AI-Driven Resource Allocation in Edge-Fog Computing: Leveraging Digital Twins for Efficient Healthcare Systems

Use Case: Cardiovascular Diseases in Mauritania

Brahim Ould Cheikh Mohamed Nouh[1], Rafika Brahmi[2], Sidi Cheikh[3]*,
Ridha Ejbali[4], Mohamedade Farouk Nanne[5]
Research Unit. CSIDS of FST, University of Nouakchott, Nouakchott, Mauritania[1,3,5]
Research Unit. (RTIM), National School of Engineers, University of Gabes, Gabes 6029, Tunisia[2,4]

***Abstract*—The evolution of healthcare, driven by remote monitoring and connected devices, is transforming medical service delivery. Digital twins, virtual replicas of patients, enable continuous monitoring and predictive analysis. However, the rapid growth of real-time health data presents major challenges in resource allocation and processing, especially in cardiac event prediction scenarios. This paper proposes an artificial intelligence-based approach to optimize resource allocation in a fog-edge computing environment, with a focus on Mauritania. The system integrates a deep learning model (CNN-BiLSTM), which achieves 98% accuracy in predicting cardiovascular risks from physiological signals, combined with a Deep Q-Network (DQN) to dynamically decide whether tasks should run at the edge or in the fog. Using IoT sensors, real-time health data is collected and processed intelligently, ensuring low latency and rapid response. Digital twins provide a synchronized virtual representation of the physical system for real-time supervision. This architecture improves resource utilization, reduces processing delays, and enhances responsiveness to critical medical conditions, supporting more accurate cardiac event prediction and timely intervention, especially in resource-constrained environments.**

Keywords*—Edge computing; fog computing; digital twin; deep learning; CNN-BiLSTM; Deep Q-Network (DQN); resource allocation; cardiac event prediction; healthcare; Artificial Intelligence (AI); Internet of Things (IoT); real-time

I. Introduction

Cardiovascular diseases, which claim millions of lives each year, remain one of the leading causes of mortality worldwide [1]. In Mauritania, the prevalence of cardiovascular disease (CVD) mortality is estimated at 16%, making it the leading cause of death from non-communicable diseases (NCDs) [2]. Hypertension (HTN), affecting 27% of the Mauritanian population [3], is the primary contributor to the burden of strokes, ischemic heart diseases, and hypertensive cardiopathies. The prevention, detection, and treatment of hypertension remain insufficient due to a lack of public awareness about risk factors, symptoms, and complications of the disease, as well as weaknesses in the healthcare system [4]. Implementing a decision support system [5] that facilitates early detection, alongside efficient resource management and rapid intervention in cardiac emergencies is crucial to improving patient survival rates [6] and achieve the target of a 33% reduction in premature mortality by 2030 [7].Moreover, a survey conducted among cardiologists at the National Cardiology Center (CNC) reveals strong support for these innovative solutions [8].However, the effective management of real-time data from medical monitoring devices remains a significant challenge, particularly in distributed environments (Edge or Fog Computing) where computational resources are often limited [9]. Edge-Fog Computing environments, positioned near IoT devices, allow for decentralized data processing, thus reducing latency and bottlenecks associated with data transfer to the cloud [10]. Optimizing resource allocation in such distributed systems is a central challenge. Dynamic resource management—including bandwidth, computing power, and storage—is crucial, especially when handling critical real-time data streams, such as those generated by biometric sensors and cameras in cardiac monitoring systems [11]. To address these challenges, integrating Edge-Fog Computing systems with artificial intelligence (AI) approaches and digital twins paves the way for intelligent and scalable healthcare systems that can adapt to the dynamic needs of patients and infrastructure [12] [13]. In this context, our work proposes an innovative approach to optimizing resource allocation in Edge-Fog Computing environments, specifically designed to enhance the prediction of cardiac events. It combines advanced AI models, including a hybrid CNN-LSTM model for cardiac event prediction and a Deep Q-Network (DQN) for dynamic resource allocation. This system aims to establish a real-time health monitoring framework capable of predicting patients' cardiac health status, determining the optimal location for task processing—whether at the edge or fog—and delivering rapid responses in critical situations. Moreover, integrating digital twins into this architecture enables comprehensive system supervision, providing a platform for real-time monitoring and predictive analysis [14]. These digital twins not only simulate system behavior under varying conditions [fdgth] but also continuously optimize resource allocation decisions [15]. Preliminary results indicate that this approach effectively handles workload variations, improves system performance, and supports rapid response to critical situations. The main contributions of our research study are as follows:

1) AI-Driven heart attack risk prediction at the edge: Development and implementation of a CNN-BiLSTM deep

*Corresponding authors.

learning model for heart attack prediction, enabling real-time monitoring and accurate risk assessment directly on edge devices.

2) Dynamic resource allocation optimisation: Implementation of a reinforcement learning Deep Q-Network (DQN) model to optimise resource management. This model dynamically determines whether data, including video streams in critical situations, should be processed locally on edge devices or offloaded to the fog layer in resource-intensive scenarios.

3) Integrating digital twin technology: Use of digital twins for cardiac monitoring in healthcare to refine the accuracy of heart attack predictions, optimise resource allocation and improve system performance through real-time monitoring, notification in critical situations and continuous optimisation based on replicated data.

The rest of this paper is organized as follows: Section II discusses related work. In Section III, we presents the proposed framework. Furthermore, Section IV is the results and discussion. The conclusion and the paper's potential future directions are presented in Section V.

II. Related Work

Many authors have carried out studies relevant to our research. In this section, the key studies are organized into sub-paragraphs with clear headings for improved readability and are summarized below:

A. Resource Allocation in Fog and Edge Computing for Healthcare

Talaat et al. [16] introduced EPRAM, a method combining Deep Reinforcement Learning (DRL) and Probabilistic Neural Networks (PNN) to enhance resource allocation and heart disease prediction in Fog environments. The system includes modules for data preprocessing, resource allocation, and effective prediction, significantly reducing latency and improving load balancing. Aazam et al. [17] focused on task offloading in Edge Computing using machine learning (ML) models such as kNN, Naive Bayes, and SVC. Although their models improved processing efficiency in medical scenarios (including COVID-19-related cases), they did not report specific performance metrics. Khan et al. [18] proposed a dynamic resource allocation algorithm for IoHT applications. Their results demonstrated a 45% reduction in delay, 37% reduction in energy consumption, and 25% reduction in bandwidth usage compared to existing approaches.

B. Machine Learning-Based Medical Data Processing

Amzil et al. [19] developed ML-MDS, a medical data segmentation method that achieved 92% accuracy while reducing latency by 56%. Similarly, Ullah et al. [20] used fuzzy reinforcement learning to design energy-efficient healthcare IoT systems. Hanumantharaju et al. [21] applied Random Forest and Naive Bayes algorithms for heart disease prediction. Scrugli et al. [22], on the other hand, achieved over 97% accuracy using a CNN to detect arrhythmia disorders.

C. Deep Learning and Synthetic Data for Cardiac Events

Rajapaksha et al.[23] used LSTM models with synthetic data to predict cardiac arrests, achieving 96% accuracy. Tang et al.[24] introduced SH-CSO, an optimization algorithm that achieved 96. 16% precision for heart disease and 97. 26% for the diagnosis of diabetes. Dritsas et al.[25] compared several deep learning models on a heart attack prediction dataset. Their hybrid model outperformed others with 91% accuracy, 89% precision, and 90% recall.

D. Hybrid Deep Learning Models for Cardiac Prediction

Hossain et al.[26] used a hybrid CNN-LSTM model, achieving up to 74.15% accuracy. Sudha et al.[27] achieved 89% using a similar approach. Verma et al.[28] proposed the FETCH system, which combines Fog Computing, IoT, and DL to enhance real-time cardiac monitoring. Elsayed et al.[29] integrated CNN and Fog Computing for image-based diagnosis, achieving near-perfect accuracy (99.88%) on X-ray images.

E. Architectures and Comparative Studies

Tripathy et al. [30] proposed an architecture combining quartet deep learning and edge devices, evaluated using the FogBus framework based on performance indicators such as congestion and accuracy. Scrugli et al.[22] compared several ML algorithms (LR, SVM, NB, KNN, RF, GB) to identify the best one for early heart failure detection, especially within cloud computing environments. The Table I provides an overview of studies focusing on edge–fog systems in the healthcare domain.

The Table [I] provides an overview of studies focusing on Edge–Fog systems in the healthcare domain, highlighting the key limitations identified in each study.

III. Proposed Framework

We proposed a multi-layered framework for remote healthcare monitoring and resource allocation, including IoT sensors, edge computing, fog and digital twin technology, to predict heart attacks in real time and allocate resources efficiently. IoT sensors collect key physiological data, including parameters such as heart rate, type of chest pain and cholesterol levels, alongside video input from a camera during critical events. Edge devices are used to run pre-trained deep learning models to predict a heart attack, and activate the camera as needed. During the same time, a deep Q-Network (DQN) decides if the data is processed locally or offloaded to the fog layer. Predictions and video frames are transmitted to a digital twin, which not only monitors the patient's health but also diagnoses the situation based on the collected data. If the digital twin detects an emergency, such as a potential heart attack, it automatically notifies the medical staff, family members, and ambulance teams, enabling prompt intervention and refines resource allocation through historical analysis. Fig. 1 illustrates the architecture of this system, highlighting the seamless flow from data collection to decision making.In the following sections, each layer of the proposed architecture will be detailed in the following sections.

TABLE I. OVERVIEW OF STUDIES FOCUSING ON EDGE-FOG SYSTEMS FOR HEALTHCARE

Reference	Focus	AI Technique / Architecture	Limitations
(Aazam et al.) [2021] [17]	underscores the significance of intelligent decision-making in resource-constrained environments for enhancing	kNN,naive Bayes (NB), SVC/ Edge-Cloud	Algorithmic Limitations : The study does not fully address how resource allocation is managed dynamically across middleware entities
(scrugli et al.) [2021] [22]	explore the implementation of a system for at-the-edge cognitive processing of ECG data.	CNN/ Edge-Cloud	Limited Scope of Generalization,
(khan et al.) [2022][18]	This paper proposes workload-aware efficient resource allocation and load balancing in the fog-computing environment for the IoHT.	algo/fog-Cloud	Overemphasis on Simulation: The study is largely validated through simulations, which might not fully replicate the complexity of real-world healthcare scenarios.
(talaat et al.) [2022][16]	the EPRAM paper significantly advances the understanding and implementation of resource allocation and prediction in fog computing, particularly for smart healthcare systems, by introducing a comprehensive and effective methodology.	PNN,RL/ Fog-Cloud	it lacked specific implementation details. to confirm its effectiveness in real-world healthcare FC deployments.
(verma et al.) [2022][28]	combines fog computing with IoT and deep learning to enable efficient healthcare monitoring and diagnosis	Random Forests, Gradient Boosting/Fog-Cloud	does not address dynamic resource allocation strategies effectively.
(elhadad et al.) [2022][31]	Immediate notification handling in healthcare monitoring	Algorithmic Pattern Recognition/Fog-Cloud	lacks comprehensive strategies for managing limited computational and energy resources on fog nodes effectively. This could hinder scalability for high-demand healthcare applications
(hanumantharaju et al.) [2022][21]	develop a novel fog-based healthcare system for Mechanized Diagnosis of Heart Diseases using ML algorithms	Random Forest, Naive Bayes/Fog-Cloud	-Dynamic Resource Allocation: The dynamic and often unpredictable nature of healthcare demands is not fully accounted for, which could lead to inefficiencies in resource utilization during peak usage periods. -Lack of Real-World Validation
(hossain et al.) [2023][26]	Combined CNN and LSTM to identify Cardiovascular disease	CNN, LSTM	Lack of Real-Time Deployment Considerations Neglect of Resource Allocation
(sudha et al.) [2023][27]	Combined CNN and LSTM to identify Cardiovascular disease	CNN, LSTM	Deployment challenges include optimizing resources in real-time environments.
(elsayed et al.) [2023][29]	intersection of fog computing and modified CNNs in the domain of healthcare image analysis	CNN/Fog-Cloud	Resource Constraints in Fog Computing and need to implement an effective resource allocation strategy
(tripathy et al.) [2023][30]	The approach uses a quartet deep learning framework combined with fog and edge computing to process healthcare data closer to the user, reducing dependency on cloud services.	DQN/Fog-Cloud	The paper emphasizes the efficiency of the fog platform but does not delve deeply into adaptive resource allocation strategies.
(rajapaksha et al.) [2023][23]	developed predictive model in identifying the likelihood of developing cardiac	LSTM	Lack of Real-Time Testing
(ullah et al.) [2024][20]	Treduce delays in processing and transmitting healthcare data	FIS,RL,NN/Fog-Cloud	Problem of dynamic resource allocation
(dritsas et al.) [2024][25]	apply and compare the performance of five well-known Deep Learning (DL) models, to a heart attack prediction dataset.	MLP,CNN,RNN,LSTM, GRU	Computational Overhead: hybrid architectures, are computationally intensive. how these models can be deployed in resource-constrained environments, such as edge or fog computing.
(tang et al.) [2024][24]	create a model for detecting diabetes and cardiovascular diseases by integrating AI and IoT	SH-CSO algorithm/Fog-Cloud	The aspect of resource allocation is not addressed, especially given that fog nodes are limited in resources.
(dayana et al.) [2024][32]	The paper emphasizes the importance of ML methods for early detection, diagnosis, and prevention, aiming to reduce mortality rates and healthcare costs associated with heart disease	LR,SVM,NB,KNN,RF,GB	The paper does not adequately address the practical limitations of deploying cloud-driven machine learning models in environments with limited resources
(amzil et al.) [2024][19]	an ML-based approach to improve health data classification and reduce latency in healthcare systems	k-fold random forest	- Limited Focus on Real-Time Validation - Lack of Dynamic Resource Allocation

A. IoT and Sensor Layer

The IoT and Sensor Layer plays a pivotal role in the continuous collection of real-time data from the patient, utilizing a variety of physiological and camera sensors. Physiological sensors continuously monitor key health parameters, including heart rate (HR), blood pressure (BP), and cholesterol levels. The data collected from these sensors serves as the primary input for evaluating the patient's health condition and is subsequently fed into the predictive AI model for heart attack prediction and other critical health assessments. In emergency situations, the camera captures video feeds that offer visual context regarding the patient's physical state. This visual data complements the physiological measurements and enhances the overall understanding of the patient's condition, particularly during critical events.

B. Edge Computing Layer

The Edge Computing Layer is the central layer in this work, responsible for processing the patient's health data from physiological sensors using an AI model for heart attack prediction. The camera is activated only in critical situations to capture video frames, ensuring privacy. A Deep Q-Network (DQN) model is used to decide whether to process the data locally on the Raspberry Pi or offload it to the Fog Layer, optimizing resource usage. Once processed, all data, including health metrics and video frames, are transmitted from the Fog

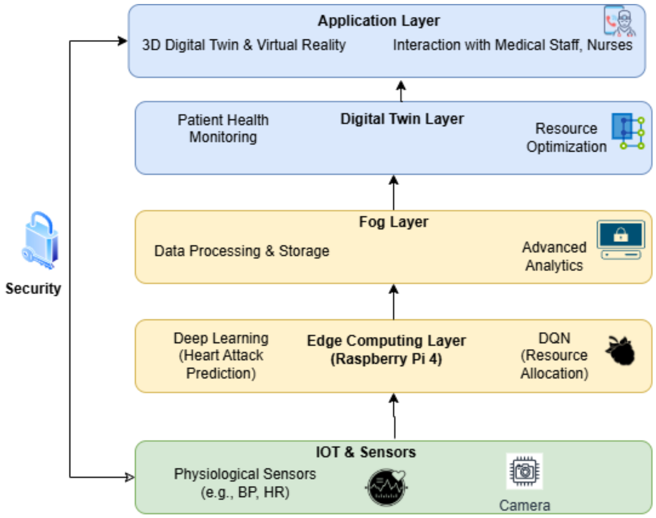

Fig. 1. Multi-layer architecture of the proposed framework for heart attack prediction and resource allocation, integrating IoT sensors, edge computing, fog, and digital twin technology.

or Raspberry Pi to the Digital Twin Layer. This data allows for the continuous update of the virtual model, supporting real-time health monitoring and decision-making.

1) AI Driven heart attack risk prediction at the edge: We trained an IA model for heart attack prediction using a hybrid convolutional neural network (CNN) and bidirectional long-short-term memory (BiLSTM) architecture. This model was specifically designed to predict heart attacks based on physiological data, including heart rate, blood pressure, and cholesterol levels. The model was trained and deployed on a Raspberry Pi 4B, which features a quad-core Cortex-A72 processor and 4GB of RAM, providing sufficient computational power for edge-based inference.

a) Dataset: In this study, the data set from the UCI machine learning repository dataset is used . Data in the dataset are collected from the Hungarian Institute of Cardiology, Cleveland clinic foundations. It consists of information on patient records both normal and abnormal. This database contains 76 attributes, with a total of 303 observations. The attributes are age, sex, resting blood pressure, cholesterol, etc. And the data set consists of six missing values. In 303 observations, 138 are normal persons, and 165 are abnormal persons, i.e., sufered from heart disease.

b) CNN-BiLSTM Architecture: Our proposed hybrid CNN-BiLSTM model leverages Convolutional Neural Networks (CNNs) for feature extraction and Bidirectional Long Short-Term Memory (BiLSTM) layers for sequential learning, effectively capturing both spatial and temporal dependencies to enhance prediction accuracy. The architecture, illustrated in Fig. 2, consists of a CNN layer followed by a dropout of 0.5, a BiLSTM layer with 64 units, and a fully connected layer. The model was trained for 200 epochs with a learning rate of 0.0025, utilizing the softmax activation function.

c) CNN: CNN has been effectively used in image processing, face recognition and time series analysis, among other applications[33].It is possible to construct CNN architecture by stacking three primary layers: convolution, pooling, and fully

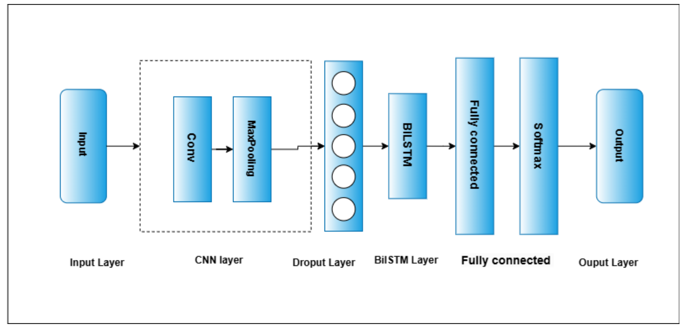

Fig. 2. CNN-BILSTM architecture.

connected (FC). Every convolution layer has a set of learnable filters whose objective is to automatically extract local characteristics from the input matrix using the learned filters. It is possible to minimize the complexity of the computational load and improve model performance by using filters that execute convolution operations based on two essential notions, namely weight sharing and local connection, which may be achieved via filters [34].

d) BiLSTM: As an extension to RNNs, Long Short-Term Memory (LSTM) is introduced to remember long input data and thus the relationship between the long input data and output is described in accordance with an additional dimension (e.g., time or spatial location). An LSTM network remembers long sequence of data through the utilization of several gates such as: 1) input gate, 2) forget gate, and 3) output gate. The deep-bidirectional LSTMs (BiLSTM) networks are a variation of normal LSTMs, in which the desired model is trained not only from inputs to outputs, but also from outputs to inputs. More precisely, given the input sequence of data, a BiLSTM model first feed input data to an LSTM model (feedback layer), and then repeat the training via another LSTM model but on the reverse order of the sequence of the input data (i.e., Watson-Crick complement [35].

In this work we proposed A hybrid model for predicting heart disease using CNN and BiLSTM algorithms.

e) Evaluation metrics: The model's performance was evaluated using metrics such as accuracy, precision, recall, and F1-score, ensuring its effectiveness in real-time heart attack prediction.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

$$\mathrm{T}_{\text{inference}}(\mathrm{ms}) = \mathrm{T}_{\text{out}} - \mathrm{T}_{\text{inp}} \quad (5)$$

Where:

- TP (True Positive): The number of correctly identified heart attack cases, where the model accurately predicts a heart attack event.
- FN (False Negative): The number of heart attack cases that were not predicted by the model, indicating missed detections of actual heart attacks.
- FP (False Positive): The number of instances where the model incorrectly predicts a heart attack, leading to false alarms for non-heart attack events.
- TN (True Negative): The number of correctly identified non-heart attack cases, where the model accurately predicts the absence of a heart attack.
- T_{inp}: The timestamp when the physiological data (e.g., heart rate, blood pressure) is fed into the prediction model for analysis.
- T_{out}: The timestamp when the heart attack prediction result is generated, marking the point at which the model's decision is outputted for clinical assessment.

2) Allocation resources model using DQN (Deep Q-Network): Resource allocation in an Edge-Fog environment presents a significant challenge due to the diverse nature of tasks, fluctuating workloads, and the stringent demands for low latency. Achieving an optimal balance between local processing (Edge) and offloading to the Fog requires quick, adaptive decision-making to ensure minimal latency, maximize resource efficiency, and control costs effectively. The Deep Q-Network (DQN) emerges as a promising solution, enabling autonomous learning to make optimal decisions in complex and dynamic environments [36]. In the context of healthcare, particularly in heart attack prediction, intelligent Edge-Fog resource management can enhance prediction accuracy and, more importantly, save lives by ensuring the rapid and reliable processing of critical data.

- DQN Concepts: The Deep Q-Network (DQN) is a reinforcement learning algorithm that combines Q-learning, a table-based control method, with deep neural networks. Q-learning problems are typically framed as Markov Decision Processes (MDPs), which consist of pairs of states (s_t) and actions (a_t). State transitions occur with a transition probability (p), a reward (r), and a discount factor (γ). The transition probability p reflects the likelihood of transitioning between states and receiving associated rewards. According to the Markov property, the next state and reward depend only on the previous state (s_{t-1}) and action (a_{t-1})[37]. Traditional Q-learning struggles to handle large-scale or continuous-space MDPs due to the *curse of dimensionality* in the Q-table. To address this issue, DeepMind introduced the DQN algorithm, which approximates the Q-table using deep neural networks. In DQN, the Q value of each action can be predicted by simply inputting the current state (s_τ) into the network, simplifying computation. The DQN uses a deep neural network $Q(s, a; \omega)$, parameterized by weights ω, to approximate the value function $Q(s, a)$. In this framework, the agent is responsible for learning, while the environment provides the interaction context [38]. The primary objective of the agent is to learn optimal actions that maximize cumulative rewards. The agent selects actions (a_τ) and trains the neural network, while the environment updates the state (s_t) and computes the reward (r_t). The DQN employs two neural networks , the evaluation network (*eval-net*) and the target network (*target-net*), which share the same architecture [39]. The *eval-net* estimates Q values, while the *target-net* provides stable Q values as targets for training. The Q values are updated using a modified Bellman equation:

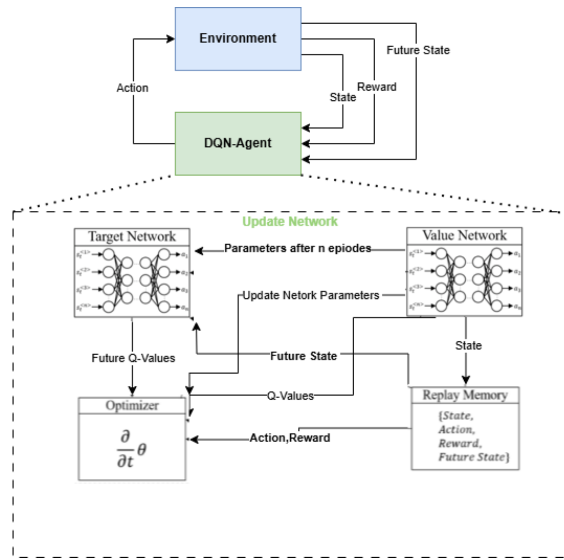

Fig. 3. Concept of DQN.

$$Q'(s_t, a_t) = Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right] \quad (6)$$

where $Q(s_t, a_t)$ and $Q'(s_t, a_t)$ are the current and updated Q values for a given action a in state s at time t, α is the learning rate (typically a small positive value), r_{t+1} is the reward received after performing the action, γ is the discount factor (close to but less than 1), and $\max_a Q(s_{t+1}, a)$ represents the highest estimated Q value for the next state s_{t+1}. This approach allows DQN to learn effectively in complex environments by leveraging the power of deep neural networks. The specific process is shown in figure [3]

- System model and problem formulation: This hybrid model aims to optimize resource allocation in an Edge-Fog Computing environment for an efficient healthcare system. It combines reinforcement learning with the physical constraints of the Raspberry Pi's resources.
- System Variables: (R_a, C_a, B_a) are the resources in RAM, CPU, and Bandwidth, respectively, available on the Raspberry Pi board, (L_t) is the maximum

acceptable latency for processing data, (T_m) is the maximum operating temperature of the Raspberry Pi, and (P) is the prediction result (0 = normal, 1 = critical).

- Consumption Variable: $(R_u,\ C_u,\ B_u\ )$ are the resources in RAM, CPU, and Bandwidth, respectively, necessary for local processing, L_c is the current time measure, and T_c is the current temperature of the Raspberry Pi.

The resource constraints to ensure the optimal functioning of the system are given by:

$$\begin{aligned}\text{RAM: } R_u \leq R_a, \quad \text{CPU: } C_u \leq C_a, \quad \text{Bandwidth: } B_u \leq B_a,\\ \text{Latency: } L_c \leq L_t, \quad \text{Temperature: } T_c \leq T_m.\end{aligned} \tag{7}$$

The reward function R assesses the effectiveness of resource allocation:

$$\begin{aligned}R = \alpha_1 \cdot \text{RAM\_efficiency} + \alpha_2 \cdot \text{CPU\_efficiency}\\ + \alpha_3 \cdot \text{Bandwidth\_efficiency} - \beta \cdot \text{Latency\_penalty}\end{aligned}$$

où :

$$\begin{aligned}\text{RAM\_efficiency} &= \frac{R_a - R_u}{R_a}\\ \text{CPU\_efficiency} &= \frac{C_a - C_u}{C_a}\\ \text{Bandwidth\_efficiency} &= \frac{B_a - B_u}{B_a} \quad \text{(si offload vers Fog)}\\ \text{Latency\_penalty} &= \max(0, L_c - L_t)\end{aligned}$$

The coefficients α_1, α_2, α_3, and β are adjusted according to the relative importance of the resources. The allocation decision a is made as follows:

- If all constraints are satisfied locally:

$$\begin{aligned}R_u \leq R_a, \quad C_u \leq C_a, \quad T_c \leq T_m,\\ B_u \leq B_a, \quad L_c \leq L_t.\end{aligned} \tag{8}$$

  then $a = 0$ (Local processing).

- Otherwise, if one or more constraints are not satisfied, or if the reward is lower locally, then $a = 1$ (Transfer to Fog).

Require: Discount factor γ, exploration rate ϵ, replay memory capacity P, heart attack prediction model HA_model, DQN model DQN_model.

C. Fog Layer

In our system, after the Deep Q-Network (DQN) model runs on the Raspberry Pi to determine whether data should be processed locally or offloaded, the Fog Layer becomes essential. When the Raspberry Pi is unable to process more complex data, such as video frames captured by the camera, it transmits this data to the Fog Layer. The Fog Layer then processes these larger, more computationally demanding

Algorithm 1 DQN-Based Resource Allocation for Heart Attack Prediction (DQNRAP)

Initialize replay memory D to capacity P.
Initialize evaluation network with parameters θ.
Initialize target network with parameters $\theta' = \theta$.
Connect to Azure IoT Hub for Digital Twin synchronization.
Configure interval $T_{pred} = 0.1$ sec, buffer size $N_{threshold} = 2$.
Initialize camera to standby mode.
for each episode k **do**
  Initialize initial state s_1 by collecting sensor data (IMU, temperature, heart rate).
  for each step t **do**
    Collect real-time sensor data $Inputdata$.
    Predict heart attack status:

$$prediction = HA\_model.predict(Inputdata)$$

    Threshold Prediction:

$$heart\_status = \begin{cases}1 & \text{if } prediction > 0.5 \quad \text{(Critical)}\\ 0 & \text{otherwise (Normal)}\end{cases}$$

    Update Buffer with $heart\_status$.
    if $\sum(buffer[-N_{threshold}:]) = N_{threshold}$ **then**
      Activate camera for 1-minute video capture.
      Set $camera\_status = 1$.
    end if
    Construct State:

$$s_t = [RAM, CPU, Latency, Bandwidth, Temperature]$$

    Generate random number $h \in [0, 1]$.
    if $h < \epsilon$ **then**
      Randomly select action a_t.
    else
      $a_t = \arg\max_a Q(s_t, a; \theta)$.
    end if
    Execute action a_t (local processing or fog offloading).
    Observe reward r_t and next state s_{t+1}.
    Store Transition (s_t, a_t, r_t, s_{t+1}) in D.
    Update Evaluation Network (Algorithm 2).
    if $t\%C == 0$ **then**
      Reset Target Network: $\theta' = \theta$.
    end if
    Synchronize data with Azure IoT Hub:

$$\begin{aligned}payload = \{&timestamp,\ heart\_status,\ camera\_status,\\ &RAM,\ CPU,\ Latency,\ Bandwidth,\ a_t\}\end{aligned}$$

  end for
end for

Algorithm 2 Evaluation Network Update

1: Sample mini-batch (s, a, r, s') from D.
2: Compute target Q' value:

$$Q' = r + \gamma \max_{a'} Q(s', a'; \theta')$$

3: Update Q-network by minimizing loss:

$$Loss = (Q(s, a; \theta) - Q')^2$$

Algorithm 3 Routing Decision

1: **if** $a_t = 1$ **then**
2: Process video locally (Raspberry Pi).
3: **else**
4: Offload data to Fog.
5: **end if**

datasets, enabling efficient data handling and ensuring that the local resources are not overwhelmed. This approach optimizes the overall system performance by leveraging the Fog Layer's ability to handle more intensive computations.

D. Digital Twin Layer

The Digital Twin Layer generates a real-time virtual model of the patient, continuously updated with data from IoT sensors to monitor health status and optimize resource allocation. Leading cloud platforms, such as Amazon Web Services (AWS) and Microsoft Azure, offer solutions for building digital twins. In our work, we utilize Azure * to develop a digital twin that simulates the patient's heart condition, visualizing processed data from the camera and storing historical records. This enhances overall system prediction accuracy and improves resource allocation through predictive analytics. The Fig. 4 represents a JSON program fragment of the prediction of the IA heart attack model and also the resource allocation if data of camera will be processed at edge or transferred to the fog.the digital twin will be used to monitor the heart status and control the process of data between the edge and the fog computing .

```
{
  "id": "dtmi:example:CardiacHealthTwin;1",
  "@type": "Interface",
  "displayName": "Cardiac Health Twin",
  "contents": [
    {
      "@type": "Property",
      "name": "HeartDisease",
      "schema": "integer",
      "description": "Indicates whether the patient
        has heart disease (1 for yes, 0 for no)."
    },
    {
      "@type": "Property",
      "name": "ResourceAllocation",
      "schema": "integer",
      "description": "Allocation of processing
        resources: 0 for Edge, 1 for Fog."
    }
  ]
}
```

Fig. 4. JSON Program fragment of the patient status for Azure DT.

E. Application Layer

The Application Layer leverages 3D digital twin models and virtual reality to enhance patient monitoring and emergency response. The digital twin continuously updates with real-time data, providing a visual representation of the patient's heart condition. When critical situations are detected, the system automatically notifies medical staff and family members for immediate intervention. Beyond monitoring, the digital twin plays a vital role in refining the heart attack prediction and resource allocation models by analyzing historical data and improving decision accuracy. This ensures better healthcare management and faster response in emergencies.

IV. Results and Discussion

A. Result of Heart Attack Prediction Model

This section presents the experimental results obtained from testing the heart attack prediction model on both a PC and edge devices. Initially, the CNN-BiLSTM model was evaluated on the PC to assess its performance. Following this, the model was transferred to the Raspberry Pi 4B, where its accuracy, size, and inference time were validated.

TABLE II. Classification Report for the Heart Attack Prediction Model

Class	Precision	Recall	F1-score
0	0.9722	1.0000	0.9859
1	1.0000	0.9608	0.9800
Accuracy		0.9835	
Macro avg	0.9861	0.9804	0.9830
Weighted avg	0.9839	0.9835	0.9834

The results presented in Table II, Fig. 5, and 6 highlight the performance of our model, which achieved an overall accuracy of 98.35%. As shown in the confusion matrix in Fig. 5), the model correctly classified all instances of Class 0, resulting in a perfect classification rate of 100%. However, Class 1 achieved a slightly lower accuracy of 96.08%, with 3.92% of instances misclassified as Class 0.The classification report in Table II further demonstrates the effectiveness of our trained model. For Class 0 (No Attack), the precision is 97.22%, recall is 100%, and the F1-score is 98.00%. For Class 1 (Heart Attack), the model achieves a precision of 1.000, recall of 96.08%, and an F1-score of 98.00%.The macro and weighted averages further confirm that the model handles both classes effectively, exhibiting minimal bias while maintaining high precision, recall, and F1-scores. Additionally, the ROC curve Fig. 6 showcases the model's near-perfect ability to distinguish between the two classes, with an impressive Area Under the Curve (AUC) of 0.99.

1) Comparison of proposed heart attack prediction models with related works: To validate the effectiveness of our proposed hydrid CNN-BiLSTM heart attack prediction model, we compared it with existing models in the literature. This

*https://azure.microsoft.com/fr/products/digital-twins/

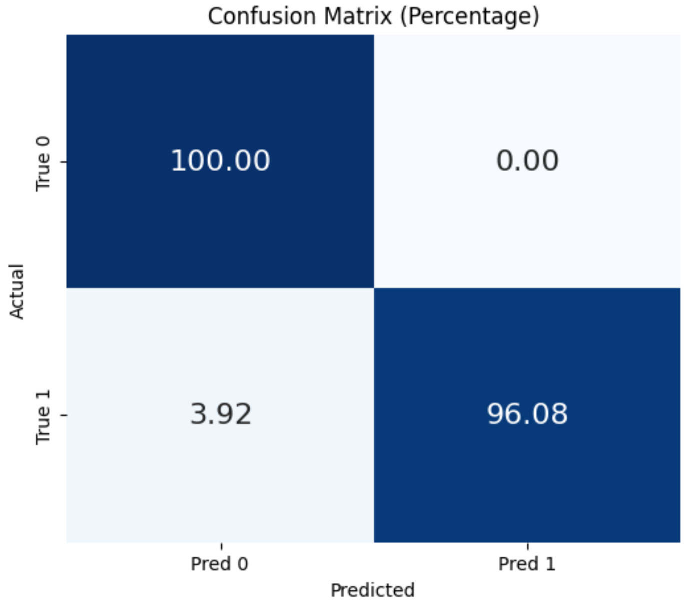

Fig. 5. Confusion matrix for the heart attack prediction model.

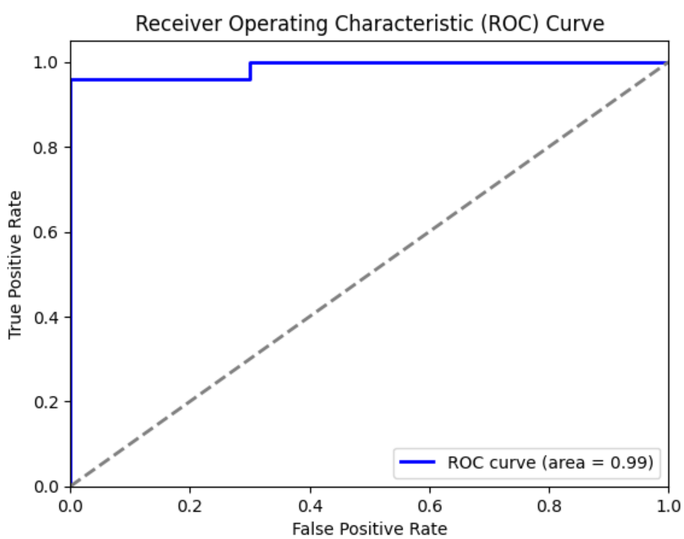

Fig. 6. Confusion matrix for the heart attack prediction model.

comparison highlights differences in model architectures using the same data set. Table III summarizes the results of the comparison using the accuracy metric of key related works.

2) Comparison of heart attack prediction models using the same Cleveland dataset: The results in III indicate that our proposed CNN-BiLSTM model achieves superior accuracy compared to traditional architectures, benefiting from the combination of convolutional and bidirectional long short-term memory (BiLSTM) layers. This hybrid architecture enhances feature extraction and temporal modeling, leading to more precise predictions.

TABLE III. COMPARISON OF HEART ATTACK PREDICTION MODELS

Date	Authors	Model Architecture	Accuracy (%)
2022[40]	Abdelghani et al.	LR algorithme	82,6
2023[27]	Sudha V K et al.	CNN-LSTM	89
2024[25]	Dritsas et al.	CNN-GRU	91
2024[41]	Remya et al.	CNN- UMAP algorithm	91
2024[42]	Bouqentar et al.	SVM	92
2023[34]	Shrivastava et al.	CNN-BiLSTM	96.66
2024	**Ours proposed Methode**	**CNN-BiLSTM**	**98.34**

B. *Result of Allocation Resource Model Using DQN*

The result of Fig. 7 shows the evolution of average cumulative rewards over episodes in the DQN model. Initially, the agent receives low rewards, but gradually improves its decisions. After approximately 200 episodes, the rewards stabilize around 50, indicating that the agent has learned an optimal strategy and that its learning process has effectively converged. Fig. 8 represents the decay of epsilon ϵ, a key parameter in DQN that regulates the balance between exploration and exploitation.

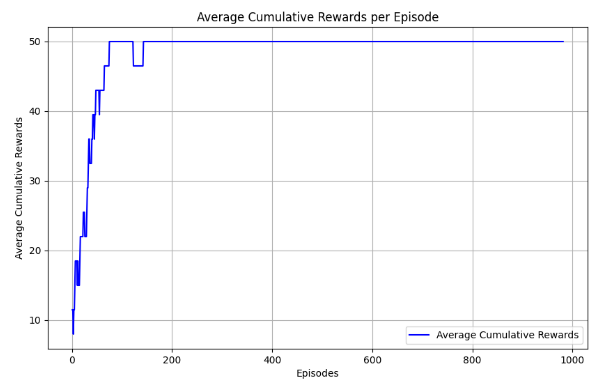

Fig. 7. Average cumulative rewards.

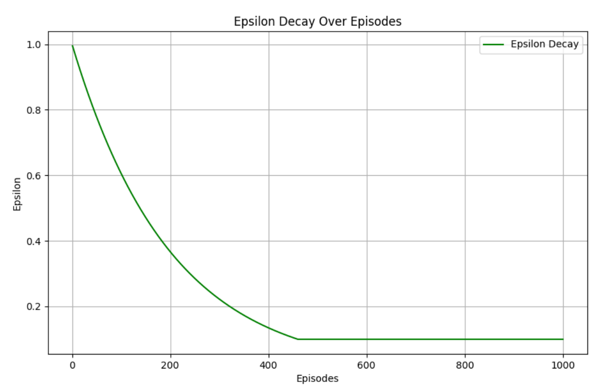

Fig. 8. Epsilon decay.

At the beginning, ϵ is high (~ 1), allowing the agent to explore various actions. As training progresses, ϵ decreases, encouraging the agent to rely more on decisions that have produced the best rewards. After (~ 400) episodes, ϵ becomes very low, which means that the agent has learned enough

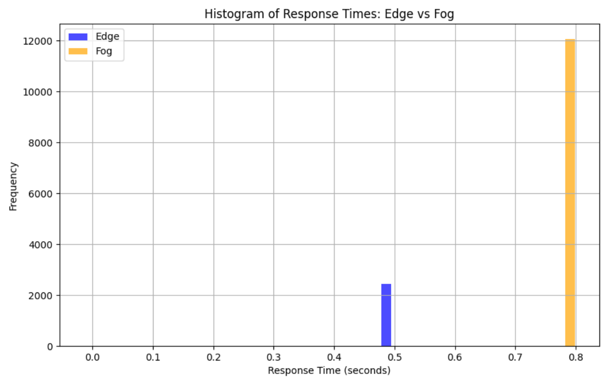

Fig. 9. Response times.

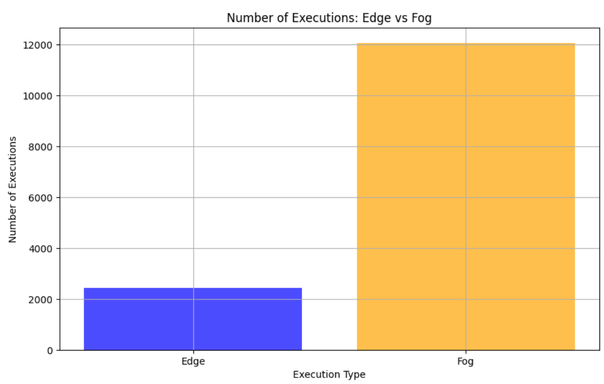

Fig. 10. Execution count.

and now relies primarily on optimal choices. These two main results complement each other: the decrease in ϵ explains the stabilization of cumulative rewards, confirming that the DQN model learns progressively, efficiently and optimally.

The study 9 displays a response time histogram comparing Edge and Fog processing. The Edge responses are significantly faster (0.5s), whereas Fog responses take longer (0.8s), indicating a higher processing delay in the Fog environment. The 10 shows that the majority of executions (over 12,000) take place in the Fog, compared to only 2,500 in the Edge.

C. Discussion

The results indicate excellent performance of the prediction model based on a hybrid CNN-BiLSTM network, with high precision and reliability metrics (see Fig. 5 and Fig. 6). This level of performance is crucial in critical scenarios such as cardiac event prediction, where false negatives could have severe consequences.In terms of resource allocation, the Deep Q-Network model demonstrates fast and efficient learning capabilities (Fig. 7 and Fig. 8), dynamically adapting to maximize system performance. Most processing decisions were offloaded to the Fog (Fig. 10), which suggests that critical tasks require more computing power than what is available at the Edge. Regarding latency, the results in Fig. 9 show that the proposed approach ensures fast processing—an essential factor in real-time medical monitoring scenarios. By integrating artificial intelligence and deep learning techniques into a Fog/Edge architecture, the system succeeds in ensuring both diagnostic accuracy and responsiveness while optimizing resource utilization.

V. Conclusion and Future Work

In this study, we proposed an intelligent and adaptive system for real-time cardiac event prediction and resource allocation in Edge-Fog Computing environments. By integrating a deep learning-based CNN-BiLSTM model for heart attack prediction and a Deep Q-Network (DQN) for dynamic resource management, our approach demonstrates the potential to enhance real-time monitoring and response efficiency in healthcare applications. Furthermore, the incorporation of digital twins into our architecture enables continuous system optimization and predictive analysis, reinforcing the reliability and adaptability of the proposed framework. Experimental results indicate that our model effectively manages workload distribution, reduces latency, and improves decision-making for critical healthcare scenarios. The ability to dynamically allocate resources between edge and fog computing environments ensures optimal system performance, even under fluctuating workloads. Our approach is highly relevant to the context of Mauritania, where cardiovascular diseases represent a significant public health challenge. It also aligns with the national goal of reducing the mortality rate from these diseases by 33% by 2030.

Future work will focus on enhancing the system's capabilities by developing an AI model at the fog level to process video data transferred from edge devices, optimizing real-time analysis and reducing latency. Additionally, we aim to improve the digital twin component to refine AI model efficiency and enhance system adaptability, leading to better overall performance. To further strengthen privacy and scalability, we will incorporate federated learning techniques, enabling decentralized model training without compromising sensitive patient data. Ultimately, our research paves the way for a more responsive and intelligent healthcare infrastructure, capable of providing real-time cardiac monitoring with high accuracy while optimizing computational resources effectively.

References

[1] A. A. Nancy, D. Ravindran, D. R. Vincent, K. Srinivasan, and C.-Y. Chang, "Fog-based smart cardiovascular disease prediction system powered by modified gated recurrent unit," *Diagnostics*, vol. 13, no. 12, p. 2071, 2023.

[2] World Health Organization, "Profil des maladies non transmissibles : Mauritanie," 2025, consulté le 15 février 2025. [Online]. Available: https://cdn.who.int/media/docs/default-source/country-profiles/ncds/mrt-fr.pdf?sfvrsn=68b09b9_36&download=true

[3] Centre National de Cardiologie, "Rapport sur les maladies cardiovasculaires en mauritanie," 2025, consulté le 15 février 2025. [Online]. Available: https://cnc.mr/fr/node/1276

[4] M. de la Santé de la Mauritanie, "Plan stratégique de lutte contre les maladies non transmissibles en mauritanie," 2021, consulté le 15 février 2025. [Online]. Available: https://www.iccp-portal.org/system/files/plans/MRT_B3_s21_PLAN_MNT_Version%20Finale%20%281%29.pdf

[5] R. Arthi and S. Krishnaveni, "Optimized tiny machine learning and explainable ai for trustable and energy-efficient fog-enabled healthcare decision support system," *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, p. 229, 2024.

[6] N. Nissa, S. Jamwal, and S. Mohammad, "Early detection of cardiovascular disease using machine learning techniques an experimental study," *Int. J. Recent Technol. Eng*, vol. 9, no. 3, pp. 635–641, 2020.

[7] M. de la Santé de la Mauritanie, "Plan national de développement sanitaire (pnds) 2022-2030, volume 1," 2024, consulté le 15 février 2025. [Online]. Available: https://p4h.world/app/uploads/2024/10/PNDS-2022-2030-Volume1_Mauritanie.x23411.pdf

[8] G. Forms, "Formulaire de collecte de données," https://docs.google.com/forms/d/1E0VIsU8r-693UH4humkXiGUBydQP74wQOYXT6LD-mq0/edit, 2025, consulté le 24 mars 2025.

[9] A. Hazra, P. Rana, M. Adhikari, and T. Amgoth, "Fog computing for next-generation internet of things: fundamental, state-of-the-art and research challenges," *Computer Science Review*, vol. 48, p. 100549, 2023.

[10] R. Das and M. M. Inuwa, "A review on fog computing: issues, characteristics, challenges, and potential applications," *Telematics and Informatics Reports*, vol. 10, p. 100049, 2023.

[11] S. Khan, T. Arslan, and T. Ratnarajah, "Digital twin perspective of fourth industrial and healthcare revolution," *Ieee Access*, vol. 10, pp. 25 732–25 754, 2022.

[12] A. A. Mutlag, M. K. Abd Ghani, O. Mohd, K. H. Abdulkareem, M. A. Mohammed, M. Alharbi, and Z. J. Al-Araji, "A new fog computing resource management (frm) model based on hybrid load balancing and scheduling for critical healthcare applications," *Physical Communication*, vol. 59, p. 102109, 2023.

[13] R. Brahmi, N. Boujnah, and R. Ejbali, "Elaboration of innovative digital twin models for healthcare monitoring with 6g functionalities," *IEEE Access*, vol. 12, pp. 109 608–109 624, 2024.

[14] A. Vallée, "Digital twin for healthcare systems," *Frontiers in Digital Health*, vol. 5, p. 1253050, 2023.

[15] A. K. Jameil and H. Al-Raweshidy, "Ai-enabled healthcare and enhanced computational resource management with digital twins into task offloading strategies," *IEEE Access*, 2024.

[16] F. M. Talaat, "Effective prediction and resource allocation method (epram) in fog computing environment for smart healthcare system," *Multimedia Tools and Applications*, vol. 81, no. 6, pp. 8235–8258, 2022.

[17] M. Aazam, S. Zeadally, and E. F. Flushing, "Task offloading in edge computing for machine learning-based smart healthcare," *Computer networks*, vol. 191, p. 108019, 2021.

[18] S. Khan, I. A. Shah, N. Tairan, H. Shah, and M. F. Nadeem, "Optimal resource allocation in fog computing for healthcare applications," *Comput. Mater. Contin*, vol. 71, no. 3, pp. 6147–6163, 2022.

[19] A. Amzil, M. Abid, M. Hanini, A. Zaaloul, and S. El Kafhali, "Stochastic analysis of fog computing and machine learning for scalable low-latency healthcare monitoring," *Cluster Computing*, pp. 1–21, 2024.

[20] A. Ullah, S. Yasin, and T. Alam, "Latency aware smart health care system using edge and fog computing." *Multimedia Tools & Applications*, vol. 83, no. 11, 2024.

[21] R. Hanumantharaju, K. Shreenath, B. Sowmya, and K. Srinivasa, "Fog based smart healthcare: a machine learning paradigms for iot sector," *Multimedia Tools and Applications*, vol. 81, no. 26, pp. 37 299–37 318, 2022.

[22] M. A. Scrugli, D. Loi, L. Raffo, and P. Meloni, "An adaptive cognitive sensor node for ecg monitoring in the internet of medical things," *IEEE Access*, vol. 10, pp. 1688–1705, 2021.

[23] L. T. W. Rajapaksha, S. M. Vidanagamachchi, S. Gunawardena, and V. Thambawita, "An open-access dataset of hospitalized cardiac-arrest patients: Machine-learning-based predictions using clinical documentation," *BioMedInformatics*, vol. 4, no. 1, pp. 34–49, 2023.

[24] Z. Tang, Z. Tang, Y. Liu, Z. Tang, and Y. Liao, "Smart healthcare systems: A new iot-fog based disease diagnosis framework for smart healthcare projects," *Ain Shams Engineering Journal*, vol. 15, no. 10, p. 102941, 2024.

[25] E. Dritsas and M. Trigka, "Application of deep learning for heart attack prediction with explainable artificial intelligence," *Computers*, vol. 13, no. 10, p. 244, 2024.

[26] M. M. Hossain, M. S. Ali, M. M. Ahmed, M. R. H. Rakib, M. A. Kona, S. Afrin, M. K. Islam, M. M. Ahsan, S. M. R. H. Raj, and M. H. Rahman, "Cardiovascular disease identification using a hybrid cnn-lstm model with explainable ai," *Informatics in Medicine Unlocked*, vol. 42, p. 101370, 2023.

[27] V. Sudha and D. Kumar, "Hybrid cnn and lstm network for heart disease prediction," *SN Computer Science*, vol. 4, no. 2, p. 172, 2023.

[28] P. Verma, R. Tiwari, W.-C. Hong, S. Upadhyay, and Y.-H. Yeh, "Fetch: a deep learning-based fog computing and iot integrated environment for healthcare monitoring and diagnosis," *IEEE access*, vol. 10, pp. 12 548–12 563, 2022.

[29] A. Z. Elsayed, K. Mohamed, and H. Harb, "Revolutionizing healthcare image analysis in pandemic-based fog-cloud computing architectures," *arXiv preprint arXiv:2311.01185*, 2023.

[30] S. S. Tripathy, M. Rath, N. Tripathy, D. S. Roy, J. S. A. Francis, and S. Bebortta, "An intelligent health care system in fog platform with optimized performance," *Sustainability*, vol. 15, no. 3, p. 1862, 2023.

[31] A. Elhadad, F. Alanazi, A. I. Taloba, and A. Abozeid, "Fog computing service in the healthcare monitoring system for managing the real-time notification," *Journal of Healthcare Engineering*, vol. 2022, no. 1, p. 5337733, 2022.

[32] T. N. Dayana *et al.*, "A comprehensive review of heart disease prediction using cloud-driven machine learning," in *2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS)*. IEEE, 2024, pp. 160–167.

[33] Y. Li, Y. He, and M. Zhang, "Prediction of chinese energy structure based on convolutional neural network-long short-term memory (cnn-lstm)," *Energy Science & Engineering*, vol. 8, no. 8, pp. 2680–2689, 2020.

[34] P. K. Shrivastava, M. Sharma, A. Kumar *et al.*, "Hcbilstm: A hybrid model for predicting heart disease using cnn and bilstm algorithms," *Measurement: Sensors*, vol. 25, p. 100657, 2023.

[35] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of lstm and bilstm in forecasting time series," in *2019 IEEE International conference on big data (Big Data)*. IEEE, 2019, pp. 3285–3292.

[36] X. Xiong, K. Zheng, L. Lei, and L. Hou, "Resource allocation based on deep reinforcement learning in iot edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1133–1146, 2020.

[37] M. SUGIMOTO, R. UCHIDA, H. MATSUFUJI, S. TSUZUKI, H. YOSHIMURA, K. KURASHIGE, and M. DEGUCHI, "An experimental study for development of multi-objective deep q-network–in case of behavior algorithm for resident tracking robot system–," in *Proc. of ICESS*, 2020, pp. 7–16.

[38] Y. Wang, L. Chen, H. Zhou, X. Zhou, Z. Zheng, Q. Zeng, L. Jiang, and L. Lu, "Flexible transmission network expansion planning based on dqn algorithm," *Energies*, vol. 14, no. 7, p. 1944, 2021.

[39] S. Lang, F. Behrendt, N. Lanzerath, T. Reggelin, and M. Müller, "Integration of deep reinforcement learning and discrete-event simulation for real-time scheduling of a flexible job shop production," in *2020 winter simulation conference (WSC)*. IEEE, 2020, pp. 3057–3068.

[40] B. A. Abdelghani, S. Fadal, S. Bedoor, and S. Banitaan, "Prediction of heart attacks using data mining techniques," in *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2022, pp. 951–956.

[41] S. Remya, "Heart disease prediction using cnn with various feature selection approaches," *Indian Journal Of Science And Technology*, vol. 17, no. 38, 2024.

[42] M. A. Bouqentar, O. Terrada, S. Hamida, S. Saleh, D. Lamrani, B. Cherradi, and A. Raihani, "Early heart disease prediction using feature engineering and machine learning algorithms," *Heliyon*, vol. 10, no. 19, 2024.