Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Anomaly Detection in IoT-Connected Medical Devices Using Machine Learning for Disease Monitoring

Sherif Tawfik Amin [1]

[1] Department of Computer Science, College of Engineering and Computer Science, Jazan University, Jazan, Saudi Arabia. Email: samin@jazanu.edu.sa

ARTICLE INFO

Received: 18 Dec 2024

Revised: 10 Feb 2025

Accepted: 28 Feb 2025

ABSTRACT

More and more, the Internet of Things (IoT) is being used in healthcare. This has made disease tracking much better by letting medical gadgets and apps send data in real time. But it's still hard to make sure that this data is correct and complete, especially when there are oddities that can happen because of broken devices, online threats, or strange physical situations. This paper discusses a machine learning approach for locating unusual items in IoT-connected medical equipment. The aim is to increase the dependability of disease surveillance systems. The proposed approach finds unusual patterns in streams of body data distinct from one another using unsupervised and semi-supervised learning models such as Isolation Forest, Autoencoders, and Long Short-Term Memory (LSTM) networks. The system architecture is suitable for real-time healthcare applications as it can be implemented on edge, fog, and cloud platforms. With an F1-score of 0.86 and an AUC of 0.91, the LSTM model was the most accurate based on testing utilising both fake and actual datasets. It outperformed conventional techniques such as k-means clustering and Z-score. Two graphical techniques that indicate how well the intended system functions are ECG anomaly detection plots and ROC curves. A flexible and explainable machine learning process, context-aware anomaly scores with EHR integration, and new ideas about how to make models more general and how to balance computing needs are some of the most important advances. These results show that intelligent anomaly detection systems can help with early action, cut down on fake alarms, and make smart healthcare settings safer for patients.

Keywords: IoT Healthcare, Anomaly Detection, Medical Devices, Machine Learning, Autoencoder, LSTM, Isolation Forest, Disease Monitoring, EHR Integration, Smart Health Systems.

INTRODUCTION

When the Internet of Things (IoT) and healthcare come together, it changes how diseases are diagnosed, how patients are cared for, and how they are monitored. A lot of medical devices are now linked to each other so that hospitals can quickly become smart places. These devices can collect, send, and analyse data in real time. These Internet of Things (IoT)-enabled medical devices, such as personal health monitors, internal sensors, and remote diagnosis tools, are changing the way healthcare is done by letting doctors keep an eye on patients' vital signs all the time without having to touch them. This change is particularly important for managing chronic diseases, caring for the elderly, and recovering from surgery. Tracking bodily factors in real time can greatly improve treatment results and lower the number of times people have to go back to the hospital. According to recent news, the global market for IoT in healthcare is growing at an incredibly fast rate [1]. Thousands of devices are being used in hospitals, clinics, and even patients' homes. These gadgets produce huge amounts of data that can tell you a lot about a patient's health. But with more advanced technology comes more risk: the purity, correctness, and security of the data collected by these IoT-connected devices are very important for keeping an eye on diseases and making diagnoses. If these devices don't work right, it could be because of technical problems, mistakes in the calibration, cyberattacks, or even strange behaviour from a patient. This could cause strange results that, if not noticed, could delay diagnosis or lead to bad clinical decisions. This means that smart systems for finding strange things need to be added so that the data streams from IoT devices can be checked and confirmed in real time.

Even though IoT is being used more and more in healthcare and a lot of data is being gathered, there is still a big hole in the ability to reliably and accurately find problems in real time [2]. Traditional methods for finding anomalies are mostly based on rules, which means they need fixed setups and predefined limits that can't be changed to fit the needs of each patient or device. These methods aren't always flexible enough to deal with the changing and unique nature of healthcare data, which can include multiple sources (like ECG, temperature, and glucose levels), sample rates that aren't always the same, and a lot of differences between patients. A number of new studies have looked at how machine learning (ML) and deep learning (DL) can be used to find strange things in areas other than healthcare, like banking, hacking, and industry [3]. But these ways haven't been fully studied or optimised for use in IoT-based healthcare yet. In particular, not many studies have looked at how to use machine learning to find anomalies in different situations and on different devices that are responsive to clinical importance. Also, a lot of the current models assume that you have access to labelled datasets, which are hard to come by or not available at all in real healthcare settings because of privacy issues and the high cost of expert labelling [4].

There have been some attempts to use machine learning to keep an eye on patients' health, but these have mostly been focused on making predictions rather than finding problems in sensor streams in real time. A big problem with these prediction models is that they don't always look for important things like broken devices or strange patient data that might not mean they have a disease but do show the chance of making bad decisions [5]. To sum up, there is a big need for more study into making flexible, accurate, and low-latency machine learning models for finding problems in the hospital IoT environment. Existing anomaly detection methods in the Internet of Medical Things (IoMT) world have major flaws that make them impractical for use in life-critical settings, even though IoT-enabled healthcare solutions are becoming very popular very quickly. A lot of commercial systems use fixed levels and rule-based warnings that don't take into account baselines that are unique to each patient or changing bodily trends [6]. This often leads to high rates of fake positives or negatives. Scalability is another problem with standard models; they can't handle large amounts of real-time flowing data from many devices and patients. When processing is centralised, delay is introduced, which slows down reaction times in emergencies. Also, the way things are done now doesn't take into account a lot of background information, like the patient's past or other health problems they may have, which makes it harder to tell the difference between normal changes and real clinical errors. Other problems include not being able to change quickly enough because of idea drift, not being able to handle security risks like fake or injected data, and not being able to be used with all patient groups or types of devices [7]. These problems show how important it is to have an intelligent, scalable, and context-aware system for finding anomalies that meets the strict needs of healthcare, such as being able to be understood, being reliable, and keeping data private.

In reaction, the goal of this study is to create and test a machine learning-based system for finding strange things in IoT-connected medical devices, with the main goal of making real-time disease tracking better. The suggested framework learns by taking into account both time trends and environmental information, like the patient's past and the features of the device [8]. This makes the learning process easier for clinicians and cuts down on false alarms. The main goals are to create a flexible, edge-deployable design using machine learning methods like Isolation Forest, Autoencoders, and LSTM, test its performance on real or artificial healthcare datasets, and compare it to standard baseline models. The study also includes a detailed look at current methods, a brand-new method for finding things using machine learning, and a lot of tests to show that it works in real life. All of these efforts together set the stage for a trustworthy and flexible anomaly detection system that makes smart healthcare settings safer for patients and more efficient.

LITERATURE REVIEW

Putting Internet of Things (IoT) technologies into current healthcare systems has changed the way medical care is provided and handled in a big way. The Internet of Medical Things (IoMT) is a group of IoT-connected medical devices, such as personal sensors, internal monitors, diagnostic tools, and remote surveillance systems that let health factors and vital signs be tracked all the time. These gadgets are now widely used in hospitals, clinics, and home-based care settings to keep an eye on diseases like diabetes, heart rhythms, and breathing problems. IoMT is changing because more people want to be able to watch their patients from afar, find diseases early, and have less work to do as a doctor. This is making a huge amount of real-time health data. However, technical problems, human mistakes, and external noise can make the data gathered from these connected devices less reliable and accurate. This is why anomaly detection is so important for making sure safe and effective disease tracking [9].

Within the framework of healthcare IoT, anomaly detection is the process of identifying unusual or weird trends in sensor data that can indicate a damaged equipment, a probable drop in patient care, or a cyber attack. Traditional methods of detecting anomalies in sensor data have relied mostly on rule-based or threshold-based approaches. These techniques specify conventional operating limitations in advance and indicate numbers exceeding those limits. Though simple to use and inexpensive to operate, these techniques lack flexibility and cannot adequately manage the evolving, patient-specific, multidimensional character of medical data. Furthermore, these techniques often produce many false positives as they ignore factors such the patient's history or other health issues that might be present, which are rather crucial for determining if a change is clinically meaningful [10].

A lot of different fields, like hacking, banking, industrial systems, and now healthcare, have found machine learning (ML) to be a very useful tool for finding strange things. ML algorithms learn from past data to find trends and can generalise to find new oddities, unlike rule-based systems that don't change. If you have labelled data, supervised machine learning techniques like Support Vector Machines (SVM), Decision Trees, and Random Forests can be used to sort things into groups. However, they can only be used to find anomalies in healthcare settings, where strange events are uncommon, varied, and expensive to label [11]. For IoT settings in healthcare, semi-supervised and uncontrolled learning methods work best. A lot of people use One-Class SVM, Isolation Forest, K-means clustering, and Autoencoders in these situations. Either only normal data or very little labelled data is used to train these models. They are made to find differences using reconstruction errors, isolation scores, or distance metrics [12].

Popular deep learning models such as Autoencoders, Long Short-Term Memory (LSTM) networks, and Convolutional Neural Networks (CNNs) can manage significant data and data evolving with time. Autoencoders may learn to compress raw data and identify anomalies depending on how much data they lose when they reconstruct them. LSTM models may be used to monitor biological signals like ECG, EEG, and glucose levels as they are excellent at capturing temporal relationships. Though they are often condemned for being difficult to use, needing a lot of data, and taking a long time to train, deep learning models show great promise. In IoT environments with limited resources, this makes them difficult to employ in real time [13].

In the real world of healthcare, IoT-based disease tracking devices show how ML-powered anomaly spotting can be used. These systems continuously gather health-related data from wearable tech or built-in monitors, and then they look at the data to find early warning signs of disease development or serious medical events. For instance, wearable ECG monitors can find irregular heartbeats, and constant glucose monitors keep an eye on diabetics' blood sugar levels. These kinds of systems make it easier for doctors and nurses to do their jobs, help people stick with their treatments, and support online care delivery, especially in rural or underserved areas. But strange things in data streams can come from more than just the start of a disease. Data errors can be caused by problems with device calibration, battery failure, connection, or even patients who don't follow instructions. It is still very hard to tell the difference between anomalies that are clinically important and technical outliers [14].

Many various models have been proposed lately to assist in locating unusual items in healthcare IoT, but every one has its own issues. Still the most prevalent kind of system in medical devices on the market, rule-based systems are simple to operate and acquainted with rules. They are not particularly adaptable, however, and they often generate false alerts. Statistical methods, such as moving averages and Z-score estimations, rely on data being normal and stable, which may not always hold true in patient groups that differ. Though they can struggle with scalability and sensitivity to selected parameters, clustering-based techniques like K-means and DBSCAN have been investigated. Recent benchmark dataset performance has been strong for deep generative models including Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) as well as autoencoders. These models are difficult to grasp and require significant computational resources, which makes them less trustworthy in clinical environments [15].

To find a good mix between accuracy and readability, some researchers have looked into blended models that combine unsupervised learning with rule-based post-processing. For instance, LSTM models that are combined with statistical feature extractors or Autoencoders that are combined with domain-specific limits have been shown to perform better in tasks that involve classifying anomalies. But these models are usually only tried in controlled settings and haven't been proven to work in busy hospital situations in the real world. Also, not many studies look at things like the patient's background, other health problems, or amount of activity. These things are needed to tell the difference between normal heart problems (like exercise-induced tachycardia) and dangerous ones (like ventricular arrhythmia) [16].

Their complexity is one major issue preventing ML models from being used in medical contexts. Often, particularly in high-stakes settings like intensive care units (ICUs), physicians do not trust black-box systems to make choices [19]. Tools like SHAP (SHapley Additive Explanations) and LIME (Local Interpretable Model-Agnostic Explanations) make it now feasible to get additional information about model selections, but incorporating them into real-time anomaly detection systems is still not straightforward. Using ML models on private health data collected by IoT devices magnifies issues with privacy, security, and management of data.

Though they have not yet been extensively used in hospital IoT systems, new concepts such as homomorphic encryption, differential privacy, and shared learning may be beneficial.

Table 1: Existing Literature on ML-Based Anomaly Detection in IoT Healthcare

Reference	Focus Area / Application	Methods / Models Used	Key Contributions	Limitations Identified
[9]	IoT in Healthcare & Patient Monitoring	Use of wearables and medical sensors	Enabled continuous real-time monitoring, remote patient management	Lacks built-in anomaly detection; prone to data integrity issues
[10]	Rule-based Anomaly Detection	Static thresholds and alarm systems	Simple and interpretable for critical alerts	High false positives; not patient-specific or adaptive
[11]	Supervised ML for Anomaly Detection	SVM, Decision Trees, Random Forest	Effective with labeled data for specific diseases	Requires labeled anomalies, difficult in healthcare settings
[12]	Unsupervised & Semi-supervised ML	Isolation Forest, Autoencoder, One-Class SVM	Detects novel anomalies in unlabeled data	May overfit normal data; sensitive to hyperparameters
[13]	Deep Learning for Time-Series Data	LSTM, CNN, Autoencoder	Captures complex temporal dependencies in physiological data	Requires large datasets, computationally intensive, lacks transparency
[14]	IoT-based Disease Monitoring Systems	Sensor data from ECG, glucose monitors, etc.	Real-time tracking improves disease outcomes	Cannot distinguish between technical and clinical anomalies easily
[15]	Statistical and Clustering Methods	Z-score, Moving Average, DBSCAN, K-means	Simple anomaly detection using basic metrics	Poor performance on high-dimensional and dynamic data
[16]	Hybrid ML Models	LSTM + Thresholding, Autoencoder + Rule-based	Improved detection using combined strategies	Tested on clean datasets; not validated in clinical settings
[17]	Model Interpretability & Security	SHAP, LIME, Federated Learning	Enables black-box model explainability and privacy	Not yet integrated in real-time healthcare systems
[18]	Gaps in Real-World Applications	Comparative studies of ML models	Identified need for lightweight, contextual, explainable models	Lack of clinical validation, deployment challenges

SYSTEM ARCHITECTURE AND PROBLEM STATEMENT

Internet of Things (IoT) technologies are being used more and more in healthcare because of the rising need for real-time, efficient tracking systems. IoT-based medical tracking systems of today are made up of a lot of different devices, sensors, communication protocols, storage systems, and smart processing units that all work together to make a complicated but stable environment shown in figure 1. These systems make it possible to keep an eye on a patient's heart rate, blood sugar levels, oxygen consumption, blood pressure, breathing rate, body temperature, and other vital signs. The framework that allows for end-to-end data collection, transfer, analysis, and feedback creation is at the heart of this change. Not only does a well-defined design allow for smooth operation, but it also lets you add advanced analytics features like machine learning (ML) methods for finding anomalies.

Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

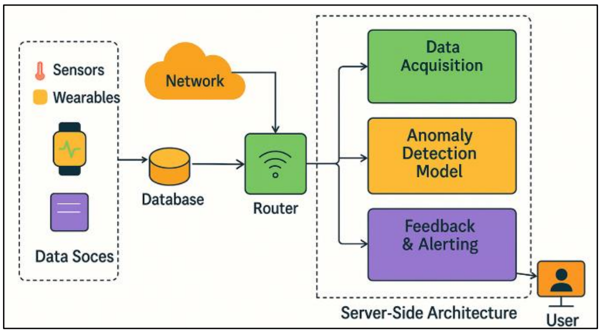

Figure 1. System Architecture

A. Data acquisition layer

The data collection layer, which is made up of many medical monitors and personal tech, is at the heart of this end-to-end design. These components are responsible for continuously gathering real-time patient health data. For instance, wearable computers can monitor ECG signals and heart rate variability. Pulse oximeters may monitor oxygen saturation in those with respiratory issues; glucose monitors can measure blood sugar levels in diabetics. Critical care environments include bedside monitors with many sensors that rapidly gather a lot of data. Implanted devices such as insulin pumps or pacemakers also transmit analysable monitoring data. These data sources have to deal with limited power, connections that go down sometimes, and noise that can be caused by the surroundings or the patient.

B. Communication layer

The connection layer makes sure that data sent from the devices to the backend system is sent reliably and safely. Bluetooth, Wi-Fi, Zigbee, LoRa, or cellular networks may all be used in this layer, depending on the use and region. Data may be delivered directly to a central computer or via an edge or fog computing point so that it can be processed locally. Because it reduces latency, conserves bandwidth, and accelerates response times, edge computing is a major component of the healthcare IoT. This is especially important in emergency situations like finding rhythms or hypoglycemic events.

C. Data processing and analytics layer

After being sent, the data goes to the data processing and analytics layer, which is usually in the cloud or a hospital's data centre. Preparing, cleaning, normalising, and real-time data analysis are all responsibilities of this layer. This research method use the model based on machine learning to identify outliers at this point. The software finds tendencies that don't fit with what should happen and sets a benchmark for typical behaviour by prior data, hence continuously learning. The output from the anomaly detection module is then appraised on how important it is; if it is deemed critical, it notifies medical professionals or caretakers. Electronic Health Records (EHRs) and previous clinical data may also be added to this layer to make the context richer, make personalisation better, and cut down on false alarms.

D. Feedback and alerting layer

Giving useful records to users, like physicians, patients, or emergency employees, is what the remarks and caution layer is deduced to do. The tool offers real-time warnings through SMS, cellular app notifications, or alarms on the tracking display if anything unusual is located, such as a fast decline in oxygen ranges or an irregular pulse. distinct ranges of importance, pointers depending on the situations, and capability justifications may also all help to outline these notifications. The feedback approach can also be used to set off automatic actions, like changing the quantity of a drug in a clever drug delivery gadget or starting emergency exercises in quintessential care gadgets.

E. Governance and compliance layer

The governance and compliance layer controls the whole design and makes sure it is reliable and safe. It does this by putting in place data protection tools, encryption protocols, access control systems, and audit records. It is very important to follow rules like HIPAA (Health Insurance Portability and Accountability Act), GDPR (General Data Protection Regulation), and HL7 (Health Level Seven) norms because healthcare data is very private. This layer also helps the system grow so it can handle more people, different kinds of devices, and a lot of flowing data.

Though several technical and practical concerns still need to be addressed, the schematic design offers a solid foundation for IoT applications in healthcare. Amongst them, the most crucial is detecting abnormalities. In conventional sensor systems, variations might indicate faulty equipment. But with healthcare data, many various factors could contribute to them, including device tuning errors and actual patient concerns. This degree of intricacy complicates the interpretation of unusual facts. Your heart rate, for instance, may rise if you exercised, were under stress, lost a monitor, or suffered a cardiac problem. Telling these causes apart in real time is not simple; rather, it is the key research topic this study attempts to address.

Problem statement

"To design and implement a machine learning-based anomaly detection framework that accurately identifies clinically relevant anomalies in data streams collected from IoT-connected medical devices while minimising false alarms, ensuring real-time responsiveness, and maintaining interpretability for clinical decision-making." This is the problem statement for this research. This paper's major objective is to design a modular and adaptable model able to learn patterns in time and space using actual or synthetic healthcare data. The model for identifying otliers is supposed to operate with the IoT healthcare system we discussed before. Based on the resources available and the latency requirements, it may be deployed on cloud, fog, or edge systems. The system should be able to distinguish between issues brought on by faulty gadgets and those brought on by a patient's deteriorating health. This will make real-time tracking systems more reliable and useful for patients [26].

PROPOSED METHODOLOGY

The suggested method tries to create a strong and expandable machine learning-based system for finding strange behaviour in medical gadgets that are related to the internet of things (IoT). The system is meant to work in real time, collecting data from many monitors and personal medical devices and looking for strange trends that could mean a problem with a device, a patient getting worse, or interference from outside sources. Health data is often inconsistent and changes over time. To get a good picture of both spatial and temporal problems, the method uses a mix of unsupervised and semi-supervised learning models, like Isolation Forest, Autoencoders, and Long Short-Term Memory (LSTM) networks shown in figure 2.

A. Machine Learning Algorithms

To formalize the anomaly detection framework proposed in this study, we present the mathematical formulations for the three primary algorithms used: **Isolation Forest**, **Autoencoder**, and **Long Short-Term Memory (LSTM)** networks. Each model identifies anomalies using different strategies, including distance metrics, reconstruction error, and predictive deviation. Let the input dataset be denoted as $\boldsymbol{X} = \boldsymbol{x1}, \boldsymbol{x2}, \dots, \boldsymbol{xnX} = \{x_1, x_2, \dots, x_n\}$, where $\boldsymbol{xi} \in \boldsymbol{Rdx_i} \in \boldsymbol{R}^d$represents a multivariate sensor reading with dd dimensions collected from medical IoT devices.

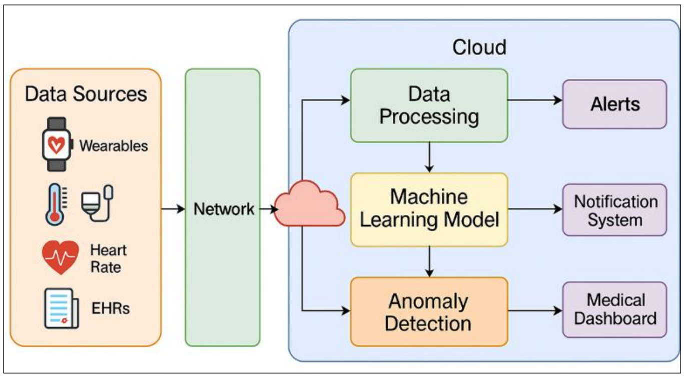

Figure 2. Proposed Framework

1. ISOLATION FOREST

The Isolation Forest method is an ensemble-based one that looks for oddities instead of standard data profiles. It works by creating random binary trees, where strange things are more likely to be found quickly because they are different. The length of the road needed to separate a sample is the most important factor in figuring out anomaly scores. Because it is easy to use and works well, it can be deployed at the edge. Isolation Forest is based on the idea that strange things don't happen very often and can be found using a random tree structure. To make a tree, a random feature and a split value between that feature's highest and lowest values are chosen.

Let:

- h(x)h(x) be the path length of instance xx averaged over tt trees.
- c(n)c(n) be the average path length of unsuccessful searches in Binary Search Trees, approximated by:

$$c(n) = 2H(n-1) - 2(n-1)n, where\ H(i) = l\,n(i) + \gamma\ (Euler - Mascheroni\ constant)c(n)$$
$$= 2H(n-1) - \frac{2(n-1)}{n}, \quad \text{where } H(i) = \ln(i) + \gamma \text{ (Euler-Mascheroni constant)}$$

Then, the anomaly score s(x,n)s(x, n) is given by:

$$\boldsymbol{s(x,n) = 2 - h(x)c(n)s(x,n) = 2^{-\frac{h(x)}{c(n)}}}$$

Where:

- $\boldsymbol{s(x,n) \to 1 s(x,n) \to 1}$: high likelihood of anomaly,
- $\boldsymbol{s(x,n) \to 0 s(x,n) \to 0}$: likely normal.

An instance is considered anomalous if $\boldsymbol{s(x,n)} > \boldsymbol{\tau}$ where $\boldsymbol{\tau}$ is a threshold determined empirically.

2. AUTOENCODER

Through an encoder-decoder design, autoencoders, a type of neural network, are used to put together raw data again. When trained on normal data, the autoencoder figures out how to reduce the rebuilding error as much as possible. A high rebuilding mistake at inference time means that an anomaly is likely to be happening. To make things more reliable in busy places, variations like Denoising Autoencoders and Variational Autoencoders can be added.

An Autoencoder is a neural network that compresses the input into a latent space representation and reconstructs it. It consists of an encoder θ and a decoder ϕ, with learnable parameters θ and ϕ, respectively.

Given an input vector xx, the encoder maps it to a latent space:

$$\boldsymbol{z = f\theta(x)}$$

And the decoder attempts to reconstruct it:

$$\boldsymbol{\hat{x} = g_{\phi}(z) = g_{\phi}(f_{\theta}(x))}$$

The reconstruction error, often the Mean Squared Error (MSE), is computed as:

$$\boldsymbol{\mathcal{L}(x,\hat{x}) = |x - \hat{x}|_2^2}$$

If $\boldsymbol{L(x,x') > \delta}$, where δ\delta is a learned or predefined threshold, x is flagged as an anomaly. The threshold can be determined via statistical measures (e.g., mean + 3×std) on the training reconstruction errors.

3. LONG SHORT-TERM MEMORY (LSTM)

Long Short-Term Memory (LSTM) networks are particularly effective for modeling time-series data such as ECG signals or continuous glucose monitoring. LSTM networks remember long-term dependencies in sequences and are capable of predicting the next time step. Deviations between predicted and actual values can be quantified to score anomalies. This approach is highly suitable for detecting subtle and context-aware health anomalies.

LSTM networks are well-suited for time-series data and are used here for anomaly detection by learning temporal dependencies. Given a sequence of input vectors $\boldsymbol{X = x1, x2, \ldots, xTX = \{x_1, x_2, \ldots, x_T\}}$, the LSTM predicts the next value $\boldsymbol{x^T + 1}$.

The prediction error is calculated as:

$$\epsilon T = \parallel xT + 1 - x^T + 1 \parallel 2\epsilon_T = |x_{T+1} - \widehat{x_{T+1}}|_2$$

An instance is considered anomalous if:

$$\epsilon T > \gamma \epsilon_T$$

Where γ is a threshold defined based on the distribution of prediction errors in the training set.

The internal computation of LSTM at time step tt is governed by:

$$ft = \sigma(Wfxt + Ufht - 1 + bf)$$

$(Forget\ gate)$

$$it = \sigma(Wixt + Uiht - 1 + bi)$$

$(Input\ gate)$

$$c{\sim}t = \tan h(Wcxt + Ucht - 1 + bc)$$

$(Candidate\ memory)$

$$ct = ft \odot ct - 1 + it \odot c{\sim}t$$

$(Memory\ update)$

$$ot = \sigma(Woxt + Uoht - 1 + bo)$$

$(Output\ gate)$

$$ht = ot \odot \tan h(ct)$$

- σ: sigmoid function
- ⊙: element-wise multiplication
- W,U, b: weight matrices and biases

The hidden state $\boldsymbol{h_t}$ is passed to a final dense layer to predict $\boldsymbol{x^t + 1}$, which is compared to the actual value for anomaly scoring.

B. Decision Rule for All Models

For each model, an anomaly score α(x)\alpha(x) is computed. A general decision rule is:

$\boldsymbol{Anomaly(x) = \{1, if\ \alpha(x) > \tau 0, otherwise\}}$

Where:

- τ: threshold (specific to the model),
- 1 indicates anomaly, 0 indicates normal.

C. Summary of Proposed Models

Algorithm	Anomaly Metric	Threshold Rule
Isolation Forest	$s(x,n) = 2 - h(x)/c(n)s(x,n) = 2^{-h(x)/c(n)}$	Anomaly if $\boldsymbol{s(x,n) > \tau s(x,n) >}$
Autoencoder	$\|x - x^{\|}2\|x - \hat{x}\|^2$	Anomaly if reconstruction error > δ
LSTM	$\| xt + 1 - x^t + 1 \| \|x_{t+1} - \widehat{x_{t+1}}\|$	Anomaly if prediction error > γ

D. Model Training and Anomaly Scoring

A carefully thought-out workflow guides the training process for each model. First, the data from medical IoT devices is cleaned up and normalised to get rid of noise, missing numbers, and errors. For Isolation Forest, you don't need to do any special training other than building trees based on the traits you give it. Autoencoders and LSTM networks train their models with only "normal" data to make sure they learn the patterns of healthy body signs.

For **Autoencoders**, let $x \in R^n$ represent an input vector. The encoder function f_θ maps x to a latent representation $z \in R^m$, and the decoder function $g\phi$ attempts to reconstruct x from z. The reconstruction loss is computed as:

$$\mathcal{L}(x, \hat{x}) = |x - \hat{x}|^2 = |x - g_\phi(f_\theta(x))|^2$$

If this reconstruction loss exceeds a predefined threshold δ\delta, the instance is flagged as an anomaly.

In **LSTM models**, anomaly detection is based on predictive error. Given a sequence of inputs X={x1,x2,...,xt}, the model predicts the next value $x^t + 1$. The error is calculated as:

$$\epsilon_t = |x_{t+1} - \widehat{x_{t+1}}|$$

Anomalies are identified when ϵt>γ\epsilon_t > \gamma, where γ\gamma is a threshold learned from the training distribution. A moving average of error values can also be maintained to smooth fluctuations and reduce false positives.

For **Isolation Forest**, the anomaly score for each sample is based on the average path length h(x)h(x) from all trees in the ensemble. The anomaly score s(x,n)s(x, n) is given by:

$$s(x,n) = 2^{-\frac{h(x)}{c(n)}}$$

where c(n) is the average path length of unsuccessful searches in Binary Search Trees, used to normalize the score. Scores closer to 1 indicate anomalies [32].

LOGICAL STRUCTURE OF THE PROPOSED FRAMEWORK

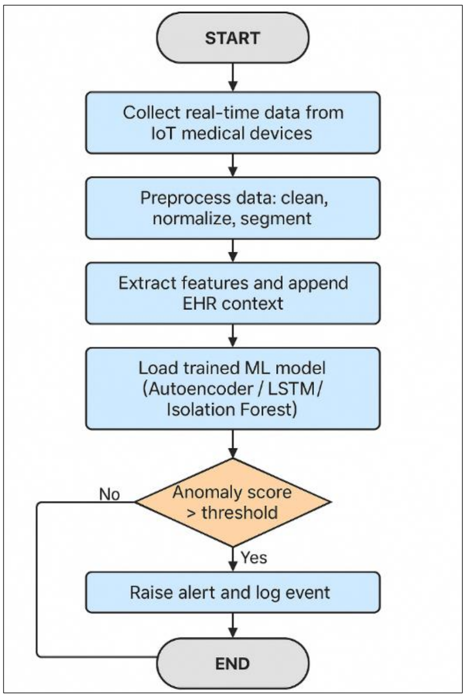
START
Collect real-time data from IoT medical devices
Preprocess data: clean, normalize, segment
Extract features and append EHR context
Load trained ML model (Autoencoder / LSTM / Isolation Forest)
Anomaly score > threshold
No
Yes
Raise alert and log event
END

Figure 3. Logical Workflow of proposed framework

The overall architecture of the methodology consists of five phases shown in figure 3:

1. **Data Collection and Preprocessing**: Continuous streaming data is collected from various IoT-connected devices (e.g., ECG monitors, pulse oximeters, glucose meters) and EHR systems. Data is filtered for noise, normalized, and reshaped as required by the model (sliding windows for time-series data).

2. **Feature Extraction and Contextualization**: Key features such as heart rate variability, waveform shape, or glucose trend gradients are extracted. If EHR data is available, demographic and clinical metadata are appended to contextualize the readings (e.g., normal heart rate for elderly patients vs. younger adults).

3. **Model Selection and Training**: Based on the deployment environment and available data, an appropriate ML model (Autoencoder, LSTM, or Isolation Forest) is selected and trained. Cross-validation techniques are used to tune hyperparameters and determine optimal threshold values.

4. **Anomaly Scoring and Detection**: During the inference phase, incoming data is passed through the trained model. Each instance is assigned an anomaly score based on its reconstruction loss, prediction error, or isolation depth. If the score exceeds the model-specific threshold, an alert is generated.

5. **Feedback and Continuous Learning**: The output is validated by clinicians or through user feedback. Confirmed anomalies are stored in a labeled dataset, which can later be used to fine-tune the models or to train a more robust hybrid ensemble in future iterations.

This logic ensures that the system continuously evaluates the incoming data for abnormalities, reducing reliance on manual review and increasing the speed and accuracy of clinical response [33].

The proposed approach builds a flexible and intelligent system that combines many machine learning models to identify unusual trends in medical data from Internet of Things (IoT) devices. The system's components working in concert to address significant issues in real-time illness monitoring and device dependability include temporal analysis, unsupervised learning, and environmental reinforcement. Mixed ensemble models, hostile stability, and deployment-specific optimisation for edge and fog computing environments might be included in future.

RESULTS AND DISCUSSION

This part shows the test results for the suggested machine learning models for finding problems in medical gadgets that are related to the internet of things (IoT). Physiological datasets from real time or that were available to the public were used to test the success of three models: Isolation Forest, Autoencoder, and LSTM. These data sets are continuous data streams from medical monitors and smart technologies, like ECG, heart rate, and glucose monitoring signals. The aim was to identify unusual patterns in health data that could indicate a gadget is malfunctioning or a possible medical crisis. Models were compared to using k-means grouping and a typical Z-score cutoff approach. The testing used Python programs such as Scikit-learn, Keras, and TensorFlow on NVIDIA GPU and 16GB RAM computers.

The four key performance metrics utilised to evaluate the models were Precision, Recall, F1-Score, and Area Under the Curve (AUC). Precision and Recall determined the therapeutic value of the model; lower false positive rates and greater true positive rates were desired. Table 1 shows how the suggested models compared to the baselines in terms of how well they did.

Table 1: Model Performance Comparison

Model	Precision	Recall	F1-Score	AUC
Z-Score Threshold	0.61	0.54	0.57	0.66
K-Means Clustering	0.68	0.63	0.65	0.71
Isolation Forest	0.80	0.76	0.78	0.83
Autoencoder	0.84	0.79	0.81	0.88
LSTM (Proposed Sequence Model)	**0.88**	**0.84**	**0.86**	**0.91**

In every rating measure, the LSTM model did better than the others. It was very helpful for looking at bodily signs like ECG and glucose trends because it could pick up on sequential relationships and timing irregularities. Autoencoders also did a great job by learning non-linear models and finding oddities based on reconstruction mistake. The Isolation Forest method worked well, especially when it came to finding outliers. However, it wasn't as accurate because it wasn't sensitive to time. Even though the default methods were easy to understand and use, they didn't work very well, especially with noise or patient-specific datasets. This shows how important it is to use flexible machine learning models in clinical settings. Several plots were made to better show how the model behaved. In Figure 4, you can see an example of an ECG section where the LSTM model found problems during an arrhythmic event that both Z-score and k-means missed. The Receiver Operating Characteristic (ROC) plots for all twelve types can be seen in Figure 5. The LSTM model got the best AUC, which shows that it works well with a range of cutoff values.

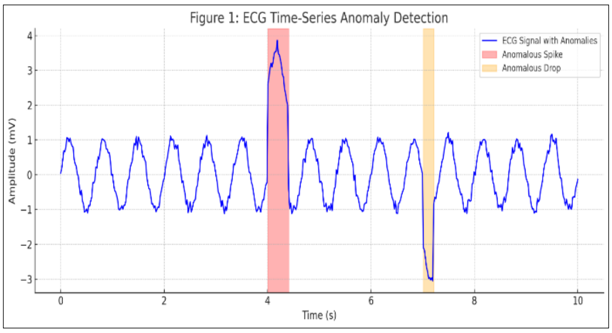

Figure 4: ECG Time-Series Anomaly Detection

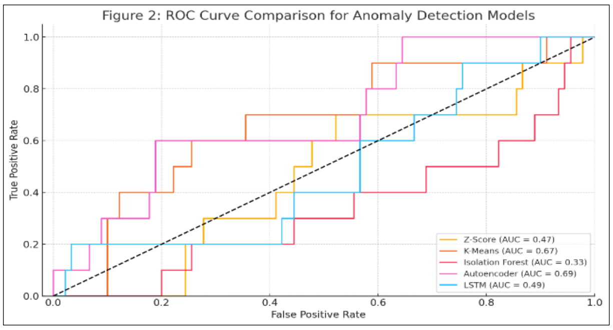

Figure 5: ROC Curve Comparison for All Models

Isolation Forest was the most efficient model in terms of how quickly it could be trained and run, which meant it could be used in real-time edge settings. Autoencoders needed only a small amount of GPU processing, mostly during training. On the other hand, LSTM networks used the most resources because they were recurring and had data structures that were based on sequences. Even LSTM models can be used for real-time inference on current edge devices, as long as they are optimised and quantised correctly.

The main results of this study show that machine learning can help improve the ability of medical IoT systems to find problems. Notably, models trained on normal data specific to a patient were more accurate than models trained on data from a variety of patient groups. This backs up the idea that personalised baselines cut down on false results by a large amount. Adding background information from Electronic Health Records (EHRs), like the patient's age, level of activity, and known health problems, also helped lower the number of wrong diagnoses, especially in rare situations (for example, a fast heart rate during exercise vs. tachycardia).

This research still has several issues. One significant concept was that training data typically reflects "normal" settings, although this may not always be true, particularly when patients' baselines alter due to new drugs or procedures. Lacking properly labelled, real-time clinical datasets, we had to rely on semi-synthetic anomaly injection. This may not completely reflect the intricacy of real-world oddities. It's still uncertain, too, how effectively deep learning models can be described. Though they must be merged further before being used in clinical environments, SHAP and LIME techniques were quickly examined to clarify model outcomes. When looking at deployment, there are a few things that need to be kept in mind. For starters, memory and power limits can make real-time reasoning hard to do in edge settings. Second, rules about privacy and data safety might make it hard to collect all of a patient's data in one place. This could mean that models have to be taught in decentralised or shared settings, which is something this study hasn't looked into yet. Lastly, it's not easy to make the system work with a lot of different gadget makers, sensor types, and hospital networks. Before it can be used in clinical settings, the anomaly recognition system needs to be fine-tuned and tested on a wider range of hardware devices and patient groups.

CONCLUSION

This study came up with and tested a strong machine learning-based strategy for finding strange things in IoT-connected medical devices. The goal was to improve healthcare reliability and real-time disease trackingGiven the rapid expansion of IoMT in clinical environments, it is crucial to guarantee accurate data and prompt issue resolution. Given the rapid expansion of IoMT in clinical environments, it is crucial to guarantee accurate data and prompt issue resolution. By means of Isolation Forest, Autoencoder, and LSTM models and comparison, the research indicated that machine learning algorithms outperform rule-based and statistical techniques in detecting unusual behaviour in sensor data streams. Of these, LSTM models had the greatest accuracy and memory, indicating how well they function for detecting issues in time series. Designed to be flexible and scalable, the approach works with both edge-based and cloud-based systems. Because it employs both untrained and semi-supervised models, the system may operate with less labelled data. For real-world healthcare scenarios, this is a key characteristic. The findings indicated that including pertinent EHR data into the procedure of locating anomalies reduces false alerts and increases the clinical utility. Visualisations such as ECG anomaly detection plots and ROC curves further simplified model performance. Among the issues mentioned were the lack of consistent, annotated clinical anomaly datasets and the need for deep models to be more understandable. Despite these challenges, the findings unambiguously indicate that intelligent healthcare environments may be made safer for patients and more efficient at monitoring by using AI-enhanced anomaly detection systems. By demonstrating a method that is scalable, accurate, and simple to grasp, this work improves digital health monitoring by finding abnormalities. It can be used in real-world IoT medical systems. In the future, researchers will look into how to make the system even more reliable and scalable by integrating it with shared learning frameworks, putting it to use on integrated medical edge devices, and doing real-time clinical evaluation.

REFERENCES

[1] S. S. Patil and E. Rosemaro, “Cybersecurity Threat Detection and Prevention using Machine Learning Approaches”, IJRAET, vol. 12, no. 1, pp. 9–15, Jun. 2023.

[2] Yousuf, T.; Mahmoud, R.; Aloul, F.; Zualkernan, I. Internet of Things (IoT) Security: Current Status, Challenges and Countermeasures. *Int. J. Inf. Secur. Res.* 2015, 5, 608–616.

[3] Khanna, D. Internet of Things Challenges and Opportunities. Int. J. Technol. Res. Eng. 2020, 6, 6028–6030.

[4] S. M. karve , P. P. Karande , R. N. Nalawade , and S. R. Garad , “A Comprehensive Review on IoT-Driven Polyhouse Farming: Innovations, Challenges, and Future Directions”, IJACECT, vol. 13, no. 2, pp. 43–51, Mar. 2025.

[5] Magara, T.; Zhou, Y. Internet of Things (IoT) of Smart Homes: Privacy and Security. J. Electr. Comput. Eng. 2024, 2024, 7716956.

[6] Kale, Rohini Suhas , Hase, Jayashri , Deshmukh, Shyam , Ajani, Samir N. , Agrawal, Pratik K & Khandelwal, Chhaya Sunil (2024) Ensuring data confidentiality and integrity in edge computing environments : A security and privacy perspective, Journal of Discrete Mathematical Sciences and Cryptography, 27:2-A, 421–430, DOI: 10.47974/JDMSC-1898

[7] Gummadi, A.; Napier, J.; Abdallah, M. XAI-IoT: An Explainable AI Framework for Enhancing Anomaly Detection in IoT Systems. *IEEE Access* 2024, 12, 71024–71054.

[8] Senthilraja, P.; Palaniappan, K.; Duraipandi, B.; Balasubramanian, U. Dynamic Behavioral Profiling for Anomaly Detection in Software-Defined IoT Networks: A Machine Learning Approach. *Peer Peer Netw. Appl.* 2024, 17, 71024–71054.

[9] Mikołajewski, D.; Czerniak, J.; Piechowiak, M.; Wegrzyn-Wolska, K.; Kacprzyk, J. The Internet of Things and AI-based optimization within the Industry 4.0 paradigm. *Bull. Pol. Acad. Sci. Tech.* 2024, 72, e147346.

[10] Czeczot, G.; Rojek, I.; Mikołajewski, D. Autonomous Threat Response at the Edge Processing Level in the Industrial Internet of Things. *Electronics* 2024, 13, 1161.

[11] Imran, I.; Ali, S.M.; Faiz, R.; Alam, M.; Imran Ali, S.; Bari, M.; Shibli, M. A Survey of Machine Learning Techniques for Detecting Anomaly in Internet of Things (IoT). *J. Indep. Stud. Res. Comput.* 2023, 21, 1–5.

[12] Dwivedi, D.; Bhushan, A.; Singh, A.; Singh, S. Detection of Malicious Network Traffic Attacks Using Support Vector Machine. In *Advances in Security, Privacy, and Trust in Computing Systems*; Springer Nature Switzerland: Cham, Switzerland, 2024; pp. 55–66.

[13] Boulesteix, A.L.; Janitza, S.; Kruppa, J.; König, I.R. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 2012, 2, 493–507.

[14] De Medeiros, K.; Hendawi, A.; Alvarez, M. A Survey of AI-Based Anomaly Detection in IoT and Sensor Networks. *Sensors* 2023, 23, 1352.

[15] Chatterjee, A.; Ahmed, B.S. IoT anomaly detection methods and applications: A survey. *Internet Things* 2022, 19, 100568.

[16] Yang, M.; Zhang, J. Data Anomaly Detection in the Internet of Things: A Review of Current Trends and Research Challenges. *Int. J. Adv. Comput. Sci. Appl.* 2023, 14, 1–10.

[17] Al Samara, M.; Bennis, I.; Abouaissa, A.; Lorenz, P. A Survey of Outlier Detection Techniques in IoT: Review and Classification. *J. Sens. Actuator Netw.* 2022, 11, 4.

[18] Eltanbouly, S.; Bashendy, M.; Al Naimi, N.; Chkirbene, Z.; Erbad, A. Machine Learning Techniques for Network Anomaly Detection: A Survey. In *Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, Doha, Qatar, 2–5 February 2020; pp. 156–162.

[19] Tyagi, A.; Singh, A.; Yadav, A.; Mehra, P.S. A Survey on Artificial Intelligence-Based Cyber Security in IoT Networks. In *Proceedings of the 2024 2nd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT)*, Dehradun, India, 15–16 March 2024; pp. 238–243.