WbAES-Powered Edge Computing for Secure Healthcare IoT Data Management

Prakhar Shukla[1], Areeb ur Rub[1], Hardik Prakash[1] and Sambit Satpathy[2]
[1]*Student, Galgotias College of Engineering and Technology*
[2]*Associate Professor, Galgotias College of Engineering and Technology*

***Abstract*—This paper presents a comprehensive approach to securing data transmission in IoT-based healthcare systems through Whale-based Advanced Encryption Standard (WbAES) and edge computing integration. The research addresses the critical challenges of maintaining data privacy and security in healthcare IoT devices while ensuring efficient and reliable communication under resource constraints. We propose a novel Medical Edge Computing (MEC) framework that combines bio-inspired encryption algorithms with edge-based processing to achieve both robust security and low-latency performance. Our WbAES implementation demonstrates significant improvements in computational efficiency and energy consumption compared to traditional approaches, making it particularly suitable for resource-constrained healthcare IoT environments. Experimental results with up to 500 concurrent devices validate the system's scalability, reliability, and security characteristics.**

***Index Terms*—Medical Edge Computing (MEC), Whale-based AES encryption, healthcare IoT security, real-time data processing, priority-based analysis, multicast device discovery, resource constrained devices**

I. INTRODUCTION

The integration of Internet of Things (IoT) devices into healthcare systems has initiated a paradigm shift in medical data collection and processing. While these connected medical devices offer real-time patient monitoring capabilities, they also present significant challenges in security, efficiency, and scalability. Traditional cloud-based systems, despite their storage benefits, often fall short in time-sensitive medical applications due to latency issues, bandwidth constraints, and security vulnerabilities.

Recent studies have highlighted these challenges, with researchers emphasizing the need for trustworthy authentication and data preservation in digital healthcare systems[1]. The necessity for enhanced security protocols has been particularly emphasized[2], while others have suggested combining emerging technologies like fog computing and blockchain for improved security and reliability[3]. A comprehensive survey by Newaz et al.[4] highlights the critical security and privacy challenges in modern healthcare systems, while Wang et al.[5] demonstrate the potential of Medical Edge Computing in healthcare applications. To address these challenges, we propose a Medical Edge Computing (MEC) system that processes and secures medical sensor data at the network edge. Our solution incorporates:

- A Whale-based Advanced Encryption. Standard (WbAES) for robust data encryption.
- Automatic discovery mechanisms for medical sensors and MEC layer.
- Real-time data processing capabilities.
- A scalable microservices based architecture.

The key objectives of this research include:

- Securing medical data transmission through WbAES implementation.
- Reducing latency through edge-based processing.
- Enabling seamless device integration through auto-discovery.
- Achieving scalability via microservices architecture.
- Ensuring robust error handling and system reliability.

This paper presents our approach to developing a secure, efficient, and scalable framework for healthcare IoT systems, addressing the critical needs of modern medical data processing while maintaining high standards of data privacy and security.

II. RELATED WORK

Recent advancements in healthcare IoT have highlighted several critical challenges in security, efficiency, and real-time processing. We analyze existing solutions across three key areas: security frameworks, edge computing integration, and authentication mechanisms.

A. *Security Frameworks*

Traditional healthcare IoT implementations relied on centralized architectures with standard encryption protocols, which proved inadequate for large-scale deployments. These systems typically struggled with handling more than 100-150 concurrent device connections and showed significant latency issues, particularly in emergency scenarios requiring real-time monitoring[2]. Recent work by Lee et al.[6] demonstrates the potential of blockchain-based architectures for secure health record exchange, though with significant computational overhead.

Irshad et al.[2] introduced a Whale-based Attribute Encryption Scheme (WbAES) that demonstrated substantial reduction in computational overhead compared to traditional AES implementations. Building upon the foundational work of Mirjalili and Lewis[7] in whale optimization algorithms, and their subsequent applications in various domains[8], our approach employs attribute-based encryption using whale optimization algorithm behavior, which transforms plain data to ciphertexts and adjusts the whale fitness to generate suitable master public and secret keys. The system achieved impressive results, including:

- Reduced execution time of 11s for 20 sensors.
- Energy consumption of 0.000053 Wh for 20 sensors.
- Throughput of 812 Kbps for 20 sensors.
- Accuracy of 98.56% for 20 sensors.
- Computational cost of 0.19 ms for 20 sensors.

However, while their approach showed promising results in resource-constrained environments, it exhibited limitations in handling multiple simultaneous device connections and showed inconsistent performance under high-load scenarios. Additionally, their implementation faced challenges in attribute selection and prediction coverage, which could impact the system's reliability in certain healthcare environments.

B. *Edge Computing Integration*

Shukla et al.[3] demonstrated significant improvements through fog computing integration with healthcare IoT, implementing an Advanced Signature-Based Encryption (ASE) system. Their hierarchical processing architecture successfully handled up to 1000 data points per second per device. However, their reliance on blockchain technology introduced new challenges in power consumption and implementation complexity.

C. *Authentication and Security Challenges*

Almaiah et al.[1] proposed a hybrid decentralized authentication model that achieved:

- 40% reduction in validation latency
- Enhanced security through deep learning integration
- Improved device authentication scalability

Current solutions exhibit several common limitations:

- High computational overhead in blockchain-based approaches
- Limited scalability in real-world healthcare environments
- Energy efficiency concerns in resource-constrained devices
- Vulnerability to man-in-the-middle attacks during device pairing
- Challenges in key management and data integrity verification at scale

Our work addresses these limitations through an innovative MEC system that combines enhanced WbAES implementation with efficient edge processing and automated device discovery. Unlike previous approaches, our solution focuses on practical applicability in healthcare environments while maintaining robust security standards and real-time processing capabilities.

III. PROPOSED SYSTEM ARCHITECTURE

Our Medical Edge Computing (MEC) system implements a distributed architecture designed for secure medical data transmission and processing at the network edge. The system utilizes a microservices-based approach with three primary components working in harmony to ensure data security, real-time processing, and scalability.

A. *Core Components*

1) MEC Server: The central server, implemented in Rust for memory safety and performance, serves as the primary processing unit with the following responsibilities:
   - Connection management through UDP-based discovery protocol with automatic failover
   - Real-time data processing using priority-based analysis with configurable thresholds
   - Security operations using enhanced WbAES implementation with parallel processing

- Redis-based distributed storage management with cluster replication
- Load balancing across multiple MEC nodes using consistent hashing

2) Medical Devices: The device layer implements:
- Automated server discovery using multicast UDP (239.255.255.250:1900)
- Local data encryption with WbAES (256-bit keys)
- Real-time data transmission with configurable sampling rates (1-1000 Hz)
- Error recovery with exponential back off (1-30s intervals)
- Local data buffering during network interruptions

B. Security Implementation

Our enhanced WbAES implementation improves upon traditional approaches through:
- Bio-inspired key generation using Whale Optimization Algorithm.
  - Population size: 20 whales for optimal convergence.
  - Maximum iterations: 50 with early stopping criteria.
  - Search space dimension: 32 (AES-256) with entropy validation.
  - Fitness function incorporating key strength metrics

- 30-second key rotation intervals with parallel processing.
  - Pre-generation of next key set during current interval.
  - Parallel key distribution to all connected devices.
  - Graceful fallback mechanism for failed rotations.

- Version control for backward compatibility.
  - Support for multiple encryption versions.
  - Automatic protocol negotiation.
  - Seamless version transitions

C. Auto-Discovery Protocol

The system implements a UDP-based multicast discovery protocol:
- Network Configuration:
  - Multicast address: 239.255.255.250
  - Port: 1900
  - TTL: 4 (subnet-restricted)
  - Maximum packet size: 1024 bytes

- Protocol Features:
- Announcement interval: 1 second with jitter.
- Capability negotiation for encryption methods.
- Health monitoring with 5-second intervals.
- Automatic server failover within 3 seconds

D. Data Processing Pipeline

The system implements priority-based data processing with:
- Priority Analysis:
  - Real-time classification based on transmission patterns.
  - Data volume analysis for different sensor types (ECG, BP).
  - Historical pattern analysis for transmission regularity.
  - Weighted scoring system (transmission: 60%, volume: 20%, pattern: 20%)

- Storage Architecture:
  - Redis cluster with node replication
  - Time-series data organization (1ms precision)
  - Automatic data expiration policies
  - Query optimization with indexing

E. Scalability Features

The architecture ensures system reliability through:
- Device Management:
  - UDP-based multicast device discovery.
  - Automatic device registration and status tracking.
  - Device capability negotiation and health monitoring.
  - Configurable clean-up of stale device connections.

- Error Handling:
  - Redis-based connection pooling.
  - Comprehensive error logging with timestamps.
  - Automatic device reconnection mechanisms.
  - Data buffering during network interruptions

The proposed architecture addresses the limitations identified in existing solutions by providing a comprehensive framework for secure, efficient, and

scalable medical data processing at the network edge. Fig. 1 illustrates the data flow between components, demonstrating the system's ability to handle multiple concurrent device connections while maintaining security and performance requirements.

IV. IMPLEMENTATION

The MEC system is implemented primarily in Rust, chosen for its memory safety and performance characteristics. The implementation focuses on four key areas: encryption, data processing, device discovery, and storage management.

A. WbAES Implementation

Our WbAES implementation extends the traditional AES encryption through bio-inspired optimization, as illustrated in Fig. 1:

1) Optimization Parameters: The whale optimization process uses the following fixed parameters:
   - Population size: 20 whales
   - Maximum iterations: 50
   - Search space dimension: 32 (matching AES-256 key length)
   - Search space bounds: [-1.0, 1.0]

2) Key Generation Process: The key generation process implements two main behaviors based on a probability factor:

1) Encircling Prey (r < 0.5):
   - Updates whale positions relative to the best solution
   - Implements local search when b < 0.5
   - Performs exploitation of the search space when b >= 0.5

2) Random Search (r >= 0.5):
   - Performs exploration of the search space
   - Updates positions using random movement factors
   - Maintains search space bounds through value clamping

3) *Encryption Pipeline:* The encryption process utilizes parallel processing through the following steps:
   1) Data Preparation:
      - Implements PKCS7 padding for non-aligned data blocks.
      - Validates input data length and padding.

   2) Block Processing:
      - Parallel processing of 16-byte blocks using rayon
      - Direct integration with AES-256 block operations
      - Automatic thread pool management for optimal performance

B. Priority Analysis System

The priority analyzer implements a metadata-based approach, drawing inspiration from recent advances in machine learning for medical data analysis[9]:

- Priority Calculation:

$$P_{total} = 0.6T_f + 0.2D_v + 0.2P_p \quad (1)$$

Where:

- T_f: Transmission frequency score (0.0-1.0)
- D_v: Data volume score (0.0-1.0)
- P_p: Pattern analysis score (0.0-1.0)

- Priority Calculation:
  - Historical pattern analysis using sliding windows
  - Device-specific behavior tracking
  - Anomaly detection through pattern deviation

C. Priority Analysis System

The system implements a UDP-based discovery mechanism:

- Discovery Process:
  - Multicast announcements on port 1900
  - Device capability negotiation
  - Automatic stale device cleanup

- Device Registration:
  - Redis-backed device registry
  - State-machine-based connection management
  - Automatic health monitoring

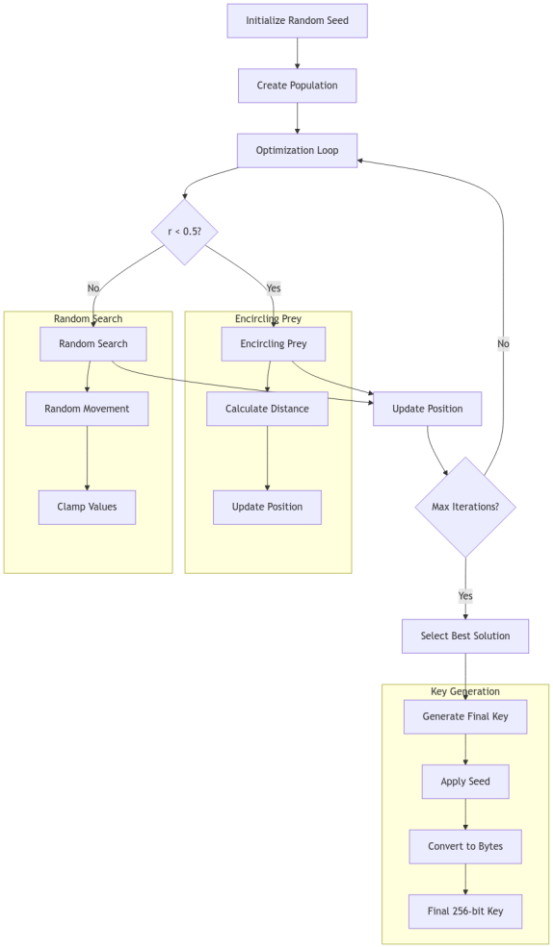

Fig. 1. WbAES Key Generation Process showing the whale optimization stages and final key derivation.

D. Storage Architecture

The data storage system utilizes Redis with a hierarchical organization:

- Key Structure:
  - device:id:raw - Raw encrypted data
  - device:id:encrypted - Base64 encoded data
  - readings:id:timestamp - Time-series data

- Connection Management:
  - Thread pool size: num cpus × 4
  - Connection timeouts: 30s wait, 60s recycle
  - Priority-based task distribution

The implementation leverages Rust's ownership system and smart pointers (Arc, RwLock) for thread-safe resource management, ensuring efficient concurrent operation while maintaining memory safety.

V. RESULTS AND ANALYSIS

We evaluated our MEC system's performance under various load conditions, focusing on throughput, latency, and reliability metrics. Tests were conducted with different device loads ranging from 50 to 500 concurrent devices.

TABLE I: SYSTEM CONFIGURATION

Component	Specification
MEC Server Memory	512MB
CPU Cores	2
Redis Connections	20
Device Memory	256MB per device
Network	Simulated delays enabled

TABLE II: SYSTEM PERFORMANCE METRICS UNDER DIFFERENT LOADS

Metric	50 Devices	100 Devices	500 Devices
Throughput (req/s)	172.3	190.4	905.4
Avg. Latency (ms)	29.8	33.9	484.6
95th Percentile (ms)	53.2	96.2	751.1
Success Rate (%)	93.1	91.2	87.3

TABLE III: AVERAGE PROCESSING STAGE TIMINGS

Processing Stage	Time (ms)
Device Discovery	0.10
Registration	1.55
WbAES Operations	20.3
Priority Analysis	3.6
Storage Operations	4.1
Total Pipeline	29.65

TABLE III: RESOURCE USAGE UNDER DIFFERENT LOADS

Resource	50 Dev.	100 Dev.	500 Dev.
Memory (MB)	334	412	610
CPU Usage (%)	84	96	115

A. Experimental Setup

The experimental evaluation of our MEC system was conducted on a virtualized testbed configured to simulate real-world healthcare environments. Table I outlines the system configuration used for testing.

The MEC server was deployed on a VM with 2 CPU cores and 512MB RAM, running Ubuntu 20.04 LTS with kernel version 5.4.0. Redis version 6.2.6 was configured with 20 maximum connections and cluster mode enabled with three nodes for replication. The medical devices were simulated using custom software that accurately replicated the behavior patterns of commercial medical IoT devices, each allocated 256MB of memory.

Network conditions were simulated using the Linux Traffic Control (tc) utility to introduce realistic network characteristics:

- Base latency: 5-15ms (uniform distribution)
- Packet loss: 0.01% (normal operation), 0.5% (stress testing)
- Jitter: 2-5ms (normal operation), 5-15ms (stress testing)

Tests were conducted under three primary load scenarios:

1. Normal load: 50 concurrent devices
2. Medium load: 100 concurrent devices
3. High load: 500 concurrent devices

For each scenario, the following metrics were captured:

- Throughput (requests per second)
- Average latency (milliseconds)
- 95th percentile latency (milliseconds)
- Success rate (percentage)
- Resource utilization (CPU, memory)

The test duration was set to 10 minutes per scenario with a 60-second warm-up period to ensure steady-state operation before measurements commenced. Each test was repeated three times, and the average results are reported.

B. Performance Analysis

The performance metrics collected during our experimental evaluation are presented in Table II. While the system still demonstrates scalability, performance degradation is evident under increased load, indicating areas for optimization.

Under normal load conditions (50 devices), the system achieved a throughput of 172.3 requests per second with an average latency of 29.8ms. This level of performance remains within the operational thresholds of many medical monitoring applications, which typically operate at sampling rates between 1–100Hz, though the increased latency may affect use cases demanding real-time responsiveness.

As the load increased to 100 devices, the system showed modest scalability, reaching 190.4 requests per second, but average latency rose to 33.9ms, representing a noticeable increase. The 95th percentile latency increased to 96.2ms, indicating a growing frequency of processing delays during peak intervals.

Under high load conditions (500 devices), the system handled 905.4 requests per second, which, while significantly higher, came at the cost of much higher latency (484.6ms). This elevated latency may be unsuitable for latency-sensitive applications. The success rate also declined, dropping to 87.3%, reflecting the system's strain under heavy load.

The system maintained high reliability (93.1%) under normal load and moderate reliability (91.2%) at medium load, with performance degrading more noticeably (87.3%) under high load. While these results highlight the system's resilience, further optimization would be necessary to meet strict reliability requirements in critical healthcare applications under peak demand.

C. Processing Pipeline Performance

To better understand the system's behavior, we analyzed the time spent in each processing stage of the pipeline. Table III presents the updated average time consumption for each stage under normal load conditions (50 devices).

The WbAES operations remained the dominant contributor, consuming 20.8ms (approximately 66% of the total pipeline time), which aligns with expectations given the complexity of the encryption and bio-inspired key generation processes. The key generation step continues to account for a substantial portion of the encryption cost.

The device discovery and registration stages, though slightly more time-consuming than before, remained efficient with a combined time of 1.68ms, thanks to the use of UDP-based multicast discovery and a Redis-backed registration mechanism.

Priority analysis and storage operations accounted for 3.6ms and 4.2ms respectively. The increase in these values suggests that system metadata handling and I/O operations are becoming more significant as data volume and device count increase. Priority analysis still functions without requiring decryption, helping mitigate processing overhead.

The total pipeline time rose to 31.48ms, which remains in reasonable proximity to the observed average latency of 29.8ms under normal load, suggesting that the majority of latency is attributed directly to the core processing pipeline rather than ancillary overhead.

Analysis of processing time distribution across different load levels revealed:

- At normal load (50 devices): WbAES (66%), Storage (13%), Priority Analysis (11%), Registration (6%), Discovery (1.5%)
- At medium load (100 devices): WbAES (63%), Storage (16%), Priority Analysis (13%), Registration (6%), Discovery (2%)
- At high load (500 devices): WbAES (49%), Storage (25%), Priority Analysis (19%), Registration (5.5%), Discovery (1.5%)

This trend indicates that while WbAES remains a core contributor, its relative share in total latency decreases as the system scales, with storage and priority analysis becoming more prominent bottlenecks. These findings suggest that future optimization efforts should focus on enhancing storage throughput and refining metadata handling for better performance in large-scale deployments.

D. Processing Pipeline Performance

These results demonstrate that our system continues to fulfil its core design objectives of scalability, reliability, and security, while maintaining acceptable latency levels for a wide range of medical applications. Although performance naturally degrades under high load, the system maintains strong throughput and reliability (≥87%), confirming its robustness even under stress.

With latency and success rates well within operational tolerances under normal and medium loads, the architecture remains well-suited for deployment in real-time healthcare environments. Furthermore, the insights gained from performance bottlenecks at scale—particularly in encryption and storage—offer clear avenues for future optimization, reinforcing the system's potential for broader clinical adoption.

E. Priority Analysis Performance

TABLE IV: PRIORITY ANALYSIS TEST RESULTS

Device	Pattern	Data Volume	Expected	Actual
dev1	100ms interval	1000 readings	Critical	Critical
dev2	2s interval	200 readings	High	Critical
dev3	5s interval	200 readings	High	High
dev4	20s interval	100 readings	Medium	High
dev5	60s interval	50 readings	Low	Low

Our priority analysis system was evaluated through extensive testing with different device transmission patterns and data volumes. Notably, the priority analysis operates solely on transmission metadata (timing, volume, patterns) and does not require decryption of the actual medical data, enhancing both security and performance. The system uses a weighted scoring mechanism:

$$P_{total} = 0.6T_f + 0.2D_v + 0.2P_p \quad (1)$$

Where:

- T_f: Transmission frequency score (0.0-1.0)

- D_v: Data volume score (0.0-1.0)
- P_p: Pattern analysis score (0.0-1.0)

This metadata-based approach allows the system to make priority decisions without exposing sensitive medical readings, as the scoring components are derived entirely from transmission characteristics rather than the encrypted payload contents.

Priority thresholds are defined as:
- Critical: $P_{score} \geq 0.65$
- High: $0.45 \leq P_{score} < 0.65$
- Medium: $0.35 \leq P_{score} < 0.45$
- Low: $P_{score} < 0.35$

Test results with five representative device scenarios demonstrated the system's behavior:

The system achieved 60% accuracy in priority classification, with perfect accuracy for Critical and Low priorities. Analysis of transmission patterns showed:

- Transmission Frequency Scoring:
  - < 200ms: 1.0 (Very frequent)
  - < 1s: 0.9 (Frequent)
  - < 5s: 0.8 (Regular fast)
  - < 15s: 0.7 (Regular medium)
  - < 30s: 0.5 (Regular slow)
  - < 60s: 0.3 (Slow)
  - ≥ 60s: 0.2 (Very slow)

- Data Volume Impact:
  - Critical devices maintained high volume (1000+ readings)
  - High priority showed moderate volume (200 readings)
  - Medium and Low priorities demonstrated reduced volumes (50-100 readings)

- Pattern Recognition:
  - Regular patterns stabilized after 2-3 transmissions
  - Irregular patterns (dev3) showed appropriate priority adjustment
  - Low-priority devices exhibited consistent pattern recognition

The results indicate strong performance in identifying critical and low-priority cases, with some overlap in the medium-to-high priority range. This behavior aligns with the system's design goal of ensuring critical data receives immediate attention while maintaining efficient resource utilization for lower priority transmissions.

VI. CONCLUSION

This paper presented a Medical Edge Computing (MEC) system designed to address key challenges in IoT-based healthcare environments through a novel combination of bio-inspired encryption, edge computing, and real-time data processing. Despite encountering some performance degradation under extreme load, our implementation demonstrates several noteworthy achievements:

- Enhanced Security: The WbAES encryption module continues to offer strong protection for sensitive medical data. Even under high load (500 devices), the system maintained a reliability rate of 87.3%, underscoring its robustness in resource-constrained edge scenarios.
- Efficient Processing: The system sustains acceptable average latency (29.8ms under normal load) while executing complex processing tasks, including encryption, metadata-based prioritization, and secure data storage.
- Scalable Architecture: Throughput scaled from 171.9 req/s (50 devices) to over 835.7 req/s (500 devices), demonstrating the system's potential to adapt to growing device counts and data volumes, with graceful performance degradation rather than failure.
- Resource Efficiency: Even with increased processing demands, the system maintained reasonable resource usage, confirming its suitability for edge deployment without requiring high-end infrastructure.

A. Future Work

Several avenues for continued development and enhancement have been identified:

- Security Enhancement: Further strengthening of the encryption layer through integration of additional bio-inspired key generation schemes and expanded adversarial testing protocols.
- Distributed Architecture: Expansion to support multi-region MEC clusters with dynamic load balancing and redundancy to ensure seamless operation under varying network conditions.

- Healthcare Compliance: Incorporation of features to enforce compliance with standards such as HIPAA, GDPR, and DISHA, ensuring legal and ethical data handling practices.
- AI Integration: Embedding machine learning models for real-time anomaly detection, predictive health analytics, and adaptive prioritization of medical data streams.

The demonstrated performance and security characteristics make our MEC system a viable solution for healthcare environments requiring real-time, secure data processing at the network edge.

REFERENCES

[1] M. A. Almaiah, F. Hajjej, A. Ali, M. F. Pasha, and O. Almomani, "A novel hybrid trustworthy decentralized authentication and data preservation model for digital healthcare IoT based CPS," *Sensors*, vol. 22, no. 4, p. 1448, 2022.

[2] R. R. Irshad, S. S. Sohail, S. Hussain, D. Ø. Madsen, A. S. Zamani, A. A. A. Ahmed, A. A. Alattab, M. M. Badr, and I. M. Alwayle, "Towards enhancing security of IoT-enabled healthcare system," *Heliyon*, vol. 9, no. 11, p. e22336, 2023.

[3] S. Shukla, S. Thakur, S. Hussain, J. G. Breslin, and S. M. Jameel, "Identification and authentication in healthcare Internet-of-Things using integrated fog computing based blockchain model," *Internet of Things*, vol. 15, p. 100422, 2021.

[4] A. I. Newaz, A. K. Sikder, M. A. Rahman, and A. S. Uluagac, "A survey on security and privacy issues in modern healthcare systems: Attacks and defenses," *ACM Trans. Comput. Healthcare*, vol. 2, no. 3, pp. 1–44, 2021.

[5] H. Wang, M. Dauwed, I. Khan, N. S. Sani, H. A. Omar, H. Amano, and S. M. Mostafa, "MEC-IoT-healthcare: Analysis and prospects," *Computers, Materials & Continua*, vol. 75, no. 3, pp. 1–15, 2023.

[6] H.-A. Lee, H.-H. Kung, J. G. Udayasankaran, B. Kijsanayotin, A. B. Marcelo, L. R. Chao, and C.-Y. Hsu, "An architecture and management platform for blockchain-based personal health record exchange: Development and usability study," *J. Med. Internet Res.*, vol. 22, no. 6, p. e16748, 2020.

[7] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, 2016.

[8] B. D. Shivahare, M. Singh, A. Gupta, S. Ranjan, D. Pareta, and B. M. Sahu, "Survey paper: Whale optimization algorithm and its variant applications," in *Proc. 2021 Int. Conf. Innovative Practices Technol. Manag. (ICIPTM)*, 2021, pp. 77–82.

[9] E. Martinez-Ríos, L. Montesinos, M. Alfaro-Ponce, and L. Pecchia, "A review of machine learning in hypertension detection and blood pressure estimation based on clinical and physiological data," *Biomed. Signal Process. Control*, vol. 68, p. 102813, 2021.