# A Preventive Health Monitoring System for Urban Communities Using IOT

**Soma Yashwanth[1], Yathavakilla Sanjana[2], Ekkaldevi Ditya[3], Permula Hitesh Hemendra[4]**
*[1234] AIML, Sreyas Institute of Engineering and Technology, Hyderabad, India–500068*

**How to Cite this Article**:
Yashwanth, S., Sanjana, Y., Ditya, E. & Hemendra, P. H. (2026). A Preventive Health Monitoring System for Urban Communities Using IOT. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(04).
https://doi.org/10.55041/ijcope.v2i4.538



https://doi.org/10.55041/ijcope.v2i4.538

## 1.0 Abstract

Preventive healthcare is really important these days in big cities where people have a lot of health problems because of the way they live. A lot of health issues happen because peoaple do not see the early warning signs they do not go to the doctor when they should or they do not keep track of their health regularly. So healthcare systems usually focus on treating people when they're already very sick. To fix this problem this project is making a health monitoring system that uses Internet of Things technology to keep track of a persons health all the time and remind them to take care of themselves.

This system uses Internet of Things sensors to collect health information like heart rate how much a person moves and the air quality around them. This information is then sent to a computer. Analyzed to see how a persons health is changing over time. By looking at this information in a way and using intelligent methods the system can find unusual changes in a persons health and send them a message or give them some advice. This helps people take care of themselves before they get really sick which reduces the chance of having health problems.

The system is made to be simple and not too expensive so it is easy to use and good for people who live in cities and also for people who do not have a lot of access to healthcare. Keeping track of health all the time helps people know more about their health and take care of themselves. It also helps doctors make better decisions.

The preventive healthcare system is a thing because it helps people stay healthy and it shows how Internet of Things solutions can make it easier for people to take care of their health. This project is also related to Sustainable Development Goal 3 which is about having good health and well-being by promoting preventive healthcare and using Internet of Things technology to make health monitoring better.

**Keywords:** *Good Health and Well-Being (SDG 3), Health Monitoring System, Internet of Things (IoT), Preventive Healthcare, Real-Time Data Analysis, Smart Healthcare Systems.*

## 2.0 Introduction

Health problems are not waiting until old age anymore. Between desk jobs, processed food, and packed schedules, people are dealing with chronic conditions heart disease, diabetes, hypertension earlier and with less time to manage them. The standard response is a doctor visit when something feels wrong, or an annual checkup if you are disciplined. That mostly works, until it does not.

The gap between appointments is the real problem. A lot can go wrong in a short period. Heart rate spikes, temperature changes, early warning signs they come and go unnoticed. By the time a patient sits in a clinic, the moment has passed or things have gotten worse. Research puts 30–40% of serious complications in the category of preventable with earlier detection, though continuous monitoring through traditional means would require hospitalization or around-the-clock nursing care.

Access makes everything harder. Rural patients drive hours for a consultation. Urban patients wait weeks and pay more than they should. Neither system is built for catching problems early. IoT is a reasonable answer to this. Sensors can monitor heart rate, body temperature, and other vitals around the clock, then push that data to a phone, a dashboard, or a doctor’s system no visit required. The hardware is inexpensive: an ESP32 or similar microcontroller, a few sensors, a cloud connection. When readings drift outside normal thresholds, the system flags it and alerts whoever needs to know.

Preventive care is less dramatic than it sounds. You are not catching a heart attack in progress you are noticing that something is trending in the wrong direction before it becomes an emergency. Worth being direct about the landscape: health monitoring devices already exist, and most are either expensive, limited to one parameter, or complicated enough that patients abandon them. Those devices only display readings. This system analyzes the data, checks it against thresholds, and triggers alerts when something is off. That difference matters.

There is also a data angle that gets underestimated. A single clinic reading is a snapshot. Weeks of continuous data is a pattern. Doctors working with longitudinal health data can spot trends that no single appointment would reveal a slow upward creep in resting heart rate, temperature fluctuations that track with stress or poor sleep, baseline shifts that precede a flare-up. Patients who show up with actual data get better answers.

The system described here is not a replacement for medical care worth saying plainly, because these tools get oversold. A sensor cannot diagnose anything. An alert is not a prescription. What it does is reduce the information gap between patient and provider, and give people a reason to act before something gets bad enough to be unmissable.

## 3.0 Proposed System

The system is straightforward: a low-cost IoT setup that tracks a person's vitals continuously, without requiring them to leave home. The target users are elderly people and patients managing chronic conditions people for whom "just go to the doctor" is not always a realistic option.

### 3.1 Overview

The system monitors heart rate and body temperature. Sensors attached to the patient feed readings to a microcontroller an ESP32 or Arduino which processes the numbers and pushes them to a cloud platform or mobile app. That is the whole pipeline. It is not complicated by design.

What makes it useful is that it runs constantly. It is not checking once a day or logging data for a weekly review it polls at regular intervals, watching for drift. When something goes outside the normal range, whether that is a fever climbing or a heart rate behaving oddly, it sends an alert. The caregiver or doctor finds out quickly, rather than at the next scheduled visit when the window may have already closed.

### 3.2 System Architecture

The system is built in layers. Each component has a defined job collecting data, processing it, sending it somewhere, and triggering an alert when needed. Nothing overlaps, nothing is redundant.

#### 3.2.1 Sensor Layer (Data Acquisition)

This layer collects physiological data from the patient using two sensors:

- Heart Rate Sensor (MAX30102 or equivalent photoplethysmography sensor)
- Temperature Sensor (LM35 or DS18B20 digital temperature sensor)

These sensors convert physical parameters pulse and heat into electrical signals that the microcontroller can read and process.

Heart rate is calculated in beats per minute (BPM) using the following formula:

$$\text{Heart rate (BPM)} = (60 \times N) / T$$

Where:

- N = Number of pulses detected
- T = Time interval in seconds

#### 3.2.2 Processing Layer (Microcontroller Unit)

The processing layer uses a microcontroller such as the ESP32 or Arduino Uno, which acts as the central processing unit. It handles signal conditioning, analog-to-digital conversion (ADC), and data filtering to clean up noisy sensor output.

To improve reliability, multiple sensor readings are averaged:

$$\text{Mean Value} = \Sigma x / n$$

Where:

- X = Sensor readings
- n = Number of observations

### 3.2.3 Communication Layer (Data Transmission)

This layer sends processed data to the cloud using the ESP32's built-in Wi-Fi. The system supports two protocols HTTP for straightforward REST-based uploads, and MQTT for lightweight publish-subscribe messaging that works better on slower connections.

Data rate is calculated to evaluate transmission efficiency:

$$\text{Data Rate} = \text{Total Data Transmitted} / \text{Time}$$

### 3.2.4 Cloud Layer (Storage and Analysis)

The cloud layer implemented on Firebase or an equivalent platform receives incoming readings, stores them for historical review, and makes them available for visualization. This is where threshold checks happen. If a reading crosses the safe limit defined below, an alert is triggered:

$$\text{Alert Condition: } x > x\_max \text{ or } x < x\_min$$

Where:

- x = Current sensor value
- x_max, x_min = Safe limits

### 3.2.5 Application Layer (User Interface)

The application layer is the part patients and caregivers actually see — a mobile app or web dashboard that displays current readings, historical graphs, and alert history. The interface is kept simple on purpose. Anyone who can use a smartphone can use this.

A basic health risk indicator quantifies how far a current reading sits from the normal reference value:

$$\text{Risk Score} = (x - x\_normal) / x\_normal$$

Where:

- x = current value
- x_normal = Normal reference value

### 3.2.6 Alert Module

The alert module fires when the microcontroller detects a reading outside the safe range. Notifications go out via the app or SMS. The specific conditions that trigger an alert are:

$$\text{If } (HR > 100 \text{ BPM}) \Rightarrow \text{Alert}$$
$$\text{If } (Temp > 38°C) \Rightarrow \text{Alert}$$

### 3.3 Working of the System

The system operates in a continuous loop:

1. Each reading is passed to the microcontroller for signal conditioning, ADC conversion, and averaging.
2. The microcontroller compares the processed value against the predefined safe thresholds.
3. Sensors attached to the patient's body capture heart rate and temperature at regular intervals.
4. If the reading is within range, it is uploaded to the cloud and the cycle repeats.
5. If an abnormal value is detected, an alert is sent immediately to the patient, a family member, or a physician.

This loop runs without interruption, so any dangerous change in condition gets flagged the moment it happens not hours later at the next scheduled check.

### 3.4 System Modules

| Module | Component | Function |
|---|---|---|
| **Sensor Module** | MAX30102, LM35/DS18B20 | Captures raw heart rate and body temperature signals |
| **Data Processing** | ESP32 / Arduino Uno | ADC conversion, signal conditioning, threshold comparison |
| **Communication** | Wi-Fi (HTTP / MQTT) | Transmits processed readings to cloud platform |
| **Cloud Storage** | Firebase / ThingSpeak | Stores data, runs threshold checks, enables visualization |
| **Alert Module** | App notification / SMS | Notifies user or caregiver when readings cross safe limits |
| **User Interface** | Mobile app / Web dashboard | Displays live readings, alert history, and trend graphs |

*Table 1: System Modules and Their Functions*

### 2.5 Key Features

The system includes several features designed to make it practical for everyday home use:

| Feature | Description |
|---|---|
| **Real-Time Monitoring** | Continuous tracking of heart rate and body temperature |
| **Instant Alerts** | Notifications sent the moment readings cross defined thresholds |
| **Remote Access** | Patient data viewable from anywhere via app or browser |
| **Low Cost** | Built entirely with off-the-shelf, inexpensive components |
| **User-Friendly** | No technical knowledge required for setup or daily use |

| | |
|---|---|
| **Expandable Design** | Additional sensors (SpO2, blood pressure) can be added later |

*Table 2: Key Features of the Proposed System*

### 3.6 Advantages

Compared to existing monitoring solutions, the proposed system offers several concrete benefits. It cuts down on unnecessary hospital visits by flagging problems at home before they escalate. Early detection gives patients and clinicians more time to respond. Monitoring that runs in the background imposes no daily burden on the user. And because the cost is low, it is accessible to patients who cannot afford commercial medical-grade wearable. Family members also gain peace of mind knowing that something is watching even when they cannot be present.

### 3.7 Novelty of the Approach

What makes this system different from what is already on the market is the combination: affordable hardware, multi-parameter sensing, and active alerting in a single package designed for home use. Most existing solutions are either expensive clinical devices, single-parameter wearables, or consumer fitness trackers that display readings without checking them against health thresholds. This system sits in the gap between those categories. It monitors more than one thing, costs less than commercial alternatives, and does something when a number goes wrong rather than just logging it.

### 3.8 Future Enhancements

The current setup monitors two parameters. With modest additions, it could cover blood pressure, blood oxygen saturation (SpO2), and blood glucose. The cloud pipeline could feed into a machine learning layer that learns what is normal for a specific patient over time and flags deviations that are unusual for that person even if they fall within population-level averages. Integration with hospital information systems would let the data flow directly into a clinician's workflow. None of this is far out of reach; it is a matter of adding modules to an architecture that was designed to accommodate them.

### 4.0 Experimental Setup

The hardware side of the setup is deliberately basic. A MAX30102 pulse oximetry sensor handles heart rate collection. An LM35 or DS18B20 sensor covers body temperature. Both connect to an ESP32 microcontroller, which was the practical choice for this project because it has built-in Wi-Fi no separate communication module required. The physical assembly runs on a breadboard using standard jumper wires and a USB power supply, the kind of setup any electronics student could put together in an afternoon.

Sensor sampling runs at 1 Hz for heart rate (one reading per second) and every five seconds for temperature. At that rate, the ESP32 generates roughly 12 data points per minute per parameter. Each reading goes through a basic averaging filter five consecutive readings are mean-averaged before the result is considered a valid data point. This smooths out spikes that are electrical noise rather than actual physiology, which matters when you are deciding whether to trigger an alert.

The software runs on the Arduino IDE using embedded C/C++. Processed readings are uploaded to Firebase Realtime Database via HTTP. Firebase was chosen because it handles concurrent reads well and the free tier is sufficient for the data volumes involved. From there, a lightweight web dashboard built using standard HTML/CSS and the Firebase JavaScript SDK pulls the data and renders live readings and historical line graphs. No additional backend server is needed; Firebase handles authentication and database access directly.

Alert thresholds were set based on standard clinical reference ranges: heart rate flagged if it exceeds 100 BPM or drops below 50 BPM, body temperature flagged if it exceeds 38°C (indicating possible fever). When a reading crosses either limit, the system fires an alert through the app and, optionally, via an SMS gateway. Alert latency was measured during testing: the median time from threshold crossing to notification delivery was under four seconds on a stable Wi-Fi connection.

Testing was carried out over a ten-day period with five volunteers, across different conditions rest, light activity, and mild exertion. Readings were cross-checked against a clinical pulse oximeter and a standard digital thermometer at periodic intervals. The comparison gave a baseline for accuracy. No formal clinical trial was conducted, and the system is not intended as a diagnostic-grade device, but the consistency of results across conditions gave reasonable confidence that the sensor pipeline is behaving as expected.

| Component | Model Used | Purpose | Unit Cost (Approx.) |
|---|---|---|---|
| **Microcontroller** | ESP32 Dev Board | Processing & Wi-Fi | ₹350 |
| **Heart Rate Sensor** | MAX30102 | Pulse detection | ₹180 |
| **Temperature Sensor** | LM35 / DS18B20 | Body temp measurement | ₹40 |
| **Cloud Platform** | Firebase (free tier) | Storage & alerts | Free |
| **IDE / Language** | Arduino IDE (C/C++) | Firmware development | Free |
| **Dashboard** | Web (HTML/CSS/JS + Firebase SDK) | User interface | Free |

*Table 3: Hardware and Software Components Used in the Experimental Setup*

## 5.0 Results

The system was evaluated over a ten-day monitoring period involving five adult volunteers. Sensor readings were collected continuously and cross-verified at regular intervals against a reference clinical pulse oximeter and a calibrated digital thermometer. The comparison was informal this was not a clinical trial but it was enough to assess whether the sensor pipeline was producing trustworthy output.

Heart rate readings from the MAX30102 sensor stayed within ±3 BPM of the reference device for most participants, with occasional spikes during sensor repositioning or movement artifacts. Body temperature readings tracked within 0.3°C of the reference thermometer under resting conditions. Accuracy drifted slightly during active use, which is expected skin-surface temperature measured by a contact sensor is not the same as oral or tympanic temperature, and slight discrepancies under exertion are a known limitation of this sensor category.

Alert latency the time from a threshold-crossing reading to notification delivery averaged 3.8 seconds on a stable Wi-Fi connection, with a maximum measured delay of 6.2 seconds across all tests. No missed triggers were observed during the testing window. Every reading that crossed a threshold produced an alert; no false negatives were detected, though a small number of motion-artifact readings did produce brief false alarms before the averaging filter smoothed them out.

Cloud logging worked reliably throughout the test period. Firebase handled all uploads without loss, and the web dashboard displayed readings within one to two seconds of their upload timestamp. The historical graphs made it straightforward to spot patterns that individual readings would not reveal — one participant's resting heart rate climbed steadily over three days during a period of reported poor sleep, something that would not have been visible from a single measurement.

| Parameter | Sensor Reading (Avg.) | Reference Value (Avg.) | Mean Deviation |
|---|---|---|---|
| **Heart Rate (BPM)** | 75.4 | 74.1 | ±2.8 BPM |
| **Body Temperature (°C)** | 36.9 | 36.7 | ±0.28°C |

*Table 4: Sensor Accuracy Compared to Reference Devices (10-Day Average)*

| Test Condition | Alerts Triggered | Alerts Delivered | Avg. Latency (s) |
|---|---|---|---|
| High Heart Rate (>100 BPM) | 12 | 12 | 3.6 |
| Low Heart Rate (<50 BPM) | 3 | 3 | 3.9 |
| High Temperature (>38°C) | 7 | 7 | 4.1 |
| Combined (HR + Temp) | 4 | 4 | 3.8 |

*Table 5: Alert System Performance During Testing*

The ablation tests ran alongside the main evaluation. Disabling the alert module had the most telling effect: data kept logging, but no one found out when something went wrong. A monitoring system that does not notify anyone is just a data recorder. Removing cloud connectivity took away remote access entirely, which defeats the point for home or rural use. Without the data averaging step, sensor output became noticeably jittery —

individual readings fluctuated by as much as ±8 BPM rather than ±3, making threshold decisions less reliable. Each test made the same argument from a different angle: pull out any one piece and something important breaks, even if the rest keeps running.

| Feature | Proposed System | Fitbit Sense | Commercial Patient Monitor | Smartwatch (Basic) |
|---|---|---|---|---|
| **Cost** | < ₹700 | ~₹24,000 | ~₹85,000+ | ~₹3,000+ |
| **Heart Rate Monitoring** | Yes | Yes | Yes | Yes |
| **Temperature Monitoring** | Yes | Limited | Yes | No |
| **Threshold Alerts** | Yes (auto) | Limited | Yes | No |
| **Cloud Data Logging** | Yes | Yes | Yes | No |
| **Home Use Without Technical Support** | Yes | Yes | No | Yes |
| **Expandable Sensors** | Yes | No | Partial | No |

*Table 6: Comparison with Existing Health Monitoring Solutions*

### 6.0 Applications, Discussion, and Ablation Study

### 6.1 Applications

The most obvious users are elderly people and those managing chronic conditions — people who would otherwise need frequent hospital visits just to confirm their numbers are still in range. Hospitals can use it too, for patients who do not need constant bedside supervision but still need watching, freeing up staff for cases that require hands-on attention. In rural areas, where the nearest clinic might be an hour away, a sensor that flags an abnormal reading at home gives people a heads-up they would not otherwise have.

None of this replaces a doctor, but it shifts the model slightly: from showing up when something is already wrong, to knowing earlier that something is heading in the wrong direction. That gap is where a lot of preventable harm happens.

### 6.2 Discussion

The results held up. Sensor readings stayed close to standard reference values, which is good enough for the kind of continuous home monitoring this system is designed for. It is not a clinical-grade device, but it does not need to be. When readings crossed set thresholds, alerts went out quickly no meaningful delay, no missed triggers. The cloud logging worked as expected, and the graphs made it easy to spot trends that a single reading would not reveal: a gradual temperature climb, a heart rate running slightly high all afternoon.

That is where continuous monitoring earns its keep. Not in catching dramatic spikes, but in making slow drift visible before it becomes a problem. The system does not eliminate the need for medical care — it gives both patient and doctor better information to work with when that care happens.

### 6.3 Ablation Study

To understand how much each component contributes, each part of the system was disabled independently and the effect was observed. Removing the alert module was the most telling: sensors kept running, data kept logging, but no one found out when something went wrong. A monitoring system that does not notify anyone is just a data recorder, and a data recorder is not useful in a home health context where the whole point is catching problems early.

Removing cloud connectivity had a similar effect: everything worked locally, but remote access disappeared, which defeats much of the purpose for home or rural use where the caregiver may not be physically present. Without the data averaging step, the sensor output became jittery and harder to trust individual readings fluctuated enough that the threshold logic would have generated false positives. Each component test made the same point from a different direction: the system only works as a whole. Remove any one piece and something important breaks, even if the rest is technically still running.

| Component Disabled | Observed Effect | Conclusion |
|---|---|---|
| **Alert Module** | Data logged; no notifications sent on threshold crossing | Real-time alerting is essential; without it the system is just a passive recorder |
| **Cloud Connectivity** | Local processing continued; remote access and logging lost | Cloud layer is critical for home and remote use cases |
| **Data Averaging Filter** | Sensor output became jittery; false alerts increased | Averaging is necessary to suppress noise-driven false positives |
| **Temperature Sensor** | Only heart rate remained; fever detection not possible | Multi-parameter monitoring adds meaningful coverage |

*Table 7: Ablation Study — Effect of Removing Individual System Components*

### 7.0 Conclusion

Health monitoring should not stop at the clinic door. The gap between appointments is where a lot of preventable harm happens, and IoT offers a practical way to fill it not with expensive clinical-grade equipment, but with inexpensive sensors, standard microcontrollers, and cloud tools that are already freely available.

The system described here does three things: it monitors heart rate and body temperature continuously, it checks those readings against safe thresholds, and it sends an alert when something goes wrong. That is a short list, but it covers the essentials. Testing over ten days with five participants showed sensor accuracy within clinically acceptable margins for a home monitoring context, alert delivery within four seconds of a threshold crossing, and no missed triggers across the test period.

The design is deliberately modest. Off-the-shelf components, embedded C firmware, a free-tier cloud database, and a basic web dashboard. The point was never to build something impressive it was to build something that works, that patients will actually use, and that costs little enough to be accessible to people who cannot afford commercial medical wearables. On those measures, the system delivers.

The ablation tests confirmed that each component earns its place. Remove the alerts and it becomes a passive recorder. Remove the cloud and it loses the remote access that makes it useful outside hospitals. Remove the data averaging and false alarms become a problem. The system only works as a whole.

What comes next is a matter of adding to an architecture designed to accommodate growth. Blood pressure and SpO2 sensors are straightforward additions. A machine-learning layer that learns what is normal for a specific patient rather than relying on population averages would improve threshold precision. Integration with hospital systems would close the loop between home monitoring and clinical care. None of that is far out of reach. For now, the baseline works: a sensor, a microcontroller, a cloud connection, and an alert. That is enough to give patients something they currently do not have a reason to pay attention before things get bad.

**References**

[1] Population Reference Bureau, "2014 World Population Data Sheet," [Online]. Available: http://www.prb.org/Publications/Datasheets/2014/2014-world-population-data-sheet.aspx

[2] OECD/European Commission, "Sweden – A Good Life in Old Age? Monitoring and Improving Quality in Long-Term Care," June 2013. [Online]. Available: http://www.oecd.org/els/health-systems/Sweden-OECD-EC-GoodTime-in-Old-Age.pdf

[3] F. Yuan Jie, Y. Yue Hong, X. Li Da, Z. Yan, and W. Fan, "IoT-Based Smart Rehabilitation System," IEEE Transactions on Industrial Informatics, vol. 10, pp. 1568–1577, 2014.

[4] X. Boyi, X. Li Da, C. Hongming, X. Cheng, H. Jingyuan, and B. Fenglin, "Ubiquitous Data Accessing Method in IoT-Based Information System for Emergency Medical Services," IEEE Transactions on Industrial Informatics, vol. 10, pp. 1578–1586, 2014.

[5] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An Information Framework for Creating a Smart City through Internet of Things," IEEE Internet of Things Journal, vol. 1, pp. 112–121, 2014.

[6] Jawbone Inc., "Jawbone Fitness Trackers," accessed April 2015. [Online]. Available: https://jawbone.com/up/trackers

[7] FitBit Inc., "Flex: Wireless Activity + Sleep Wristband," accessed April 2015. [Online]. Available: https://www.fitbit.com/flex

[8] Apple Inc., "Apple Watch," accessed April 2015. [Online]. Available: https://www.apple.com/watch/

[9] A. Pantelopoulos and N. Bourbakis, "A Survey on Wearable Sensor-Based Systems for Health and Prognosis," IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 40, no. 1, pp. 1–12, Jan. 2010.

[10] D. Son et al., "Multifunctional Wearable Devices for Diagnosis and Therapy of Movement Disorders," Nature Nanotechnology, pp. 1–8, 2014.

[11] A. Page, O. Kocabas, T. Soyata, M. Aktas, and J.-P. Couderc, "Cloud-Based Privacy-Preserving Remote ECG Monitoring and Surveillance," Annals of Noninvasive Electrocardiology, 2014.

[12] R. Paradiso, G. Loriga, and N. Taccini, "A Wearable Health Care System Based on Knitted Integrated Sensors," IEEE Transactions on Information Technology in Biomedicine, vol. 9, no. 3, pp. 337–344, Sept. 2005.

[13] Narassima M.S., Anbuudayasankar S.P., Shriram K. Vasudevan, Abhinavaram J., "Physicians' and Users' Perceptions Towards Wearable Health Devices," Indonesian Journal of Electrical Engineering and Computer Science, vol. 5, no. 1, pp. 234–242, Jan. 2017.

[14] A. Redondi, M. Chirico, L. Borsani, M. Cesana, and M. Tagliasacchi, "An Integrated System Based on Wireless Sensor Networks for Patient Monitoring, Localization and Tracking," Ad Hoc Networks, vol. 11, pp. 39–53, 2013.

[15] N. A. Abu Bakar, W. M. Wan Ramli, and N. H. Hassan, "The Internet of Things in Healthcare: An Overview, Challenges and Model Plan for Security Risks Management Process," Indonesian Journal of Electrical Engineering and Computer Science, vol. 15, no. 1, pp. 414–420, July 2019.

[16] C. Doukas and I. Maglogiannis, "Bringing IoT and Cloud Computing towards Pervasive Healthcare," in Proc. Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), IEEE, 2012, pp. 922–926.

[17] S. Amendola, R. Lodato, S. Manzari, C. Occhiuzzi, and G. Marrocco, "RFID Technology for IoT-Based Personal Healthcare in Smart Spaces," IEEE Internet of Things Journal, vol. 1, no. 2, pp. 144–152, 2014.

[18] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An Information Framework for Creating a Smart City through Internet of Things," IEEE Internet of Things Journal, vol. 1, no. 2, pp. 112–121, 2014.

[19] M. Bazzani, D. Conzon, A. Scalera, M. Spirito, and C. Trainito, "Enabling the IoT Paradigm in E-Health Solutions through the VIRTUS Middleware," in Proc. IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), June 2012, pp. 1954–1959.

[20] A. Benharref and M. Serhani, "Novel Cloud and SOA-Based Framework for E-Health Monitoring Using Wireless Biosensors," IEEE Journal of Biomedical and Health Informatics, vol. 18, no. 1, pp. 46–55, Jan. 2014.

[21] C. Rolim et al., "A Cloud Computing Solution for Patient's Data Collection in Health Care Institutions," in Proc. Second International Conference on eHealth, Telemedicine, and Social Medicine (ETELEMED), Feb. 2010, pp. 95–99.

[22] M. Lan et al., "Wanda: An End-to-End Remote Health Monitoring and Analytics System for Heart Failure Patients," in Proc. ACM Conference on Wireless Health (WH '12), New York, NY, 2012, pp. 9:1–9:8.

[23] L. Wei et al., "A Practical Tool for Visualizing and Data Mining Medical Time Series," in Proc. 18th IEEE Symposium on Computer-Based Medical Systems, June 2005, pp. 341–346.

[24] Y. Mao et al., "Medical Data Mining for Early Deterioration Warning in General Hospital Wards," in Proc. IEEE 11th International Conference on Data Mining Workshops (ICDMW), Dec. 2011, pp. 1042–1049.

[25] V. Ukis, S. Tirunellai Rajamani, B. Balachandran, and T. Friese, "Architecture of Cloud-Based Advanced Medical Image Visualization Solution," in Proc. IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), Oct. 2013, pp. 1–5.