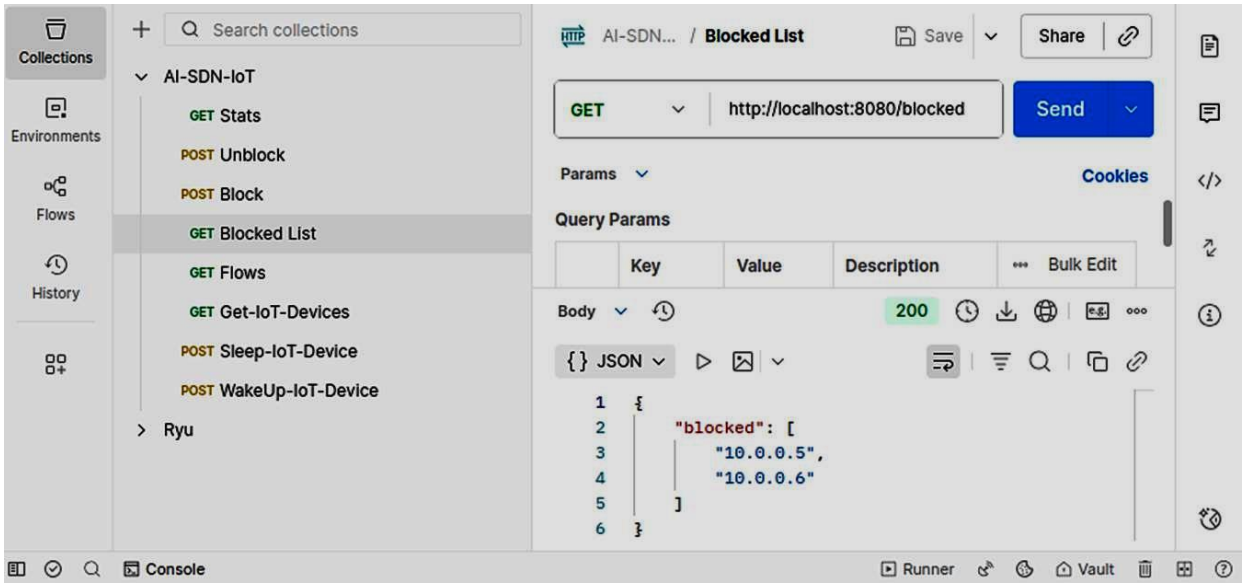


Fig. 10. REST API request window of the framework in Postman

3. Case study

The purpose of the case study experiment is to apply artificial intelligence algorithms to dynamically distribute access to a limited resource and to verify the functionality and usability of the framework for this kind of task. The IoT system is emulated by an IoT TokenServer API with a strict request-per-minute limit, while the client hosts may represent IoT devices or an IoT hub/gateway that generate requests to this service.

The algorithm of the proposed model of the experiment (Fig. 11) consists of three main stages: data preparation, AI module preparation, and testing with result evaluation. Each stage is divided into subtasks, enabling the trained AI model to be integrated into the SDN environment sequentially.

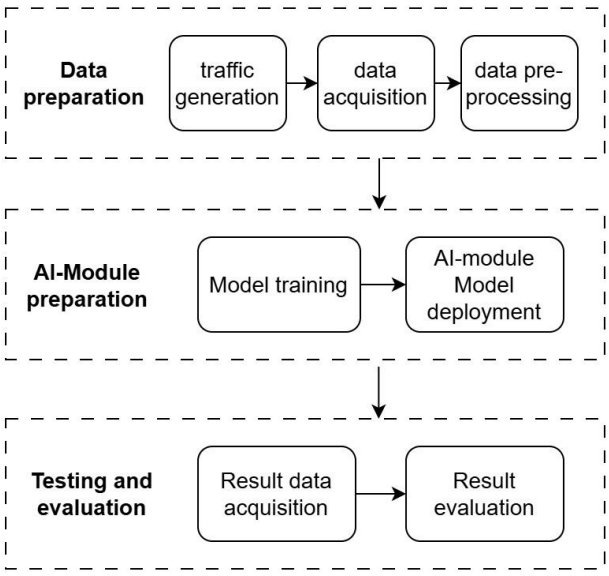


Fig. 11. Model of the experiment

- *Data Preparation*. To generate traffic, Python modules were developed that, as described in Table 1, simulate client behavior in two modes - normal (stable clients, h2-h4) and aggressive (peak load clients, h5-h6) - and send requests to the limited resource. The limited resource is represented by an IoT token-server API connected to host h1, with a limit of requests per minute. Data collection is performed by an event logger on the SDN controller and on h1, which periodically gathers traffic characteristics and exports them to CSV files. The log files contain key parameters summarized in Table 2.

The log files are divided into two types: API_log, which reflects direct HTTP requests from clients to the token server, and training_log, which records aggregated metrics collected from the SDN controller for subsequent AI module training. The combination of these two sources establishes a connection between the actual state of the resource, the controller's decisions, and client behavior over time. Merged into a single dataset, these files formed the foundation for further training and testing of the machine learning models.

The generated dataset included both stable request patterns and bursty overloads, ensuring sample representativeness. This contributed to effective model training and performance evaluation across different scenarios. Data preprocessing included standardizing data formats, handling missing values, and partitioning into training and test sets without temporal overlapping.

Table 2

Description of log file fields for training

Field	Description
ts	Timestamp (ms, Unix epoch) of the recorded request
src_ip	IP address of the source of traffic.
status	Server HTTP response code (200, 429)
count_60s	Number of requests from the client during the last minute
global_limit	Current resource limit (requests per minute)
total_60s	Total number of all requests to the resource during the last minute
share_prev	Client's share of resource usage
active_clients	Number of active clients
label_fair_prev	Ground truth label indicating the correctness of the previous resource allocation
reset_in_sec	Time (in seconds) remaining until the next global limit reset

- *AI-Module Preparation.* Based on the collected log files, a training dataset was formed to predict resource overuse (over_resource) in the SDN environment. To reduce dimensionality and eliminate correlated features, feature selection was performed, retaining only temporal and dynamic traffic parameters (count_60s, global_limit, active_clients, share_prev, status, total_60s). A logistic regression model from the scikit-learn package was trained on 80% of the dataset, with the remaining 20% used for testing. Logistic regression was chosen for its interpretability, robustness on small datasets, and ability to efficiently model the binary resource state (within_resource / over_resource) with minimal computational overhead, which is critical for real-time SDN operation. After achieving stable results, the model was saved in .joblib format for integration into the Python AI module script. For verification, the model was tested on new experimental data obtained during a repeated traffic run. The prediction results were saved in CSV files for subsequent comparison with the actual values (label_fair_prev).

During initialization, the AI model was loaded from the .joblib file. At each monitoring interval, a vector representing the current resource state was generated and passed to the model for class prediction: 0 - within_resource, 1 - over_resource. In the event of a predicted overload, the module dynamically added a drop-flow rule on the controller for the corresponding traffic source, restricting access to the resource for a specified timeout period (until the limited resource was refreshed). After the blocking interval ended, the rule was automatically removed. Thus, the prepared and deployed AI module enabled the integration of machine learning into the SDN controller's decision-making

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

loop, allowing adaptive real-time management of access to limited resources.

To formalize the resource management logic, the adaptation process is defined as a per-client quota function $q_i(t)$ (1), representing the individual access quota (allowed requests per minute) for the client *i* in the current control window:

$$q_i(t) = f(x_i(t)), \quad (1)$$

where $x_i(t)$ is the client feature vector built from recent traffic statistics and previous allocations:

$$x_i(t) = [\text{count\_60s}, \text{total\_60s}, \text{active\_clients}, \text{global\_limit}, \text{share\_prev}, \text{quota\_prev}, \text{is\_blocked}] \quad (2)$$

Based on vector (2), the trained AI model estimates a risk score $p_i(t)$ (3) (probability of resource overuse).

$$p_i(t) = g(x_i(t)), \qquad p_i(t) \in [0,1], \quad (3)$$

where $g(\cdot)$ is the trained ML classifier that outputs the overuse risk score, and T_{hight} is a predefined high-risk threshold (in our experiments $T_{hight} = 0.70$).

If $p_i(t) \geq T_{hight}$, the client's quota is adapted by applying a penalty step, otherwise, the baseline fair-share allocation is maintained, and any released capacity is redistributed to low-risk clients. The resulting quota vector $\{q_i(t)\}$ is dynamically updated and enforced via the SDN/REST control interface.

- *Testing and evaluation.* To assess AI integration, data were collected under two scenarios: Baseline (without AI) and AI-enabled. Logs from the limited resource and the SDN controller were aggregated into 5-minute windows (time-based grouping), and for each client (h2–h6), the number of attempts, successful requests, share of attempts, and success rate were calculated. The aggregated CSV files were used to generate comparative charts and compute summarized performance metrics.

For the trained logistic regression model that predicts limited resource overuse, standard evaluation metrics were applied: Accuracy (the overall proportion of correct model decisions), Precision (the proportion of correctly predicted positive cases among all positive predictions made by the model), Recall (the proportion of correctly identified actual positive cases among all true positives), and F1-score (the harmonic mean between Precision and Recall) (4) - (7).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (4)$$

where TP – (True Positive) - the model correctly predicted a resource overuse;

TN – (True Negative) - the model correctly predicted the absence of resource overuse;

FP – (False Positive) - the model incorrectly indicated a resource overuse that did not occur;

FN – (False Negative) - the model failed to detect an actual resource overuse.

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (5)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (6)$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (7)$$

The calculations were performed using the classification_report() and confusion_matrix() functions from the scikit-learn library. The results are presented as confusion matrices in both absolute (Fig. 12) and normalized values (Fig. 13). Overall, the model correctly identifies approximately 93% of within_resource cases and about 99% of over_resource cases, confirming its suitability for operation at the SDN control level.

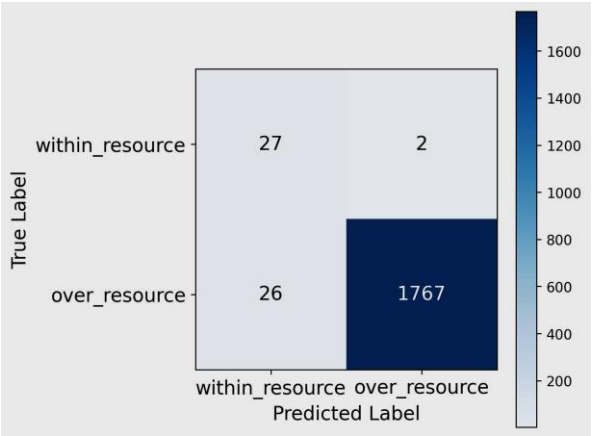


Fig. 12. Confusion matrix for the logistic regression model in absolute values

The impact of AI integration is illustrated by a two-panel scatter plot (Fig. 14): the x-axis represents the attempt share, and the y-axis shows the success share; the red dashed line y = x indicates ideal proportionality.

- *Without AI* (Fig. 14.1), the points are noticeably scattered relative to the y = x line: clients with a higher share of attempts do not consistently achieve a proportional share of successes; imbalances are visible

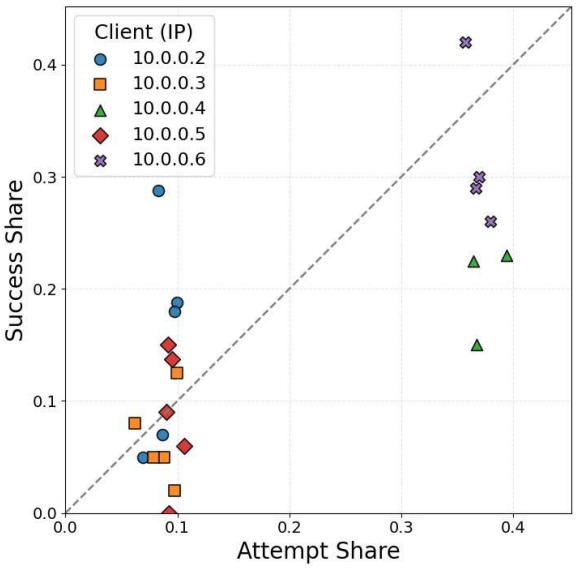
when an active client either receives excessive resources or fails to obtain sufficient successful requests during peak loads.

Fig. 13. Confusion matrix for the logistic regression model in normalized form

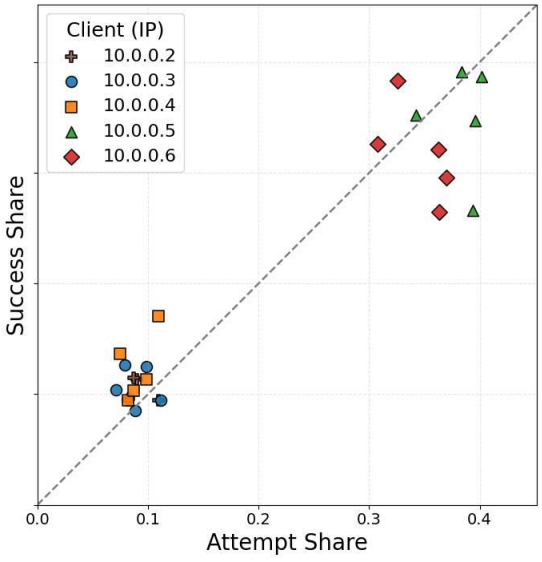
- *With AI* (Fig. 14.2), the cloud of points aligns more closely along the y = x line, indicating a trend toward proportional distribution: clients generating a higher share of attempts receive a comparable share of successes, while smaller clients are not "pushed out" of the resource during bursts and continue to receive their guaranteed resource quota.

The AI module stabilizes the distribution of requests among clients, maintaining fairness accuracy within the range of 90-100% (Fig. 15), whereas in the baseline case (without AI), accuracy fluctuates between 0-85%, indicating improved proportionality and predictability of access to the limited resource.

In this setup, explainability is achieved through the structural transparency of logistic regression, enabling direct inspection of how each traffic feature (such as "count_60s" or "active_clients") contributes to the predicted overload probability. This ensures that resource allocation decisions are not "black box" outcomes but



1)



2)

Fig. 14. Attempt vs. Success share for the scenarios: 1 – without AI; 2 – with AI

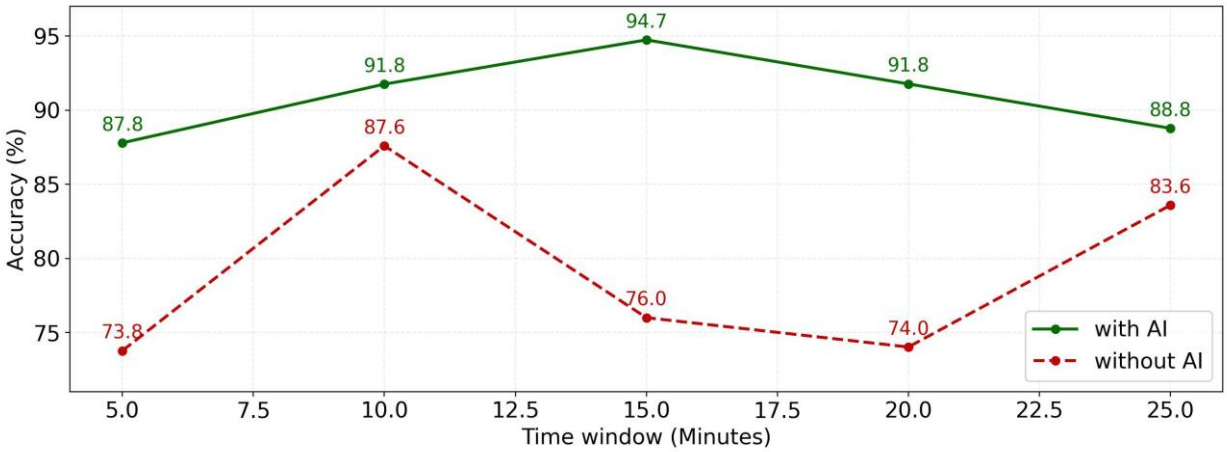


Fig. 15. Fairness accuracy over time

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

are explicitly linked to network state parameters via defined risk thresholds. Furthermore, trustworthiness is supported by a validation approach that relies on reproducible feature extraction and quantitative evaluation based on classification and error-rate metrics. To ensure operational safety, the control policy is bound by minimum service quotas and incremental penalty steps, preventing radical actions during transient network fluctuations. At the same time, full audit logging provides accountability for every decision made by the AI module.

4. Discussion

By integrating flexible resource management, AI-enabled traffic analysis, and distributed security mechanisms, the developed framework guarantees efficient operation even in large-scale, heterogeneous environments. Let us examine how the proposed framework tackles the main challenges faced by SDN in the context of IoT, namely:

- *Energy management*. The use of programmable interfaces in SDN networks [27] allows adaptive energy consumption based on current load and network performance requirements. At the application layer of the framework, an AI-based infrastructure employing deep reinforcement learning is used for power management. DRL utilizes statistical data and Q-learning algorithms [28] to optimize energy usage. The choice of EMQX as the MQTT broker implementation is justified by its ability to efficiently handle millions of connections and deliver MQTT messages with minimal latency, as confirmed by the study in [29];

- *Scalability*. The integration of the SDN approach into IoT simplifies this process for IoT networks by distributing controllers horizontally or hierarchically at the control layer, while maintaining centralized control over each distributed component [30]. SDN controllers enable dynamic changes to network topology without requiring physical modifications to its structure, simplifying the integration of new IoT devices into the overall ecosystem. The use of cloud infrastructure allows computational resources to be scaled to meet current network demands, eliminating the need to expand physical infrastructure. This allows a flexible response to changing loads by adding or removing resources in real time;

- *Security management*. AI-based approaches - specifically deep learning (DL) with CNN and LSTM - are used at the control layer for network traffic analysis and anomaly detection, enabling efficient identification of malicious traffic and appropriate threat response [17]. The flexibility of SDN allows the isolation of suspicious devices and the modification of routing paths to minimize risks. The integration of AI algorithms enhances the network's ability to predict threats, automate responses, and mitigate attacks [31]. In the event of an attack, SDN supports rapid network recovery, threat isolation, and uninterrupted IoT operation [32]. The advancement of blockchain technology - a decentralized ledger that immutably stores information in sequentially linked blocks has opened promising applications in IoT. [33, 34]. Physical integration is achieved using blockchain components (gateways, lightweight nodes, and computational nodes) to minimize the risks of unauthorized access and data tampering when accessing the cloud or establishing MQTT connections;

- *Load balancing*. The integrated AI traffic management module within the SDN controller uses supervised learning [35] and deep reinforcement learning (DRL) approaches to optimize routing and evenly distribute network load, ensuring stable performance even during peak conditions. The SDN paradigm, which separates the control plane from the data plane, enables dynamic traffic redistribution and latency minimization, both of which are essential for large-scale heterogeneous networks [36]. As a result, SDN helps maintain the stable operation of network nodes even as data volumes increase;

- *Interoperability.* The use of OpenFlow switches enables the integration of IoT devices of various types into a unified, manageable network. SDN promotes interoperability in IoT through flexible resource configuration [37]. For example, unified APIs enable integration of devices that use different communication protocols. IoT controllers that support the MQTT protocol operate on the Publish/Subscribe model, ensuring the universal exchange of control commands regardless of device platform.

The results of the simulated experiment showed that integrating the AI module into the SDN controller's decision-making loop not only ensures high accuracy in detecting resource overuse but also measurably improves the proportionality and stability of access distribution among clients. The average fairness of request distribution increased from 79.2% to 90.98% (an increase of 11.78 percentage points, 14.87% relative), which is significant for practical limited-resource SDN scenarios. Future research will focus on comparing different machine learning models (Random Forest, SVM, neural networks, etc.), utilizing alternative datasets with diverse traffic profiles, and evaluating the effectiveness of each approach in real-time, limited resource access management tasks.

5. Conclusions

The integration of artificial intelligence into software-defined networks for the Internet of Things enables adaptive resource management, enhances network security, and improves energy efficiency. The use of

SDN allows centralized control of traffic routing and increases the scalability of IoT networks.

The proposed hybrid Cloud-SDN-IoT framework successfully addresses the key tasks faced by modern SDN-IoT systems: energy management, scalability, enhanced security, load balancing, and interoperability among heterogeneous devices. It combines the computational power of the cloud, the local analytical capabilities of the SDN control layer, and the ability to dynamically balance traffic. The framework enables fair and stable access control to a limited IoT resource (the TokenServer API) under different traffic patterns, as experimentally validated in an emulation environment. The use of machine learning algorithms for network traffic analysis and anomaly detection enhances security by reducing the risk of attacks on the IoT infrastructure. The integration of blockchain technologies into the framework facilitates secure access management and transaction verification within the SDN environment, minimizing the risks of data tampering and unauthorized access. The use of the MQTT protocol enables efficient control of IoT devices, particularly for energy optimization and security purposes. By building the experiment on open-source components as Linux, the Ryu controller, and Mininet, the framework is fully reproducible, cost-free, and accessible to the research community. The developed tool is valuable for further scientific experiments, testing artificial intelligence algorithms, and validating hypotheses in the field of SDN-IoT without the need for specialized hardware.

In the conducted experiments, the AI-enabled control loop increased the average fairness of request distribution from 79.2% to 90.98% (an increase of 11.78 percentage points, 14.87% relative), demonstrating improved proportional access to the limited IoT TokenServer API while preserving stable, real-time request regulation.

Future research will focus on testing the framework on large synthetic datasets to evaluate its efficiency under controlled conditions, and then on modeling real operational scenarios to analyze performance, scalability, and reliability under various network conditions. Another important direction is to explore ways to reduce the cost of commercial deployment using cloud-based applications and to mitigate latency issues caused by the geographical distance of cloud data centers by expanding the use of edge and fog computing to enable partial local processing of IoT data.

Overall, the proposed architecture of the Hybrid Cloud-SDN-IoT framework enables the study and implementation of various policies for ensuring security and access to limited resources in information and communication networks and complex ecosystems, as well as the development of advanced smart algorithms for adaptive resource management and their distribution among system components and users.

Contributions of authors: conceptualization, formulation of tasks - **Heorhii Vorobets;** general structure of the work, comparison and drafting of manuscript - **Anatolii Banar**; visualization, software, verification - **Anatolii Banar;** analysis of results, review and editing - **Heorhii Vorobets, Anatolii Banar.**

Conflict of Interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, author ship or otherwise, that could affect the research and its results presented in this paper.

Financing

This study was conducted without financial support.

Data Availability

The manuscript contains no associated data.

Use of Artificial Intelligence

The authors confirm that they did not use artificial intelligence methods while creating the presented work.

All the authors have read and agreed to the published version of this manuscript.

References

1. Banar, A., & Vorobets', H. Perspektyvni napryamy rozvytku, udoskonalennya i zastosuvan' merezhi SDN na osnovi metodiv shtuchnoho intelektu [Promising Directions for Developing, Improving and Application SDN Networks Based on Artificial Intelligence Methods]. *Visnyk Khmel'nyts'koho natsional'noho universytetu. Seriya: Tekhnichni nauky - Herald of Khmelnytskyi National University. Technical Sciences*, 2025, vol. 355, no. 4, pp. 15-21. DOI: 10.31891/2307-5732-2025-355-1. (In Ukrainian).

2. Rukkas, K., Morozova, A., Uzlov, D., Kuznietcova, V., & Chumachenko, D. Optimizing Information Support Technology for Network Control: A Probabilistic-Time Graph Approach. *Radioelectronic and Computer Systems*, 2024, no. 2, pp. 85-97. DOI: 10.32620/reks.2024.2.08.

3. Chaganti, R., Suliman, W., Ravi, V., & Dua, A. Deep Learning Approach for SDN-Enabled Intrusion Detection System in IoT Networks. *Information*, 2023, vol. 14, article no. 41. DOI: 10.3390/info14010041.

4. Zabeehullah, Haq, Q. M. U., Arif, F., Khan, N. A., Anwar, M. S., & Alhalabi, W. A Secure AI Framework for Intelligent Traffic Prediction and Routing in SDN-Based Consumer Internet of Things. *IEEE Trans-*

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

actions on Consumer Electronics, 2025, vol. 71, no. 2, pp. 6294-6306. DOI: 10.1109/TCE.2025.3552609.

5. Zahoor, S., & Mir, R. N. Resource Management in Pervasive Internet of Things: A Survey. *Journal of King Saud University - Computer and Information Sciences*, 2021, vol. 33, no. 8, pp. 921-935. DOI: 10.1016/j.jksuci.2018.08.014.

6. Shuaib, M., Bhatia, S., Alam, S., Masih, R. K., Alqahtani, N., Basheer, S., & Alam, M. S. An Optimized, Dynamic, and Efficient Load-Balancing Framework for Resource Management in the Internet of Things (IoT) Environment. *Electronics*, 2023, vol. 12, no. 5, article no. 1104. DOI: 10.3390/electronics12051104.

7. Pandey, S., Chaudhary, M., & Tóth, Z. An Investigation on Real-Time Insights: Enhancing Process Control with IoT-Enabled Sensor Networks. *Discover Internet of Things*, 2025, vol. 5, article no. 29. DOI: 10.1007/s43926-025-00124-6.

8. Rawat, P., Rawat, G. S., Rawat, H., & Chauhan, S. Energy-Efficient Cluster-Based Routing Protocol for Heterogeneous Wireless Sensor Network. *Annals of Telecommunications*, 2025, vol. 80, no. 1, pp. 109-122. DOI: 10.1007/s12243-024-01015-7.

9. Prasad, V. K., Dansana, D., Bhavsar, M. D., Acharya, B., Gerogiannis, V. C., & Kanavos, A. Efficient Resource Utilization in IoT and Cloud Computing. *Information*, 2023, vol. 14, no. 11, article no. 619. DOI: 10.3390/info14110619.

10. Lilhore, U. K., Simaiya, S., Sharma, Y. K., Rai, A. K., Padmaja, S. M., Vajid, K. N., Kumar, V., Alroobaea, R., & Alsufyani, H. Cloud-Edge Hybrid Deep Learning Framework for Scalable IoT Resource Optimization. *Journal of Cloud Computing*, 2025, vol. 14, no. 1, article no. 5. DOI: 10.1186/s13677-025-00729-w.

11. Banar, A., & Vorobets', H. Khmarni SDN-kontrolery z pidtrymkoyu ShI: arkhitektura, masshtabovanist' ta bezpeka - porivnyal'ne doslidzhennya [AI-Enabled Cloud SDN Controllers: Architecture, Scalability, and Security - A Comparative Study]. *Bezpeka infokomunikatsiynykh system ta Internetu rechey - Security of Infocommunication Systems and Internet of Things*, 2025, vol. 3, no. 1, article no. 01011. DOI: 10.31861/sisiot2025.1.01011. (In Ukrainian).

12. Siddiqui, S., Hameed, S., Shah, S. A., Ahmad, I., Aneiba, A., Draheim, D., & Dustdar, S. Toward Software-Defined Networking-Based IoT Frameworks: A Systematic Literature Review, Taxonomy, Open Challenges and Prospects. *IEEE Access*, 2022, vol. 10, pp. 70850-70901. DOI: 10.1109/ACCESS.2022.3188311.

13. Udayaprasad, P. K., Shreyas, J., Srinidhi, N. N., Dilip Kumar, S. M., Dayananda, P., Askar, S. S., & Abouhawwash, M. Energy Efficient Optimized Routing Technique with Distributed SDN-AI to Large Scale I-IoT Networks. *IEEE Access*, 2024, vol. 12, pp. 2742-2759. DOI: 10.1109/ACCESS.2023.3346679.

14. Bekri, W., Jmal, R., & Chaari Fourati, L. Distributed Secured and Trustworthy IoT Framework Based on SDN, AI, and Blockchain. *Cluster Computing*, 2025, vol. 28, article no. 1065. DOI: 10.1007/s10586-025-05770-7.

15. Mishra, S. R., Shanmugam, B., Yeo, K. C., & Thennadil, S. SDN-Enabled IoT Security Frameworks - A Review of Existing Challenges. *Technologies*, 2025, vol. 13, no. 3, article no. 121. DOI: 10.3390/technologies13030121.

16. Sayegh, H. R., Dong, W., & Al-Madani, A. M. Enhanced Intrusion Detection with LSTM-Based Model, Feature Selection, and SMOTE for Imbalanced Data. *Applied Sciences*, 2024, vol. 14, no. 2, article no. 479. DOI: 10.3390/app14020479.

17. Wani, A., Revathi, S., & Khaliq, R. SDN-Based Intrusion Detection System for IoT Using Deep Learning Classifier (IDSIoT-SDL). *CAAI Transactions on Intelligent Technology*, 2021, vol. 6, no. 3, pp. 281-290. DOI: 10.1049/cit2.12003.

18. Kulkarni, M., Baddeley, M., & Haque, I. Embedded vs. External Controllers in Software-Defined IoT Networks. *Proceedings of the 2021 IEEE 7th International Conference on Network Softwarization (NetSoft)*, Tokyo, Japan, IEEE, 2021, pp. 298-302. DOI: 10.1109/NetSoft51509.2021.9492688.

19. Theodorou, T., & Mamatas, L. DENIS-SDN: Software-Defined Network Slicing Solution for Dense and Ultra-Dense IoT Networks, 2023. Available at https://arxiv.org/abs/2312.13662 (accessed 27.09.2025). DOI: 10.48550/arXiv.2312.13662.

20. Maleh, Y., Qasmaoui, Y., El Gholami, K., Sadqi, Y., & Mounir, S. A Comprehensive Survey on SDN Security: Threats, Mitigations, and Future Directions. *Journal of Reliable Intelligent Environments*, 2023, vol. 9, no. 2, pp. 201-239. DOI: 10.1007/s40860-022-00171-8.

21. Candal-Ventureira, D., Fondo-Ferreiro, P., Gil-Castiñeira, F., & González-Castaño, F. J. Quarantining Malicious IoT Devices in Intelligent Sliced Mobile Networks. *Sensors*, 2020, vol. 20, no. 18, article no. 5054. DOI: 10.3390/s20185054.

22. Lv, J., Babbar, H., & Rani, S. AI-Driven Resource Management for Energy-Efficient Aerial Computing in Large-Scale Healthcare SDN-IoT Systems. *IEEE Internet of Things Journal*, 2025, vol. 12, no. 13, pp. 23536-23549. DOI: 10.1109/JIOT.2025.3556943.

23. Masood, F., Khan, W. U., Alshehri, M. S., Alsumayt, A., & Ahmad, J. Energy Efficiency Considerations in Software-Defined Wireless Body Area Networks. *Engineering Reports*, 2024, vol. 6, no. 3, article no. 12841. DOI: 10.1002/eng2.12841.

24. Arevalo-Herrera, J., Mendoza, J., Torre, J., Zona-Ortiz, T., & Ramirez, J. Assessing SDN Controller Vulnerabilities: A Survey on Attack Typologies, Detection Mechanisms, Controller Selection, and Dataset Application in Machine Learning. *Wireless Personal Communications*, 2025, vol. 140, pp. 739-775. DOI: 10.1007/s11277-025-11748-w.

25. Ye, R., Ouyang, Y., & Che, X. Security and Attack Prevention in Software-Defined Network. *Proceedings of the 2024 International Conference on Tele-*

communications and Power Electronics (TELEPE), Frankfurt, Germany, IEEE, 2024, pp. 824-828. DOI: 10.1109/TELEPE64216.2024.00154.

26. Dhadhal, H., & Kotak, P. Leveraging Datasets for Effective Mitigation of DDoS Attacks in Software-Defined Networking: Significance and Challenges. *Radioelectronic and Computer Systems*, 2024, no. 2, pp. 136-146. DOI: 10.32620/reks.2024.2.11.

27. Santo, Y., Dalmazo, B. L., Cordeiro, W., Abelém, A., & Riker, A. A SDN-Based Approach for Conflicting Energy Profiles on Partially Sustainable IoT Networks. *Proceedings of the 2024 IEEE 10th World Forum on Internet of Things (WF-IoT)*, Ottawa, ON, Canada, IEEE, 2024, pp. 906-911. DOI: 10.1109/WF-IoT62078.2024.10811403.

28. Talib, M. M., & Croock, M. S. AI-Enhanced Power Management System for Buildings: A Review and Suggestions. *Journal Européen des Systèmes Automatisés*, 2023, vol. 56, no. 3, pp. 383-391. DOI: 10.18280/jesa.560304.

29. Kashyap, M., Dev, A. K., & Sharma, V. Implementation and Analysis of EMQX Broker for MQTT Protocol in the Internet of Things. *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, 2024, vol. 10, article no. 100846. DOI: 10.1016/j.prime.2024.100846.

30. Hamad, D., Yalda, K., Tapus, N., & Okumus, I. T. Enhancing IoT Scalability and Security through SDN. *Romanian Journal of Information Technology and Automatic Control)*, 2024, vol. 34, no. 2, pp. 113-126. DOI: 10.33436/v34i2y202409.

31. Kotsiopoulos, T., Radoglou-Grammatikis, P., Lekka, Z., Mladenov, V., & Sarigiannidis, P. Defending Industrial Internet of Things Against Modbus/TCP Threats: A Combined AI-Based Detection and SDN-Based Mitigation Solution. *International Journal of Information Security*, 2025, vol. 24, article no. 157. DOI: 10.1007/s10207-025-01076-2.

32. Karmakar, K. K., Varadharajan, V., Nepal, S., & Tupakula, U. SDN-Enabled Secure IoT Architecture. *IEEE Internet of Things Journal*, 2021, vol. 8, no. 8, pp. 6549-6564. DOI: 10.1109/JIOT.2020.3043740.

33. Ishaq, K., & Khan, F. Blockchain in the IoT Industry: A Systematic Literature Review, 2023. Available at https://arxiv.org/abs/2308.13613(accessed 27.09.2025). DOI: 10.48550/arXiv.2308.13613.

34. Zhao, D., Zhang, D., Pei, Q., Liu, L., & Yue, P. Blockchain-Based Security Deployment and Resource Allocation in SDN-Enabled MEC System. *IEEE Internet of Things Journal*, 2024, vol. 11, no. 24, pp. 40417-40430. DOI: 10.1109/JIOT.2024.3455425.

35. Masood, F., Khan, W. U., Jan, S. U., & Ahmad, J. AI-Enabled Traffic Control Prioritization in Software-Defined IoT Networks for Smart Agriculture. *Sensors*, 2023, vol. 23, no. 19, article no. 8218. DOI: 10.3390/s23198218.

36. Rostami, M., & Goli-Bidgoli, S. An Overview of QoS-Aware Load Balancing Techniques in SDN-Based IoT Networks. *Journal of Cloud Computing: Advances, Systems and Applications*, 2024, vol. 13, article no. 89. DOI: 10.1186/s13677-024-00651-7.

37. Marshoodulla, S. Z., & Saha, G. An Approach towards Removal of Data Heterogeneity in SDN-Based IoT Framework. *Internet of Things*, 2023, vol. 22, article no. 100763. DOI: 10.1016/j.iot.2023.100763.

Received 16.07.2025, Received in revised form 08.09.2025
Accepted date 17.11.2025, Published date 08.12.2025

АДАПТИВНЕ УПРАВЛІННЯ ОБМЕЖЕНИМИ РЕСУРСАМИ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ В ЕКОСИСТЕМАХ SDN-IOT

А. Ю. Банар, Г. І. Воробець

Предметом дослідження є інтеграція методів штучного інтелекту (ШІ) у програмно-конфігуровані мережі (SDN) для адаптивного керування доступом до обмежених ресурсів в інфраструктурі екосистем Інтернету речей (IoT). **Метою** роботи є розробка моделі та архітектурного рішення гібридного Cloud-SDN-IoT фреймворку з вбудованими ШІ-компонентами для інтелектуального розподілу мережевих та обчислювальних ресурсів та експериментальна перевірка покращення розподілу доступу до обмеженого IoT-ресурсу враховуючи різні шаблони трафіку у середовищі емуляції. Основні **завдання** дослідження: 1) аналіз сучасних підходів до енергоефективного управління ресурсами та безпеки в SDN-IoT мережах; 2) створення архітектури гібридного Cloud-SDN-IoT фреймворку, який поєднує централізоване керування SDN-мережі з гнучкістю хмарної інфраструктури; 3) розробка методології експерименту з використанням компонентів машинного навчання для покращення розподілу доступу до ресурсу та зменшення нерівномірності навантаження між конкуруючими клієнтами; 4) оцінювання ефективності системи відповідно до поставлених завдань та справедливого розподілу обмеженого IoT ресурсу шляхом аналізу розподілу запитів і точності розпізнавання перевищення доступу до ресурсу. У роботі запропоновано удосконалену трирівневу модель архітектури SDN із застосуванням ШІ аналітики: інфраструктурний рівень IoT-вузлів, рівень управління SDN та хмарний прикладний рівень. Експериментальна частина реалізована у віртуальному середовищі Linux, Mininet + Ryu, а навчена модель ШІ приймає рішення про розподіл обмеженого ресурсу. **Результати** експериментів показали, що інтеграція ШІ-модуля у контур SDN-контролера підвищує точність детекції перевищення доступу до ресурсу, зменшує нерівномірність навантаження між клієнтами та покращує стабільність розподілу запитів у реальному часі. **Висновки**. Наукова новизна отриманих результатів полягає у створенні моделі

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

відтворюваної гібридної архітектури Cloud-SDN-IoT, яка забезпечує адаптивне керування доступом до обмежених ресурсів IoT-вузлів шляхом поєднання централізованого SDN-контролю з прогнозною аналітикою ШІ, що дозволило визначити та реалізувати підходи для забезпечення ключових критеріїв SDN мережі (управління енергоспоживанням, забезпечення масштабованості, підвищення рівня безпеки, балансування навантаження та досягнення інтероперабельності). Інтеграція ШІ в контур керування SDN підвищив середній показник справедливості розподілу запитів з 79,2% до 90,98% (приріст становить 11,78 в абсолютному значенні та 14,87% у відносному), що показує покращення пропорційного доступу до обмеженого IoT TokenServer API із збереженням стабільного регулювання запитів у реальному часі. **Практичне значення** полягає у можливості застосування запропонованого підходу для оптимізації доступу до обмежених хмарних сервісів, API, енергоресурсів або IoT-пристроїв у системах «розумного міста», охорони здоров'я чи промислових мережах. **Подальші дослідження** передбачають розширення ШІ-компонентів різними моделями машинного навчання, формування нових наборів даних і порівняльну оцінку ефективності кожної моделі в завданні динамічного управління ресурсами SDN-IoT і відтворенні у реальних умовах.

Ключові слова: програмно-конфігуровані мережі; Інтернет речей; штучний інтелект; керування ресурсами; хмарна інфраструктура; SDN-контролер; машинне навчання.

Банар Анатолій Юрійович – асп. каф. радіотехніки та інформаційної безпеки, Чернівецький Національний університет імені Юрія Федьковича, Чернівці, Україна.

Воробець Георгій Іванович – канд. фіз.-мат. наук, доц., зав. каф. комп'ютерних систем та мереж, Чернівецький Національний університет імені Юрія Федьковича, Чернівці, Україна.

Anatolii Banar – PhD Student at the Department of Radioengineering and Information Security, Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine,
e-mail: banar.anatolii@chnu.edu.ua, ORCID: 0009-0006-7817-2058.

Heorhii Vorobets – PhD, Associate Professor, Head of the Department of Computer Systems and Networks, Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine,
e-mail:g.vorobets@chnu.edu.ua, ORCID: 0000-0001-8125-2047, Scopus Author ID: 8581629600.