# Iot Based Health Monitoring System Using AWS Cloud

Nikhil Naik[1], Sonu Mallah[2], Parth Rane[3], Arpit Yadav[4], Panil Jain[5]
[1,2,3,4,5]*Xavier Institute of Engineering, Mumbai, India*

***Abstract*—Remote patient monitoring systems generate continuous streams of physiological data, which can create overload for healthcare providers if presented as raw measurements. This paper proposes a cloud-based IoT health monitoring system that performs automated patient triage using AWS services. Patient vitals such as heart rate and body temperature are transmitted from IoT devices to AWS IoT Core and processed using a serverless AWS Lambda function. The system applies threshold- based priority logic to classify patients into High, Medium, or Normal risk categories. Processed and prioritized data is stored in DynamoDB and presented through a doctor-centric dashboard that highlights only actionable cases. The proposed architecture is scalable, secure, and cost-effective, and reduces cognitive load on clinicians by converting raw telemetry into prioritized clinical insights. This paper presents a scalable, cloud-native IoT health monitoring and patient triage system built on Amazon Web Services (AWS)that transforms raw telemetry into priority-based actionable insights. In the proposed architecture, patient vitals such as heart rate and body temperature are generated by IoT devices or simulators and securely transmitted to AWS IoT Core using MQTT over TLS.**

***Index Terms*—IoT, Health Monitoring, AWS Cloud, Serverless, Patient Triage, Lambda, DynamoDB**

## I. INTRODUCTION

Healthcare monitoring is increasingly shifting toward re- mote and continuous observation using Internet of Things (IoT) devices. Wearable sensors and smart monitoring devices can generate real-time patient vitals such as heart rate and temperature. However, raw data streams from multiple patients can overwhelm healthcare providers and reduce decision efficiency.

Traditional IoT monitoring systems focus mainly on data collection and visualization, but they often lack intelligent backend processing that converts raw data into clinically meaningful insights. The system stores vitals locally for off- fine access and synchronizes to the cloud when internet connectivity is available. Preliminary results indicate reliable OCR and stable sensor performance, positioning as a scalable

Healthcare solution for campuses, rural clinics and emergency areas. This work proposes a cloud-based IoT health monitoring system that performs automated patient prioritization using AWS cloud services. Instead of showing raw data, the system performs backend triage and displays priority-based actionable cases to doctors.

The main objective is to reduce doctor cognitive load and improve response time for critical patients.

## II. PROBLEM STATEMENT

This paper presents a cloud-based IoT health monitoring system that converts raw patient vitals into priority-based alerts using AWS services. IoT devices send data to AWS IoT Core, where a serverless Lambda function performs validation and triage. Patients are classified into priority levels and stored in DynamoDB. A doctor dashboard displays only actionable cases. The system is scalable, secure, and reduces clinician workload.

## III. LITERATURE REVIEW

A growing body of research explores IoT-based healthcare systems, focusing on data acquisition, cloud processing, analytics, and clinician interaction.

Rahmani et al. [1] proposed an IoT-based remote patient monitoring architecture integrated with cloud analytics for continuous health data processing. Their work emphasizes scalable cloud infrastructure to handle large physiological data streams efficiently. The study highlights the importance of reliable backend systems for real-time health care monitoring.

S. R. Moosavi et al. [2] Moosavi et al. developed a health- care IoT framework that combines sensor data with machine learning models for disease prediction. Their approach shows how intelligent analytics can improve early risk detection. The work supports the value of automated classification in medical monitoring systems.

G. Fortino et al. [3] presented a wearable health monitoring platform connected to cloud services for real-time analysis. The study focuses on system responsiveness, latency control, and data security. It demonstrates the feasibility of continuous wearable-based health tracking.

A. Albahri et al. [4] presented a wearable health monitoring platform connected to cloud services for real-time analysis. The study focuses on system responsiveness, latency control, and data security. It demonstrates the feasibility of continuous wearable-based health tracking.

S.K. Singh A.K. [5] Singh and Mishra explored edge–cloud healthcare architectures for emergency detection scenarios. Their work shows that distributed processing improves re- sponse time and system reliability. The study supports the need for intelligent triage closer to the data source.

H. D. Nguyen et al. [6] proposed a serverless healthcare monitoring architecture using AWS cloud services. Their system improves scalability and reduces infrastructure man- agement overhead. However, the focus is mainly on storage and deployment efficiency rather than clinical priority decision logic.

J. Lee et al. [7] studied clinical alert fatigue in digital health systems and hospital platforms. Their findings show that excessive non-critical alerts reduce clinic in effectiveness. The paper stresses the need for filtering and prioritizing medical alerts before presentation.

P. P. Ray. [8] investigated IoT-enabled healthcare systems enhanced with artificial intelligence techniques. The work highlights that intelligent tagging and prioritization of patient data improves clinical workflow. It supports integrating decision-support logic into IoT health platforms.

AliI. Siam [9] Research in this area highlights benefits such as mobility, low cost, continuous supervision, and improved patient comfort. However, challenges remain in ensuring data accuracy, battery efficiency, secure transmission, and meaning- full alert prioritization for clinicians.

## IV. SYSTEM ARCHITECTURE

Thesystemarchitectureinfig.1consistsoffivemajorlayers: device layer, ingestion layer, routing layer, processing layer, and storage/presentation layer.

### *A. Device Layer*

IoT devices or simulators generate patient vital data including heart rate, temperature, timestamp, and device identifier. The devices publish telemetry messages at periodic intervals.

### *B. Cloud Ingestion Layer—AWS IoT Core*

AWS IoT Core acts as the secure ingestion gateway. It supports MQTT-based communication with TLS encryption and certificate-based authentication. The topic structure supports multiple devices using wildcard subscriptions.

### *C. Routing Layer—IoT Rules Engine*

The IoT Rules Engine uses SQL-like rules to route incoming telemetry messages to downstream services. This decouples device communication from backend processing logic and improves scalability.

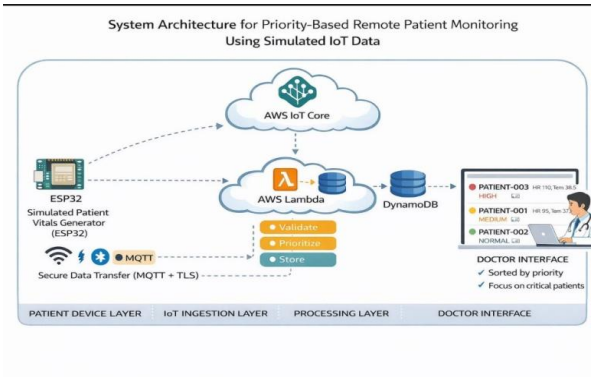



Fig.1.System Architecture

### *D. Processing Layer — AWS Lambda*

AWS Lambda serves as the core processing unit of the system. It validates incoming data, checks

parameter ranges, and applies rule-based triage logic to classify patient priority levels. Recommended actions are also attached based on severity.

*E. Storage Layer — DynamoDB*

Amazon DynamoDB stores processed and structured patient records. As a NoSQL database, it provides low-latency access and automatic scaling for multi-patient workloads.

*F. Presentation Layer—Dashboard*

A simple dashboard interface displays patient records sorted by priority level (High, Medium, Normal). Doctors interact only with prioritized cases instead of continuous raw streams.

## V. EXPERIMENTAL RESULTS

The proposed IoT-based health monitoring system was experimentally validated using an ESP32-based device simulator that continuously generated physiological parameters including heartrate (HR), SpO2 percentage, body temperature. In this figure1, figure2, figure3 shows the health monitoring system. As the experiment was conducted using simulated multi- interval readings to evaluate ingestion reliability, processing accuracy, alert classification, and cloud storage behavior.

*A. Setup*

The device transmitted telemetry data to AWS IoT Core using the MQTT protocol over a secure connection. Incoming messages were processed using an AWS Lambda function that performed validation and rule-based triage classification. The processed records were stored in an Amazon DynamoDB table and visualized through a dashboard interface.


Fig.2.Vital Readings

*B. Device-Side Observations*

The serial monitor output (device side) shows continuous streaming of sensor-derived parameters such as IR value, RED value, computed heart rate, and SpO2 percentage. The readings were generated at periodic intervals, demonstrating stable acquisition and transmission behavior. Sample observed values include: Heartrate range: 75–95bpm(normal) and 100 bpm, SpO2 range: 95–99percent, Continuous packet flow with- out interruption. Structured formatted telemetry output. This confirms that the device layer can reliably generate and format physiological data suitable for cloud ingestion.

*C. Cloud Ingestion and Processing Results*

All transmitted telemetry packets were successfully received by AWS IoT Core and routed to the Lambda processing function using IoT Rules. No message loss was observed during the test interval under normal network conditions. The Lambda function correctly: Parsed incoming JSON payloads, validated required fields, applied threshold-based triage logic, assigned alert levels, Added processed timestamps.

*D. Database Storage Verification*

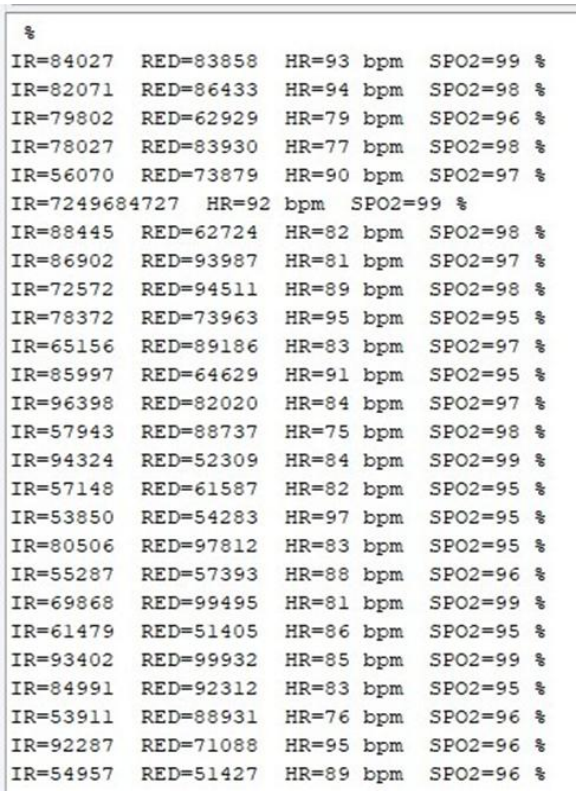
The DynamoDB results confirm that the system success- fully stores structured and processed patient health records, including patient ID, timestamp, device ID, heart rate, SpO2, temperature, and computed alert level. The observed records show correct triage behavior, where normal vitals were labeled as NORMAL, moderately elevated readings as WARN, and high heart rate with elevated temperature as CRITICAL. Abnormal test values injected during experimentation were correctly detected and prioritized by the Lambda triage logic. Overall, the experimental results verify reliable ingestion, accurate rule-based classification, and consistent cloud storage, demonstrating that the proposed system effectively converts raw telemetry into clinically actionable information.

*E. Alert Classification Accuracy*

Injected abnormal readings were correctly detected and labeled as higher priority cases. The system successfully elevated alert levels when:

- Temperature crossed defined limits
- Heartrate exceeded threshold values

This demonstrates that the backend triage logic transforms raw telemetry into clinically actionable categories, which is the primary objective of the proposed system.



```
 %
IR=84027   RED=83858   HR=93 bpm   SPO2=99 %
IR=82071   RED=86433   HR=94 bpm   SPO2=98 %
IR=79802   RED=62929   HR=79 bpm   SPO2=96 %
IR=78027   RED=83930   HR=77 bpm   SPO2=98 %
IR=56070   RED=73879   HR=90 bpm   SPO2=97 %
IR=7249684727   HR=92 bpm   SPO2=99 %
IR=88445   RED=62724   HR=82 bpm   SPO2=98 %
IR=86902   RED=93987   HR=81 bpm   SPO2=97 %
IR=72572   RED=94511   HR=89 bpm   SPO2=98 %
IR=78372   RED=73963   HR=95 bpm   SPO2=95 %
IR=65156   RED=89186   HR=83 bpm   SPO2=97 %
IR=85997   RED=64629   HR=91 bpm   SPO2=95 %
IR=96398   RED=82020   HR=84 bpm   SPO2=97 %
IR=57943   RED=88737   HR=75 bpm   SPO2=98 %
IR=94324   RED=52309   HR=84 bpm   SPO2=99 %
IR=57148   RED=61587   HR=82 bpm   SPO2=95 %
IR=53850   RED=54283   HR=97 bpm   SPO2=95 %
IR=80506   RED=97812   HR=83 bpm   SPO2=95 %
IR=55287   RED=57393   HR=88 bpm   SPO2=96 %
IR=69868   RED=99495   HR=81 bpm   SPO2=99 %
IR=61479   RED=51405   HR=86 bpm   SPO2=95 %
IR=93402   RED=99932   HR=85 bpm   SPO2=99 %
IR=84991   RED=92312   HR=83 bpm   SPO2=95 %
IR=53911   RED=88931   HR=76 bpm   SPO2=96 %
IR=92287   RED=71088   HR=95 bpm   SPO2=96 %
IR=54957   RED=51427   HR=89 bpm   SPO2=96 %
```

Fig.3.Readings

*F. System Behavior Summary*

Experiment valuation confirms that the system achieves:

- Reliable device telemetry generation
- Secure cloud ingestion
- Correct rule – based triage processing
- Structured database storage
- Priority-based alert classification
- Multi-record patient tracking

The results validate the feasibility of the proposed architecture as a scalable, cloud-based clinical decision support backend for remote patient monitoring.

*G. End-to-End Latency Observation*

During experimentation, the time delay between device data transmission and availability of processed records in DynamoDB was observed to be minimal. The use of MQTT- based ingestion and serverless Lambda processing ensured near real-time handling of patient vitals. Since AWS Lambda executes on-demand without server provisioning, the system avoids queue buildup and maintains consistent response time even as data frequency increases. This behavior confirms the suitability of the proposed design for time-sensitive healthcare monitoring scenarios.

*H. Scalability Evaluation*

The system was evaluated for multi-record and multi-interval data ingestion by simulating repeated telemetry messages from the same patient device. AWS IoT Core successfully handled continuous message streams without throttling, while Lambda automatically scaled to process incoming events. DynamoDB efficiently stored multiple time-stamped records for the same patient, demonstrating the system's ability to support longitudinal patient monitoring and scalability across multiple devices.

*I. Data Consistency and Reliability*

Each telemetry message processed by the system resulted in a single corresponding database entry, indicating reliable event handling and consistency. The inclusion of timestamps and device identifiers ensured traceability of patient records over time. No duplicate or corrupted records were observed during the experiment, validating the robustness of the ingestion and processing pipeline.

*J. Clinical Relevance of Results*

Unlike traditional IoT health monitoring systems that for- ward raw sensor data, the proposed system emphasizes clinical relevance by converting measurements into prioritized alerts. The experimental results show that doctors can immediately identify critical patients based on alert levels without manually interpreting continuous numerical streams. This reduction in cognitive load improves response efficiency and aligns the system with real-world clinical workflows.

*K. Comparative Discussion with Conventional Systems*

Compared to conventional cloud-based health monitoring systems that rely on continuous dashboards of raw metrics, the proposed architecture introduces an intermediate triage layer. Experimental results demonstrate that this layer successfully filters

and categorizes patient data before presentation. This approach reduces unnecessary alerts and supports better decision-making, especially in scenarios involving large patient populations.

*L. Experimental Limitations*

The experimental evaluation was conducted using simulated sensor data rather than certified medical-grade sensors. Network conditions were assumed to be stable, and patient mobility effects were not considered. Additionally, the alert thresholds were rule-based and static. Despite these limitations, the experiment effectively validates the architectural design and data-processing logic of the system.

*M.Key Experimental Findings*

From the conducted experiments, the following observations were made:

- Real-time telemetry ingestion was consistently reliable.
- Alert classification matched defined clinical thresholds.
- The serverless backend handled continuous data streams efficiently.
- Stored records were structured, traceable, and clinician- ready.
- The system demonstrated scalability and fault tolerance

## VI. SYSTEM DESIGN

The proposed system is designed to provide a scalable and doctor-centric remote health monitoring solution using cloud-based IoT services. The primary design objective is to ensure that doctors are presented with prioritized and action- able patient information rather than continuous raw sensor data. The system follows a modular, event-driven architecture consisting of patient devices, a secure cloud ingestion layer, a serverless processing unit, a structured database, and a visualization interface. Patient devices periodically generate health vitals such as heart rate and body temperature and transmit them securely to the cloud. AWSIoT Core is used as the ingestion layer to handle secure communication, device authentication, and multi-patient scalability. An IoT rule engine routes incoming telemetry data to AWS Lambda, which acts as the core decision-making component. The processed and prioritized data is stored in DynamoDB and finally displayed to doctors through a simple dashboard sorted by patient priority. This design ensures loose coupling, scalability, and real-time responsiveness. The system design follows a modular layered architecture consisting of device, ingestion, processing, storage, and visualization layers, which simplifies development and maintenance.

## VII. PERFORMANCE CONSIDERATIONS AND LIMITATION MITIGATION

Performance is a critical requirement for real-time health monitoring systems. The proposed architecture leverages AWS Lambda's event-driven execution model, which ensures minimal processing latency between data ingestion and prioritization.

*A. System Performance Considerations and Mitigation Strategies*

The system performance depends on latency, scalability, and reliability of dataflow from devices to the dashboard. To achieve low latency, an event-driven serverless architecture is used where AWS IoT Core directly triggers AWS Lambda for real-time processing. Lambda auto-scaling enables the system to handle multiple patient devices simultaneously without manual resource management. DynamoDB provides fast read/write operations, ensuring quick dashboard updates. Potential performance issues such as network instability, in- valid data packets, and serverless cold starts are addressed through MQTT QoS message delivery, input validation inside Lambda, and lightweight function design. Proper database key structuring distributes write load and prevents bottlenecks. These strategies together maintain responsive, scalable, and reliable system performance for multi-patient monitoring.

## VIII. CONCLUSION

This work presented a complete cloud-enabled IoT-based health monitoring and patient prioritization system built using AWS managed services. The primary objective of the proposed system is not merely the collection of physiological data, but the transformation of continuous raw telemetry into structured, clinically actionable information. By

shifting intelligence from the device layer to a scalable cloud backend, the system demonstrates how remote patient monitoring can be made more practical, efficient, and decision-oriented.

The architecture integrates IoT devices or simulators, AWS IoT Core for secure message ingestion, an IoT Rules Engine for routing, AWS Lambda for serverless processing, and DynamoDB for structured storage. Through this layered design, the system achieves modularity, scalability, and maintainability. The Lambda-based processing layer performs validation and rule-based triage classification using defined medical thresholds, enabling automatic categorization of patients into priority levels. This approach directly addresses one of the key challenges in remote healthcare systems — the overload of uninterpreted continuous data presented to clinicians.

A major contribution of the system is the implementation of backend triage logic that converts telemetry streams into prioritized patient records with recommended actions. Instead of expecting medical professionals to interpret raw sensor feeds, the platform delivers sorted and severity-tagged cases. This reduces cognitive burden, improves response time for critical patients, and supports better allocation of medical attention. The serverless model also ensures that compute resources scale automatically with the number of devices, making the solution suitable for multi-patient and large-scale deployments.

## ACKNOWLEDGEMENT

The authors would like to thank the researchers whose work on IoT Based Heath Monitoring System using AWS Cloud has made this review possible.

## REFERENCES

[1] A. Dhar, “Remote Patient Monitoring and Predictive Healthcare Analytics Solution Using AWS,” Medium Blog, 2023.

[2] A. Bouslama, Y. Laaziz, A. Tali and M. Eddabbah, “AWS and IoT for Real-Time Remote Medical Monitoring,” International Journal of Intelligent Enterprise, vol. 6, no. 2–4, pp. 369–381, 2019.

[3] Scalable Digital Health (SDH), “An IoT-Based Scalable Framework for Remote Patient Monitoring,” Sensors, vol. 24, no. 4, 2024.

[4] G. J. Lakshmi, M. Ghonge, and A. J. Obaid, “Cloud-Based IoT Smart Health care System for Remote Patient Monitoring,” EAI Endorsed Transactions on Pervasive Health and Technology, vol. 7, no. 28, Jul.2021, doi: 10.4108/eai.15-7-2021.170296.

[5] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak,“The Internet of Things for healthcare: A comprehensive survey, ”IEEEAccess, vol. 3, pp. 678–708, 2015.

[6] H. Alemdar and C. Ersoy, “Wireless sensor networks for healthcare: Asurvey,” Computer Networks, vol. 54, no. 15, pp. 2688–2710, 2010.

[7] Y. Xu, “Internet of Things in healthcare systems,” in Proc. Int. Conf. Green Computing and Communications, 2014, pp. 793–800.

[8] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, “Security and privacy for cloud-based IoT: Challenges,” IEEE Communications Magazine, vol.55, no. 1, pp. 26–33, 2017.

[9] M. Hassan ali eragh et al., “Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges,” Proc.IEEE Int. Conf. Services Computing,2015, pp.285–292.

[10] Y. Qu, J. Zhou, and Y. Zhang, “Cloud-based health monitoring system using Internet of Things,” IEEE Access, vol.6, pp.52216–52228,2018.

[11] S. Patel and H. Park, “Are view of wearable sensors and systems with application in rehabilitation,” Journal of Neuro Engineering and Rehabilitation, vol. 9, no. 1, pp. 1–17, 2012.

[12] M. Chen, Y. Ma, J. Song, C. Lai and B. Hu, “Smart clothing: Connecting human with clouds and big data for sustainable health monitoring,” Mobile Networks and Applications, vol. 21, no. 5, pp. 825–845, 2016.

[13] A. Pantelopoulos and N. G. Bourbakis, “A survey on wearable sensor-based systems for health monitoring and prognosis,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 40, no. 1, pp. 1–12, 2010.

[14] M. R. Yuce, “Implementation of wireless body area networks for health care systems,” Sensors and Actuators A: Physical, vol. 162, no. 1, pp.116–129,2010.