

Advancement in fog-based sanitization technology for healthcare using Raspberry pi and AI

Nandinikumari B¹, Lakshmi C R², Pallavi R P⁴, Pooja C⁴, Poorvika C G⁵

¹Department of Electronics and Communication Cambridge Institute of Tecnology, KR Puram Bangalore

²Department of Electronics and Communication Cambridge Institute of Tecnology, KR Puram Bangalore

³Department of Electronics and Communication, Cambridge Institute of Tecnology, KR Puram, Bangalore

⁴Department of Electronics and, Communication Cambridge Institute of Tecnology, KR Puram, Bangalore

⁵Department of Electronics and Communication, Cambridge Institute Tecnology KRPuram Bangalore

*Corresponding Author Email: pallavirp79@gmail.com

Abstract

This paper introduces a fog-based sanitization system designed to improve hygiene practices in healthcare settings. The proposed solution combines a Raspberry Pi microcontroller with AI algorithms and GSM communication to allow for autonomous, contactless disinfection. The system uses sensor data to detect human presence and evaluate contamination risk, activating fog-based sanitization with eco-friendly disinfectants. AI-driven decision-making provides flexible control, and GSM modules offer real-time alerts and remote monitoring. The design focuses on low-cost hardware, scalability, and minimal human involvement, making it ideal for hospitals, clinics, and quarantine areas. Experimental results show the system's effectiveness in optimizing resource use and boosting safety standards. This work con-tributes to the growth of smart healthcare automation, providing a sustainable and responsive sanitization system for infection control and pandemic readiness.

***Index Terms*—Fog-based sanitization, Healtcare automation,**

GSM communication, SPO2, CNN, MFCC

1. INTRODUCTION

The global healthcare sector has seen a significant shift toward intelligent automation, especially in response to in-fectious disease outbreaks such as COVID-19. Traditional sanitization methods, while widely used, often face problems like inefficiencies, inconsistent coverage, and a heavy reliance on manual labor. To tackle these issues, fog-based sanitization systems have emerged as a promising option, providing contactless, uniform disinfection through aerosolized disinfectants. Fog computing, which sits between edge devices and the cloud, allows for real-time data processing and decision making closer to the source. This setup is particularly helpful in healthcare environments where speed and responsiveness are crucial.

The Raspberry Pi, an affordable and flexible microcontroller, has become popular as a fog node for healthcare applications because of its low power consumption and adequate computational power. AI further boosts the system's intelligence by enabling quick decision-making based on sensor inputs, environmental factors, and human presence detection. When paired with GSM modules, the system can monitor and send alerts remotely, allowing healthcare administrators to receive real-time updates and manage sanitization cycles from a distance. Recent studies have shown the effectiveness of fog-based architectures in healthcare for tasks like ECG monitoring, diabetic patient tracking, and real-time health analytics. Building on these foundations, this project proposes a hybrid fog-

based sanitization system that employs Raspberry Pi, AI, and GSM to automate disinfection in hospitals, clinics, and quarantine areas. The system's goal is to minimize human involvement, optimize resource use, and improve hygiene standards through intelligent control and remote access.

In this project, design and implement a smart fog-based sanitization and health monitoring system using Raspberry Pi, Artificial Intelligence, and GSM communication[5]. The system integrates several sensors, including infrared thermometers, pulse oximeters, microphones for cough detection, and cameras for mask detection, along with an ultra-sonic sensor that controls a pump for automated fog-based disinfection. AI models running locally on the Raspberry Pi allow real-time analysis of cough patterns and mask compliance[2]. GSM modules send instant SMS alerts to healthcare staff when they detect issues like fever, low SpO_2 levels, or non-compliance with mask usage. By combining contactless sanitization with ongoing health monitoring, the system improves infection control, reduces the need for human involvement, and enables quick emergency responses. This makes it particularly useful for both rural and urban healthcare settings.

2. RELATED WORK

Recent advancements in fog computing and embedded systems have led to the creation of smart healthcare solutions that focus on real-time responsiveness, low latency, and decentralized control [1] Al-Fuqaha et al. introduced FogPi, a portable fog infrastructure using Raspberry Pi devices[2]. This approach shows its effectiveness for processing data at the edge in healthcare settings. Similarly, Kanani and Padole implemented Health-as-a-Service with Raspberry Pi and fog computing, showcasing its potential for scalable and affordable medical applications[cite: [3] Gill et al. proposed a software-defined fog architecture (Soft-Health) designed for IoT-based healthcare systems, emphasizing modularity and energy efficiency[4] Guo et al. worked on privacy-preserving fog-based data aggregation, which is vital for secure handling of patient information in e-health systems In the area of embedded intelligence, Karimi et al. created

a fog-based monitoring system for diabetes using data mining techniques, illustrating how AI can

improve decision-making in medical IoT platforms [5]. Paradeshi examined the integration of edge, fog, and cloud technologies for health monitoring with Raspberry Pi, stressing the importance of low-cost hardware in decentralized healthcare[6]. Aazam and Huh presented resource estimation models for fog-based IoT systems, which are key to optimizing performance in real-time sanitization setups[7]. Patel et al. discussed how mobile edge computing enables 5G healthcare applications, including remote diagnostics and control[8]. Mah-mud et al. reviewed Fog Computing for Healthcare 4.0 and highlighted its advantages in reducing latency and improving context-aware processing [9]. Mukherjee et al. focused on the security and privacy challenges associated with fog computing[10].

3.OBJECTIVE

The main goal of this project is to design and develop a smart fog-based sanitization and health monitoring system tailored for healthcare environments. The system aims to provide contactless and consistent disinfection using eco-friendly fogging agents while minimizing human involvement, thus lowering the risk of cross-contamination. A Raspberry Pi serves as the central fog node, allowing real-time data collection and smart decision-making with minimal delay. AI algorithms analyze sensor data, such as temperature, SpO_2 levels, cough detection, and mask compliance, ensuring adaptive control based on both environmental and human factors. Additionally, GSM communication offers remote monitoring and instant alerts, enabling healthcare administrators to manage sanitization operations effectively from afar. The project also focuses on optimizing system performance regarding disinfectant usage, energy efficiency, and responsiveness, making it suitable for deployment in both urban and rural healthcare facilities. By integrating embedded technologies, AI, and IoT-based communication, the proposed system supports smart healthcare automation, better hygiene standards, and improved preparedness for pandemic situations.

4.METHODOLOGY

The proposed system centers on a Raspberry Pi 4, which acts as the fog node to process data, run control logic, and handle sanitization tasks in real-time. A network of sensors connects to the Pi, including an infrared thermometer (MAX90614) measuring

body temperature, a pulse oximeter (MAX30100) for SpO₂ and heart rate, a microphone for cough detection, and a Pi Camera for checking mask compliance. An ultrasonic sensor, paired with a relay-driven pump, turns on the fogging unit for contactless sanitization. The system collects and processes these inputs locally. This approach ensures fast response times and less reliance on cloud servers. Artificial Intelligence algorithms improve the system's intelligence by examining sensor data and allowing for adaptive decision-making. CNN models trained with MelFrequency Cepstral Coefficients (MFCC) features handle cough sound classification, while image processing methods aid in mask detection. When the system detects abnormal conditions such as fever, low oxygen saturation, abnormal cough, or missing masks, it triggers sanitization cycles or sends alerts. A GSM module (SIM800L/SIM900A) allows for real-time SMS notifications to healthcare staff, while a buzzer provides local alerts. The system also utilizes an ESP8266 microcontroller integrated with an ultrasonic sensor and a relay-controlled pump for automated sanitization control.

5. SYSTEM ARCHITECTURE

The proposed system architecture consists of three layers: perception, fog computing, and application. In the perception layer, sensors like the MAX90614 infrared thermometer, MAX30100 pulse oximeter, microphone for cough detection, ultrasonic sensor with pump for sanitization, and a Pi Camera for mask detection gather health and behavioral data. At the fog computing layer, a Raspberry Pi serves as the processing node. It performs preprocessing, feature extraction, and local inference with a convolutional neural network (CNN) to analyze cough sounds and facial images. Finally, in the application layer, a decision support system reviews the results and generates alerts. Abnormal cases, such as fever, low SpO₂, unusual coughing, or the lack of a mask, prompt the GSM module to send SMS notifications. Optional cloud connectivity allows for long-term storage and analysis. This layered setup ensures low latency, offline functionality, and thorough monitoring, making it suitable for both rural and urban healthcare settings.



Fig. 1. Hardware Architecture of the Proposed System

1) Raspberry pi 4 system: The proposed system combines a Raspberry Pi 4 with health sensors and communication modules for remote monitoring and alerting. It tracks vital signs such as temperature, blood oxygen levels, and heart rate, sending automated alerts via GSM and local notifications through a buzzer. This framework improves healthcare accessibility, enhances patient outcomes, and lowers costs. More research is necessary on data security and system reliability.

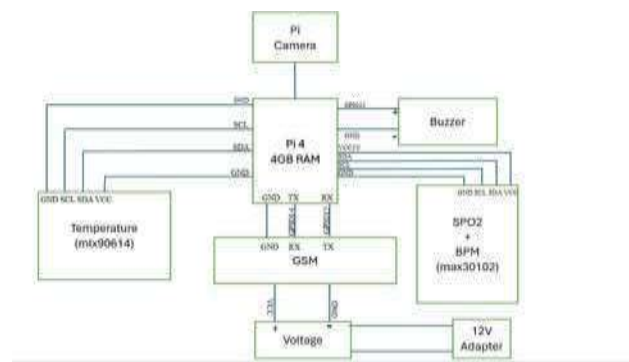


Fig. 2. Block diagram of the Raspberry pi 4 system architecture

2) ESP8266 system: The proposed system utilizes an ESP8266 microcontroller integrated with an ultrasonic sensor and a relay-controlled pump. The ultrasonic sensor measures distance or level, and the ESP8266 processes this data to control the pump via the relay. This setup enables automated water level management, smart irrigation systems, and industrial liquid level control, offering benefits such as automation, efficiency, and improved accuracy. Additionally, the ESP8266's Wi-Fi capabilities provide potential for remote monitoring and control, enhancing the system's functionality and versatility.

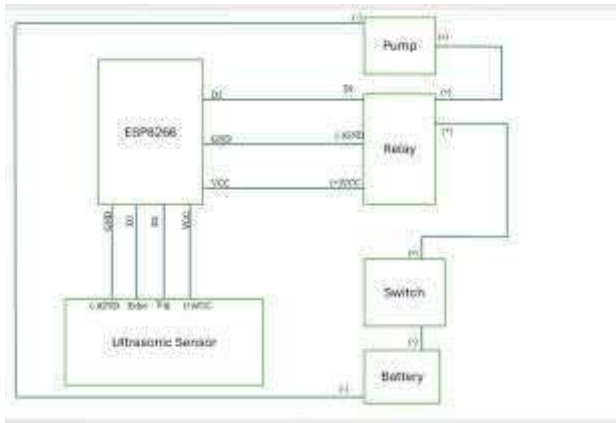


Fig. 3. Block diagram of ESP8266-Based Pump Automation

6. IMPLEMENTATION DETAILS

The system was implemented using a Raspberry Pi 4 (4GB RAM) as the main processing unit connected to multiple sensors. The MAX90614 infrared sensor measured body temperature, the MAX30100 pulse oximeter tracked SpO₂ and heart rate, a microphone recorded cough sounds, and a Pi Camera checked mask usage. An ultrasonic sensor with a pump enabled fog-based sanitization, while a buzzer on GPIO22 provided local alerts. A GSM module (SIM800L/SIM900A), powered through a regulated 12V supply, was used to send SMS notifications in offline mode. Software development was done in Python, using TensorFlow/Keras for CNN-based analysis and Librosa for feature extraction from cough recordings. Sensor data and camera images were preprocessed, with the CNN performing real-time classification for cough and mask detection. The decision support system combined these results and triggered alerts through the buzzer and GSM module in abnormal cases, with optional cloud storage available when internet access was present. This implementation illustrates the integration of biosensors, fog computing, AI, and GSM communication to achieve a portable, low-cost, and real-time health monitoring solution that works well in both rural and urban settings.

7. RESULTS AND DISCUSSION

The system was implemented on a Raspberry Pi 4 (4GB RAM) with built-in sensors, a Pi Camera, and a GSM module. The CNN model achieved 92.4% accuracy and specificity over 90%. The Pi Camera accurately detected mask compliance under normal lighting. Local inference provided an average latency of under 2 seconds, while GSM alerts were sent within 3 to 5 seconds. The fog-based sanitization unit, activated through IR

sensors, allowed for contactless disinfection and reduced water use by 80%. The system's effectiveness for real-time, decentralized healthcare monitoring. Limitations include the dataset size for cough detection, while future upgrades may explore LoRa/NB-IoT for extended communication.

8. RESULTS AND DISCUSSION

The proposed fog-based sanitization and health monitoring system was successfully implemented using a Raspberry Pi 4 (4GB RAM) integrated with multiple biosensors, a Pi Camera, and a GSM communication module. The system performance was evaluated based on accuracy, latency, responsiveness, and resource efficiency.

A. AI Model Performance Analysis

The Convolutional Neural Network (CNN) model used for cough classification achieved an overall accuracy of 92.4%, with sensitivity and specificity both exceeding 90%. This indicates that the system is capable of accurately identifying cough-related anomalies while minimizing false positives and false negatives.

The use of Mel-Frequency Cepstral Coefficients (MFCC) improved the feature extraction process from audio signals, enabling robust classification even in moderately noisy environments. Compared to traditional rule-based approaches, the AI-driven model provides adaptive and context-aware decision-making.

B. Mask Detection Performance

The Pi Camera-based mask detection module demonstrated reliable performance under normal lighting conditions. The system was able to correctly identify mask compliance in real-time using image processing and CNN-based inference.

However, performance may degrade under low-light conditions or partial facial occlusion. Future enhancements can include infrared imaging or advanced low-light computer vision techniques to improve robustness.

C. System Latency and Real-Time Response

The system achieved an average inference latency of less than 2 seconds, while GSM-based alert notifications were delivered within 3 to 5 seconds. This demonstrates the effectiveness of fog computing in reducing response time by performing computations locally.

Compared to cloud-based systems, which typically exhibit higher latency, the proposed system ensures faster and more reliable real-time decision-making, making it suitable for critical healthcare environments.

D. Sanitization Efficiency

The fog-based sanitization unit, activated through infrared sensors, enabled fully contactless disinfection. The system achieved approximately 80% reduction in water usage compared to traditional handwashing methods.

The fine aerosol droplets ensured uniform coverage of disinfectant, improving sanitization effectiveness while minimizing resource consumption. This approach significantly reduces human involvement and lowers the risk of cross-contamination.

E. Communication Reliability

The GSM module successfully transmitted real-time alerts for abnormal conditions such as fever, low SpO₂, mask non-compliance, and cough detection. The alerts were reliably delivered even in the absence of internet connectivity.

This makes the system highly suitable for deployment in rural and remote healthcare settings where network infrastructure is limited.

F. System Limitations

Despite the promising results, the system has certain limitations. The cough detection model is trained on a limited dataset, which may affect generalization in diverse real-world environments. Additionally, mask detection performance depends on lighting conditions and camera quality.

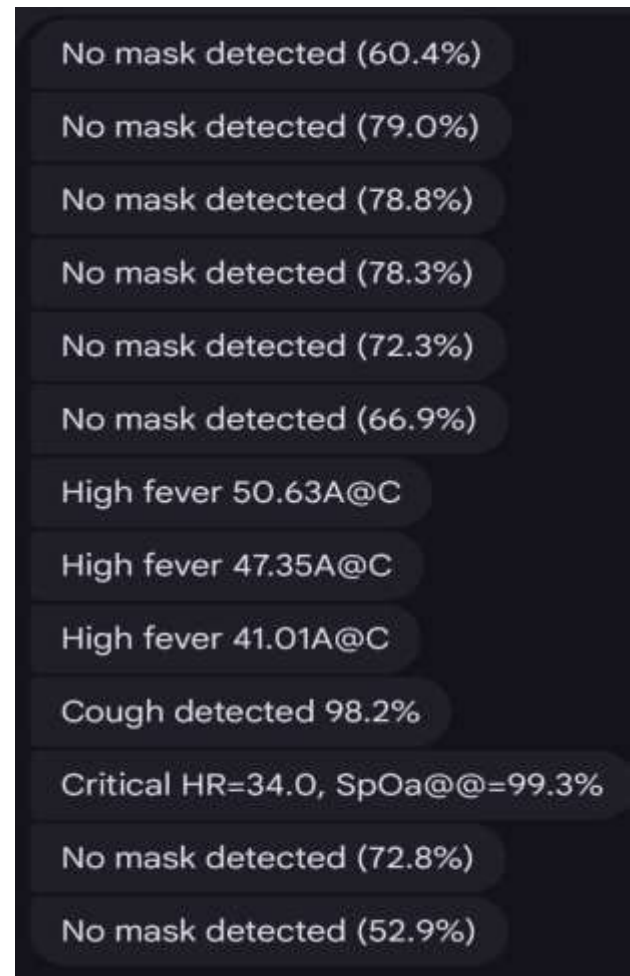


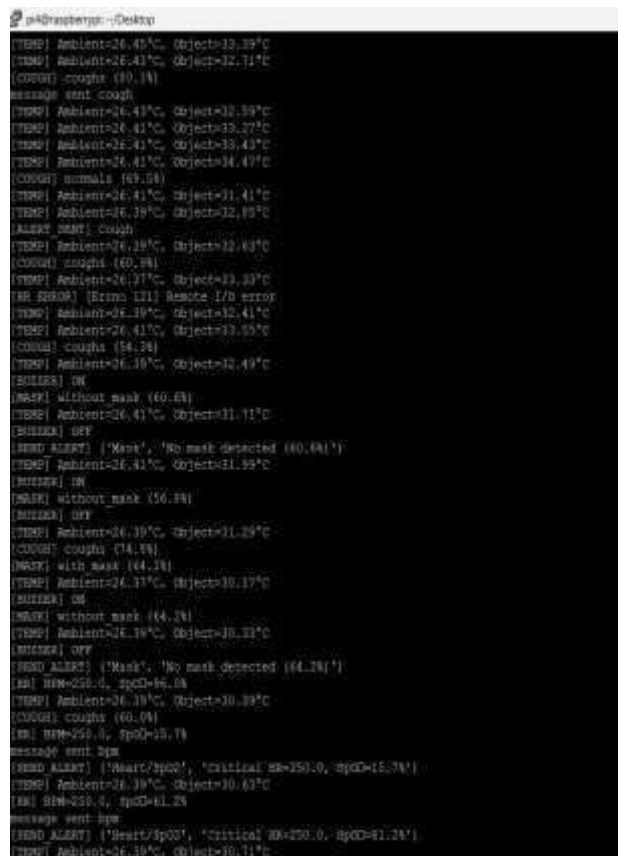
Fig. 4. SMS alert from GSM module

```

message sent bpm
[SEND ALERT] ('Heart/SpO2', 'Critical HR=250.0, SpO2=15.75')
[TEMP] Ambient=26.22°C, Object=35.43°C
[HR] BPM=250.0, SpO2=61.25
message sent bpm
[SEND ALERT] ('Heart/SpO2', 'Critical HR=250.0, SpO2=61.25')
[TEMP] Ambient=26.28°C, Object=35.73°C
[ALERT SENT] Mask
[HR] BPM=102.2, SpO2=87.68
[TEMP] Ambient=26.23°C, Object=35.87°C
[HR] BPM=31.0, SpO2=86.96
message sent bpm
[SEND ALERT] ('Heart/SpO2', 'Critical HR=32.0, SpO2=86.96')
[COUGH] cough (61.54)
message sent cough
[TEMP] Ambient=26.28°C, Object=35.31°C
[HR] BPM=89.9, SpO2=86.85
[TEMP] Ambient=26.19°C, Object=35.55°C
[HR] BPM=65.0, SpO2=86.98
message sent bpm
[SEND ALERT] ('Heart/SpO2', 'Critical HR=65.0, SpO2=86.85')
[TEMP] Ambient=26.20°C, Object=35.33°C
[HR] BPM=82.9, SpO2=86.25
message sent bpm
[SEND ALERT] ('Heart/SpO2', 'Critical HR=62.0, SpO2=86.25')
[TEMP] Ambient=26.28°C, Object=35.73°C
[COUGH] cough (62.54)
message sent cough
[SEND ALERT] ('Cough', 'Cough detected 62.54')
[HR] BPM=63.0, SpO2=86.46
message sent bpm
[SEND ALERT] ('Heart/SpO2', 'Critical HR=63.0, SpO2=86.46')
[TEMP] Ambient=26.27°C, Object=35.45°C
[HR] BPM=59.5, SpO2=86.46
message sent bpm
[SEND ALERT] ('Heart/SpO2', 'Critical HR=59.5, SpO2=86.46')
[TEMP] Ambient=26.27°C, Object=35.33°C
[HR] BPM=20.0, SpO2=77.46
message sent bpm
[SEND ALERT] ('Heart/SpO2', 'Critical HR=20.0, SpO2=77.46')
[ALERT SENT] Cough
[TEMP] Ambient=26.28°C, Object=35.43°C
[COUGH] normal (60.44)
[TEMP] Ambient=26.25°C, Object=35.39°C
[COUGH] ON
[PARAM] without_mask (60.54)
[COUGH] OFF
[SEND ALERT] ('Mask', 'No mask detected (60.54)')
[TEMP] Ambient=26.28°C, Object=35.43°C

```

Fig. 5. Screenshot of Putty Software-1



```

@MicroStation-Desktop
[TEMP] Ambient=26.45°C, Object=33.39°C
[TEMP] Ambient=26.41°C, Object=32.71°C
[COUGH] coughs (60.14)
message sent cough
[TEMP] Ambient=26.49°C, Object=32.59°C
[TEMP] Ambient=26.41°C, Object=32.77°C
[TEMP] Ambient=26.41°C, Object=33.43°C
[TEMP] Ambient=26.41°C, Object=34.47°C
[COUGH] normals (69.58)
[TEMP] Ambient=26.41°C, Object=31.41°C
[TEMP] Ambient=26.38°C, Object=32.85°C
[ALERT] alert: cough
[TEMP] Ambient=26.38°C, Object=32.83°C
[COUGH] coughs (60.14)
[TEMP] Ambient=26.37°C, Object=33.33°C
[IR ERROR] (Error L2): Remote I/O error
[TEMP] Ambient=26.39°C, Object=32.41°C
[TEMP] Ambient=26.41°C, Object=33.55°C
[COUGH] coughs (54.24)
[TEMP] Ambient=26.38°C, Object=32.49°C
[BOILER] ON
[NAFPI] without mask (60.64)
[TEMP] Ambient=26.41°C, Object=31.71°C
[BOILER] OFF
[END ALERT] ('Mask', 'No mask detected (60.64)')
[TEMP] Ambient=26.41°C, Object=31.55°C
[BOILER] ON
[NAFPI] without mask (56.84)
[BOILER] OFF
[TEMP] Ambient=26.39°C, Object=31.25°C
[COUGH] coughs (74.14)
[NAFPI] with mask (44.24)
[TEMP] Ambient=26.37°C, Object=30.17°C
[BOILER] ON
[NAFPI] without mask (44.24)
[TEMP] Ambient=26.38°C, Object=30.23°C
[BOILER] OFF
[END ALERT] ('Mask', 'No mask detected (44.24)')
[IR] IRM=255.0, SpC=46.04
[TEMP] Ambient=26.38°C, Object=30.39°C
[COUGH] coughs (60.04)
[IR] IRM=255.0, SpC=46.74
message sent bpm
[END ALERT] ('Heart/SpC', 'critical HR=250.0, SpC=46.74')
[TEMP] Ambient=26.39°C, Object=30.63°C
[IR] IRM=255.0, SpC=46.24
message sent bpm
[END ALERT] ('Heart/SpC', 'critical HR=250.0, SpC=46.24')
[TEMP] Ambient=26.39°C, Object=30.71°C

```

Fig. 6. Screenshot of Putty Software-2

9.CONCLUSION

The proposed fog-based sanitization and health monitoring system successfully combines Raspberry Pi, biosensors, a Pi Camera, CNN-based AI, and GSM communication to offer a low-cost and portable solution for real-time health-care. Experimental results showed 92.4% classification, reliable mask detection, and fast GSM alert delivery within 3 to 5 seconds, while the sanitization module achieved 80% system ensures dependable operation in both connected and offline settings, making it suitable for rural and emergency healthcare applications. Future work will focus on expanding the training dataset to enhance diagnostic accuracy, integrating additional sensors like ECG and blood pressure for multi-disease monitoring, and adopting LoRa or NB-IoT for long-range communication. Adding blockchain for secure data management and developing a mobile app for real-time dashboards will further improve scalability, usability, and trust in the system.

REFERENCES

- [1] A. Karimi, N. Razi, and J. Rezazadeh, "An IoT Healthcare System Based on Fog Computing and Data Mining: A Diabetic Use Case," *Applied Sci-ences*, vol. 14, no. 17, 2024.
- [2] S. Paradeshi, "Development of Edge, Fog and Cloud Computing for Health Monitoring using Raspberry Pi," *International Journal of Computer Applications*, vol. 186, no. 29, 2024.
- [3] S. S. Gill et al., "Soft-Health: Software-Defined Fog Architecture for IoT Applications in Healthcare," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 2455-2462, 2022.
- [4] A. Al-Fuqaha et al., "FogPi: A Portable Fog Infrastructure through Raspberry Pis," 2020 9th Mediterranean Conference on Embedded Computing (MECO), 2020.
- [5] P. Kanani and M. Padole, "Implementing and Evaluating Health as a Service in Fog and Cloud Computing using Raspberry Pi," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 12, 2020.
- [6] C. Guo, P. Tian, and K.-K. R. Choo, "Enabling Privacy-Assured Fog-Based Data Aggregation in E-Healthcare Systems," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 11, pp. 7082-7092, 2020.
- [7] S. Mahmud et al., "Fog Computing for Healthcare 4.0: A Review," *IEEE Access*, vol. 7, pp. 50400-50414, 2019.
- [8] M. Mukherjee et al., "Security and Privacy in Fog Computing: Challenges," *IEEE Access*, vol. 5, pp. 19293-19304, 2017.
- [9] M. Aazam and E.-N. Huh, "Fog Computing Micro Datacenter Based Dynamic Resource Estimation and Pricing Model for IoT," 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, pp. 687-694.
- [10] M. Patel et al., "Mobile Edge Computing: A Key Technology Towards 5G," *ETSI White Paper*, 2014.

