Optimized Resource Management and Security Enhancement in Fog Computing using Advanced Q-Learning Approaches

Kusuma G. S.
Department of Electronics and Communication Engineering, The Oxford College of Engineering, Bangalore, Karnataka, India
kusuma.gs1988@gmail.com (corresponding author)

Manju Devi
Department of Electronics and Communication Engineering, The Oxford College of Engineering, Bangalore, Karnataka, India
manju3devi@gmail.com

Received: 17 March 2025 | Revised: 17 April 2025 | Accepted: 22 April 2025

 DOI: https://doi.org/10.48084/etasr.10995

ABSTRACT

Fog computing enhances cloud computing functionalities at the network edge, providing diminished latency and improved data processing. Optimizing resource allocation and energy consumption in this dispersed infrastructure is essential for fulfilling the requirement for efficient operations. This research presents a novel energy-efficient resource optimization algorithm designed for data centers in fog computing settings. The suggested methodology utilizes the Enhanced Q-Learning (EQL) and Secure Q-Learning (SQL) algorithms to achieve dynamic resource allocation and ensure operational efficacy. The proposed technique addresses energy consumption issues and guarantees optimal resource utilization by integrating adaptive incentive functions and safe state-action mechanisms, while ensuring robust security. The simulation outcomes with CloudSim exhibit substantial enhancements in energy efficiency, system throughput, and resource optimization relative to conventional approaches.

Keywords-fog computing;Q-learning; Reinforcement Learning (RL); Enhanced Q-Learning (EQL); Secure Q-Learning (SQL)

I. INTRODUCTION

Fog computing connects edge devices to centralized cloud resources, enabling low-latency services and distributed data processing close to end-users [1, 2]. Fog data centers handle dynamic, regionally distributed workloads, but struggle with resource management, energy efficiency, and maintaining performance standards. Inefficient use of resource increases both running expenses and environmental effects, making energy-aware solutions essential. Traditional resource allocation techniques, often rule-based or heuristic, lack the flexibility to accommodate real-time workload changes and typically ignore security issues, including intrusion and misuse. This leads to inefficiencies and compromises system performance and dependability.

This paper uses Reinforcement Learning (RL), in particular Q-Learning, for dynamic resource optimization in order to overcome these constraints. Adaptive reward mechanisms in Enhanced Q-Learning (EQL) are used to increase energy efficiency and system performance. At the same time, Secure Q-Learning (SQL) adds security measures to guarantee safe resource allocation and prevent intrusion. Together, EQL and SQL provide a powerful, energy-efficient, and secure system for controlling fog computing systems.

The primary contributions of the paper are as follows:

- Energy-efficient EQL: Introduces a novel framework that integrates sophisticated, adaptive reward systems to dynamically manage energy consumption while maintaining performance balance. EQL utilizes real-time feedback to guarantee that fog data centers achieve maximum resource usage, significantly reducing energy waste and operating expenses.
- Robust SQL: Augments the conventional Q-Learning method by incorporating sophisticated security protocols to safeguard against hostile attacks. SQL protects the resource management process, assuring the integrity and dependability of decision-making against potential threats such as intrusion and malicious resource manipulation, while maintaining the overall security posture of the system.

- Thorough validation and performance assessment: Assesses the efficacy of EQL and SQL using stringent CloudSim simulations in the context of fog computing. The findings indicate significant enhancements in energy efficiency and resource optimization, surpassing conventional techniques in terms of energy consumption, system throughput, and resilience to security threats. This thorough assessment highlights the practical applicability and scalability of the proposed approach in real-world fog systems.

The main objective of this research is to introduce a self-adaptive resource management framework that guarantees optimal utilization of fog computing resources, enhances energy efficiency, and safeguards the system against security threats. This work employs comprehensive simulations with CloudSim to assess the efficacy of the suggested technique in terms of energy savings, system throughput, and security robustness. The results underscore the promise of using reinforcement learning techniques to establish a more secure, energy-efficient, and robust fog computing environment.

Authors in [1] highlight the drawbacks of conventional cloud computing for IoT, and offer a fog computing architecture to reduce latency and enhance data handling at the edge. Although it improves security and performance, issues such as network variability, interoperability, and infrastructure adequacy remain. Although energy consumption and the limited deployment of emerging technologies persist as challenges, fog computing is essential for latency-sensitive IoT applications [2]. Furthermore, a decentralized ESBAC framework [3] has been demonstrated to manage EMR access securely in fog-IoT environments using smart contracts, despite the potential for scalability and integration issues.

EEOIT [4] enhances the security of IoT task execution by identifying hostile fog nodes; however, its efficacy is dependent on accurate threat detection. A study of a fog-based IDS employing machine learning [5] demonstrates a high capacity for intrusion detection, despite limitations imposed by the dataset and computational resources. According to authors in [6], fog computing also enhances IoT scalability and privacy, although it adds complexity and vulnerabilities in managing security protocols and standards. The evolution of IoT [7] has driven edge/fog computing, despite the lack of standards and ongoing security issues that hinder effective deployment.

A survey [8] presents architectures and issues in achieving interoperable communications and shows the functionality of fog in IoT applications, including smart transportation and healthcare. Although there are still issues with standardization, fog/edge computing [9] offers low-latency processing and security with inventions such as autoscaling and encryption.

Authors in [10] propose a secure resource allocation model using linear optimization that balances security and performance in fog environments, but can be complicated and simulation-dependent. Fog computing [11, 12] in healthcare supports local processing and blockchain-based security, but has issues with record management and interoperability.

Despite being limited in their general relevance, authors in [13] assess AES encryption in fog-IoT, revealing trade-offs in key size and efficiency. iFogSim2 [14] enhances fog resource allocation, thereby reducing latency and energy consumption. However, its efficacy is constrained by scalability and simulation dependency. Finally, fog computing in IoT [15] increases responsiveness and reduces latency, but raises new security concerns. The study includes future research directions, security solutions, and fog-based architectures.

II. PROPOSED METHODLOGY

The proposed study aims to address two significant difficulties in fog computing: enhancing energy efficiency and guaranteeing strong security in resource allocation within decentralized infrastructures. We present two novel algorithms, energy-efficient EQL and robust SQL, which combine dynamic resource allocation with adaptive security measures to guarantee excellent performance and durability. Figure 1 shows the block diagram of the proposed system.

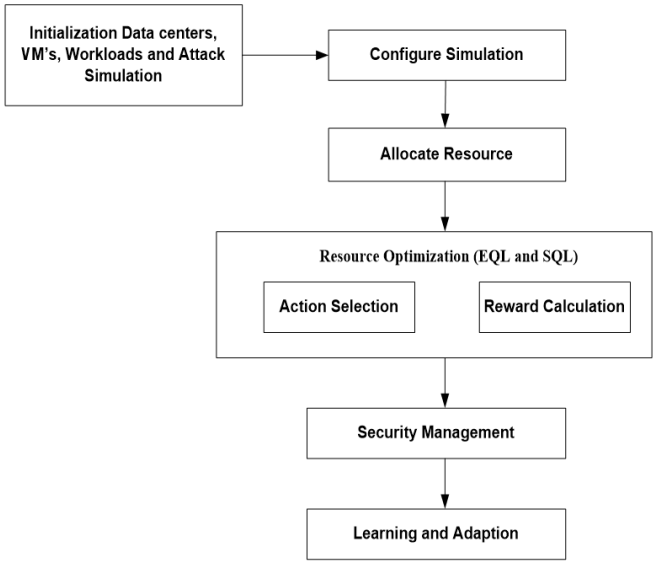

Fig. 1. Proposed model.

A. Energy-Efficient Enhanced Q-Learning

In fog computing environments, EQL is a dynamic resource allocation system that adapts to changing workload and energy consumption patterns. It promotes energy efficiency and reduces unnecessary consumption by dynamically adjusting resource distribution based on real-time performance indicators through the use of an adaptive reward mechanism. Additionally, EQL continuously improves its policy through adaptive learning, which improves resource utilization in a variety of contexts.

The proposed work aims to reduce energy usage while maintaining good management of changing workloads. EQL's adaptive method guarantees that the system can dynamically adjust resources, resulting in significant energy savings.

B. Robust Secure Q-Learning

Security in fog computing is a paramount issue due to the decentralized architecture of the infrastructure. The system's susceptibility to cyberattacks and nefarious actions could undermine its performance and integrity. SQL mitigates these difficulties by incorporating security features into the Q-learning process, guaranteeing that resource allocation prioritizes both performance and the system's security posture. The primary characteristics of SQL comprise:

- Dynamic security score: Each host within the fog infrastructure is allocated a securityScore that indicates its security condition. The score changes dynamically based on real-time variables, including vulnerability fixes, active attacks, and security events. A higher score signifies a more secure host, whereas hosts with lower security scores are penalized in the Q-learning process.
- Compromised detection: The isCompromised flag designates hosts as compromised if their securityScore falls below a specified threshold. These hosts are then excluded from the resource allocation to preserve system integrity. This mitigates threats such as illegal access or malicious manipulation of resources.
- Secure reward function: In SQL, the reward function evaluates not only system performance, but also the security condition of the host. Hosts exhibiting enhanced securityScores receive enhanced allocation priority, guaranteeing that resource allocation favors secure hosts over those that are susceptible.
- Compromised mitigation: SQL incorporates techniques to identify security breaches, including failed authentication attempts or unauthorized access to resources. Upon detection of such breaches, the system can take actions such as diminishing resource allocation to the compromised hosts or notifying administrators of the security threats.
- Logging and security alerts: The SQL framework records critical security incidents, including identified attacks or compromised hosts. These logs can activate alarms, allowing administrators to respond promptly to potential risks. The system also generates error reports and security alerts, guaranteeing that all security threats are recorded for subsequent examination.

C. Integration of Enhanced Q-Learning and Secure Q-Learning algorithms

The suggested solution integrates EQL and SQL to address energy efficiency and security concurrently. This integration guarantees that energy efficiency is prioritized while maintaining the integrity of the system. Resource allocation automatically adjusts to real-time workloads, taking into account both performance requirements and the security condition of hosts. Secure decisions involve the selection of hosts with elevated security ratings, while eschewing compromised hosts that could potentially compromise system integrity. By integrating EQL and SQL, the system achieves a harmonious strategy for resource optimization, enhancing energy efficiency and bolstering security while preserving performance.

D. Algorithm Overview

The algorithm functions through the following essential steps:

1. Initialization: The system configures all hosts with default security scores and resources. The Q-learning method is initiated with the establishment of a preliminary policy for resource distribution.
2. Monitoring and evaluation: A real-time evaluation of system metrics (e.g., CPU utilization, memory utilization) and security status (i.e., security score) is conducted. The system periodically evaluates the security posture of all hosts and computes the reward for each potential action (i.e., resource allocation decision).
3. Action selection: The system determines the subsequent action (i.e., resource allocation choice) based on the current Q-values, employing the exploration-exploitation tradeoff. The security rating of each host is incorporated into the reward function, guaranteeing that secure hosts receive precedence.
4. Reward calculation: The reward function is dynamically updated, taking into account performance metrics (throughput, CPU, RAM) and security parameters (security score, compromised status). The system calibrates the reward in accordance with the energy expenditure and security threats associated with each decision.
5. Security management: In the event that a host's security score declines to a specified level, it is designated as compromised and removed from resource allocation. The system is designed to record any instance of security violations or anomalous actions.
6. Learning and adaption: The system employs Q-learning to perpetually enhance its resource allocation strategy. The system progressively learns to enhance energy efficiency while mitigating security threats.

The suggested system seeks to enhance energy efficiency through the implementation of dynamic energy-saving strategies, hence decreasing energy consumption compared to static or heuristic-based methods. It also improves security by favoring secure hosts and eliminating compromised ones, thereby increasing the resilience of fog infrastructures against cyber-attacks. The solution is versatile for many fog computing scenarios. The research introduces a technique for enhancing energy efficiency and security in fog computing through the amalgamation of energy-efficient EQL and robust SQL algorithms. This method guarantees effective resource distribution and emphasizes security, safeguarding infrastructure against attacks. The system provides a scalable and adaptable solution for enhancing performance, security, and sustainability in fog computing environments.

III. SIMULATION AND EVALUATION SETUP

We assess the performance and efficacy of the proposed energy-efficient EQL and robust SQL algorithms in fog computing environments through simulations utilizing the

existing cloud simulation tool, CloudSim. CloudSim facilitates the modeling and simulation of cloud and fog environments, rendering it an optimal selection for our assessment framework. This section describes the simulation environment, setup, and assessment criteria employed to evaluate the efficacy of the proposed strategy.

A. Simulation Environment

The simulation environment is intended to emulate the attributes of fog computing infrastructures, including both performance and security measurements. The environment consists of many fog data centers (hosts), each exhibiting distinct performance attributes (CPU, memory, storage) and security conditions. The system's key characteristics are as follows:

- Hosts (data centers): Each host signifies a computational unit within the fog environment. Hosts can be defined by many performance metrics, including CPU capacity, memory, and storage. Moreover, each host possesses a dynamic security score, shaped by elements like vulnerability patches, identified threats, and system health.
- Virtual Machines (VMs): The simulation encompasses many VMs installed on the hosts. VMs signify applications or services operating on the fog infrastructure. VMs possess specific resource requirements, including CPU and memory, which are dynamically allocated according to the prevailing system conditions and workload.
- Workload: The system emulates fluctuating workloads, wherein several VMs exhibit distinct requirements for CPU, memory, and network bandwidth. The workload is variable and evolves over time, emulating real-world situations where resource demands fluctuate.
- Security simulation: Each host is allocated a security score that may be influenced by active attacks or vulnerabilities. Hosts may be designated as compromised if their security score drops below a specified level, signifying their unsuitability for resource distribution.

B. Configuration of the Simulation

CloudSim is set up using the following parameters:

- Quantity of hosts: The number of fog nodes (hosts) utilized in the simulation. Each host possesses a designated quantity of CPU, memory, and storage space.
- Quantity of VMs: The number of VMs operating on each host, exhibiting diverse resource requirements for CPU and memory.
- Workload patterns: A combination of stable and variable workloads, exemplifying standard fog computing applications such as IoT services, real-time analytics, and data processing jobs.
- Security metrics: The security rating for each host, which affects the reward computation in the SQL algorithm. Security metrics are continuously revised depending on simulated attack scenarios or effective vulnerability fixes.
- Attack simulation: Periodic random attacks are executed to diminish the security score of certain hosts, emulating breaches, malware incidents, or denial of service attacks.

The parameters for 20 different jobs in a scheduling system are presented in Table I. Each row represents a single task, and columns provide the following information:

- Req ID is an identity that is specific to each job.
- Million Instructions Per Second (MIPS) is the number of instructions required by the task.
- Time of execution is the estimated time the task is expected to take to complete.
- Arrival time is the time that the work is entered into the system or becomes available for processing.
- Deadline is the time by which the task must be completed in order to meet its requirements or to avoid trouble with the authorities.

TABLE I. JOB SCHEDULING PARAMETERS WITH 20 JOBS

Req ID	MIPS	Time of execution	Arrival time	Deadline
0	1000	4	1	10
1	2000	5	2	15
2	1000	4	3	10
3	2000	3	4	13
4	1000	4	8	18
5	2000	5	7	25
6	1000	4	5	20
7	2000	3	7	23
8	1000	3	7	13
9	1000	3	4	15
10	1000	3	9	19
11	1000	4	5	24
12	2000	3	7	21
13	1000	3	7	11
14	1000	3	4	20
15	1000	3	9	16
16	1000	4	5	22
17	2000	5	9	19
18	1000	5	7	17
19	1000	4	8	23

C. Evaluation Metrics

The suggested algorithms, EQL and SQL, are assessed according to the following performance and security metrics:

- Energy efficiency: The aggregate energy consumption of the fog infrastructure is quantified throughout the simulation duration. Energy conservation is a key objective of EQL, and its efficacy is assessed by comparing it with conventional static or heuristic resource allocation methodologies. Energy consumption is determined by (1):

$$Energy\ Consumption = \sum_{i=1}^{n}(CPU\ Usage_i + Memory\ Usage_i) * Energy\ Factor \quad (1)$$

where n represents the number of VMs, and $Energy\ Factor$ denotes the energy consumption per unit of CPU or memory usage.

- System throughput: This statistic assesses the overall performance of the system by quantifying the entire work accomplished by the fog infrastructure during a certain timeframe. It is quantified as the quantity of jobs or data handled per unit of time. High throughput signifies that the system is effectively managing workloads. The equation for assessing system throughput in a fog computing context is expressed in (2):

$$Throughput = \frac{Total\ Number\ of\ Tasks\ Processed}{Total\ Time} \quad (2)$$

where $Total\ Number\ of\ Tasks\ Processed$ is the aggregate number of tasks, jobs, or data units effectively processed by the fog infrastructure during a certain timeframe. $Total\ Time$ is the cumulative time over which system performance is evaluated, often expressed in seconds, minutes, or hours. This formula determines the pace at which tasks are processed by the system, with increased throughput signifying efficient workload management within the specified time period.

D. Simulation Results

The simulation results are examined according to the aforementioned assessment metrics. Comparisons are conducted between conventional methods and the suggested methodologies (EQL, SQL, and the integrated solution) to ascertain the advantages of dynamic, energy-efficient, and secure resource allocation. Key findings include:

- Energy savings: The proposed EQL-based method is anticipated to substantially decrease energy consumption relative to static allocation solutions by dynamically reallocating resources according to real-time needs and system performance.
- Enhanced security: The SQL-based methodology fortifies the fog architecture by eliminating compromised hosts from resource allocation, thereby preventing potential attacks from impacting the system.
- System throughput: Both EQL and SQL algorithms are anticipated to maintain high system throughput, guaranteeing efficient management of dynamic workloads without jeopardizing security or performance.

The simulation and assessment framework offers a thorough methodology for assessing the performance and security of the proposed EQL and SQL algorithms in fog computing environments.

IV. RESULTS AND DISCUSSION

The results reveal how the proposed integrated method, the SQL method, and the EQL method compare in terms of energy consumption when using different workloads. Table II provides a comprehensive overview of the energy consumption statistics.

TABLE II. COMPARISON OF ENERGY EFFICIENCY

Algorithm	No of jobs	Energy consumed (J)
EQL	10	9,620
SQL	10	6,997
Integrated method	10	6,100
EQL	15	12,760
SQL	15	8,902
Integrated method	15	8,000
EQL	20	16,950
SQL	20	11,941
Integrated method	20	10,930
EQL	25	19,800
SQL	25	14,300
Integrated method	25	13,100
EQL	30	23,750
SQL	30	16,900
Integrated method	30	15,200

The integrated method and SQL have been demonstrated to provide significant energy savings in comparison to the EQL algorithm. However, the integrated method consistently exhibits superior energy efficiency and demonstrates additional gains over the separate SQL. To illustrate, when 30 jobs are considered, the integrated method exhibits a 35.5% energy reduction in comparison to the EQL, as opposed to approximately 29.5% for SQL.

The SQL algorithm's scalability and efficacy in energy-efficient resource management are highlighted by the fact that its energy-saving benefits become increasingly apparent as the workload grows. The integrated method further extends this trend, maintaining superior efficiency with increasing job loads.

A visual comparison of the energy consumption of the three techniques-EQL, SQL, and the integrated method-is shown in Figure 2. The figure reveals a clear decline in energy use as the number of jobs increases. The integrated method is the most efficient technique under all examined loads.

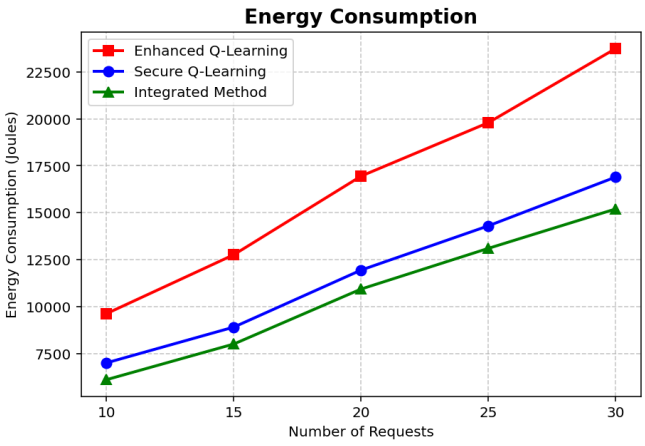

Fig. 2. Comparison graph for energy consumption.

Furthermore, the results provide a comparative analysis of the throughput of SQL, EQL, and the proposed integrated method for different workloads. The throughput values are summarized in Table III. The integrated method always outperforms SQL in terms of throughput, with gains ranging from about 5% to about 9%, as shown in Figure 3. This implies that the integration not only maintains the scalability and adaptability of SQL, but also improves it more in high-load situations.

TABLE III. COMPARISON OF THROUGHPUT

Algorithms	No of Jobs	Throughput
EQL	10	1
SQL	10	11
Integrated method	10	11.6
EQL	15	1.3
SQL	15	16
Integrated method	15	17.2
EQL	20	1.5
SQL	20	20.5
Integrated method	20	22.4
EQL	25	1.7
SQL	25	25
Integrated method	25	27.2
EQL	30	2.0
SQL	30	30
Integrated method	30	32.8

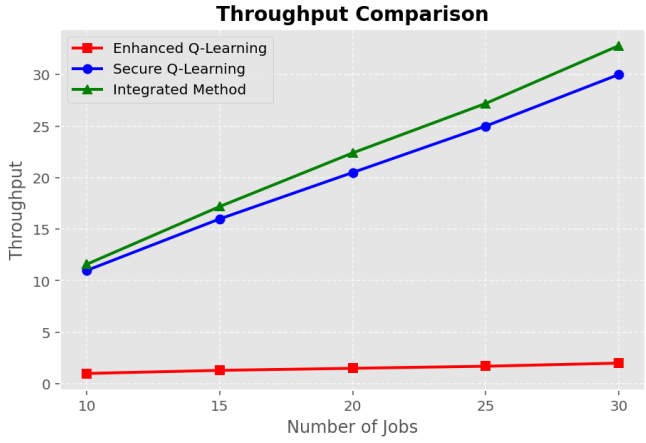

Fig. 3. Comparison graph for throughput.

V. CONCLUSION

The research presents an innovative approach for energy-efficient and secure resource management in fog computing environments, integrating energy-efficient Enhanced Q-Learning (EQL) with robust SQL (SQL) algorithms. The EQL dynamically modifies resource allocation to reduce energy consumption using adaptive incentive systems, whereas the SQL integrates security measurements to detect and eliminate compromised hosts. CloudSim simulations demonstrated notable advancements in energy efficiency, system throughput, resource optimization, and improved resistance to security attacks. The EQL-SQL framework maintained increased performance and optimal resource utilization despite fluctuating workloads and attack scenarios. This methodology balances performance improvement with cybersecurity in fog computing, rendering it applicable to diverse IoT, edge computing, and real-time data processing scenarios. Future work will concentrate on expanding this technique to encompass multi-cloud and hybrid fog-cloud systems, integrating more intricate attack scenarios, and enhancing the balance between security measures and system performance. Advanced machine learning methodologies, such as deep reinforcement learning, can enhance the decision-making capabilities of the framework.

REFERENCES

[1] G. Khan, K. K. Gola, R. Kanauzia, and S. Kumar, "Secure Architecture to Support IoT based on Fog Computing," *Procedia Computer Science*, vol. 215, no. C, pp. 608–617, Jan. 2022, https://doi.org/10.1016/j.procs.2022.12.063.

[2] P. Vashisht, S. B. Bajaj, and A. Narang, "Energy-Efficient Fog Computing: A Review and Future Directions," *International Journal of Innovative Research in Computer Science and Technology*, vol. 12, no. 2, pp. 135–139, Apr. 2024, https://doi.org/10.55524/ijircst.2024.12.2.24.

[3] N. Ranjan, B. Balkhande, M. Ghuge, B. S. Dakhare, B. R. Singh, and A. Shukla, "Efficient and Secure Blockchain-Based Access Control for Fog-Assisted IoT Cloud in Electronic Medical Records Sharing," *Journal of Electrical Systems*, vol. 20, no. 1s, pp. 138–151, Mar. 2024, https://doi.org/10.52783/jes.759.

[4] A. N. Alvi, B. Ali, M. S. Saleh, M. Alkhathami, D. Alsadie, and B. Alghamdi, "Secure Computing for Fog-Enabled Industrial IoT," *Sensors*, vol. 24, no. 7, Apr. 2024, Art. no. 2098, https://doi.org/10.3390/s24072098.

[5] P. Kumar, G. P. Gupta, and R. Tripathi, "A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 10, pp. 9555–9572, Oct. 2021, https://doi.org/10.1007/s12652-020-02696-3.

[6] S. Sicari, A. Rizzardi, and A. Coen-Porisini, "Insights into security and privacy towards fog computing evolution," *Computers & Security*, vol. 120, Sep. 2022, Art. no. 102822, https://doi.org/10.1016/j.cose.2022.102822.

[7] C. Avasalcai, I. Murturi, and S. Dustdar, "Edge and Fog: A Survey, Use Cases, and Future Challenges," in *Fog Computing: Theory and Practice*, A. Zomaya, A. Abbas, and S. Khan, Eds. Hoboken, NJ, USA: John Wiley & Sons, Ltd, 2020, pp. 43–65, https://doi.org/10.1002/9781119551713.ch2.

[8] A. Hazra, P. Rana, M. Adhikari, and T. Amgoth, "Fog computing for next-generation Internet of Things: Fundamental, state-of-the-art and research challenges," *Computer Science Review*, vol. 48, May 2023, Art. no. 100549, https://doi.org/10.1016/j.cosrev.2023.100549.

[9] T. Kim, S. Yoo, and Y. Kim, "Edge/Fog Computing Technologies for IoT Infrastructure," *Sensors*, vol. 21, no. 9, May 2021, Art. no. 3001, https://doi.org/10.3390/s21093001.

[10] K. Massey, N. Moazen, and T. Halabi, "Optimizing the Allocation of Secure Fog Resources based on QoS Requirements," in *2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, Washington, DC, USA, 2021, pp. 143–148, https://doi.org/10.1109/CSCloud-EdgeCom52276.2021.00035.

[11] V. K. Quy, N. V. Hau, D. V. Anh, and L. A. Ngoc, "Smart healthcare IoT applications based on fog computing: architecture, applications and challenges," *Complex & Intelligent Systems*, vol. 8, no. 5, pp. 3805–3815, Oct. 2022, https://doi.org/10.1007/s40747-021-00582-9.

[12] C. Awasthi, M. Nawal, and P. K. Mishra, "Security Concerns of Fog Computing in Field of Healthcare using Blockchain: A Review," in *2021 International Conference on Communication information and*

Computing Technology, Mumbai, India, 2021, pp. 1–5, https://doi.org/10.1109/ICCICT50803.2021.9510166.

[13] W. Wu, "Application and Effectiveness of IoT Edge and Fog Computing Technologies in Smart Energy Development with the Use of Encryption Algorithms and Security Systems," *Computing and Informatics*, vol. 43, no. 5, pp. 1029–1052, Oct. 2024, https://doi.org/10.31577/cai_2024_5_1029.

[14] A. M. Ghalwah and G. A. Al-Sultany, "Leveraging Community-based Approaches for Enhancing Resource Allocation in Fog Computing Environment," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 20372–20378, Feb. 2025, https://doi.org/10.48084/etasr.9206.

[15] M. Burhan *et al.*, "A Comprehensive Survey on the Cooperation of Fog Computing Paradigm-Based IoT Applications: Layered Architecture, Real-Time Security Issues, and Solutions," *IEEE Access*, vol. 11, pp. 73303–73329, 2023, https://doi.org/10.1109/ACCESS.2023.3294479.