Mesopotamian journal of Cybersecurity
Vol.5,No.1, **pp**. 39–61
DOI: https://doi.org/10.58496/MJCS/2025/004; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

Data Mining and Machine Learning-Based Healthcare Monitoring in Cloud-IoT

Sarah Amer [1], Rania Hazim[1], Wassan Kader[1,*]

[1]*Department of Computer Engineering, University of Diyala, 32001 Diyala, Iraq.*

ARTICLE INFO

Article History
Received 01 Nov 2024
Accepted 03 Jan 2025
Published 02 Feb 2025

Keywords
Cloud
Machine learning
IoT devices
Healthcare
Data mining

ABSTRACT

Healthcare monitoring Cloud-IoT systems use data mining and machine learning methods to analyse patient data in real-time from linked devices. By offering insights for the early diagnosis of anomalies and individualized treatment suggestions, this strategy improves healthcare management. In this research first the Collect and Load the Clevant Heart Disease Dataset for Data Collection Process. Next, preprocess the loaded data using the Synthetic Minority Oversampling Technique (SMOTE), and then the feature extraction process is done using the Principal Component Analysis (PCA) Method. In this case, the characteristic must be extracted by feeding a specific column. The classification procedure is then carried out using Generative Adversarial Networks (GAN) and an optimization approach called Adaptive Moment Estimation. This is where the model executes GAN operations, and the output will be produced. The data is then transferred to an edge-cloud environment to minimize storage problems and provide instant access to critical data. This process starts with the encryption and decryption of data using Homomorphic encryption with the Laplacian technique. In addition, have taken the generated values from the GAN network as original values and encrypt them using Homomorphic encryption with the Laplacian technique. Next, the routing process is done using the leach protocol to optimize energy consumption and communication efficiency. The leach protocol is used to route among the data to divide the data into clusters and perform energy consumption. Finally, the simulation of this research is conducted by Python – 3.9.6 network simulator, and the performance of the proposed model is estimated based on various performance metrics such as accuracy at 90%, precision at 94%, authentication time, throughput at 90%, and packet delivery ratio with 94% this demonstrated that the suggested effort produced better results both in terms of quantitative and qualitative aspects.

1. INTRODUCTION

One industry recognized as a critical component for national growth and development is the healthcare industry [1]. The effect of an Internet of Things (IoT)-based healthcare system on people and society is significant. It is important to the pharmaceutical business and scientific community. Because of their complexity and heterogeneity, data are more difficult to comprehend and investigate [2]. The advancements in science and technology have coincided with progress in the healthcare sector. The development of information and communication technology (ICT) has paved the way for creative solutions in a wide range of business sectors, such as logistics, transportation, healthcare, and agriculture. One significant factor propelling ICT technical growth is the IoT, which is guiding future industries toward automation and decentralized intelligence [3]. A certain number of servers that may be utilized by the demands of the client are safeguarded by the cloud computing provider. The growth of these devices has been very rapid, and they have significantly transformed people's daily lives at every level [4].

In genetic biology research, cloud computing has been shown to be a viable and affordable method for integrating and analysing enormous amounts of data [5]. Data processing in the cloud is virtualized and is a productive approach to control and data monitoring in real time. The link that entails data streams is what these industrial devices intend. Through the use of big data, new patterns or noteworthy trends are identified. A wide range of unstructured data sources creates challenges in integrating information for analysis. The amount of data generated by IoT-enabled devices is massive. The combination of cloud computing with these devices provides new networking, scalability, and storage possibilities. These devices' restricted processing power is insufficient to handle vast amounts of healthcare data. Numerous stakeholders obtain infrastructure services and applications via cloud computing [6]. Considering unique identities (UIDs) and the capacity to transmit data across a network without the need for people or human-to-computer contact, these devices are systems of interconnected computing devices.

*Corresponding author. Email: wasanmohammed_enge@uodiyala.edu.iq

The internet of Medical Things (IoMT) is the real-world integration of IoT-enabled devices with medical technologies. The IoT facilitates the remote examination of healthcare devices and app data by transferring them to medical IT servers. To address patients' medical conditions and assist them in avoiding any more dire situations, medical professionals may access patients' health information remotely in real time via any mobile application or online platform thanks to the IoMT [7]. This computing technique offers consumers software and infrastructure services in addition to the services that customers seek over the internet. Given the noteworthy expansion of cloud computing, the number of users and requests is rising quickly. Consequently, it is critical to increase the speed and precision of cloud computing [8]. Even though today's dynamic global society depends heavily on data generated by these device applications, effectively using this information is still difficult in the medical field [9]. IoMT gadget sensors, together with human contact with these sensors, are thought to be a major source of data from which machine learning (ML) algorithms may extract characteristics to identify and learn practical patterns. Many uses in healthcare and elder care, including identifying activities for medical evaluation, fall detection, anxiety detection, tracking one's fitness, vital sign monitoring, and illness diagnosis, may benefit greatly from its usage [10]. Systems based on IoT-enabled devices and machine learning are effective because of developments in sensing, processing, spectrum utilization, and ML. Microelectronic advancements have made small, inexpensive medical sensors conceivable, which has revolutionized medical services and made these solutions viable [11].

Many efforts have been made to provide patient data remotely without visiting hospitals because of recent advancements in wireless sensor networks and the Internet of Things. It helps experts decide the best course of action to take or dispatch a certain medical assistance team. The transfer of vital patient information in an emergency may have a major influence on patient survival. Cloud computing has revolutionized computation and storage, opening new avenues for innovation in Internet of Things-based health monitoring systems [12][20]. Cloud-based and IoT-enabled device apps perform better than do conventional apps in terms of precision and effectiveness. Among the applications based on the cloud and these device technologies were financial services, healthcare, and defense. Remote locations may access medical records via cloud-based device solutions. Healthcare apps promptly gather data and modify the severity of medical factors [13]. The robotics industry uses cutting-edge technology to increase the overall sector's economic competitiveness. In the automation sector, monitoring systems play a critical role in increasing productivity, cutting costs, providing early warning systems, forecasting illness, and many other functions. Systems for monitoring are integrated with new technologies such as ML, cloud computing, and the IoT to improve their performance [14]. Many wireless applications utilize large datasets and analyse them with advanced processing techniques for greater accuracy and efficiency [15].

The main purpose of data mining methods (such as association rule mining, classification, and clustering) is to analyse data and find hidden patterns. Large private or public organizations in related or unrelated fields used to work together to perform data mining on aggregated data from all cooperative organizations (or participants) to extract helpful information for shared advantages.

Because the collaborative players operate in a dispersed environment, each cooperative participant must exchange specific data to perform data mining [16][30]. The main goal of this research is to utilize data mining and ML algorithms to improve healthcare monitoring in the cloud-IoT, which can hinder the following issues:

- **Impact of Privacy and Security:** Existing methods have inadequate privacy and security measures in healthcare data that can lead to unauthorized access, breaches, and potential misuse of sensitive patient information.
- **Increased latency and bandwidth:** The existing method does not minimize the high latency and bandwidth issues that can impede timely access to critical patient information and real-time communication among healthcare professionals.
- **Insufficient Storage:** Previous methods have inadequate storage capacity that poses challenges in managing the continuous influx of patient data, limiting historical records and hindering comprehensive longitudinal analysis.
- **Imbalanced class and data annotation:** Class imbalance in healthcare datasets and a shortage of annotated data for specialized conditions hinder the development of accurate and inclusive machine learning models.

2. RELATED WORKS

This section presents a literature review on the use of data mining and machines for healthcare monitoring in the cloud-IoT. The authors of [17] utilized “IoT, fog, and cloud technologies”, and the heart disease detection system was intelligent and effective. The obtained healthcare IoT data are preprocessed via a fuzzy inference system and filtering technique. The deep learning-recurrent neural network **(**DL-RNN) model of the “gated recurrent unit (GRU)” is then used at the fog layer for forecasting. In contrast to the outcomes of the GRU method, the recommended fuzzy inference system with an upgraded GRU exactly forecasts the danger of a heart attack from data from IoT-enabled device patients and electronic health records (EHRs). Moreover, fog computing increases the danger of security and illegal access because data are processed and kept across several edge devices. This can be avoided by setting up fog nodes with tight access control guidelines, hardware that is impossible to tamper with, and protection against physical damage and site-based attacks. Methods for authentication

and trust that were created before the emergence of fog nodes and heterogeneous device nodes are outdated. Therefore, a new framework for users, services, node authentication, and trust needs to be created. A secure way to offload tasks while guaranteeing their accuracy and integrity is required since offloading work to fog nodes has the potential to harm personal data. The user can access several fog nodes that might be sensitive. It is essential to safeguard private information confidentiality via proper privacy-preserving measures.

The authors of [18] reported that the data of cardiac patients are handled by a health observing system built on AI and these devices. The heart patient's behaviors are tracked by the system, which helps patients remain cognizant of their state. Moreover, the scheme can use ML algorithms to categorize diseases. However, the primary research constraint is that the suggested system gathers information from various sources and transfers it to the cloud for additional processing. Different useful healthcare technologies that are controlled by electronic devices such as phones and tablets, which are popular among medicinal professionals, can be added to these device-based systems to increase their capacity. These gadgets only have basic computing power and can store data locally. These devices also have poor security, which puts patient data privacy and confidentiality at risk. IoT-enabled devices that are worn on the body or implanted can monitor patients continuously and help identify potential health problems early on.

The authors of [19] proposed a prediction framework that uses ML techniques to monitor the real-time data of sensor nodes in a medical background. An IoT-enabled device smart hospital environment has been created that uses various sensors, including air quality, temperature, humidity, flame, and current sensors, to monitor and operate appliances via the internet. Three key features of these device-generated sensor data are their massive number, organized nature, and real-time nature. Predicting early defects in an IoT context is the primary goal of this research to guarantee the correctness, fidelity, dependability, and integrity of devices that are enabled by these devices. “Using a decision tree, K-nearest neighbor, Gaussian naive Bayes, and RF methodologies”, the suggested error estimate model was assessed; however, on the given dataset, RF demonstrated the maximum accuracy compared with the other methods. The outcomes demonstrated the effectiveness of using machine learning techniques on IoT-based sensors to display the hospital automation method. Among these techniques, random forest was shown to have the highest accuracy. The suggested model could be useful to the user in deciding on the suggested course of action and managing unforeseen losses caused by errors made throughout the automated process. However, security becomes a significant concern when there are several IoT devices, so it is important to consider their security.

The authors of [21] proposed a lightweight authentication method employing completely homomorphic encryption with a privacy-preserving schema. This method allows for safe online access to patient data and the sharing of that data in an encrypted format with stakeholders for various reasons. The suggested authentication method for the internet of Medical Things is simple to use and resistant to any network assault. However, a significant drawback of the suggested authentication method is its high message transfer rate. This raises the price of communication. It can be decreased in the future by the use of more potent challenge–response techniques.

The authors of [22] addresses related security issues, and this article suggested an authentication mechanism for wireless body area networks that employs certificate-less cryptography. Burrows–Abadi–Needham logic is used in a formal security study, which demonstrates the suggested protocol's resistance to common assaults. Furthermore, they use the Automatic Verification Security Protocol and evaluation model tool for safety exploration and the real-or-random model for mathematical evidence. The suggested approach is more functional and less expensive than the current protocols are, according to a thorough analysis. However, even under the best circumstances, their plan involves using a centralized server, which introduces latency.

The authors in [23] raised the bar for ensuring better healthcare services while also achieving the visualization and accountability of healthcare users," They attempted to highlight in this article the importance of machine learning-based IoT devices for patient monitoring in cloud environments. Additionally, they reported that patients’ level of proficiency using the platform was high in emergencies in particular and nearly the same in all other circumstances. As a result, their technology was more accurate in every usability test on the basis of patients' experiences, especially in emergencies. The limitation of this work is that despite the benefits, there are trust, privacy, and security concerns. These challenges must be resolved earlier, so healthcare suppliers and actors can completely embrace the suggested system.

The authors of [24] presented a "cross-architecture IoMT malware detection and classification system based on byte sequences extracted from Executable and Linkable Format", formerly termed ‘Extensible Linking Format files use an attention-based multidimensional DL technique’. The DL approach streamlines the task of developing characteristics and retrieving them from unorganized byte patterns. Furthermore, the recommendation process makes it easier to classify the ELF file's CPU architecture.

The authors of [25] explored various FL programs in diverse IoT domains, such as clever towns, healthcare, commercial IoT, and clever grid systems. It investigates how FL can cope with the demanding situations posed via the allotted nature of IoT information, which includes statistics heterogeneity, privacy issues, and communication constraints. A large portion of the survey is dedicated to analysing the methodologies and algorithms used in federated mastering for dispensed

selection-making in these devices. This encompasses a dialogue on federated optimization techniques, communique-efficient algorithms, and privacy-retaining mechanisms. The survey also delves into the function of side computing in facilitating efficient FL in IoT-enabled devices, considering the resource constraints inherent in facet gadgets. Furthermore, the paper reviews the contemporary modern-day federated learning frameworks and systems tailor-made for these device environments. It evaluates their ability to cope with real-global challenges and provides scalable solutions for disbursed decision-making. The survey concludes by identifying open research challenges and potential avenues for future traits in federated learning for these devices, emphasizing the need for novel algorithms, robust safety features, and standardized frameworks.

This paper improves on prior research by addressing fundamental obstacles in healthcare monitoring systems that combine cloud computing and the IoT. Previous research has shown the promise of the IoT and cloud technologies in improving healthcare services but has faced challenges such as a lack of privacy and security, high latency, limited storage, and dataset imbalances. To address these restrictions, this study presents multiple sophisticated methods, such as homomorphic encryption utilizing the Laplacian technique for strong data protection, the synthetic minority oversampling technique and principal component analysis for managing class imbalances and extracting features, and the merging of edge-cloud architectures to increase data transmission and storage efficiency. Furthermore, integrating generative adversarial networks with adaptive moment estimation optimization improves the precision and dependability of machine learning models, outperforming conventional methods. The significance of this study is its ability to offer a healthcare monitoring system that is more secure, efficient, and scalable, guaranteeing real-time management of patient data with the utmost privacy and accuracy, which is crucial in today's healthcare settings.

3. PROPOSED METHOD

The main purpose of the proposed method is to use ML and data mining techniques to enhance the overall performance of the cloud-based healthcare monitoring system. Fig. 1 illustrates the overall architecture of the proposed architecture. Important steps are taken as part of this process, including the following:

- Data collection
- Pre-Processing
- Classification
- Data transmission in the edge-Cloud
  - Authentication
  - Routing
  - Storage
- Healthcare monitoring in the IoT

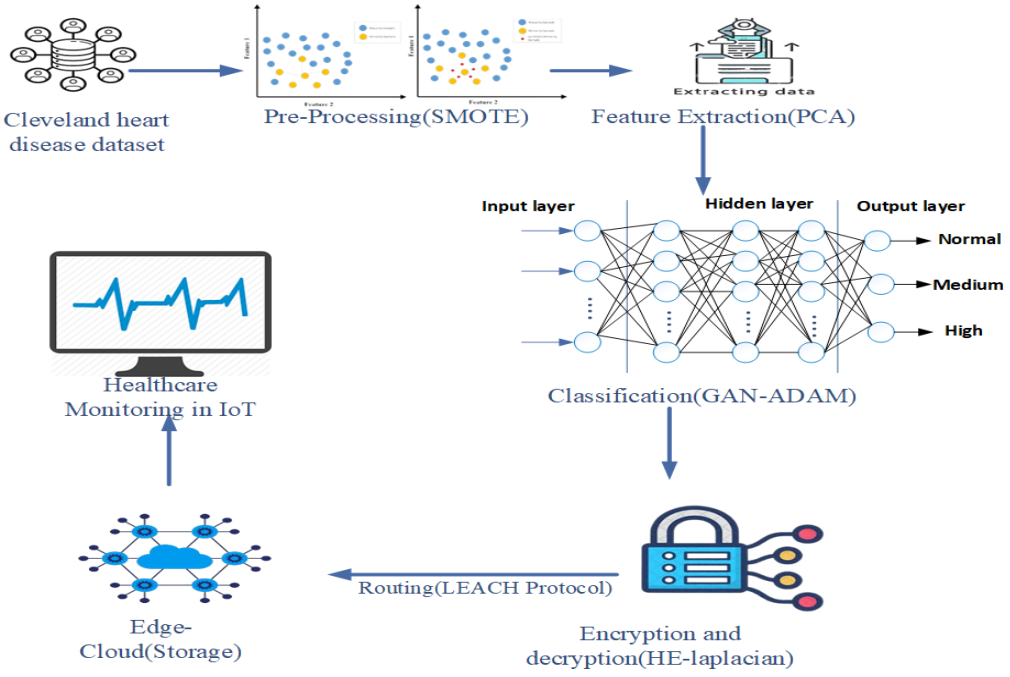

Fig 1. Overall architecture of the proposed architecture.

3.1 Data Collection

In the beginning stage, first, the **Cleveland heart disease dataset,** which is a large and broad repository that includes 303 patient records and 13 variables, is gathered. Additional investigations and ML applications in the field of cardiovascular health are provided by this dataset. Table 1. Represents the data attributes.

TABLE I. DATASET ATTRIBUTES

S.no	Attributes	Attributes size
1	Dataset size	303
2	Number of features	13
3	Distribution of positive case	70-30(47.25%), 80-20(49.18%)
4	Distribution of negative case	70-30(52.75%), 80-20(50.82%)

3.2 Pre-Processing

After the dataset is collected, the SMOTE is used to address the class imbalance in the dataset. Then, to enhance minority class illustration, this technique is used to produce a more robust and balanced dataset. After preprocessing, the features extracted via principal component analysis (PCA) were “age, sex, chest pain type, resting blood pressure, cholesterol, fasting blood sugar, resting electrocardiography, maximum heart rate achieved, exercised-induced angina, old peak, slope, num, thal and ca”**.**

3.2.1 Synthetic minority oversampling technique (SMOTE)

To overcome the class imbalance dataset, the SMOTE**.** To enhance minority class representation, this technique produces a more robust and balanced dataset.

It is one of the most exaggerated techniques. This technique creates a synthetic minority class sample generated along the line that connects one of its neighbors "k–nearest neighbor", which is a member of the minority class samples with randomly selected minority class samples. SMOTE employs a regressive approach to selection and search. Among the k closest neighbors, a threshold number of samples is chosen to create new fake minority class samples. The number of synthetic minority samples that must be produced determines the threshold's value. The procedure is repeated until the necessary number of synthetic minority class samples is produced. The SMOTE method artificially generates additional minority class samples in the space of features rather than the information space to equalize the distribution of classes. This expands the minority class's decision-making space. A new synthetic minority class sample is formed in SMOTE and is located on the line segment between y_i And $\bar{y}$ here$y_i, \bar{y} \in M_{min}$ can be described as,

$$y_{syn} = y_i + (\bar{y} - y_i).\times rand(0,1) \quad (1)$$

Where,

y_i is the minority class sample, which is to be oversampled

where $\bar{y}$ is another minority sample that is usually designated from the M_{min} samples near y_i.

The symbol .× represents elementwise multiplication

$rand(0,1)$ Specifies a random number within the interval $(0,1)$.

Despite its ability to alleviate class imbalance by creating artificial samples, SMOTE may have certain disadvantages that can affect the effectiveness of the model. A potential issue is that SMOTE can add noise by generating artificial instances that do not truly represent the actual distribution of the minority category, resulting in artificial data and the possibility of excessive fitting. This could decrease the model's ability to accurately predict outcomes on unseen data. To address these problems, SMOTE can be coupled with methods such as Tomek links or edited nearest neighbors (ENNs) to eliminate noisy or duplicate samples. Furthermore, fine-tuning SMOTE settings, including the quantity of nearest neighbors employed, can assist in producing a wider variety of highly realistic artificial examples. By recognizing and addressing these possible downsides, this research offers a more even and sturdy method for managing unbalanced datasets.

3.2.2 Principal component analysis

After the balanced datasets, the features are extracted via PCA, and then, the features of fasting resting electrocardiography, maximum heart rate achieved, exercise-induced angina, old peak, slope, num, thal, and ca”. The selected characteristics, such as age, sex, type of chest pain, resting blood pressure, and cholesterol levels, are all important clinical markers of heart health. All of these characteristics are strongly associated with risk factors for heart disease, which is crucial for developing accurate prediction models. By addressing the importance of these characteristics, the authors can show that the selection was not random but rather influenced by their established link to heart disease results. This explanation enhances the section

by demonstrating that the selected features for PCA are not only important for reducing dimensions but also necessary for creating a model that accurately identifies the key predictors of heart disease.
The goal of this is to create a new feature set with fewer dimensions than the original dataset. In doing so, a D-dimensional dataset would be changed into a new, lesser D-dimensional dataset. where$d \ <= D$.

Consider$D - dimensional\ dataset$

$$y = (y_1 , y_2 , y_3 , \dots . , y_M) \tag{2}$$

The following procedures are used to reduce the dimensions of the data via PCA:
The first step computes the mean of Y via the following formula:

$$(y) = \frac{1}{M} \sum_{j=1}^{M} . (yj) \tag{3}$$

This will support both the covariance calculation and data standardization. To permit an output free from bias, standardization scales the data such that the variables and values fall within a specified range.
The second step computes the covariance matrix as

$$Cov(y) = \frac{1}{M} \sum_{y=1}^{M} . (yj - yj')(yj - yj')S \tag{4}$$

To determine the dependencies and correlations between the features, the covariance matrix is used. The last phase is the spectral decomposition of the covariance matrix via eigenvectors $\xi_1, \xi_2, \dots . , \xi_E$ and eigenvalues $\mu_1, \mu_2, \dots , \mu_E$. The eigenvalues are sorted as$\mu_1 >= \mu_2 > = \cdots > = \mu_2$
This gives:

$$Z = (z_1, z_2, z_3, \dots . , z_q) \tag{5}$$

In this way, Z has the primary components and is the lower d-dimensional dataset. This is provided by the following formula:

$$(Z = (\xi S1(yj - yj'), \xi S2(yj - yj'), \xi S3(yj - yj'), \dots . , \xi Se(yj - yj'))S) \tag{6}$$

Z, which contains primary components, is the new dimensional representation for the original dataset Y.
Upon conducting PCA on the balanced dataset, which contains important attributes such as age, sex, and type of chest pain, the dimensionality decreases from D to a lower-dimensional form. This procedure includes finding the average for normalization, determining the covariance matrix to study feature relationships, and conducting spectral decomposition with eigenvectors and eigenvalues to detect the main components. Nevertheless, PCA has drawbacks such as being sensitive to outliers and assuming linear relationships between features, leading to potential impacts on its usefulness and the precision of the model produced.

3.3 Classification

GANs can be utilized to generate realistic and diverse samples of the data distribution, improving the model's ability to understand and classify different patterns in the dataset. “Adam is an optimization algorithm” that familiarizes the learning rates for each parameter through training. Adam optimization improves the effectiveness of the training process for these devices, enabling quicker convergence and improved handling of the complex relationships within the data. Combining generative adversarial networks with adaptive moment estimation optimization involves integrating the adversarial training of GANs with the adaptive learning rates provided by the Adam optimization algorithm. This synergy enhances the overall performance of these models for data classification. These methods produce diverse and realistic illustrations of different cardiac health types, and Adam optimization safeguards effective training, resulting in a more sophisticated and precise classification of data into high, normal, and medium classes.

3.3.1 Generative adversarial networks

GANs can be exploited to produce realistic and diverse samples for data delivery, improving the framework's ability to identify the variability of patterns in the dataset. Fig. 2 shows a block diagram of these devices. The figure of the generative adversarial network illustrates how the generator and discriminator components cooperate. The generator produces artificial data samples that emulate genuine data, whereas the discriminator compares these samples with correct data and discriminates among true and generated inputs. While in training, the generator struggles to improve its results to deceive

the discriminator, which in turn continues to enhance its ability to identify counterfeit samples. This oppositional technique leads to the generator making increasingly more true data as time progresses. The figure illustrates how the generator's results are input into the discriminator, which in turn gives input back to the generator, enabling an ongoing process of learning and enhancement.

Exploiting these models in classification tasks can face several challenges, such as mode collapse, leading to a reduction in the variability of produced samples, and training instability, which may hinder effective learning. Moreover, GANs frequently request significant computational resources and current problems in the evaluation and optimization of hyperparameters, which could influence overall efficiency.

The two networks that make up these devices are the generator and discriminator. Both networks work together and compete with each other at the same time. The discriminator network can recognize the phony data that look genuine after a sizable number of training cycles, and the "generator network" can create fake data that are real. Once false data are generated, training and prediction can be performed on real data. The generator network is divided into three levels: the first "hidden layer", which has "21 neurons and an Elu activation function" specified in equation (7), receives "input random noise".

The second "hidden layer" is the next layer. It has "24 neurons and an Elu activation function". The last layer and the output layer are the generator network that calculates the fictitious samples. It has 25 neurons in the layer, and their function is sigmoid activation. Once the false samples are created, they are put into the “discriminator network”, which likewise has 3 unique layers, and added to the genuine dataset.

$$z = \begin{cases} y & when \quad y \geq 0 \\ \beta(e^{y} - 1) & when \quad y < 0 \end{cases} \tag{7}$$

where β is a variable that may be used to control the point of saturation of the negative Elu section.

The “discriminator network”, which is also made up of 3 separate layers, receives both the real dataset and the bogus sample that was constructed from 16 neurons in the first buried layers and is triggered by an “Elu activation function”. The 2nd “hidden layer”, which has eight neurons overall and is also triggered by an “Elu activation function”, comes next. One neuron with a “sigmoid activation function” that can discriminate between actual and bogus inputs makes up the final output layer. The discriminator's output is utilized to calculate the “loss function that these models” employ. As a result, the generator's parameters update more slowly, whereas the discriminator's parameters update more quickly. The generator may create a fresh realistic dataset, and the discriminator can no longer distinguish between real and fake data after the generator and discriminator have been trained for a particular number of periods.

In the training procedure of generative adversarial networks, the generator and discriminator systems are adjusted to improve their effectiveness. Typically, the discriminator is updated more often than the generator is, sometimes multiple times per generator update. The purpose of this is that the discriminator obligation excels at discriminating between real and fake samples to efficiently lead the generator. The updates it receives are determined by a loss function that assesses its exactness in correctly classifying real and synthetic information. On the other hand, the generator is usually only updated once after several updates to the discriminator. The updates of the generator are focused on the feedback from the discriminator to make data that are problematic for the discriminator to distinguish from real data. This input is employed to change the generator's scenery to improve the authenticity of the produced samples. The gradients calculated from the loss functions of every system guide the updates, allowing the discriminator to increase its classification ability and the generator to improve the quality of its outputs.

3.3.2 Adaptive moment estimation optimization

“Adam is an optimization algorithm” that adapts the learning rates for each parameter through training. Adam optimization improves the effectiveness of the training process for the GAN-enabled model, enabling quicker convergence and improved handling of the complex relationships within the data.

It is an optimization algorithm that uses adaptive estimates of lower-order moments and makes use of “stochastic gradient data (GD)” for the objective function. By combining the gains of “AdaGrad and RMSProp”, the Adam method calculates the learning rates from the approximations of the 1st and 2nd gradient moments. One obtains the initial momentum, or mean, as follows:

$$n_j = \beta_1 n_{j-1} + (1 - \beta_1)\frac{\partial C}{\partial W} \tag{8}$$

The second momentum is computed as follows:

$$\gamma_j = \beta_2 \gamma_{j-1} + (1 - \beta_2)\left(\frac{\partial C}{\partial W}\right)^2 \tag{9}$$

β_1, β_2 represents the mean speed at which the momentum moves.
$\frac{\partial C}{\partial W}$ is the weight-dependent cost function with parameter w.
n_j *and* γ_j are biased near 0 when β_1, β_2 is nearly one.
The Adam method takes advantage of the "corrected bias" estimate of the 1st and 2nd moments in the following ways to address these biases:

$$\hat{n}_j = {}^{n_j}/_{(1-\beta_1)} \tag{10}$$

$$\hat{\gamma}_j = {}^{\gamma_j}/_{(1-\beta_2)} \tag{11}$$

The Adam update rule makes use of these instances in the following ways:

$$W_{j+1} = W_j - \alpha \frac{\hat{n}_j}{\sqrt{\hat{\gamma}_j + \varepsilon}} \tag{12}$$

where ε is utilized to prevent division into zero scenarios and where α is the learning rate.

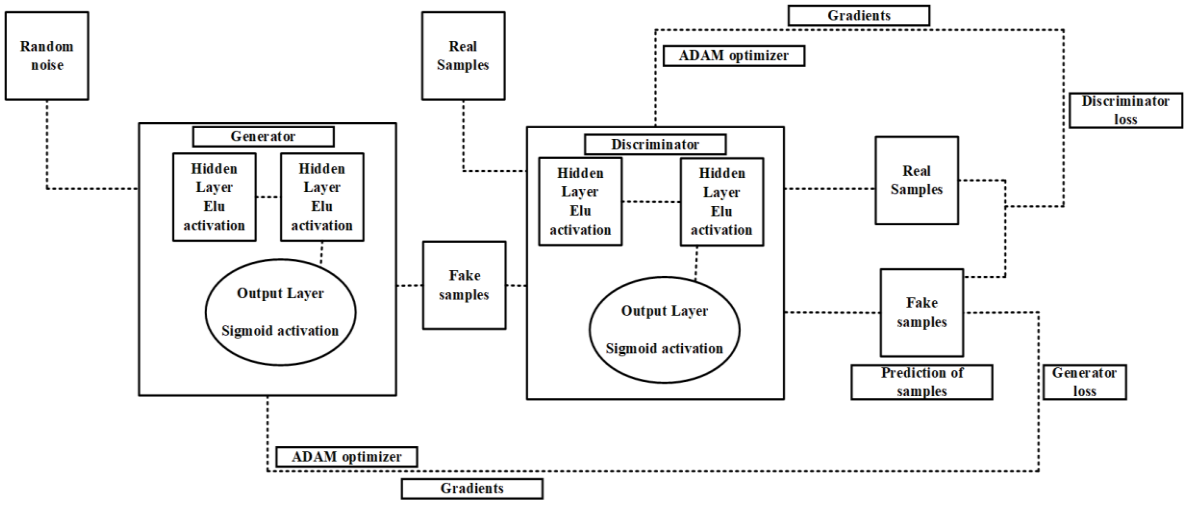

Fig 2. Block diagram of the GAN.

4. DATA TRANSMISSION AND STORAGE IN EDGE CLOUD

By utilizing a smooth integration of **edge computing and cloud storage**, the classified data are effectively sent to the edge-cloud environment. This reduces storage issues and guarantees immediate access to vital information. By using an edge-cloud architecture for storage, the strain on centralized cloud systems is lessened. This method combines the scalability of cloud storage with the real-time processing performance of edge devices to maximize storage for the constant influx of patient data.
To lessen the strain on the cloud master's station processing and storage, the cloud edge collaboration structure is suggested. Situated near a data source, the edge layer is a networked intelligent agent and offers local or nearby intelligent decision-making and services. The goal of this study is to improve cloud collaboration by breaking down the edge computation layer into three sublayers: "edge computing software as a service (EC-SaaS)", "edge computing platform as a service (EC-PaaS)", and "edge computing infrastructure as a service (EC-IaaS)", as shown in Fig. 3.
The fundamental open platform, or EC-IaaS layer, consists of AI, storage, exchange of information, and system service abilities in addition to the operating system, hardware, and container and communication openness. EC-PaaS realizes the administration and use of applications and offers a backplane for all kinds of operating software. Simultaneously, to fulfil the technical "plug-and-play" prerequisites in the healthcare system (HS) apparatus, the "plug-and-play service" is used as a "software layer" that serves as the foundation for further applications. The Data Centre, a crucial part of EC-PaaS, is intimately linked to the gathering, processing, transfer, and computation of data as well as other HS business operations. Applications are developed and deployed by the EC-Sass layer to integrate healthcare system (HIS) business requirements.

It is a particular application of HIS edge computing technology that provides “data proxy services” to facilitate seamless data interchange support operation and maintenance management through the Cloud-Edge-Client (CEC) framework. The “state perception and execution control unit” of the HS is referred to as the terminal layer. Sensing technology is used to track, gather, and interpret fundamental healthcare data, including wearable devices, gear status, and IHS equipment data. Strong and reliable organizational support for the administration and operation of the HS is provided by the cloud-management-edge-terminal design, which also offers flexible flexibility to variations in “internal and external needs”. When used in conjunction with the CEC platform, real-time processing of HS transactions and resource scheduling enable the achievement of an active sense of a patient's physical state. Furthermore, the CEC platform expedites the realization of serviceable transformation and business change in an economical way and enhances the efficiency of construction and maintenance.

The system consists of a host computer, an edge gateway, and a series of edge devices. The host computer is used to activate the edge and gateway devices so that they can process the input feature map. “Edge devices” and “Edge gateway” devices are linked to exchange information with one another.

All the IoT gadgets in the network are denoted as E, and D signifies the exchange of information edges between the devices: EE represents the edge device, and GE represents the edge gateway.

This is presented as

The overall network is expressed as:

$$M = [E, D] \tag{13}$$

This equation illustrates the holistic structure of the edge-cloud system. It shows the hierarchical layout, which consists of the host computer, edge gateway, and edge devices. The primary computer is in charge of organizing tasks, even as the threshold gateway handles neighborhood processing and device conversation. The edge devices perform real-time sensing and data gathering.

A group of linked devices is verified as:

$$E = [EE_i, GE_i, Host] \tag{14}$$

The network's communication edges are denoted by D, and the “number of IoT devices and edge computing devices” determines the value of |D|. E represents all IoT devices in the system. This equation highlights the scale of the device by depicting the general quantity of interconnected devices. Comprehending this organization is essential for comparing the network's ability and effectiveness.

The maximum values of |D| are provided as follows in the equation if |E| = n.

N is the number of devices

$$|D| = 0 < |\mathrm{D}| < \frac{n(n-1)}{2} \tag{15}$$

The equation provides an understanding of how linked the network is and how much information is shared between devices. It aids in understanding the level of connectivity between devices and the possibility of communication delays.

Let us denote the latency as δ , which includes the computational latency (δ_{cl}) and network latency(δ_{nl})

$$\delta = \delta_{cl} + \delta_{nl} \tag{16}$$

This equation separates latency into two parts: computational and network aspects. It aids in pinpointing the origins of time wasted in the system. Computational latency involves the duration for which devices handle data, whereas network latency refers to the pauses in data transmission.

In this approach, the network latency (δ_{nl}) is zero, as no data are sent to the cloud network. Therefore, the system latency

$$\delta = \delta_{cl} \tag{17}$$

The computation latency (δ_{cl}) of IoT devices is mostly determined by how the input is processed, which in turn affects how busy each processor is. In this particular method, network latency is eliminated because no data are transmitted to the cloud network. Hence, the delay is solely related to computations. This streamlining highlights the effectiveness of the edge-cloud system, which reduces the communication lag.

Estimated at time t, the computational latency is computed as follows, taking into account the job assignment indication(β).

$$\delta_{cl} = \beta\left(\frac{zi}{yi}\right) \tag{18}$$

where zi is the computational complexity of k and where yi is the computational capability of k. This formula calculates the processing delay by taking into account the work to be done and the capabilities of the processor. It shows how the time

taken to process varies on the basis of the workload and the power of the devices being used. This assists in assessing how well edge devices perform in different scenarios.
Therefore, the estimate of inference latency δ_{il} for the IoT edge network may be found by:

$$\delta_{il} = \delta_{cl} \tag{19}$$

Computational latency impacts inference latency, which is affected by input processing and device workload. Understanding how fast the system can deliver real-time insights and responses relies on this equation.
This can reveal how effectively the suggested framework functions with real-time intelligent devices.

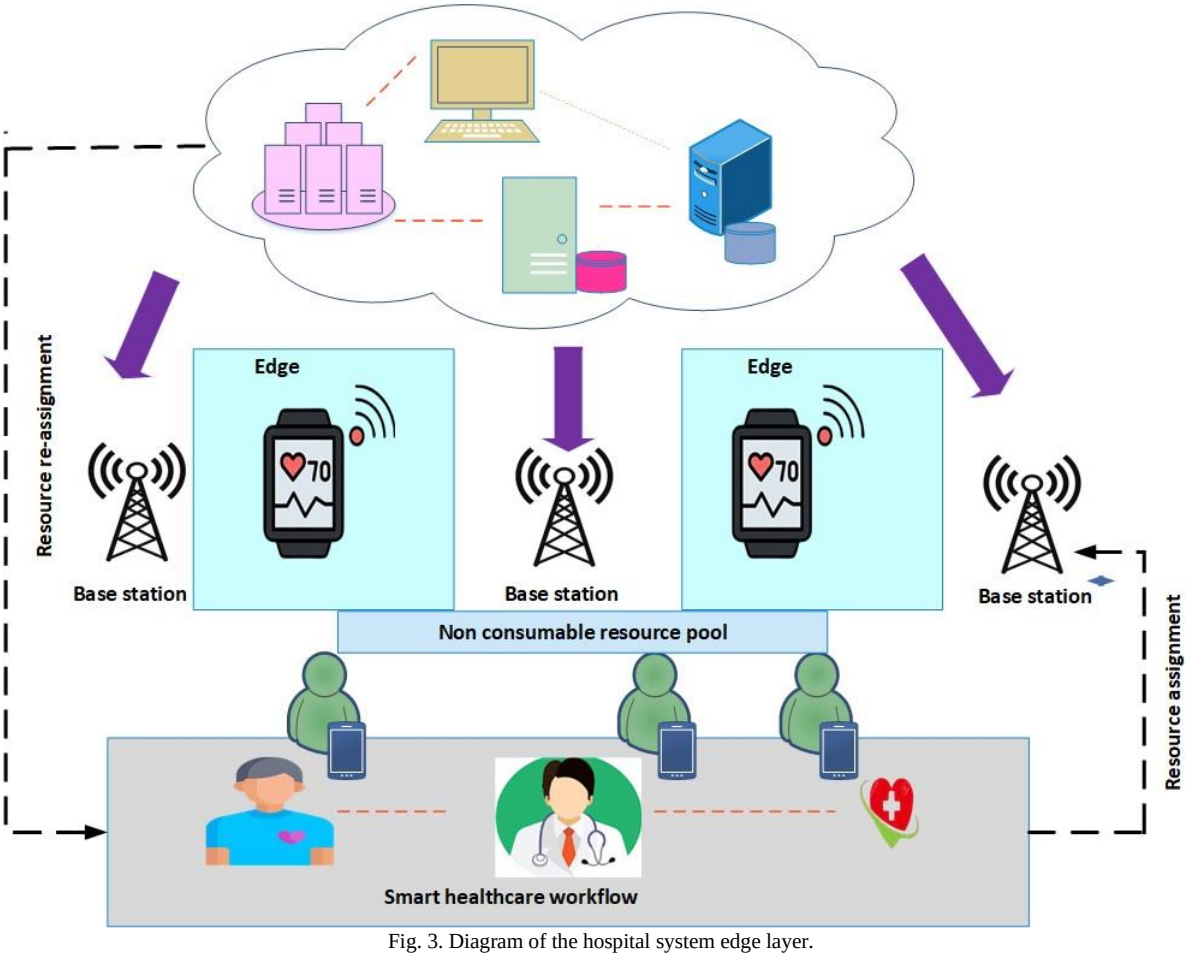

Fig. 3. Diagram of the hospital system edge layer.

4.1 Encryption and Decryption of Data

Homomorphic encryption with the Laplacian technique is used to strengthen data security during transmission. By protecting patient privacy and preventing illegal access, this cryptographic method guarantees the security of critical healthcare data.

4.1.1 Homomorphic encryption

A cryptographic technique known as homomorphic encryption enables cipher texts to be arithmetized by third parties without the need for decryption. It operates on clear text communications and yields the same outcome as encrypting.
In formal terms, an encryption system is deemed homomorphic over an operation $*$ if it is capable of supporting the following attributes:

$$E(n_1) * E(n_2) = E(n_1 * n_2) \tag{20}$$

Where n_1, n_2 belongs to N, the set of all possible messages, and E is the encryption technique.

The four algorithms that make up an HE scheme are KeyGen, Enc, Dec, and Eval. For the asymmetric configuration, "KeyGen" makes a pair (public key, private key), whereas for the symmetric version, it generates a secret key. The decryption algorithm is called Dec, whereas the encryption method is called Enc. Conventional cryptosystems use the three algorithms KeyGen, Enc, and Dec; however, for homomorphic encryption schemes, an extra method known as the Eval algorithm is needed. The algorithm is defined as follows:

$$Eval\ (f, G_1, G_2) = f(n_1, n_2) \tag{21}$$

Where f is a function that may be added or multiplied and where Dec (G_1) = m1 and Dec (G_1) =n_2.
He may be divided into three categories: partially homomorphic encryption (PHE), somewhat homomorphic encryption (SWHE), and fully homomorphic encryption (FHE). These categories are based on the quantity (limited or limitless) and the kind of operation "(addition or multiplication)". PHE permits only one type of operation to be carried out indefinitely. Refer to encryption as additive homomorphic encryption (AHE) when the operation involves addition, as in the "Paillier scheme". A scheme is multiplicative when it involves multiplication, as in the case of the RSA scheme. The SWHE permits both kinds of operations, but only a certain number of them. However, FHE permits an infinite number of both kinds of operations. The overall system performance can be greatly affected by the computational expenses of homomorphic encryption (HE). Generating keys, encrypting, and decrypting in HE can be more resource intensive than traditional cryptography because of the intricate number-theoretic algorithms and the necessity of upholding homomorphic properties. The Eval algorithm is very intense, especially in fully homomorphic encryption systems that allow both addition and multiplication, as it computes encrypted data. The added computational load may result in extended processing durations, increased resource usage, and decreased throughput, impacting system responsiveness and scalability. To reduce these effects, optimized schemes using homomorphic encryption, hardware acceleration, a combination of different methods, and effective algorithms can be used to maintain a balance between security and performance, ensuring that the system meets its operational needs.

4.1.2 Laplace mechanism

To ensure (ϑ, γ)-DP, the Laplacian mechanism that adds noise samples from $L\ (0, \alpha)$ must meet the required and sufficient conditions$\left[1 - exp\left(\frac{1}{2}\ \left(\vartheta - {}^{\Delta_1}/_{\alpha}\right)\right)\right]_+ \leq \gamma$. As seen in the following Lemma, the Laplace mechanism, in contrast to the Gaussian mechanism, is also capable of guaranteeing pure DP because of the exponential tails of the noise distribution.

Lemma: The Laplace technique ensures that γ is differentially private for $\gamma \geq {}^{\Delta_1}/_{\alpha}$, where Δ_1 is the query's l_1 sensitivity, by adding K-independent noise samples from $L\ (0, \alpha)$ to each coordinate of the query answer.

Hence, the lowest amount of noise required for γ-DP is represented by the Laplace noise of scale ${}^{\Delta_1}/_{\gamma}$. Notably, the conventional Laplace processes introduce noise by relying on only one sensitivity measure, failing to consider the possibility of varying sensitivity for each query answer coordinate.

4.2 Routing

After encryption and decryption in IoT-enabled healthcare monitoring, the "Low Energy Adaptive Clustering Hierarchy (LEACH) protocol" is used for effective data routing. Although emphasis is placed on cluster heads and the importance of energy levels, a more organized explanation of the implementation of the LEACH protocol would enhance understanding. The procedure starts by randomly choosing a cluster head (CH) depending on the node energy levels and Euclidean distance. The energy needed for the CH function is calculated to ensure that the node can manage the workload. The nodes closest to the CH are organized on the basis of their Euclidean distance, and their energy levels are monitored continuously. If the CH runs out, a fresh CH is chosen, and the cycle continues, improving energy efficiency and guaranteeing effective data transmission in the IoT-supported healthcare monitoring system. By optimizing energy consumption and communication efficiency, this protocol increases the network's data transmission reliability. Fig. 4 represents the clustering process.

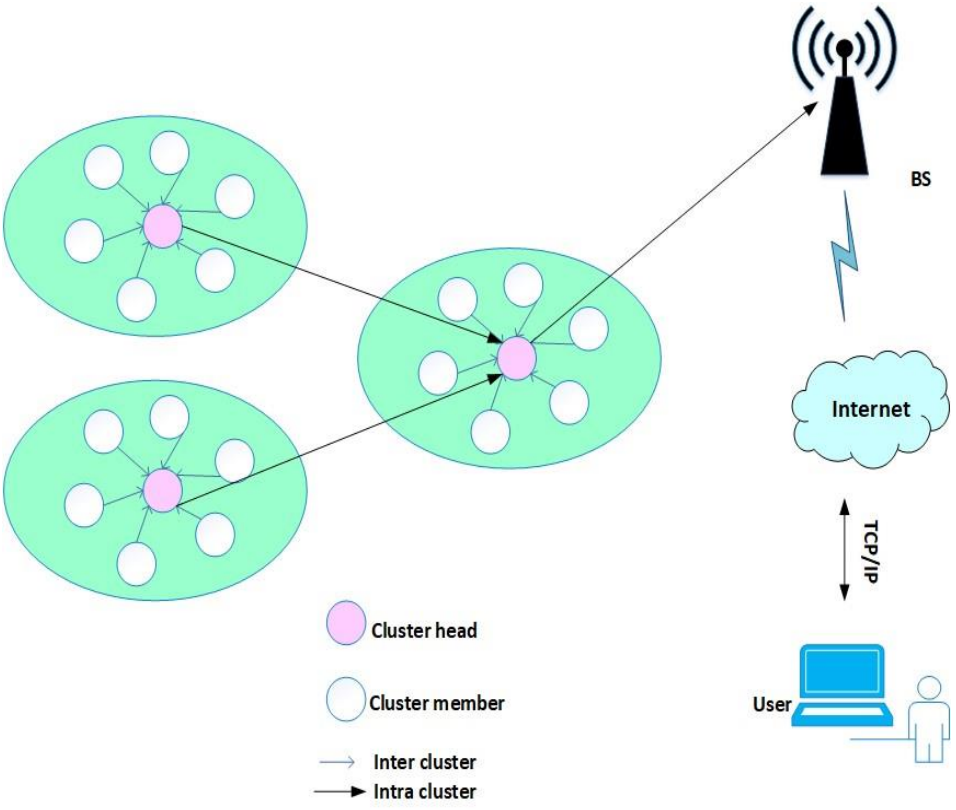

Fig. 4. Clustering process.

- ***LEACH protocol***

The weights given to the nodes determine who is the cluster leader. The weight of each node is determined by the specific energy level of the sensor node. Thus, in addition to the Euclidean distance, the cluster head must also meet the energy weightage requirement. Only when the cluster head's weight exceeds the predetermined threshold is it chosen. The primary goal of adjusting the node's weight is to enable it to oversee the cluster head's workload. It is necessary to compute the energy needed to convey the combined data.
The energy that the cluster head needs is

$$F_{CH} = m\left(K * \left(F_{elec} + F_{gt} * c^2\right)\right) \quad for\ c < 0,$$
$$F_{CH} = m\left(K * \left(F_{elec} + F_{amp} * c^4\right)\right) \quad for\ c \geq 0 \quad . \tag{22}$$

In this case, F_{CH} represents the energy used by the cluster head n denotes the number of nodes allocated to the cluster, k represents the message bits, F_{elec} signifies the energy needed to send and receive the data bit, F_{gt} and F_{amp} denote the parameters for computing the L-bit message when transmitting over free space multipath propagation, and c indicates the transmitting distance towards the sink node.
Using the k-means method, the cluster head selection may be represented by the following equation:

$$G = \sum_d^n \sum_{z \in f_d} (z_i - h_d)^2 \tag{23}$$

Z is a cluster mote, h is the head to be chosen, d is the "number of clusters", and G is the function of the k-means algorithm. The Euclidean distance can be given by

$$c\{z_i - h_d\} = (z_i - h_d)^2 \tag{24}$$

where c is the Euclidean distance of the nodes in the cluster.
Let $y = \{y_1, y_2, \ldots., y_n\}$ be the quantity of sensors placed across the network.

LEACH protocol algorithm

1. Determine how much energy the node needs to function as the CH.
2. First, choose the CH at random from the available nodes.
3. The following action repeatedly determines the Euclidean distance between the nodes.
4. Now, equation (24) is used to choose the nodes that have the same Euclidean distance.
5. Check the node's energy level after that; it should be able to perform the CH task.
6. Choose it to be the cluster head if its energy can function as one; in this case, $\geq F_{CH}$.
7. Reject it if not.
8. Now, add the nodes for that specific set of rounds to that specific CH.
9. Several rounds are completed or the procedure is terminated if the CH is lethal.
10. Send a request message once again to modify the nodes' energy level and Euclidean distance.
11. Continue from 3 until every node is deceased.

The outcome of the entire procedure is an extensive Internet of Things healthcare monitoring system. To offer effective and timely patient care, the combined technology allows on-demand access to stored data, real-time classification variations, and rapid responses to healthcare notices. The choice technique in the LEACH protocol is intricately planned to increase energy allocation and prolong the network's lifetime. The primary step's random selection of CHs is vital, as it prevents specific nodes from continually carrying the energy load, thus safeguarding a balanced distribution of energy consumption throughout the system. Chance variability supports the prevention of premature energy exhaustion in various nodes and ultimately increases the total lifetime of the system. Subsequent operations, such as finding the energy needed to utilize a CH function and calculating the Euclidean distance, ensure that a given CH can carry out its burden without failure and sustain reliable communication with nodes in a cluster. The protocol adapts to changing variations by permanently monitoring energy levels and distances, ensuring a robust and efficient system. This organized approach of choosing and reallocating CHs supports decreased energy loss, avoids system congestion, and ensures the uninterrupted, dependable functioning of these devices' healthcare monitoring systems.

5. RESULTS AND DISCUSSION

The recommended learning approach's performance evaluation experimentation analysis is accessible in this part. There are 2 subsections in this section: a simulation study and a comparative analysis.

5.1 Simulation Setup

To simulate the proposed research method, python-3.9.6 is utilized. This tool is efficient and provides all specifications for the proposed technique. Table 2 presents the system specifications.

TABLE II. SYSTEM SPECIFICATION.

Software specification	OS	Windows 10-(64 bit)
	Tool	Python – 3.9.6
Hardware specification	RAM	4 GB
	Hard Disk	500

5.2 Comparative Analysis

This section compares the proposed method to several current methods, such as the Backtracking Search-Based Deep Neural Network (BS-DNN) [26], the Substitution-Ceaser cipher and improved Elliptical Curve Cryptography (SCC-IECC) [27], the use of a blockchain as a trusted, secure, and transparent Distributed Ledger Technology (BC-STDLT) [28], and the "Modified-RPL (Routing Protocol for Low Power and Lossy Networks) method [29], to assess its effectiveness via performance metrics such as the number of epochs vs. accuracy (%), the number of epochs vs. precision (%), the number of Users vs. authentication time(s), the number of users vs. throughput (%), and the number of users vs. packet delivery ratios (%).

A. Evaluating the Impact of Preprocessing Techniques and Training-Testing Splits on Classification Performance.

This case study examines in depth a classification model that uses both a discriminator and generator to investigate the impact of preprocessing methods and various training–testing splits on model effectiveness. The objective is to decide which data partition—70--30 or 80--20—produces superior outcomes in terms of model correctness and consistency.

B. Data Preprocessing and Model Setup.

Before investigating the model performance, the dataset is subjected to conventional preprocessing steps, which involve characteristic scaling via the use of `StandardScaler` to standardize the information. The data then fall under two major heads as input characteristics and result categories. This ensures that the data reach the model uniformly, which is one of the basic necessities in obtaining reliable results.

1) 70–30 Training–Testing Split

The data of the case study are divided into 70% for training and 30% for testing. This was done because the model had to train for 200 epochs, where the loss capabilities of the discriminators as well as generators were tracked. The effects of the division act as a starting line for evaluating the model's ability to generalize new information on the basis of a small amount of training data.

Loss of Discriminator: The discriminator's ability to distinguish between real data and generated data. Loss of the generator: The potential of the generator to create data that are labelled actual with the aid of the discriminator. Test loss assesses how well the discriminator performs on the test dataset, representing the model's ability to generalize. The findings from this separation are crucial for determining the model's performance, with the maximum amount of data used for training and substantial help in testing.

2) 80–20 Training–Testing Split

In the second case, 80--20 split was used, with 80% of the data used for training and the remaining 20% for testing. This division yields additional training data, which could result in a stronger model, but it additionally reduces the scale of the test set, potentially impacting the accuracy of the generalization measurements.

Loss of the discriminator: With expanded data in training, an improvement in the study of the discriminator that can cause a lower loss is expected.

Generator Loss: Maximizing training data could improve the generator's performance, resulting in the generation of more authentic samples.

Test Loss: Evaluating the test loss in this situation involves evaluating whether the model's ability to generalize is improved or compromised with a smaller test set.

C. Comparison and Best Practices

Following the experiments using both 70–30 and 80–20 splits, the results are assessed to identify the optimal method for confirming the experimental findings. The factors that were taken into consideration were model generalization, i.e., the model's performance on the data it has not been trained on. Consistency: The stability of the loss metrics over different epochs.
Efficiency is the equilibrium between the size of training data and the reliability of testing.

70--30 Division: Commonly more effective when aiming to avoid model overfitting and ensuring enough testing data for generalization assessment. 80-20 Split: Beneficial for enhancing model performance, especially with limited data, by maximizing the training data.

The decision on whether to use a 70–30 or 80–20 split is based on the particular needs of the study. If reliability testing is prioritized, a split of 70--30 could be more suitable. Nonetheless, in cases where there is a lack of data in the dataset and the model requires additional training data, an 80--20 split may prove to be advantageous.

5.2.1 Number of epochs vs. accuracy (%)

The accuracy of a model often improves as the number of epochs increases. This is because more epochs enable the model to match the data more effectively by repeatedly fine-tuning its parameters. Equation (25) is used to determine the ratio of accurate forecasts (positive and negative) among all forecasts made. In cloud-IoT healthcare monitoring, having more epochs generally results in a better model, enhancing its ability to classify data accurately and consequently increasing the accuracy. This may be stated mathematically as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{25}$$

Positive instances that are correctly identified (e.g., cases of a health condition) are referred to as true positives (TPs). TN refers to the accurate identification of negative instances, such as cases where the health condition is not present.

False positives (FPs): the quantity of inaccurate positive forecasts (such as cases where the model mistakenly labels a health issue).
Incorrectly predicting a health condition as not existing is known as a false negative (FN).
Increased epochs facilitate the optimization process, potentially leading to higher accuracy in healthcare monitoring tasks within Cloud-IoT environments.

TABLE III. NUMERICAL OUTCOMES OF ACCURACY (%)

(x-axis) – Number of epochs	Accuracy (%)- (y-axis)		
	BS-DNN	Bi-LSTM-FIS	Proposed
1	70	71	73
20	72	74	75
40	74	75	77
60	75	77	79
80	75	76	77
100	75	77	79
120	74	76	80
140	76	80	82
160	79	81	83
180	83	85	87
200	85	87	90

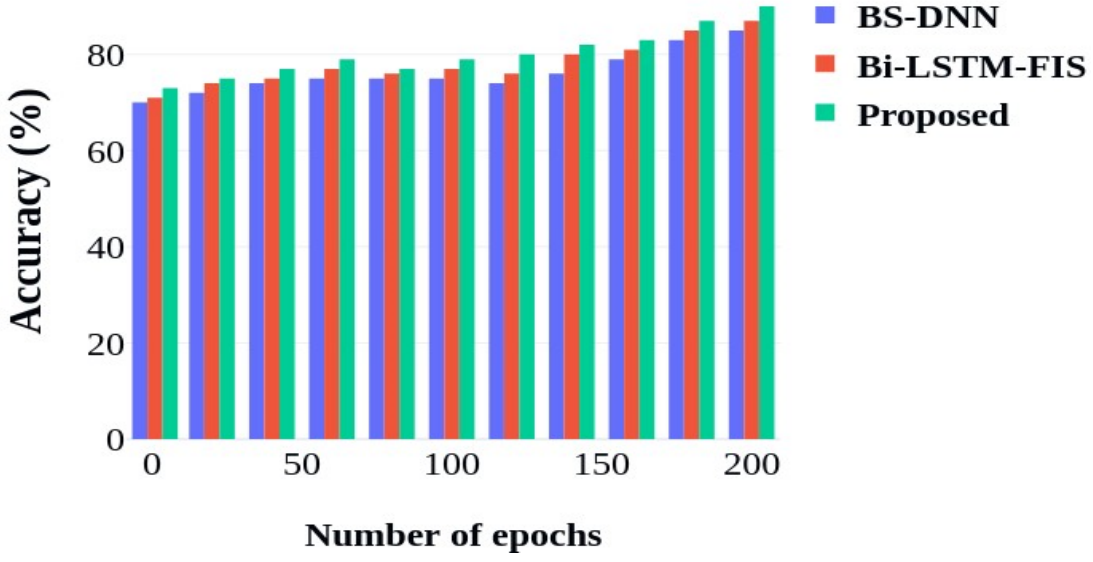

Fig. 5. Number of epochs vs. accuracy (%).

The comparison of the suggested model with the BS-DNN and Bi-LSTM-FIS models, as shown in Table 3 and Fig. 5, demonstrates the continuously superior performance of the suggested model throughout all training epochs. The suggested model, which begins with a slightly higher initial accuracy of 73% in epoch 1 than 70% for BS-DNN and 71% for Bi-LSTM-FIS, continues to outperform the other models during training. At the 20th epoch, the suggested model reached 75% accuracy, whereas BS-DNN and Bi-LSTM-FIS had lower accuracies of 72% and 74%, respectively. This initial benefit indicates that the proposed model is better at capturing key data trends from the beginning. Through training, the proposed model reliably outperforms the other models, attaining 79% accuracy at epoch 100, compared with 76% accuracy for BS-DNN and 78% accuracy for Bi-LSTM-FIS. The learning capability and adaptability of the proposed model are emphasized by its consistent improvement in terms of accuracy, with the ability to outperform the other models. At epoch 200, the proposed model reaches its peak accuracy of 90%, which is especially superior to the best accuracies of 85% for BS-DNN and 87% for Bi-LSTM-FIS. The proposed model constantly outperforms in all epochs, thereby showing a good learning process, robust design, and improved generalization capabilities, which makes it a reliable and effective choice for real-world applications where precision and model performance are at risk.

5.2.2 Number of epochs vs. precision (%)

Precision is the measure of accuracy in positive predictions. The model's ability to correctly distinguish true positives from false positives highlights its effectiveness. As the number of epochs increases, the model's accuracy improves, showing that there is an enhanced reorganization of genuine health conditions and decreased false alerts. This is especially crucial in healthcare monitoring, as precise accuracy assures that the model can accurately detect significant health concerns without needless notification. The precision in mathematics is defined as equation (26):

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{26}$$

The model's ability to identify genuine positive instances improves with more iterations across epochs, leading to increased accuracy in healthcare monitoring activities in Cloud-IoT contexts.

TABLE IV. NUMERICAL OUTCOMES OF PRECISION (%).

(x-axis) – Number of epochs	Precision (%)- (y-axis)		
	BS-DNN	Bi-LSTM-FIS	Proposed
1	75	74	76
20	76	77	79
40	77	79	81
60	79	82	83
80	80	81	83.7
100	79	82	85
120	82	84	86
140	83	85	87
160	82	85	89
180	88	89	91
200	86	90	94

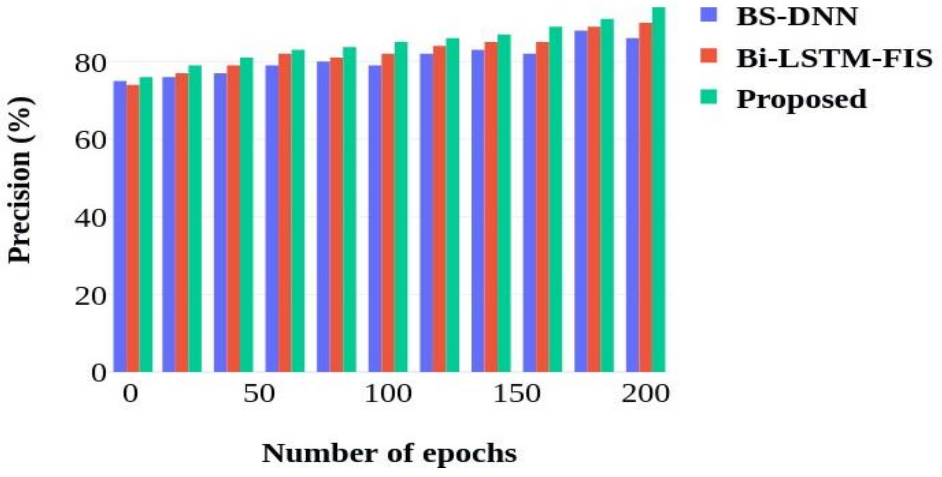

Fig. 6. Number of epochs vs. precision (%)

The evaluation of accuracy, shown in Fig. 6 and Table 4, highlights the continual dominance of the suggested model over BS-DNN and Bi-LSTM-FIS during training. Commencing from epoch 1, the suggested model displays a superior initial accuracy of 76%, as opposed to 75% of BS-DNN and 74% of Bi-LSTM-FIS, showing a more robust foundational efficiency. The early lead increases further as the model is trained, maintaining steady performance from epochs 1 to 20. At the 20th epoch, the proposed model attains a precision rate of 79%, surpassing BS-DNN's 76% and Bi-LSTM-FIS's 77%. During the training process, every model exhibits enhanced accuracy, but the suggested model stands out with a notable increase, achieving a precision of 94% by epoch 200, in contrast to 86% for BS-DNN and 90% for Bi-LSTM-FIS. The notable and consistent difference in accuracy between the suggested model and the current models during the training process demonstrates both the efficacy of the suggested method and its ability to improve practical accuracy in real-life scenarios. This learning in performance raises how the proposed model can make use of training to achieve good results, therefore making it a more enticing option for tasks with needful accuracy.

This exploration can steer future work in developing monitoring tools that are more trustworthy and efficient by illustrating that models that provide better accuracy actually will be able to identify the right real health issues and reduce false alarms. In actual implementation, the results may lead to enhanced patient outcomes through providing more exact and timely treatments and, in future investigations, may motivate further investigation into balancing computational efficiency with model effectiveness in time-constrained health conditions.

5.2.3 Number of Users vs. authentication time(s)

Because it takes more computing power to authorize identifications for every user, the authentication time (s) frequently increases as the number of users increases. This connection may be categorized as follows:

$$Authentication\ time\ =\ Base\ time\ +\ user\ count\ \times\ overhead\ per\ user \tag{27}$$

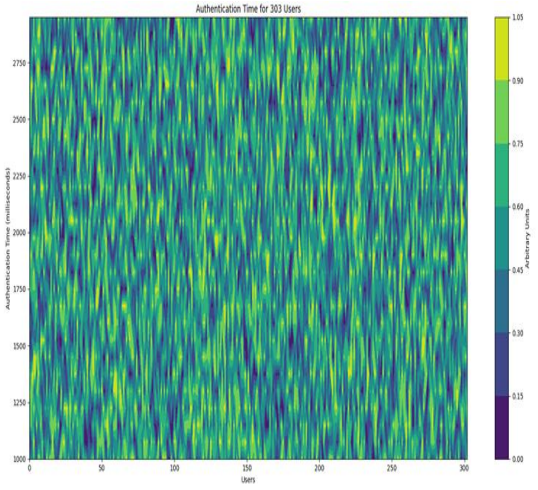

Fig. 7. Number of users vs. authentication time(s).

where:
The essential processing time for verification is represented by base time.
The measure of users trying to validate is known as the "user count."
The additional time needed for every user above and above the basic time is called "overhead per user."
In Cloud-IoT organizations, the verification time tends to increase linearly with the number of users, affecting organization responsiveness.
Figure 7 shows how the verification time decreases with increasing number of users, revealing a noticeable pattern. This unexpected result indicates that the recommended method enhances verification procedures more efficiently than do the present approaches, such as Bi-LSTM-FIS and SCC-IECC. As the number of users increases, the recommended technique effectively reduces the authentication time, possibly because of improved algorithms or optimized processing approaches that simplify user verification. On the other hand, the present techniques lack the same efficiency, resulting in longer authentication times with more users. The proposed method's performance edge shows its improved scalability and efficacy, making it a more practical option for scenarios with growing user populations.

5.2.4 Number of users vs. throughput (%)

There is frequently a nonlinear connection between the throughput (%) and the number of users. As additional users sign up, throughput first increases, taking advantage of parallelism and spreading the problem. However, owing to resource saturation and the struggle for organizational resources, throughput may eventually stagnate or even decrease. This partnership may be summarized as follows:

$$Throughput\ =\ \frac{completed\ transaction}{Total\ time}\ \times 100\% \tag{28}$$

Higher throughput is initially correlated with more users; however, in Cloud-IoT contexts, throughput% may be impacted if a system struggles to maintain the same level of efficiency beyond a particular threshold.

TABLE V. NUMERICAL OYTCOMES OF THROUGHPUT (%)

(x-axis) – Number of users	Throughput (%)- (y-axis)		
	BC-STDLT	Bi-LSTM-FIS	Proposed
1	70	72	73
50	73	75	79
100	78	80	82
150	81	83	85
200	83	85	87
250	84	86	88
300	82	85	90

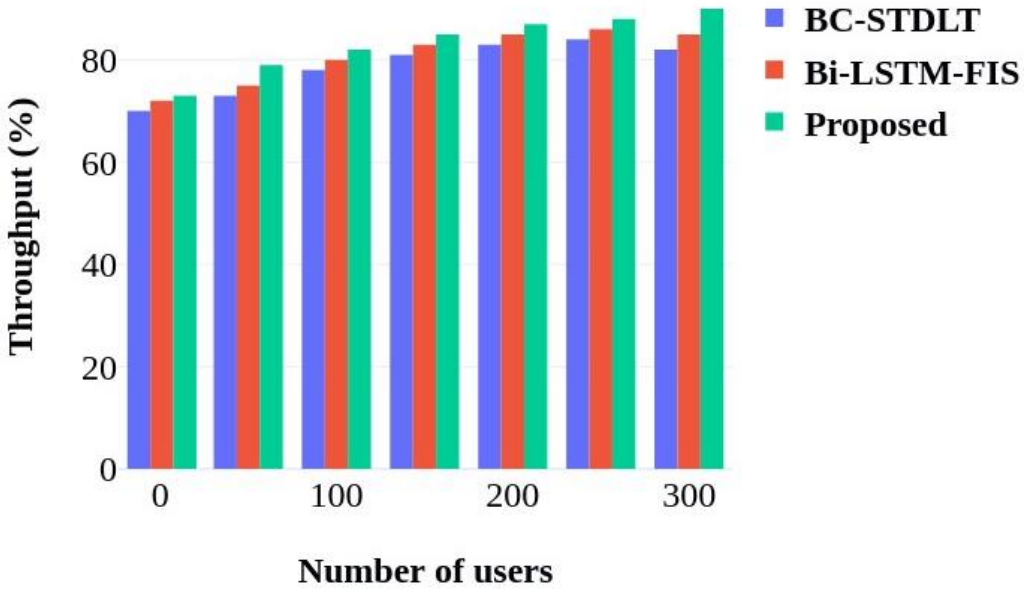

Fig. 8. Number of users vs. throughput (%).

Fig. 8 and Table 5 present a thorough analysis of throughput performance for BC-STDLT, Bi-LSTM-FIS, and the suggested model at different numbers of users. As more users join, the performance of all the models fluctuates significantly, with the suggested model consistently surpassing the other models. Beginning with one user, the suggested model reaches a throughput of 73%, exceeding BC-STDLT at 70% and Bi-LSTM-FIS at 72%. This early benefit becomes increasingly evident with a greater number of users. For example, when there are 50 users, the suggested model achieves a throughput of 79%, which is notably greater than BC-STDLT's 73% and Bi-LSTM-FIS's 75%. This pattern perserves as more user connections, demonstrating the greater scalability of the suggested model. When the user count reaches 300, the recommended model achieves a peak throughput of 90%, superior BC-STDLT at 82%, and Bi-LSTM-FIS at 85%. The continuous enhancement in processing volume highlights the efficacy of the recommended model in managing higher user volumes, positioning it as a more flexible and productive option for high-traffic settings. The model's ability to maintain high performance as the user base increases is clearly shown through both numerical data and graphical illustration, further endorsing its appropriateness for applications that require strong throughput at dissimilar user levels.

5.2.5 Number of users vs. packet delivery ratio (%)

The packet delivery ratio (%) frequently decreases as the user base increases. This is because when the user base increases, there is a greater chance of system congestion, collisions, and interference, all of which might cause packet loss. Similarly, the relationship may be summarized as follows:

$$Packet\ delivery\ ratio = \frac{Sucessfully\ delivered\ packets}{Total\ sent\ pacets} \times 100\% \quad (29)$$

As the number of users grows in the Cloud-IoT, the packet delivery ratio (%) fails because of the higher probability of packet loss due to network traffic or collisions.

TABLE VI. NUMERICAL OUTCOMES OF THE PACKET DELIVERY RATIO (%)

(x-axis) – Number of users	Packet Delivery ratio (%)- (y-axis)		
	Modified-RPL	Bi-LSTM-FIS	Proposed
1	78	81	85
50	82	85	87
100	85	86	89
150	86	87	90
200	88	90	92
250	89	91	93
300	91	92	94

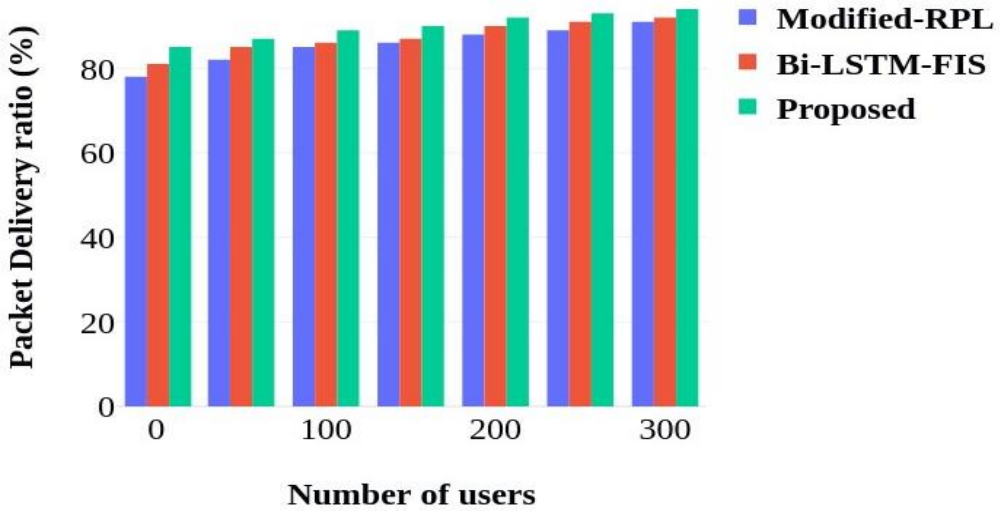

Fig. 9. Number of users vs. Packet delivery ratio (%)

Fig. 9 and Table 6 display an evaluation of the packet delivery ratio (PDR) between several user counts for Modified-RPL, Bi-LSTM-FIS, and the recommended model. The data show that as more users are added, the PDR expands for all the models, but the recommended model consistently performs better than the other models do. Beginning with a single user, the proposed model achieves 85% PDR, exceeding both Modified-RPL at 78% and Bi-LSTM-FIS at 81%. This initial benefit perserves as the number of users increases. When the number of users reaches 50, the proposed model reaches 87% packet delivery, in contrast to 82% for Modified-RPL and 85% for Bi-LSTM-FIS. This pattern persists as more users connect, highlighting the greater scalability and efficiency of the proposed model. Significantly, the suggested model attains an inspiring 94% PDR with 300 users, which is notably higher than those of Modified-RPL's 91% and Bi-LSTM-FIS's 92%. The proposed model has a consistently higher PDR with disparate user counts, which means that the model can handle increasingly demanding systems well, thus being a more dependable and scalable option to use in atmospheres where maintaining a high packet delivery ratio is vital. This demonstration shows that the model is more capable of managing higher user loads and safeguards effective and precise data transmission, thus confirming the appropriateness of the model for applications with high demand.

5.2.6 Confusion matrix

The performance of the classification model was tested via a confusion matrix that provided an extensive overview of the outcomes of each of the training and testing set classifications. The confusion matrix gives us the opportunity to look at the TP, TN, FP, and FN, thus providing even more exhaustive information about the strengths and weaknesses of our classifier, more than what a basic precision metric may show.

A confusion matrix is particularly useful when dealing with imbalanced classes or unequal fault costs. The investigation of the confusion matrix can hence cause adjustments to the model. The critical errors, for example, lowering false positives within a fraud detection mechanism, can be reduced.

For both the 70–30 and 80–20 train–test splits, in this exploration, the confusion matrix was considered. These divisions involve various ratios of the dataset allocated for training and testing, enabling us to assess the model's resilience in different scenarios.

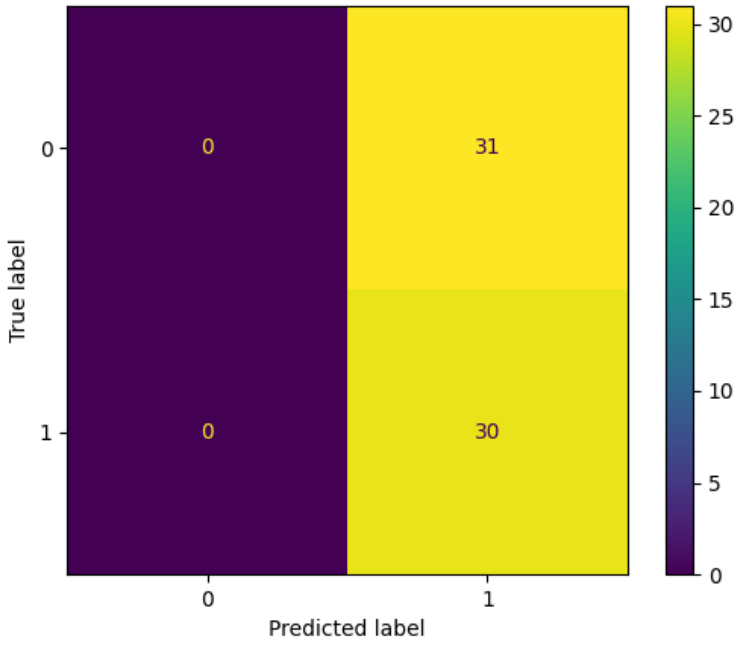

Fig. 10. Confusion matrix 70--30

Fig. 10 shows that the model accurately recognized [47.25%] positive samples and [52.75%] negative samples, demonstrating its efficacy with this particular division of data. Nevertheless, the rate of false positives indicates that certain negative cases were mistakenly identified as positive, which could have significant implications depending on the specific use case.

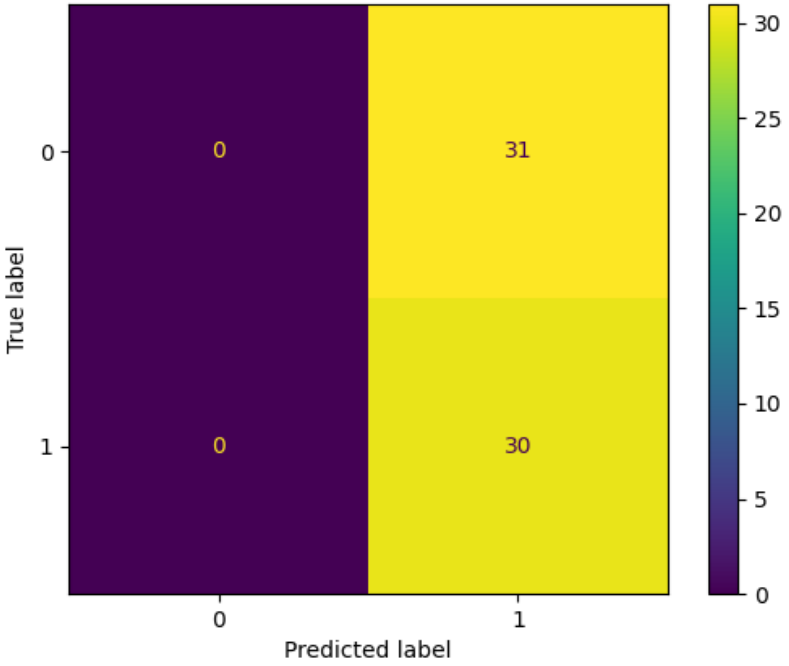

Fig. 11. Confusion matrix 80-20

Fig. 11 represents the 80--20 division; the model showed a slightly varied performance, with a true positive rate of [49.18%] and a false positive rate of [50.82%]. This change highlights how crucial it is to choose the right split between training and test data, as the model's accuracy may vary depending on the proportions used for training. Table 7 presents a comparison with other machine learning algorithms.

TABLE VII. COMPARISON OF THE PROPOSED METHOD WITH OTHER MACHINE LEARNING METHODS

	Method	Accuracy (%)
1.	Fuzzy temporal neural classifier [31]	88
2.	CNN [1]	77
3.	Decision Tree [32]	85
4.	Particle swarm intelligence [33]	80.85
5.	Proposed	90

The proposed research therefore brings many new ideas to the fore, making it unique compared with the prevailing methods in healthcare monitoring systems combined with the Cloud-IoT. The use of homomorphic encryption and the Laplacian approach is an important innovation that enhances data privacy and safety by providing strong safety for sensitive health information in transmission. The incorporation of SMOTE and PCA balances the class distribution, thereby improving feature selection for good accuracy and dependability of machine learning models. Significant achievements include incorporating edge-cloud design into data transmission and storage, increasing storage efficiency, and enabling instant access to critical data. The proposed model includes a GAN with Adam optimization to improve classification accuracy. These improvements enhance the efficacy and extensibility of monitoring organizations within healthcare systems while having impactful practical outcomes and more accurate and timely diagnostics and treatments that will lead to better patient care. The results of this learning can lead to advances in safe, effective, and scalable healthcare monitoring options.

6. CONCLUSION

First, to begin the data collection process, the relevant heart disease dataset is gathered and loaded. The SMOTE is then used to preprocess the loaded data, and the PCA Method" is used to extract features. In this case, to extract the feature, a specific column must be fed. A target was included as an illustration. Next, the adaptive moment estimation optimization approach in conjunction with GANs is used to carry out the classification procedure. In this case, the model is trained to perform GAN operations, and the model generates the intended result. This is where the transaction loss between the discriminator and generator is shown. The data are then sent to an edge-cloud environment, which minimizes storage problems and ensures quick access to critical data. The first step in this procedure is the encryption and decryption of data via the Laplacian method of homomorphic encryption. The Laplacian method of homomorphic encryption was used to encrypt the initial values generated by the GAN network. The Leach protocol is then used in the routing process to maximize communication efficiency and energy usage. In this case, the data are routed via the leach protocol to separate the data into clusters and calculate energy usage. The data are then kept on a server with edge-cloud architecture. Finally, the proposed methods perform better than the existing methods do by comparing the results of the proposed method to several current methods, such as the BS-DNN [26], SCC-IECC [27], the BC-STDLT [28], and "modified-RPL (routing protocol for low-power and lossy networks) [29], which yield better ratios, as shown in Tables 3 to 6 and Figs. 4 to 7.
In the future, improving the encryption and security framework by combining quantum-resistant cryptographic methods would be beneficial. Advancements in quantum computing may expose current encryption techniques, such as homomorphic encryption, to quantum attacks. By integrating quantum-resistant algorithms, healthcare monitoring systems in Cloud-IoT environments could be greatly enhanced in terms of security and durability, guaranteeing the safety of patient data against new quantum threats. This enhancement increases the system's ability to withstand and stay ahead of changing cybersecurity threats.

Funding

None

ACKNOWLEDGEMENT

None

CONFLICTS OF INTEREST

The author declares no conflict of interest.

References

[1] S. Jubal, S. Sharma, and A. S. Shukla, "Smart skin health monitoring using AI-enabled cloud-based IoT," *Materials Today: Proceedings*, vol. 46, pp. 10539–10545, 2021.

[2] A. Zamanifar, "Remote patient monitoring: health status detection and prediction in IoT-based health care," in *IoT in Healthcare and Ambient Assisted Living*, pp. 89–102, 2021.

[3] A. A. Nancy, D. Ravindran, P. D. Raj Vincent, K. Srinivasan, and D. Gutierrez Reina, "IoT-cloud-based smart healthcare monitoring system for heart disease prediction via deep learning," *Electronics*, vol. 11, no. 15, p. 2292, 2022.
[4] M. Azrour, J. Mabrouki, and R. Chaganti, "New efficient and secured authentication protocol for remote healthcare systems in cloud-IoT," *Security and Communication Networks*, vol. 2021, pp. 1–12, 2021.
[5] K. Lakshmanan and S. Arumugam, "An efficient data science technique for IoT-assisted healthcare monitoring system using cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 11, e6857, 2022.
[6] K. Lakshmanan and S. Arumugam, "An efficient data science technique for IoT-assisted healthcare monitoring system using cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 11, e6857, 2022.
[7] T. P. Jacob, A. Pravin, and R. R. Kumar, "A secure IoT-based healthcare framework using modified RSA algorithm and artificial hummingbird-based CNN," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 12, e4622, 2022.
[8] M. F. Khan, T. M. Ghazal, R. A. Said, A. Fatima, S. Abbas, M. A. Khan, and M. A. Khan, "An IoMT-enabled smart healthcare model to monitor elderly people using machine learning technique," *Computational Intelligence and Neuroscience*, vol. 2021, p. 1, 2021.
[9] S. Iranpak, A. Shahbahrami, and H. Shakeri, "Remote patient monitoring and classifying using the Internet of Things platform combined with cloud computing," *Journal of Big Data*, vol. 8, no. 1, p. 120, 2021.
[10] C. Chakraborty and A. Kishor, "Real-time cloud-based patient-centric monitoring using computational health systems," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 6, pp. 1613–1623, 2022.
[11] F. Sabry, T. Eltaras, W. Labda, K. Alzoubi, and Q. Malluhi, "Machine learning for healthcare wearable devices: the big picture," *Journal of Healthcare Engineering*, vol. 2022, p. 1, 2022.
[12] X. Wu, C. Liu, L. Wang, and M. Bilal, "Internet of things-enabled real-time health monitoring system using deep learning," *Neural Computing and Applications*, pp. 1–12, 2023.
[13] A. I. Siam, M. A. Almaiah, A. Al-Zahrani, A. Abou Elazm, G. M. El Banby, W. El-Shafai, et al., "Secure health monitoring communication systems based on IoT and cloud computing for medical emergency applications," *Computational Intelligence and Neuroscience*, vol. 2021, p. 1, 2021.
[14] S. H. Oleiwi, S. S. Gunasekaran, K. I. AbdulAmeer, M. A. Mohammed, and M. A. Mahmoud, "Securing real-time data transfer in healthcare IoT environments with blockchain technology," *Mesopotamian Journal of CyberSecurity*, vol. 4, no. 3, pp. 291–317, Dec. 2024.
[15] M. Uppal, D. Gupta, S. Juneja, A. Sulaiman, K. Rajab, A. Rajab, et al., "Cloud-based fault prediction for real-time monitoring of sensor data in hospital environment using machine learning," *Sustainability*, vol. 14, no. 18, p. 11667, 2022.
[16] M. K. Ahirwar, P. K. Shukla, and R. Singhai, "CBO-IE: a data mining approach for healthcare IoT dataset using chaotic biogeography-based optimization and information entropy," *Scientific Programming*, vol. 2021, p. 1, 2021.
[17] N. Domadiya and U. P. Rao, "Improving healthcare services using source anonymous scheme with privacy-preserving distributed healthcare data collection and mining," *Computing*, vol. 103, no. 1, pp. 155–177, 2021.
[18] 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.
[19] N. A. Almujally, T. Aljrees, O. Saidani, M. Umer, Z. B. Faheem, N. Abuzinadah, K. Alnowaiser, and I. Ashraf, "Monitoring acute heart failure patients using Internet-of-Things-based smart monitoring system," *Sensors*, vol. 23, no. 10, p. 4580, 2023.
[20] P. K. Dutta , Trans., "Encoding IoT for Patient Monitoring and Smart Healthcare: Connected Healthcare System Fostering Health 6.0", BJIoT, vol. 2023, pp. 48–58, Jul. 2023, doi: 10.58496/BJIoT/2023/007.
[21] M. Uppal, D. Gupta, S. Juneja, A. Sulaiman, K. Rajab, A. Rajab, M. A. Elmagzoub, and A. Shaikh, "Cloud-based fault prediction for real-time monitoring of sensor data in a hospital environment using machine learning," *Sustainability*, vol. 14, no. 18, p. 11667, 2022.
[22] S. Shreya, K. Chatterjee, and A. Singh, "A smart secure healthcare monitoring system with Internet of Medical Things," *Computers and Electrical Engineering*, vol. 101, p. 107969, 2022.
[23] A. K. Agrahari, S. Varma, and S. Venkatesan, "Two factor authentication protocol for IoT-based healthcare monitoring system," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–18, 2022.
[24] S. Balakrishnan, K. Suresh Kumar, L. Ramanathan, and S. K. Muthusundar, "IoT for health monitoring system based on machine learning algorithm," *Wireless Personal Communications*, pp. 1–17, 2022.
[25] R. T. Hameed and O. A. Mohamad, "Federated learning in IoT: A survey on distributed decision making," *Babylonian Journal of Internet of Things*, vol. 2023, pp. 1–7, 2023.

[26] R. M. Abd El-Aziz, R. Alanazi, O. R. Shahin, A. Elhadad, A. Abozeid, A. I. Taloba, and R. Alshalabi, "An effective data science technique for IoT-assisted healthcare monitoring system with a rapid adoption of cloud computing," *Computational Intelligence and Neuroscience*, vol. 2022, p. 1, 2022.

[27] M. A. Khan, M. T. Quasim, N. S. Alghamdi, and M. Y. Khan, "A secure framework for authentication and encryption using improved ECC for IoT-based medical sensor data," *IEEE Access*, vol. 8, pp. 52018–52027, 2020.

[28] B. Zaabar, O. Cheikhrouhou, F. Jamil, M. Ammi, and M. Abid, "HealthBlock: A secure blockchain-based healthcare data management system," *Computer Networks*, vol. 200, p. 108500, 2021.

[29] F. Gara, L. B. Saad, R. B. Ayed, and B. Tourancheau, "RPL protocol adapted for healthcare and medical applications," in *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 690–695, IEEE, Aug. 2015.

[30] A. S. . Abdulbaqi, A. M. . Salman, and S. B. . Tambe, "Privacy-Preserving Data Mining Techniques in Big Data: Balancing Security and Usability", SHIFRA, vol. 2023, pp. 1–9, Jan. 2023, doi: 10.70470/SHIFRA/2023/001.

[31] G. K. Kamalam and S. Anitha, "Cloud-IoT secured prediction system for processing and analysis of healthcare data using machine learning techniques," in *Advanced Healthcare Systems: Empowering Physicians with IoT-Enabled Technologies*, pp. 137–172, 2022.

[32] R. Manikandan, R. Patan, A. H. Gandomi, P. Sivanesan, and H. Kalyanaraman, "Hash polynomial two factor decision tree using IoT for smart healthcare scheduling," *Expert Systems with Applications*, vol. 141, p. 112924, 2020.

[33] S. Meti, S. Razauddin, R. Nallakumar, P. M. Mansingh, A. Z. Sameen, S. Pandey, et al., "An empirical IoT and cloud-based customizable healthcare surveillance system," *International Journal of Information Technology*, pp. 1–7, 2024.