Explainable AI for Event and Anomaly Detection and Classification in Healthcare Monitoring Systems

Menatalla Abououf, Shakti Singh, *Member, IEEE,*, Rabeb Mizouni, Hadi Otrok, *Senior Member, IEEE*

***Abstract*—Artificial intelligence (AI) has the potential to revolutionize healthcare by automating the detection and classification of events and anomalies. In the scope of this work, events and anomalies are abnormalities in the patient's data, where the former are due to a medical condition, such as a seizure or a fall, and the latter are erroneous data due to faults or malicious attacks. AI-based event and anomaly detection (EAD) and their classification can improve patient outcomes by identifying problems earlier, enabling more timely interventions while minimizing false alarms caused by anomalies. Moreover, the advancement of Medical Internet of Things (MIoTs), or wearable devices, and their high processing capabilities facilitated the gathering, AI-based processing, and transmission of data, which enabled remote patient monitoring, and personalized and predictive healthcare. However, it is fundamental in healthcare to ensure the explainability of AI systems, meaning that they can provide understandable and transparent reasoning for their decisions. This paper proposes an online EAD approach using a lightweight autoencoder (AE) on the MIoT. The detected abnormality is explained using KernelSHAP, an explainable AI (XAI) technique, where the explanation of the abnormality is used, by an artificial neural network (ANN), to classify it into an event or anomaly. Intensive simulations are conducted using Medical Information Mart for Intensive Care (MIMIC) dataset for various physiological data. Results showed the robustness of the proposed approach in the detection and classification of events, regardless of the percentage of the present anomalies.**

***Index Terms*—Explainable AI, Deep learning, Event detection, Anomaly detection, Classification, Medical Internet of Things (MIoT), SHAP.**

I. Introduction

The advancement of computing technologies and their processing capabilities have enabled the automation of various systems, such as environmental and healthcare monitoring systems [1], [2], [3]. This includes autonomous data collection, processing and decision-making. Moreover, wireless and wearable devices have been gaining a lot of attention in the medical field since they facilitate real-time, personalized and remote monitoring. A wearable device, or a medical internet of things (MIoT) in the scope of this paper, is an ecosystem of hardware and software that can capture, store, process and distribute data wirelessly to an end-user serving a specific task in the healthcare industry [4]. This is particularly useful for patient monitoring applications since it reduces the costs of physical hospital visits while providing timely protection. An MIoT utilizes various sensors to monitor a patient's physiological data or vitals, which include heart rate, blood pressure, oxygen level, etc.. It also performs some initial processing and communicates with a local processing unit (LPU) for further analysis. An LPU is a gateway, possibly the patient's smartphone, that has higher processing capabilities than an MIoT and communicates wirelessly with the healthcare provider of the patient in case a medical abnormality is detected.

The accuracy of sensor readings in patient monitoring applications is of utmost importance since inaccurate or compromised physiological readings can raise false alarms, or no alarms in critical situations. This may affect the patient's well-being as the medical professional will make decisions based on the received data and alarms. The sensors may be inaccurate for various reasons, including hardware tampering, fault, or malicious attacks. In addition, erroneous readings could also occur due to hardware limitations since MIoTs are resource-constrained and are susceptible to electromagnetic interference and calibration errors [5], [6]. On the other hand, the collected data may not be as expected or appear abnormal due to a medical condition experienced by the patient. Therefore, erroneous sensor readings, due to faults, attack, etc., are referred to as *anomalies*, while non-erroneous, but abnormal readings due to medical condition are referred to as *events*.

Medical events are predominantly abnormalities in the vitals that last a long time [7]. In addition, a correlation of abnormality is observable between the different vitals, i.e. events affect the readings of at least two. On the other hand, anomalies are usually in the form of spikes and do not show any spatial or temporal correlation across the different vitals. Using the nature of anomalies and events, many works in literature have proposed event and anomaly detection (EAD) approaches, where the abnormality detected is classified to minimize false alarms. These works either use statistical or artificial intelligence (AI) approaches to tackle the problem [5], [8], [9], [10]. Some of these works utilize sensors solely for data collection and frequently send the patient's data to the LPU for abnormality detection and classification [11], [12], [13], [14], [15]. However, this entails significant and costly communication overhead [16]. Furthermore, few existing works utilize the processing capabilities of MIoTs for local processing before raising a flag to the LPU, where the LPU performs further analysis and only alerts the medical professional in case of an event[17], [18]. However, only some of these approaches interpret the alerts raised or explain the underlying factor that led to them.

Explaining and understanding the reasoning behind a detection model's output is critical to improving the healthcare system. However, medical professionals are showing

M. Abououf, S. Singh, R. Mizouni, and H. Otrok are with the Center for Cyber Physical Systems, Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, UAE, e-mail: (menatalla.abououf@ku.ac.ae).

low acceptance of decisions made by models that cannot be explained [19]. In particular, AI-based models, while they improve prediction and classification in many applications, are considered black boxes, where the decisions made are not transparent. Hence, many tools are emerging to explain such complex models, known as explainable AI (XAI) [20]. Recently, few works have proposed XAI, mainly using Shapley Additive exPlanation (SHAP), for healthcare applications that use tabular data [21], [22], [23], [24]. However, these works are purely diagnostics, i.e. concerned with event detection and did not consider anomalies. In addition, the detection is performed on a central level after collecting the data from the patient, which is not applicable in continuous monitoring applications. Finally, one patient's data is used in a model created by data collected from various patients, and hence the model is not personalized.

To overcome the above-mentioned challenges, this work proposes a two-layer patient monitoring system. In the first layer, at the MIoT, a lightweight and unsupervised autoencoder (AE) is employed that is trained on the patient's data, to detect data abnormality in real time. The model needs to be lightweight to reduce the memory footprint and computational complexity when employed on MIoT. If an abnormality, event or anomaly, is detected, a flag, along with the patient's data, is sent to the LPU. In the second layer, at the LPU, the decision of the EAD model is explained using KernelSHAP, an estimation of SHAP. It uses a surrogate model, based on linear weighted regression, and a background dataset to explore a sample of possible coalitions to achieve an almost accurate estimate of the contributions with lower complexity. Finally, an artificial neural network (ANN) is used to classify the abnormality into events or anomalies using the SHAP values of the vitals. It alerts the medical professional in case of an event. To the best of our knowledge, this is the first work that uses XAI to explain the decision and classify anomalies and events. Hence, the contribution of this work can be summarized as follows:

- Detects medical events timely and accurately, while considering the presence of anomalies, using unsupervised AE
- Explains the underlying factor of the decision made by the model, based on the contributions of the different vitals towards the decision, using KernelSHAP
- Minimizes false alarms by accurately classifying events and anomalies based on the contribution of vitals to the abnormality using ANN

The efficacy of the proposed solution is tested on Medical Information Mart for Intensive Care (MIMIC) dataset[1]. A patient's dataset consists of ~ 25000 records, each containing readings of 6 different vital signs, namely, respiration rate, heart rate, systolic and diastolic blood pressure, pulse pressure, and oxygen saturation. The dataset is also labelled for medical events, whereas anomalies are injected for verification purposes. The proposed approach is extensively simulated using Keras [25], [26], and its performance is compared with similar works [9], [11], [13], [15].

II. Related Work

Event (and anomaly) detection has been widely tackled in the literature in the healthcare field. This could solely be for the medical diagnosis, or to also detect erroneous readings due to faulty sensors or malicious attacks. Various methods are proposed, which can be divided into statistical-based or AI-based techniques in centralized or decentralized manner. The widespread of AI-based EAD raised the question of extracting the underlying reasoning of the prediction or decision [27]. This has led to recent publications introducing explainable AI to explain the predictions. In this Section, some of the related work of event (and anomaly) detection in the medical sector, with and without XAI, is summarized and discussed.

A. Event detection with XAI

This section summarizes some of the related work that uses XAI to explain the decision of the proposed model in healthcare applications using clinical tabular data.

In [21], gradient boosted machines (GBMS) are used to predict first-time acute exacerbation of chronic obstructive pulmonary disease (AECOPD) in patients using features, such as vitals signs, symptoms, medications, etc. Then, SHAP is used to explain the model's decisions. In [22], random forest (RF) is used to diagnose chronic kidney disease using optimized and low-cost acquired clinical parameters, such as urine and blood tests. The diagnosis is then interpreted and justified using SHAP. The work in [23] proposes a feature selection methodology and utilizes RF to diagnose patients with knee osteoarthritis diseases, then uses SHAP to understand the impact of the features on the model's decision. Finally, the work in [24] proposes a novel fuzzy classification rules learning through clustering (FCRLC) for heart disease diagnosis using clinical data. Subsequently, the authors propose fuzzy classifiers to provide an interpretable knowledge base to clinician to explain the decision.

While the works mentioned above use XAI to explain the decision of the model and define the contributing features, they have the following shortcomings - 1) they are solely diagnostic. Hence, they only detect medical events and do not consider anomalies, 2) they are centralized; hence, a single source of decision making, 3) they are not continuously monitoring the patient, and 4) the models are generic to all patients and not personalized; hence, not accounting for natural differences between people's attributes.

B. Event and Anomaly detection (EAD) without XAI

This Section discusses some related work on EAD in healthcare applications using AI or statistical-based approaches.

The authors in [5] proposed an EAD mechanism that classifies the record into normal or abnormal using J48, random forests, and k-nearest neighbors. Once an abnormality is detected, regression algorithms are used to classify an anomaly from an event. Similarly, the work in [8] follows a similar pipeline, where support vector machine is proposed for classification. These works mainly depend on machine learning approaches to detect and classify anomalies and events.

[1] https://ieee-dataport.org/documents/mimic-dataset-anomaly-detection

TABLE I: Summary of the presented related-work.

Reference	Method	XAI	Decentralized	Detection target	Classification
[21], [22], [23]	ML	SHAP	×	Event	×
[24]	ML	Fuzzy classifier	×	Event	×
[5], [8]	ML	×	×	Event & anomaly	✓
[9], [10], [11], [15], [13], [14], [12]	Statistical	×	×	Event & anomaly	✓
[18]	Statistical	×	✓	Event & anomaly	✓
[17]	Statistical + game	×	✓	Event & anomaly	✓
Proposed Approach	**DL**	**SHAP**	✓	**Event & anomaly**	✓

The authors in [9] propose an approach that first detects abnormal records by correlating sensor readings using Mahalanobis distance (MD). Subsequently, kernel density estimator (KDE) is proposed to identify the abnormal sensor(s). In [10], an event and anomaly detection and classification approach is proposed, where - 1) principle component analysis (PCA) and feature extraction are used for dimensionality reduction and to identify attributes that are highly correlated, respectively, 2) multivariate anomaly detection is performed to identify spatio-temporal abnormalities, and 3) univariate detection is performed to pinpoint those abnormal attributes. The work in [11] proposed a similar approach, but majority voting is used to classify events and anomalies instead of feature extraction and correlation. Finally, tn [15], robust incremental PCA is proposed for dimensionality reduction, then events and anomalies are detected by comparing the squared prediction error with a threshold. The works mentioned above first identify an abnormal record, then pinpoint the attributes responsible for the flag.

A different approach is also adopted where the sensor readings are evaluated individually and then collectively. In [13], sequential minimal optimization regression (SMO regression) is proposed to predict the readings of the individual sensors. If the error between the predicted and original reading of multiple sensors is too high, an alert is raised. A similar approach is proposed in [14], where dynamic sliding window is used instead of SMO regression. The work in [12] proposes an approach where the sensor readings are first filtered to remove baseline wandering noise and normalized. If the peaks of the readings exceed a certain threshold, the different sensors are correlated for frequency and amplitude similarities. An alert is raised if the similarity exceeds the predefined threshold. All the previous works detect and classify events and anomalies with high performance. However, the processing takes place on the LPU, whihch requires frequent communication between the sensors and the LPU.

Few works in the literature implement a decentralized approach, where the detection happens on the sensor side, and the communication with the LPU occurs only when necessary. For instance, [18] proposes an approach where the sensor predicts the reading using exponentially weighted moving average (EWMA) and compares it with the original reading. If the difference is outside an ellipsoidal region of normal data, the data is sent to the LPU. Subsequently, the LPU exploits the correlation of data collected from the different sensors to classify the abnormality. The authors in [17] proposes a game-theoretical-based approach where the sensor decides to whether - 1) use its data to detect anomalies, 2) cooperate with neighboring data to detect anomalies, or 3) send its data to the LPU for processing. The decision is based on a trade-off between accuracy and network performance. While these works, being decentralized, are more efficient in terms of communication cost, they do not interpret the decision made by their model or explain the underlying factor that led to it.

In summary, event (and anomaly) detection is a concern in the medical field. Some of the works are solely event detection or diagnostic and do not consider the presence of anomalies, while others classify the data abnormality into events or anomalies without interpreting the decision. Table I presents a summary of the related work discussed in this section.

III. Overview of the Proposed Approach

This work proposes a medical healthcare monitoring system that: 1) detects abnormal vital readings, by an anomaly and event detection model, 2) explains the abnormal readings, and 3) differentiates between anomalies and medical events using a classification model. The proposed approach consists of four main parties, namely, cloud/server, MIoT, LPU and the healthcare provider, as shown in Fig. 1. The interactions among these parties and their roles are described next.

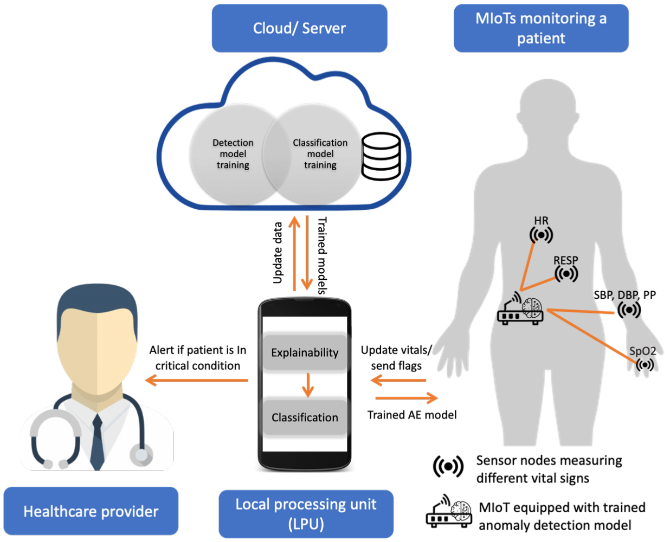

Cloud/ Server
MIoTs monitoring a patient
Detection model training
Classification model training
Update data
Trained models
HR
RESP
SBP, DBP, PP
SpO2
Alert if patient is In critical condition
Explainability
Classification
Update vitals/ send flags
Trained AE model
Sensor nodes measuring different vital signs
MIoT equipped with trained anomaly detection model
Healthcare provider
Local processing unit (LPU)

Fig. 1: Overview of the proposed approach

- **Cloud/Server**: The primary role of the cloud is to train the anomaly and event detection and classification models for each patient using their data. Customized trained models are generated to account for the natural differences in vital signs among individuals, based on gender, age, etc. [28]. The trained models are sent to the LPU of the concerned patient. In addition, the cloud stores the patient's medical history, which gets updated every period, T, by the LPU. T should be carefully chosen such that it is large enough to avoid unnecessary expensive communication for trivial data, and small enough to capture significant updates in the data.

- **LPU**: The LPU is an edge node with higher processing capabilities, possibly a smartphone. It further processes the flags received by the MIoT by explaining them using KernelSHAP and then classifies the flag into an anomaly or an event. If the flag is an event, the LPU alerts the **healthcare provider** by sending the patient's abnormal data, in addition to explaining the decision by specifying the highly contributing vitals.
- **MIoT**: The MIoT is an internet-enabled processing unit, possibly a wearable device, that is connected to or embedded with the sensors monitoring the patient's vitals. The MIoT is equipped with the EAD model received by the LPU. The main role of the MIoT is to continuously monitor the patient and send flags to the LPU in case data abnormality is detected.

Hence, the proposed monitoring system relies on three main modules, as shown in Fig. 2:

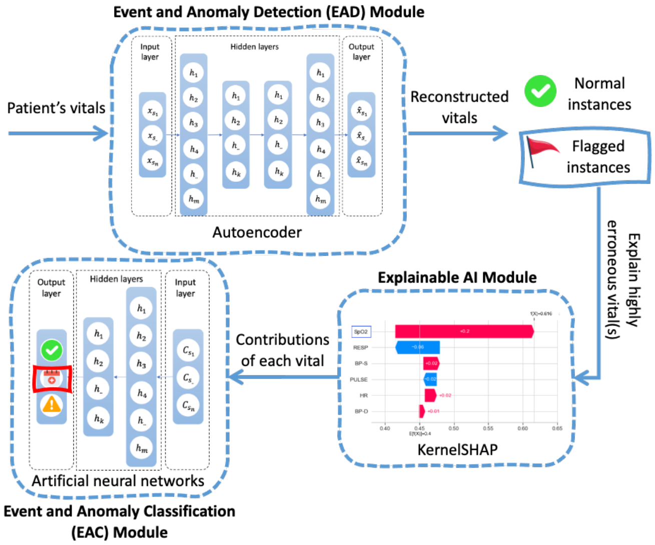


Fig. 2: Overview of the proposed monitoring system modules

- **Event and Anomaly Detection (EAD) Module**
  Referring to Figure 1, this module is trained in the cloud, and deployed on the MIoT monitoring the patient. It is responsible for generating flags in the case of data abnormality in the patient's vitals. It is based on AE, a self-supervised deep learning method based on an ANN that *encodes* the input to a latent space representation, known as a *bottleneck*, then *decodes* it back, aiming to reconstruct the original input. *The reconstruction loss* (R_loss) is the metric that measures how accurately the AE can reconstruct the original input. When data is normal, no anomalies or events are present, and R_loss should be ~ 0. However, in case of data abnormality, R_loss should be high enough to raise the flag.
- **Explainable AI (XAI) Module**
  This module is deployed on the LPU. It is responsible for explaining the flags raised by the AE model. The complexity of deep learning models makes it difficult to interpret and explain their predictions. Explainable AI enables understanding the underlying factors that led to the decision taken. This paper uses the XAI method based on SHAP (Shapley Additive exPlanation). SHAP is a game-theoretical approach that aims to calculate the local contribution of each feature towards the decision made for a particular instance, or a flag in the context of this work. This helps in identifying the influence of each vital on the final decision of the model. The Shapley value of a feature i, ϕ_i, is calculated as the weighted average of the contribution differences with all possible coalitions not containing i as follows:

  $$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M-|z'|-1)!}{M!}[f_x(z') - f_x(z' \setminus i)] \tag{1}$$

  where f is the model, x' is the simplified input, $x = h_x(x')$, where h_x is a function mapping 1 (included) or 0 (not included) to the original input. Moreover, M is the set of features, and z' is a form of coalition, where $z' \subseteq M = \{1, ..., M\}$.
  SHAP is typically used with tabular data [20]. It satisfies the following properties - 1) *local accuracy*, where the original model should match the explanation model, 2) *missingness*, where missing features in the original input must not have an impact, and 3) *consistency*, where if the model is revised such that a feature has higher impact, then that feature should have a higher SHAP, irrespective of other features [29], [30]. However, it can be deduced from (1) that SHAP is computationally expensive [20], especially with an increasing number of features. Hence, this paper uses an estimation of SHAP, called KernelSHAP [30]. KernelSHAP uses Local Interpretable Model Agnostic Explanations (LIME) and Shapley values to build linear weighted regression as an explanation model. This surrogate model uses a background dataset and explores a sample of possible coalitions to achieve an almost accurate estimate of the contributions with lower complexity [29]. Assuming n features and m instances to be explained, a rough estimate of the time complexity of KernelSHAP can be expressed as $\mathcal{O}(m * 2^n)$, where the exponential term depends on the number of possible feature subsets. The background dataset is commonly a subset of the training data to avoid the inclusion of unrealistic data instances.
  As mentioned earlier, XAI is used to explain the decision made by the model. However, the output of AE is the reconstruction of the original input, hence there is more than one output, as many outputs as the input. To overcome this issue, the work in [29] proposed a method to explain anomalies using AE. This is based

on explaining the features with the highest reconstruction error, which in turns results in explaining only a subset of the output, rather than the entire output. Using XAI with AE will be futher detailed in Section IV-A3.

- **Event and Anomaly Classification (EAC) Module** This module is deployed on the LPU. The output of the XAI module is used as an input for this module, where the flag is classified into either an anomaly or a medical event. This is because the XAI not only gives an explanation for the highly contributing vitals, but it also demonstrates a distinction between both kinds of data abnormality. This work uses ANN to classify the abnormal instances that raised flags. ANNs consist of interconnected layers - an input layer, one or more hidden layers, and an output layer. Each layer is comprised of nodes, known as neurons, which connect to each other with associated weights and biases. A node in the network applies an activation function f to the weighted sum of its inputs. The activation function introduces non-linearity into the network and allows it to model complex relationships. An activation function applied at node j in layer i can be denoted as:

$$a_ij = f(z_ij) \tag{2}$$

where z_ij represents the weighted sum of inputs to node j in layer i before applying the activation function. z_ij at each node is calculated as the dot product of the input vector and the corresponding weight vector, along with a bias term. Hence, z_ij at node j in layer i can be represented as:

$$z_ij = \sum(Wjk * a_ik) + bj \tag{3}$$

where Wjk represents the weight connecting node k in the previous layer to node j in the current layer, a_ik represents the activation of node k, and bj represents the bias term of node j.
If the node is activated, and data is propagated to the next layer. Based on the output of ANN, if the flag is classified as a medical event, the LPU alerts the healthcare provider of the patient. The main purpose of this module is to avoid raising unnecessary flags generated by anomalies.

IV. Methodology and Architecture

This section details the proposed health monitoring system of EAD and EAC with explainability.

Considering an MIoT, M_j, of patient j, and a periodic data update every T interval, the M_j can be described as a tuple, $M_j =< S_j, f_j, X_{S_j}^{(1,2,..t)}, \hat{X}_{S_j}^{(1,2,..t)} >$, where the different attributes are defined in Table II. S_j and f_j are time-independent variables, while $X_{S_j}^{(1,2,..t)}$ and $\hat{X}_{S_j}^{(1,2,..t)}$ are vectors of vectors containing the accumulated data received by sensors S_j ($X_{S_j}^{(1,2,..t)}$) and their predicted data ($\hat{X}_{S_j}^{(1,2,..t)}$), with respect to time $t \in T$. The rest of this section will detail the proposed approach in the following order - 1) models training and testing in the cloud, 2) EAD deployment on the MIoT, and 3) EAC and decision-making by LPU.

TABLE II: List of symbols used for model parameters and their definitions.

Symbols	Definition
M_j	An MIoT of patient j
S_j	Set of sensors deployed on/connected to M_j, $S_j = \{s_{j1}, s_{j2}, ..., s_{nj}\}$, where $n_j \geq 1$
f_j	The trained EAD model on M_j based on previous behavior, $f_j : \vec{x_j} \rightarrow \vec{\hat{x}_j}$
X_{S_j}	The vector of vectors of actual readings from S_j, since the last periodic update, $X_{S_j} = \{x_{S_j}^{(1)}, x_{S_j}^{(2)}, ..., x_{S_j}^{(t)}\}$ where $t \leq T$
$\hat{X}_{S_j}$	The vector of vectors of reconstructed readings obtained from f_j for all S_j since the last periodic update, $\hat{X}_{S_j} = \{\hat{x}_{S_j}^{(1)}, \hat{x}_{S_j}^{(2)}, ..., \hat{x}_{S_j}^{(t)}\}$ where $t \leq T$
n_i	Number of sensors monitoring the patient

A. Models Training and Testing on Server/Cloud

In this section, the EAD and EAC models training in the cloud and their architecture are detailed.

1) Data Description and Preparation: The historical data received by sensors monitoring a patient's vitals are merged by the timestamp into a single dataset. This work uses the Medical Information Mart for Intensive Care (MIMIC) dataset to evaluate the proposed approach. The dataset consists of ~ 25000 records for each patient. Each record contains readings for HR, PP, BP-S, BP-D, RESP and SpO2. In addition, the dataset is labelled for medical events. Table III shows the description of the used dataset. It is worth noting that the the proposed approach would work with any medical dataset and different combinations of vitals.

TABLE III: Description of the MIMIC dataset.

Feature	Type	Description
HR	float	Heart rate
PP	float	Pulse pressure
BP-S	float	Systolic blood pressure
BP-D	float	Diastolic blood pressure
RESP	float	Respiration rate
SpO2	float	Oxygen saturation
Event	binary	Normal vitals (0) or medical event (1)

The data is normalized using MinMax normalizer, as shown in Equation (4), where $\overline{x}_s$ is the normalized vital data and x_s is the original data. This is crucial to give all vitals equal importance.

$$\overline{x}_s = \frac{x_v - min(x_s)}{max(x_s) - min(x_s)} \tag{4}$$

An event is represented by a witnessed data abnormality in 2 or more vitals that takes place for a prolonged period of time. Detected events are spread out in the dataset. In order to test and evaluate the model for its ability to detect real events, these events are moved towards the end of the dataset, which will be the testing dataset. The events are moved without shuffling in order to keep the temporal sequence of the events. This did not have a negative impact on the data, as will later be evident in Section V. Finally, to keep the integrity of the data intact, the dataset is split, without shuffling, into 80% training ($X_{S_{j(train)}}$) and 20% testing ($X_{S_{j(test)}}$), ensuring that the training dataset is clean, i.e. no presence of events or anomalies.

2) Event and Anomaly Detection (EAD) Model Training and Testing: The prepared data is then used to train and evaluate the EAD model as shown in Fig. 3. As mentioned in Section III, this model is based on AE. The AE is trained using clean data ($X_{S_j(train)}$), i.e. normal vitals without events nor anomalies, to learn the normal behavior of a patient's vitals. The proposed architecture is based on a non-symmetrical deep autoencoder (NDAE) [31], consisting of an input layer, four hidden layers and an output layer, as shown in Fig. 4. An NDAE refers to an autoencoder architecture where the number of neurons or layers in the encoder and decoder parts are not equal. This is motivated by reducing the number of layers while achieving comparable detection, hence reducing complexity. The proposed NDAE can be formulated as follows:

- Encoder: Maps the input data into a lower-dimensional latent space representation. Consider an input vector X_{S_j}, and n hidden layers represented by h_n as:

$$h_1 = f_1(W_1 X_{S_j} + b_1),\ h_{2..n} = f_{2..n}(W_{2..n} h_{1..n-1} + b_{2..n}) \tag{5}$$

  where W represents the weight matrix and b represents the bias vector.
- Decoder: Attempts to reconstruct the original input from the latent space representation. Considering the reconstructed output as $\hat{X}_{S_j}$:

$$\hat{X}_{S_j} = h_n(W' h_n + b') \tag{6}$$

  where W' represents the weight matrix connecting the last hidden layer to the output layer and b' represents the bias vector of the output layer.

Subsequently, the testing data $X_{S_j(test)}$ is used to evaluate and test the trained AE. Since the MIMIC dataset is labelled for events, and not anomalies, anomalies are injected systematically in $X_{S_j(test)}$ to simulate an erroneous system. Anomalies are injected to a vital reading at time t for sensor s measuring vital of patient j, following a normal distribution as given in (7) [32], [33].

$$x_{s_j}^{(t)'} = x_{s_j}^{(t)} + zN(\mu, \sigma)\delta(t) \tag{7}$$

where $\delta(t)$ is the Dirac delta function, $N(\mu, \sigma)$ is the normal distribution function, with an average μ and standard deviation σ.

The model is then evaluated by comparing the reconstruction of the vital data with the actual data, i.e. computing R_loss. The Mean Square Error (MSE) and the Mean Absolute Error (MAE) are commonly used to compute the total R_loss. This paper uses MSE as it is more sensitive to outliers when compared with MAE. MSE is calculated as given in Equation (8) for vital signs measured at time t for patient j.

$$MSE = \frac{1}{S}\sum_{s=1}^{S}(x_{s_j}^{(t)} - \hat{x}_{s_j}^{(t)})^2 \tag{8}$$

where S is the number of vital sensors. Finally, if MSE is greater than a preset threshold TH, $MSE > TH$, the record is assumed to be abnormal, and a flag is raised.

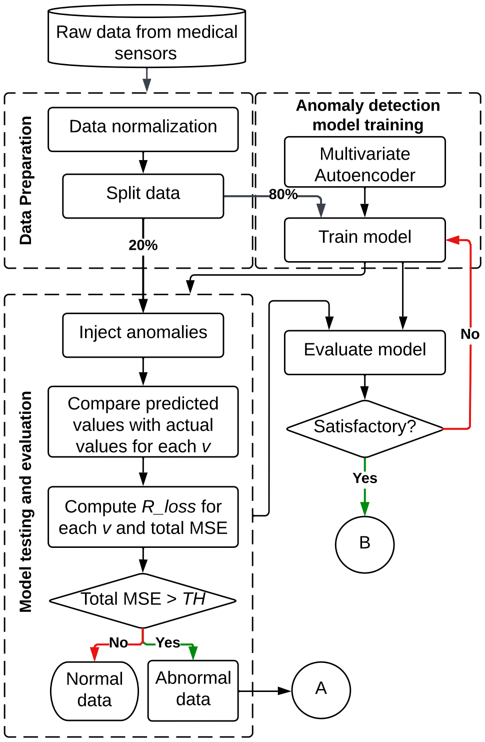

Raw data from medical sensors
Data Preparation
Data normalization
Split data
80%
20%
Anomaly detection model training
Multivariate Autoencoder
Train model
Model testing and evaluation
Inject anomalies
Compare predicted values with actual values for each v
Compute R_loss for each v and total MSE
Total MSE > TH
No
Yes
Normal data
Abnormal data
A
Evaluate model
Satisfactory?
No
Yes
B

Fig. 3: EAD model training and testing flowchart. v is a vital, R_loss and TH are the reconstruction loss and the threshold, respectively.

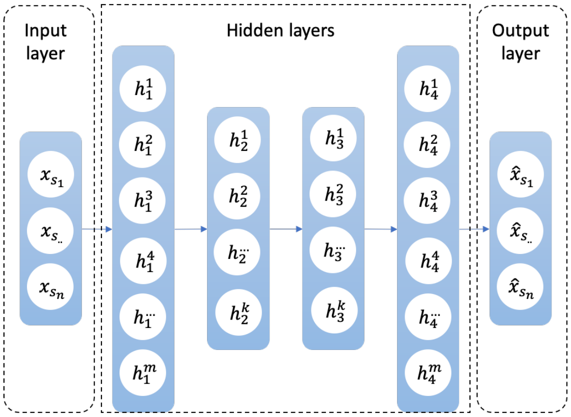


Fig. 4: Architecture of the proposed autoencoder.

Training and testing process is repeated, while tuning hyperparameters, until the model performance is satisfactory. This signifies that the model can detect all events with minimal false positives (false alarms). Next, the raised flags are used, along with explainable AI, to train an EAC model.

3) Event and Anomaly Classification (EAC) Model Training and Testing: The instances that raised a flag from $X_{S_j(test)}$, are first explained using KernelSHAP, where the explanation provides the contribution of each vital towards the flag. These contributions are then used to train the EAC model, as shown in Fig. 5. The proposed steps are as follows:

1) The reconstruction loss (R_loss), i.e. the squared difference between the reconstructed and original readings of each vital is checked. The vitals whose R_loss exceeds the MSE of that record (ES_j) are explained, where

$ES_j \subseteq S_j$. For instance, if the loss in reconstructing HR exceeds the MSE, HR is explained.

2) The background dataset, the EAD trained model, and the record that raised the flag are fed to the KernelSHAP while specifying ES_j.
3) The process is repeated for all the records that raised the flags, and the resulting contributions of the vitals for each record are saved in a dataset.
4) Categorical labels are appended to the dataset, where 2 represents an anomaly, 1 represents an event, and 0 represents a misclassification.
5) For training, the misclassified records are removed from the dataset. Subsequently, the dataset is shuffled and split into 70% training and 30% testing. These percentages are chosen because the EAC model is trained on the explained flags raised at the EAD step. These flags make up only a small percentage of the original dataset, hence, to have enough testing data to evaluate the proposed model, the testing data is chosen to be 30%.
6) The training dataset is used to train an ANN model, which is composed of an input layer, two hidden layers and an output layer. The trained model is then used along with the testing dataset to test the model.
7) The hyperparameters are tuned until a satisfactory performance is achieved. This implies that the model is able to correctly classify events with high true positive rate while minimizing false alarms, i.e. incorrectly classifying anomalies as events.

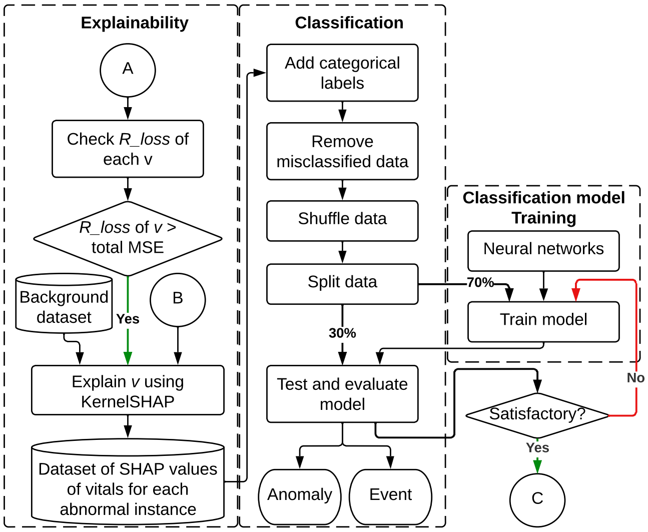


Fig. 5: Proposed approach flowchart.

Finally, the trained event model B and the anomaly detection and classifications model C (in Fig. 3 and Fig. 5) are sent and deployed on the LPU of the patient.

B. On-Device Event and Anomaly Detection (EAD) on MIoT

The EAD model is received from the LPU and deployed on the MIoT monitoring the patient. As shown in Fig. 6, the incoming readings of the different vitals are first normalized using MinMax as given in (4), then fed into the trained EAD model. The reconstructed vitals are compared with the original readings; if the MSE is greater than a preset threshold, a flag is raised and sent to the LPU, along with the patient's data. In case no flags are raised for time T, the MIoT sends the periodic updates to the LPU, which is later sent to the cloud for storage.

C. Decision Making and Classification on LPU

Figure 6 also demonstrates the processing of the flags in the LPU. First, the R_loss of each vital is compared against the MSE of the record that raised a flag. If R_loss of a vital is greater than MSE, the corresponding vital is explained using KernelSHAP. The explanation is then fed to the ANN-based EAC model to classify the abnormality of the record. If the record is classified to be an event, the patient's data is sent to alert their healthcare provider, while specifying the causing vital(s).

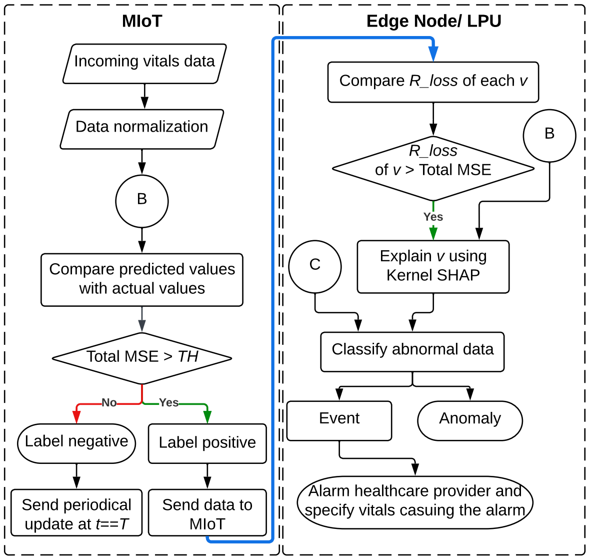

MIoT
Incoming vitals data
Data normalization
B
Compare predicted values with actual values
Total MSE > *TH*
No
Yes
Label negative
Label positive
Send periodical update at *t==T*
Send data to MIoT
Edge Node/ LPU
Compare *R_loss* of each *v*
B
R_loss of *v* > Total MSE
Yes
C
Explain *v* using Kernel SHAP
Classify abnormal data
Event
Anomaly
Alarm healthcare provider and specify vitals casuing the alarm

Fig. 6: Proposed approach flowchart.

V. Evaluation and Results

In this section, the evaluation of the proposed approach is discussed via simulations using Keras [25], [26], running on top of TensorFlow. The simulations are conducted to prove the efficacy of the approach in the detection of events and anomalies, the explanation of raised flags, and their classification into events and anomalies.

A. Experimental Setup and Metrics

As mentioned in Section III, this work uses a MIMIC dataset, labelled for medical events, consisting of ~ 25000 records for each patient. The included vitals in the dataset are HR, PP, BP-S, BP-D, RESP and SpO2. The proposed approach is evaluated on the labelled dataset of two patients, patients 221 and 230, for their availability. Each dataset is split into ~ 20000 for training and ~ 5000 for testing, accounting for 80% and 20%, respectively. The labelled events in each dataset are moved, without shuffling, to the testing dataset to test the approach on actual events. Finally, anomalies are injected in the testing dataset following (7), given that $\mu = \sigma = F/2$, where F is the frequency of injected anomalies. It is worth noting that anomalies are not injected during an event.

The training dataset is used to train the EAD model based on AE. The architecture of AE consists of one input layer, four hidden layers with rectified linear unit (ReLU) activation functions, and one output layer as shown in Fig. 4, where $n = 6$, $m = 16$ and $k = 8$. The total number of trainable parameters for the AE model is 566. The output of the AE is the reconstructed vitals and the MSE of the reconstruction loss. A predefined threshold TH of MSE, which decides whether to raise a flag is carefully chosen for each patient such that it is low enough that all the events are detected, and high enough to minimizing the false alarms. It is worth mentioning that TH would need to be revised in case the number of vitals monitoring the patient decreases or increases. The raised flags are explained using KernelSHAP, and the explanation is used as input for the EAC model based on ANN. The architecture of the ANN consists of an input layer of size $n = 6$, two dense layers of size 16 and 8, with ReLU and softmax activation functions, respectively, and a categorized output layer of size three. The output could be either 0, 1, or 2, corresponding to false alarm, event, or anomaly, respectively. The total number of trainable parameters for the ANN model is 275.

Finally, the models are evaluated using a confusion matrix for true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), where positive indicates an anomaly or an event. In addition, other metrics are derived from the confusion matrix, namely precision and recall, where precision is the ratio of correctly classified positives out of all classified positives. In contrast, recall is the ratio of correctly classified positives out of all existing positives.

B. Validation of the Event and Anomaly Detection (EAD) Model

In this section, the EAD model is compared with baseline unsupervised anomaly detection techniques: isolation forest (IF), long short-term memory autoencoder (LSTM-AE), local outlier factor (LOF) and principal component analysis (PCA). The models are compared in terms of their ability to minimize the false alarms using two datasets, for patients 221 and 230. The datasets are labelled for medical events and 20% of the testing data is injected with anomalies. The dataset of patient 221 contains 4 medical events and patient 230 contains 10 events, of different durations. It is worth mentioning that each of the models is optimized such that at least all the events that are present in a dataset are detected, hence the hyperparameters of each model are tweaked to satisfy this condition. Figure 7 shows the comparison of AE with the baseline models in terms of number of false alarms. It can be seen that AE outperformed the other models in both datasets. The difference is even more prominent with patient 230 because of the complex nature of the events. It is to be noted that PCA was also used as a comparative model, and although for patient 221 the number of false alarms was relatively low (4 false alarms), however, it was unable to detect all events of patient 230, even with lowering the threshold. Since it was not able to detect all the events, it was not included in Fig. 7.

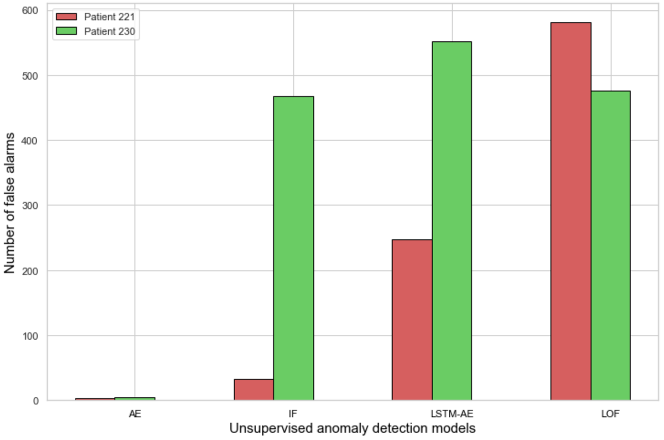


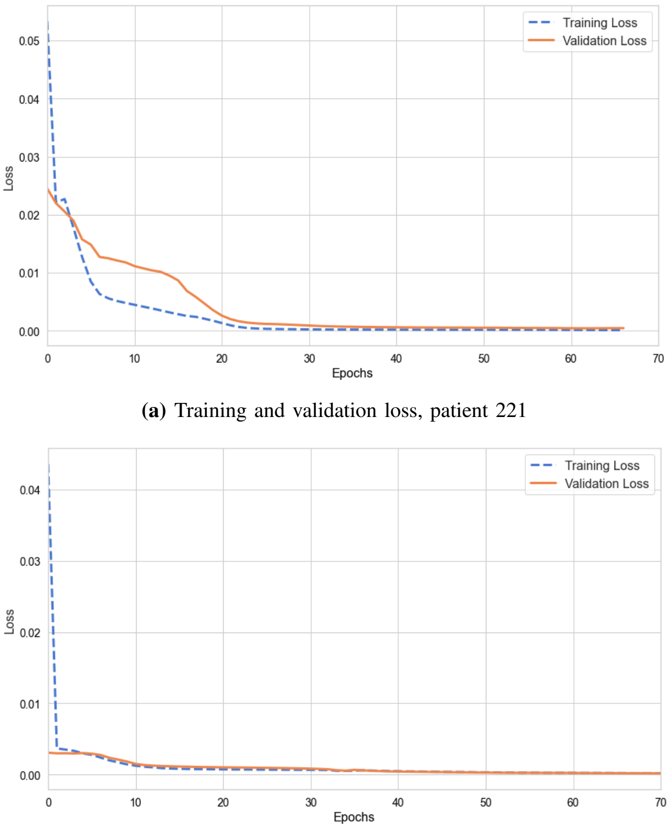
Fig. 7: Evaluation of unsupervised anomaly detection models using datasets for patients 221 and 230

Finally, the AE model is validated using learning curves which show the changes in the loss over time of experience (epochs). Figures 8a and 8b demonstrate the learning curves for AE using datasets of patients 221 and 230, respectively, to validate that the model is not an overfit nor an underfit. The Figures show that the training and validation datasets are suitably representative. Considering a validation split of 25%, the plots demonstrate that both the training and the validation loss decrease to a point of stability, with only a small gap between them, which indicates a good fit.

C. Evaluation of the Event and Anomaly Detection (EAD) Model

In this section, the EAD model is evaluated. Figure 9 demonstrates the testing dataset of the different vitals for both patients, 221 and 230, where the labelled events are shown as prolonged enclosed windows (black). The spikes observed in the vitals represent the injected anomalies that make up 20% of the testing data. It can be seen in Fig. 9a and Fig. 9b that the EAD model raised flags for all the events for both patients, especially at the beginning of the event which is crucial in case of emergency. This ensures notifying the healthcare provider early on, allowing timely intervention, if needed. In addition, the flags raised outside the labelled events are due to the injected anomalies. These flags will later be classified in order to eliminate false alarms and only alert the medical professionals in case of an event. It can also be seen that some of the anomalies did not raise flags, and this because anomalies are injected following normal distributions, and the amplitude of some anomalies are insignificant to be detected, i.e. considered as normal variation of the vitals.

Table IV quantifies the performance of the EAD model with varying percentages of injected anomalies, in terms of successfully detecting the event. If a portion of an event is detected, that means it is a true positive for that event. In the evaluation of the proposed EAD, the models for both patients were able to detect all of the events present, hence



(a) Training and validation loss, patient 221

(b) Training and validation loss, patient 230

Fig. 8: Training and validation learning curves for the AE models using datasets of patients 221 and 230, respectively

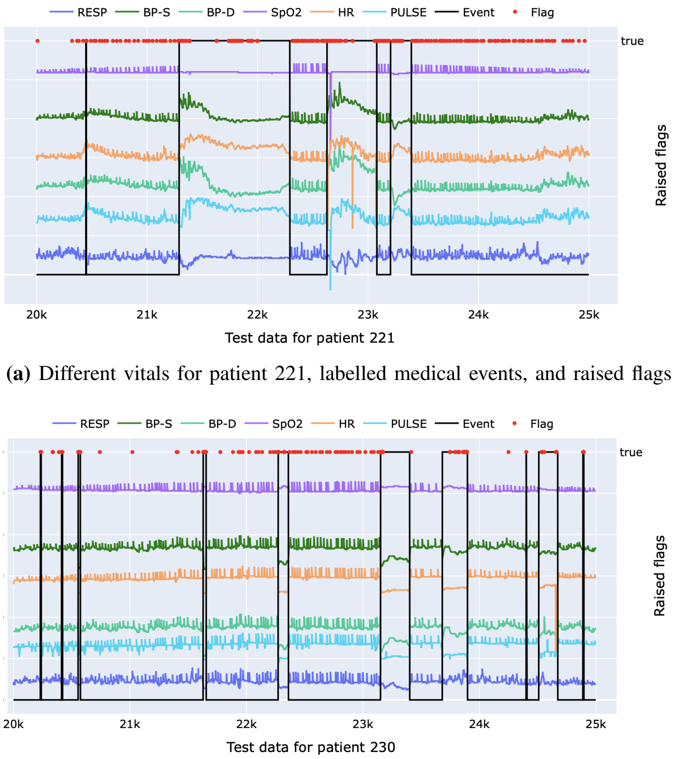
$Recall = \frac{events\_detected}{(events\_detected+event\_not\_detected)} = 1$. It is worth mentioning that a flag raised due to an anomaly is not considered a false positive at this stage, and is simply eliminated from the precision and recall calculations, as there was an actual abnormality in the data. However, it will be counted as false positive during classification, if it is misclassified as an event. Moreover, a flag raised in the absence of both an anomaly and an event is counted as a false positive, hence, precision is computed as $\frac{events\_detected}{(events\_detected+not\_anomaly/event\_detected)}$. It can be concluded that the proposed approach is robust to anomalies, as the percentage of injected anomaly did not affect its performance in detecting all the events, and displayed minimal false positives. This can be seen by the 100% recall for both patients, and the high precision ranging from $0.97 - 0.99$.

TABLE IV: Evaluation of event detection with different percentages of injected anomalies for patient 221/patient 230.

	% of Injected Anomalies			
	5%	10%	20%	30%
Precision	0.98/0.98	0.98/0.98	0.99/0.97	0.99/0.98
Recall	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0

D. Discussion of Explainability

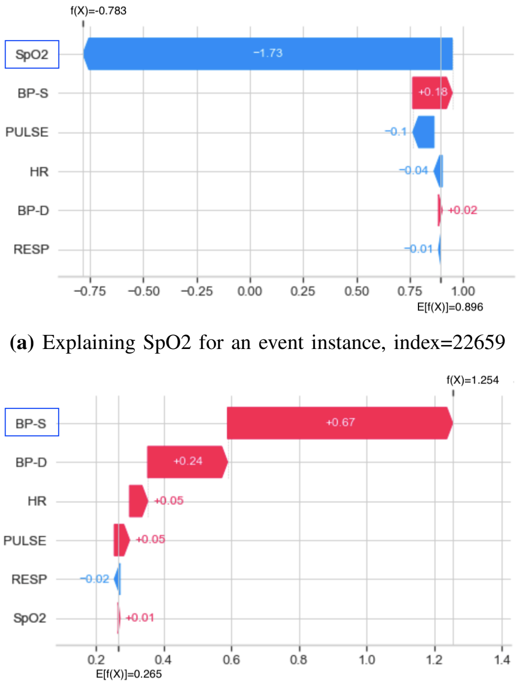
In this section, the explainability of the proposed approach is demonstrated. Figures 10 and 12 show waterfall plots of SHAP values of instances when the flags were raised by events and when they were raised by anomalies, respectively.



(a) Different vitals for patient 221, labelled medical events, and raised flags

(b) Different vitals for patient 230, labelled medical events, and raised flags

Fig. 9: flags raised (red dots) based on the EAD model for patients 221 and 230, respectively. The anomalies are represented as spikes, and make up 20% of the testing data and events are enclosed in the rectangles (black lines)



(a) Explaining SpO2 for an event instance, index=22659

(b) Explaining BP-S for an event instance, index 22746

Fig. 10: Explaining flags raised by events for patient 221

Waterfall plots help to visually demonstrate the SHAP values or the contribution of each feature in moving the predicted value away from the actual value. The sign of a SHAP value indicates whether the feature contributed in pushing the prediction lower or higher than the expected value, where negative pushes it lower and vice versa. However, in this application pushing the prediction higher or lower are of equal importance. For example, Fig. 10a shows the SHAP value of each feature, when explaining the prediction of SpO2, since this was the feature whose reconstruction error exceeded the MSE of that instance (index = 22569). It can be seen that the expected value was 0.896, while the predicted value was -0.783. In addition, the features who relatively had the highest magnitudes of SHAP values are SpO2, BP-S and PULSE. The contribution of these features could also be visually seen in Fig. 11, where in the first dotted rectangle (index = 22569) show significant drops in SpO2 and PULSE, and a peak in BP-S. Similarly, Fig. 10b shows SHAP values of the features when explaining BP-S at a different instance (index = 22746), where BP-S and BP-D relatively have the highest contribution towards the flag. In addition the second dotted rectangle in Fig. 11 show that there was a drop in both BP-S and BP-D.

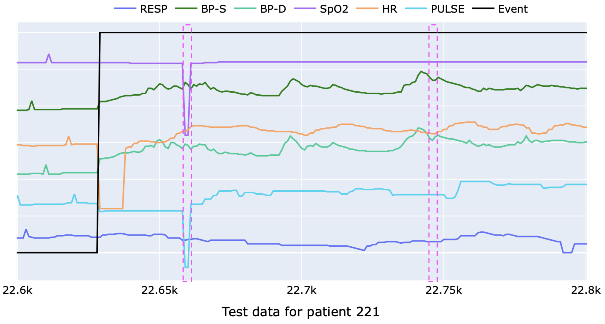
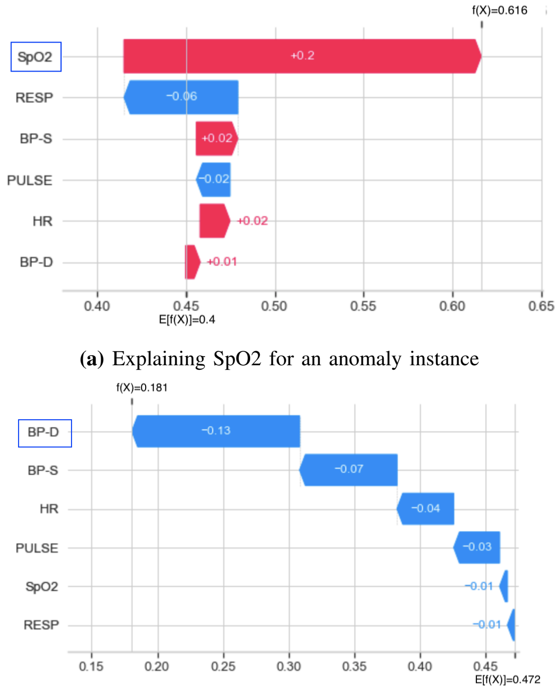


Fig. 11: The actual vital data for the two instances, where the dotted rectangles (magenta) represent indexes 22659 and 22746, respectively.

Moreover, Fig. 12a and Fig. 12b show the SHAP values of two different instances when their flags were raised by anomalies. It can be seen that in both instances, the features that have significant contributions towards the flags, were the features being explained. In fact, those features are the ones where the anomalies were injected. The results of Fig. 10 and Fig. 12 coincide with the assumption that an event shows abnormality in more than one feature, whereas this is not necessarily the case with anomalies.

Explaining events help in understanding the vitals that caused the medical event, whereas explaining the exact cause of the anomalies is not in the scope of this work. However, the discussion in this section shows that SHAP values can be used to differentiate between anomalies and events, given the different trend shown by each. Since this analysis is based on relativity and cannot be quantified, the next section will demonstrate the results of using NN-based classification of anomaly and events using the SHAP values.



(a) Explaining SpO2 for an anomaly instance

(b) Explaining BP-S for an anomaly instance

Fig. 12: Explaining flags raised by anomalies for patient 221

E. Evaluation of the Event and Anomaly Classification (EAC) Model

In this section, the performance of the EAC model is evaluated. Table V shows the precision, recall and F1-score of each class, i.e. event or anomaly, with varying percentages of injected anomalies. It can be seen that the model was able to accurately classify anomalies and events when the percentages of anomalies was $\leq 10\%$. On the other hand, when anomalies are $\geq 20\%$, the performance only marginally dropped. However, the recall of the event class, i.e. event detection, remained 1.0 regardless of the amount of anomalies.

TABLE V: Results of the EAC model with different percentages of injected anomaly for patient 221

% of anomalies	Class	Precision	Recall	F1-score
5%	Event	1.0	1.0	1.0
	Anomaly	1.0	1.0	1.0
10%	Event	1.0	1.0	1.0
	Anomaly	1.0	1.0	1.0
20%	Event	0.99	1.0	0.99
	Anomaly	1.0	0.98	0.99
30%	Event	0.98	1.0	0.99
	Anomaly	1.0	0.94	0.97

It is worth mentioning that the false positives that were discussed in Section V-C, i.e. instances with reconstruction errors above the threshold that were neither anomaly nor an event, where misclassified as anomalies. This makes them insignificant, since anomalies are ignored and only events and their explanations are used to alert the medical professional.

F. Discussion of Benchmark

In this section, the proposed approach is compared against four benchmarks that were concerned with the anomaly and event detection in medical applications. The benchmarks use the same dataset used in this work, the MIMIC dataset, and specifically for patient 221. As discussed in Section II, the first and second benchmarks, *the PCA-based approach* and *the PCA-based approach 2*, rely on PCA for dimensionality reduction, here the squared prediction errors are compared with a preset threshold for detection [11], [15]. Moreover, the third benchmark, *MD-based approach*, relies on the Mahalanobis distance (MD) of different sensor readings for abnormality detection, and then utilizes kernel density estimator (KDE) to discover the abnormal sensor [9]. The last benchmark, *SMO-based approach*, relies on sequential minimal optimization regression (SMO regression) to predict the sensor reading and the error between predicted and actual value is used for detection [13].

Table VI shows the different benchmarks and their performance. To be consistent with the benchmark, the performance metric is in terms of recall and false positive rate (FPR) of the EAD model. It can be concluded that the proposed approach outperforms [11], [15], [9] in terms of recall, and [9], [13] in terms of FPR. In addition, the proposed approach has the advantage of explaining and understanding the events before alerting the medical professional, while the benchmarks fail to do so.

TABLE VI: Comparison of the proposed approach with four different benchmarks using dataset for patient 221, with 20% injected anomalies.

	Recall	FPR
Proposed approach	**1.0**	**0.02**
PCA-based approach [11]	0.98	0.007
PCA-based approach 2 [15]	0.93	0.01
MD-based approach [9]	0.86	0.08
SMO-based approach [13]	1.0	0.05

VI. Conclusion

In conclusion, EAD in the medical field has the potential to significantly improve patient care. This paper proposed an online EAD approach using AE on MIoT. The flags raised by the AE are explained using KernelSHAP, where the use of explainable AI help ensuring that the decision-making process behind these systems is transparent and can be understood by medical professionals. Finally, the SHAP values of the features are used to classify flags into anomalies or events using ANN. Using extensive simulations and labelled real-life dataset of patients physiological data, the proposed approach showed efficacy detecting and classifying all of the medical events, even in the presence of varying percentages of injected anomalies. Finally, the proposed approach was compared with four benchmarks, were it outperformed them in terms of either recall, FPR, or both.

VII. Acknowledgements

This work was supported by the Center on Cyber Physical System as part of Khalifa University Internal Research Fund under Grant 8474000137.

References

[1] A. Alagha, S. Singh, R. Mizouni, J. Bentahar, and H. Otrok, "Target localization using multi-agent deep reinforcement learning with proximal policy optimization," *Future Generation Computer Systems*, vol. 136, pp. 342–357, 2022.

[2] M. Shurrab, S. Singh, R. Mizouni, and H. Otrok, "Iot sensor selection for target localization: A reinforcement learning based approach," *Ad Hoc Networks*, vol. 134, p. 102927, 2022.

[3] H.-N. Dai, H. Wang, G. Xu, J. Wan, and M. Imran, "Big data analytics for manufacturing internet of things: opportunities, challenges and enabling technologies," *Enterprise Information Systems*, vol. 14, no. 9-10, pp. 1279–1303, 2020.

[4] N. Akhtar, S. Rahman, H. Sadia, and Y. Perwej, "A holistic analysis of medical internet of things (miot)," *Journal of Information and Computational Science*, vol. 11, no. 4, pp. 209–222, 2021.

[5] G. Pachauri and S. Sharma, "Anomaly detection in medical wireless sensor networks using machine learning algorithms," *Procedia Computer Science*, vol. 70, pp. 325–333, 2015.

[6] S. K. Mohammed, S. Singh, R. Mizouni, and H. Otrok, "A deep learning framework for target localization in error-prone environment," *Internet of Things*, vol. 22, p. 100713, 2023.

[7] A. Ukil, S. Bandyoapdhyay, C. Puri, and A. Pal, "Iot healthcare analytics: The importance of anomaly detection," in *2016 IEEE 30th international conference on advanced information networking and applications (AINA)*. IEEE, 2016, pp. 994–997.

[8] O. Salem, A. Guerassimov, A. Mehaoua, A. Marcus, and B. Furht, "Anomaly detection in medical wireless sensor networks using svm and linear regression models," *International Journal of E-Health and Medical Communications (IJEHMC)*, vol. 5, no. 1, pp. 20–45, 2014.

[9] O. Salem, Y. Liu, and A. Mehaoua, "Anomaly detection in medical wireless sensor networks," *Journal of Computing Science and Engineering*, vol. 7, no. 4, pp. 272–284, 2013.

[10] L. Ben Amor, I. Lahyani, and M. Jmaiel, "Audit: anomalous data detection and isolation approach for mobile healthcare systems," *Expert Systems*, vol. 37, no. 1, p. e12390, 2020.

[11] L. B. Amor, I. Lahyani, and M. Jmaiel, "Data accuracy aware mobile healthcare applications," *Computers in Industry*, vol. 97, pp. 54–66, 2018.

[12] M. B. Mohamed, A. M. Makhlouf, and A. Fakhfakh, "Correlation for efficient anomaly detection in medical environment," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2018, pp. 548–553.

[13] S. A. Haque, M. Rahman, and S. M. Aziz, "Sensor anomaly detection in wireless sensor networks for healthcare," *Sensors*, vol. 15, no. 4, pp. 8764–8786, 2015.

[14] G. Smrithy, R. Balakrishnan, and N. Sivakumar, "Anomaly detection using dynamic sliding window in wireless body area networks," in *Data science and big data analytics*. Springer, 2019, pp. 99–108.

[15] L. B. Amor, I. Lahyani, and M. Jmaiel, "Pca-based multivariate anomaly detection in mobile healthcare applications," in *2017 IEEE/ACM 21st International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. IEEE, 2017, pp. 1–8.

[16] M. Tsukada, M. Kondo, and H. Matsutani, "A neural network-based on-device learning anomaly detector for edge devices," *IEEE Transactions on Computers*, vol. 69, no. 7, pp. 1027–1044, 2020.

[17] A. Arfaoui, A. Kribeche, S. M. Senouci, and M. Hamdi, "Game-based adaptive anomaly detection in wireless body area networks," *Computer Networks*, vol. 163, p. 106870, 2019.

[18] O. Salem, Y. Liu, and A. Mehaoua, "Anomaly detection in medical wsns using enclosing ellipse and chi-square distance," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 3658–3663.

[19] H. W. Loh, C. P. Ooi, S. Seoni, P. D. Barua, F. Molinari, and U. R. Acharya, "Application of explainable artificial intelligence for healthcare: A systematic review of the last decade (2011–2022)," *Computer Methods and Programs in Biomedicine*, p. 107161, 2022.

[20] A. Holzinger, R. Goebel, R. Fong, T. Moon, K.-R. Müller, and W. Samek, "xxai-beyond explainable artificial intelligence," in *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*. Springer, 2022, pp. 3–10.

[21] C.-T. Kor, Y.-R. Li, P.-R. Lin, S.-H. Lin, B.-Y. Wang, and C.-H. Lin, "Explainable machine learning model for predicting first-time acute exacerbation in patients with chronic obstructive pulmonary disease," *Journal of personalized medicine*, vol. 12, no. 2, p. 228, 2022.

[22] M. Rashed-Al-Mahfuz, A. Haque, A. Azad, S. A. Alyami, J. M. W. Quinn, and M. A. Moni, "Clinically applicable machine learning approaches to identify attributes of chronic kidney disease (ckd) for use in low-cost diagnostic screening," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 9, pp. 1–11, 2021.

[23] C. Kokkotis, C. Ntakolia, S. Moustakidis, G. Giakas, and D. Tsaopoulos, "Explainable machine learning for knee osteoarthritis diagnosis based on a novel fuzzy feature selection methodology," *Physical and Engineering Sciences in Medicine*, vol. 45, no. 1, pp. 219–229, 2022.

[24] K. Bahani, M. Moujabbir, and M. Ramdani, "An accurate fuzzy rule-based classification systems for heart disease diagnosis," *Scientific African*, vol. 14, p. e01019, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2468227621003203

[25] N. Ketkar, "Introduction to keras," in *Deep learning with Python*. Springer, 2017, pp. 97–111.

[26] N. K. Manaswi, "Understanding and working with keras," in *Deep Learning with Applications Using Python*. Springer, 2018, pp. 31–43.

[27] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, *Explainable AI: interpreting, explaining and visualizing deep learning*. Springer Nature, 2019, vol. 11700.

[28] T. Yilmaz, R. Foster, and Y. Hao, "Detecting vital signs with wearable wireless sensors," *Sensors*, vol. 10, no. 12, pp. 10 837–10 862, 2010.

[29] L. Antwarg, R. M. Miller, B. Shapira, and L. Rokach, "Explaining anomalies detected by autoencoders using shapley additive explanations," *Expert Systems with Applications*, vol. 186, p. 115736, 2021.

[30] Y.-G. Lee, J.-Y. Oh, D. Kim, and G. Kim, "Shap value-based feature importance analysis for short-term load forecasting," *Journal of Electrical Engineering & Technology*, pp. 1–10, 2022.

[31] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE transactions on emerging topics in computational intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[32] T. Luo and S. G. Nagarajan, "Distributed anomaly detection using autoencoder neural networks in wsn for iot," in *2018 ieee international conference on communications (icc)*. IEEE, 2018, pp. 1–6.

[33] M. Abououf, R. Mizouni, S. Singh, H. Otrok, and E. Damiani, "Self-supervised online and lightweight anomaly and event detection for iot devices," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 25 285–25 299, 2022.