Journal of Computer Science

Original Research Paper

Efficient Handling of Medical Data Classification in Cloud-Edge Network using Optimization Algorithm

[1]S. S. Saranya and [2]N. Sabiyath Fatima

[1]Department of CSE, School of Computing, SRM Institute of Science and Technology, Kattankulathur, Chennai, India
[2]Department of Computer Science and Engineering, B.S. Abdur Rahman Crescent Institute of Science and Technology, Vandallur, Chennai, India

Article history
Received: 01-07-2021
Revised: 30-09-2021
Accepted: 21-10-2021

Corresponding Author:
S. S Saranya
Department of CSE, SRM Institute of Science and Technology, Kattankulathur, Chennai, India
Email: saranyas6@srmist.edu.in

Abstract: Nowadays, a cloud-edge computing framework with IoT offers different medicinal facilities by classifying a massive amount of patients' health data through a Deep Neural Network. But, how to optimize task scheduling while carrying multiple tasks from multiple edge devices in real-time was still challenging. This article introduces a cooperative cloud-edge computing structure to effectively perform the fuzzy DNN classification into the edge system and handle the computationally complicated tasks of DNNs. First, the edge servers are constructed with fuzzy DNNs and cooperate with the cloud to create a cooperative cloud-edge computing paradigm. Then, an adaptive deployment method is developed using a Lion Optimization Algorithm, which supports the cloud to decide which task will be executed at the edge devices. Therefore, the study of fuzzy DNN using health data is performed for forecasting and diagnosing various diseases. Finally, the simulation outcomes reveal that the LOA achieved 37.8Jin energy use and 17.8ms latency while using 25 edge devices. Also, the fuzzy DNN achieved 85.8% accuracy for classifying the medical data and diagnosing them in the earlier stage. It concludes that LOA and fuzzy DNN are more efficient than classical optimization and classification for healthcare applications using the cloud-edge computing paradigm.

Keywords: Iot Cloud Edge Computing, Healthcare Systems, Deep Neural Network, Task Distribution, Fuzzy Classifier, Lion Optimization Algorithm

Introduction

The Internet-of-Things (IoT) is the way of developing and forming Internet-coupled things utilizing computer systems. It is improved to have many fewer effective systems like a wrist band, air-conditioner, etc. Such improved IoT items include technical reasoning capabilities for performing distributed tasks without the need for a name or individuality (Gokhale *et al*., 2018; Hassan *et al*., 2020; Shah *et al*., 2019). IoT technology is often assisted by the cloud to improve efficiency concerning the maximum resource use, storage, power and computational ability. Also, cloud or edge computing gets support from IoT technology through enhancing the possibility to manage the present world and distribute several innovative facilities in a dynamic and dispersed way.

The IoT-based cloud method will be expanded for developing innovative facilities and uses in smart settings (Botta *et al*., 2016; Lei *et al*., 2021; Kaur, 2020). The mixture of cloud/edge and IoT-based web use is effective compared to the classic cloud-based systems. Such mixed technology will be used in applications such as healthcare (Dang *et al*., 2019), defence (Fraga-Lamas *et al*., 2016) and financial services (Asadi *et al*., 2017). Mainly, the cloud-based IoT system is helpful to offer adequate facilities for healthcare purposes for forecasting and to access the files from any distant site. The forecasting system is designed by merging these technologies to efficiently forecast the patient's information, even at a remote location, which is helpful for doctors.

In IoT, huge amounts of resources generate the essential information in the real world with no conflicts like reliability to discover the practical information framework (Mahdavinejad *et al*., 2018; Adi *et al*., 2020). These are all assumed to be one of the significant challenges in IoT. However, these challenges result in a large number of changes in current innovations. In this situation, the IoT paradigm encompasses several sources of information where a combination of this information to create emerged facilities becomes a complicated problem. Smart healthcare systems engage many components which carry observed data, which

has to be combined in an automated way. The single sensor takes adequate data and however, if several sensors' information is evaluated, it may describe the essential data regarding configuration alterations.

This information fusion from many sensors enhances accurateness in practical information. Additionally, it minimizes the quantity of information to be analyzed. From this perspective, a Context-aware Data Fusion Technique (CDFT) (Saranya and Fatima, 2020) has been designed. First, information from IoT systems was collected and pre-processed to create it apparent for the merging process. The edge-based noise removal method was applied to pre-process the report, which tries to label the unlabeled features in the collected information. So, information merging was performed precisely. Then, CDFT was conducted, which uses the information from many IoT systems together based on the context. This fused information was transmitted to the cloud through edge servers. Moreover, the Deep Neural Network (DNN) algorithm has been executed to classify the received data in medical applications to detect and diagnose different diseases.

On the other hand, the fast explosion of IoT can produce a huge amount of information to be processed, including the excess capacity of the cloud server. The data is processed at the edge system to solve this problem, which lessens the cloud workloads and training time. Also, a lightweight DNN has been applied on edge servers to reduce the number of training factors in DNN and the computational burden of edge servers. In contrast, edge servers do not use adequate resources for processing and analyzing several data. So, an auto encoder-based lightweight DNN has been introduced for decreasing the information dimensionality on the edge servers. The encoder of the DNN was positioned on edge for reducing the information dimensions. After, such information was transmitted to the cloud server and restored the essential guidance for performing the classification task for healthcare purposes. But, how to accommodate multiple tasks from multiple edge devices in real-time was still challenging.

Therefore, in this study, a cooperative cloud-edge computing paradigm is designed by considering how to execute the fuzzy DNN classifier on the edge servers optimally. The main goal of this paradigm is to manage the computational tasks of DNNs and optimize the task scheduling by the lion optimization approach in the IoT scenario. Also, the edge systems are configured with DNNs by evaluating the computational intensity and latency. Besides, the cloud servers collaborate with the edge servers to build the cooperative cloud-edge computing paradigm. Then, an adaptive deployment method is designed to guarantee the load-balancing of edge servers by the LOA, which considers different factors. According to this optimization, the cloud server decides which task is executed at the edge devices. Thus, it performs the task of fuzzy DNN using health data to forecast and diagnose disorders efficiently. Additionally, it improves the classification accuracy and lessens the energy use of edge and cloud servers.

Literature Survey

A hybrid simulated Ant Colony Optimization (ACO) algorithm (Xian-Jia, 2015) has been presented for scheduling the tasks on the edge nodes based on the determination of the node's computing capacity, memory and bandwidth. However, a few parameters were not thoroughly examined to improve the efficiency of cloud-edge services. An integrated decision support method (Samuel *et al*., 2017) has been designed, which computes different heart failure traits and their contributions with the aid of an expert cardiac physician. Also, a Fuzzy Analytic Hierarchy Process (Fuzzy_AHP) method was applied for determining the global weights for the traits according to their separate contribution. After, the global weights, which indicate the contributions of the characteristics, were used for learning Artificial Neural Network (ANN) classification, which estimates the heart failure risks in patients. But it could not optimize the hidden neurons automatically and was not suitable for the vast number of instances.

An effective Privacy-Preserving Disease Prediction (PPDP) (Zhang *et al*., 2018) method has been developed in which the historical health information was encrypted and outsourced to the cloud. Then, this information was used for learning the prediction frameworks with the help of single-layer perceptron training in a privacy-preserving manner. But it has a high computation cost and error rate.

An Energy-Efficient Particle Swarm Optimization-based Clustering (EEPSOC) method (Alabdulatif *et al*., 2019) has been developed for effectively choosing the cluster heads among different IoT systems. First, the IoT systems deployed to observe the medical information were clustered and the cluster head was chosen based on the EEPSOC for data transfer. Further, an ANN-based classifier was introduced to diagnose the medical information in the cloud and recognize the severity of the syndromes. But it takes more training time and its efficiency depends on the number of instances.

A new generalized cloud healthcare system (Liu et al., 2019) has been developed using Digital Twin Healthcare (CloudDTH) to forecast, diagnose and estimate the characteristics of the fitness of individuals. Also, the conceptual framework of DTH was built for executing the facilities like real-world forecasting for the elderly. But its accuracy was not analyzed and also, the computation time was high.

For predicting and monitoring chronic kidney disease within its severity range, an IoT with a cloud-based healthcare decision support framework (Lakshmanaprabu *et al*., 2019) was built. First, the patient information was gathered with the help of IoT

systems fixed to the customer, which was accumulated in the cloud, including the associated health documents from the UCI repository. Also, the DNN classification was used to predict chronic kidney disorder and its severity range. Further, a PSO-based attribute choice scheme was applied to increase the DNN efficiency. But its complexity was high while considering more instances.

The combination of cloud-edge computing for IoT information analysis (Ghosh and Grolinger, 2019) was investigated. A deep learner was presented for reducing the amount of information on the edge with Machine Learning (ML)-based classification on the cloud. The encoder of the auto-encoder was situated on the edge to reduce the information size. The reduced information was forwarded to the cloud, where it was directly applied to ML to classify human activities. But its classification accurateness was not highly effective.

An intelligent electronic gastroscope model (Ding *et al*., 2019) was developed depending on the cloud-edge cooperative system. In this model, a Tinier-YOLO algorithm was designed depending on the k-DSC unit in the edge-computing system for identifying the lesion or finding abnormal frames. Then, the lesion ROI partition was combined with the YOLOv3 algorithm in the cloud system to enhance the modelling efficiency. On the other hand, its accuracy depends on the selection of threshold values to estimate the similarity.

A layered structure of protected edge-cloud-based medical framework (Jayaram and Prabakaran, 2020) has been suggested for real-world forecasting and treatment of disorders. Using this framework, privacy-preserving additive homomorphic cryptography was added to guarantee information privacy at the edge computing device. Additionally, the efficient pre-processing methods reduced the reaction interval and system facility between the edge and cloud systems. Further, each patient's information needs from various geographic sites were analyzed in a cloud by the dynamic weighted probabilistic classification to predict the onboard disorder. But the system response period was comparatively high.

Health Fog (Tuli *et al*., 2020) developed a new method for combining hybrid deep learning in edge computing systems and has put it to use in real-world applications for automated heart disease prediction. Using this method, healthcare was provided as a fog facility by IoT systems and the heart patient's information was effectively handled. Then, a Fog-enabled cloud model called Fog Bus was applied for deploying and evaluating Health Fog's efficiency. But, it has less robustness and accuracy since it does not enable cost-optimal implementation.

A practical framework (Bhatia *et al*., 2020) has been developed for forecasting home-centric urine-based diabetes. It has four different layers: Diabetic information collection, diabetic information categorization, diabetic-extraction and diabetic estimation and decision-making layers to predict and forecast the diabetes-oriented urine virus. Also, the probabilistic analysis of urine-based diabetes forecasting based on the degree of diabetic disease was quantified as a diabetes disease factor for prediction by the Recurrent Neural Network (RNN). Further, the occurrence of urine-based diabetes was visualized depending on the self-organized mapping process. On the other hand, the system stability and reliability were not adequate while increasing the number of instances.

A smart medical framework (Ali *et al*., 2020) has been developed for predicting heart disorder with the aid of deep ensemble learning and attribute merging schemes. Initially, attribute merging was applied to integrate the mined attributes from both sensed and electronic healthcare data for creating useful medical information. Then, the information gain method was used to remove the redundant and inappropriate features. Also, a conditional probability method was applied to determine the particular attribute weight for every label and the ensemble deep learning structure was learned to predict the heart disorder. But, it is not able to handle a large number of attributes and medical files. Also, noisy and missing data were not effectively eliminated.

A tree-based deep model (Chauhan *et al*., 2021) was developed for recognizing the facial features in the cloud system. An extra dimension was separated into a small portion and a stick was prepared for all portions. Then, the tree was represented by its branch region and stature. The leftover efficiency defined the branches, including a two-fold layer, a stack player strategy and a non-direct efficiency. The inside and out patterns were presented for the PC without concentrating on constant efficiency. But it requires more advanced algorithms for achieving load-balancing in cloud scenarios.

Proposed Methodology

In this section, the proposed algorithm is explained briefly. Fig. 1 portrays the schematic representation of the proposed cloud-edge computing framework in healthcare systems. First, the patient's health information observed by IoT systems is collected and pre-processed using the CDFT for merging such information. This merged information is then forwarded to the cloud servers via edge devices.

Then, the cloud server and edge devices collaborate to determine different system factor limits such as task latency, energy use and processing ability. The aim of LOA (Yazdani and Jolai, 2016) is to learn these determining factors for other edge devices and to find the optimal set of characteristics. Based on these optimized factors, the cloud server decides which task of fuzzy DNN (illustrated in Fig. 2) will be executed at the edge device and which task fuzzy DNN will be implemented at the cloud nodes. Thus, the medical data is classified by implementing the fuzzy DNN to predict and diagnose

different disorders on time. By optimizing the task execution, it can balance the load between cloud and edge devices effectively.

Cooperative Cloud-Edge Network Model

If the cloud server accepts fuzzy DNN from the client, it decides which task is executed at the edge devices based on the optimization criteria. If the requirements are not reached to complete the task at the edge device, then the task will be performed at the cloud nodes adaptively. The cooperative cloud-edge computing paradigm is designed to execute the fuzzy DNN operations dynamically to achieve this process. First, the cloud forwards the initial factors of the fuzzy DNN framework to the edge devices. While the learning tasks arrive, the edge device performs the task framework and transports the j^{th} layer outcome attribute map ($\mathcal{A}i^j$) to the cloud.

The cloud performs the residual layer of the fuzzy DNN and estimates the error value. The back-propagation learns this framework and the factors are mutually fine-tuned. The error value is denoted as $\mathcal{L}(p)$. Every node i comprises a local framework factor $p_i(t)$ where $t = 0,1\ldots$ is the number of iterations. At t = 0, the local factors for each node are assigned to an equal range. The local framework factor is fine-tuned after each iteration as:

$$p_i(t) = p_i(t-1) - \varsigma \cdot \nabla f\left(p_i(t-1)\right) \tag{1}$$

$\zeta > 0$ denotes the step size and $\nabla f(p)$ denotes the gradient value. After each iteration, the model factors of each node are subject to a global fine-tuning and fused as:

$$p(t) = \frac{\sum_{i=1}^{u} \mathcal{U}_i w_i(t)}{\sum_{i=1}^{u} \mathcal{U}_i} \tag{2}$$

In Eq. (2), $\mathcal{U}_i$ denotes the number of tasks executed by node i per unit period. It preserves the factor synchronization of each edge node.

Adaptive Deployment Model for Optimization-based Decision-Making System

An adaptive deployment model is presented for configuring the cloud server, which decides the task execution between the edge and cloud system. In this model, the computational cost, energy usage, processing ability and latency are analyzed based on the task arrival. The LOA is applied to optimize the fuzzy DNN factors with several limits on latency, energy use and processing ability. It is used for load-balancing of edge nodes and cloud servers.

Processing Ability Limit

In the fuzzy DNN framework, every layer executes different functions and engages a considerable storage space. Consider the computation capacity of every layer of fuzzy DNN framework execution is $\mathcal{S} = \{\mathcal{S}_1,\ldots,\mathcal{S}_n\}$ Consider the storage space engaged by every layer of the fuzzy DNN is $\mathcal{M} = \{\mathcal{M}_1,\ldots,\mathcal{M}_n\}$. Initially, assume the computation capacity of the fuzzy DNN framework and neglect the bias factor for simplification.

The computation capacity$\mathcal{S}_i$ executed by i^{th} the outcome attribute map determines the layer ($\mathcal{A}i^2$), convolution kernel $\left(\mathcal{C}k_i^2\right)$, the number of input channels (Q_{i-1}) and outcome (Q_i) entirely as $\mathcal{S}_i = \mathcal{A}i^2 \mathcal{C}k_i^2 \mathcal{Q}_{i-1} \mathcal{Q}_i$. According to the computation capacity $\mathcal{S}$ of every layer network, the sum computation capacity $\mathcal{S}t^j$ of the pre-j layer, fuzzy DNN is acquired as:

$$\mathcal{S}t^j = \sum_{i=1}^{j} \mathcal{A}i^2 \mathcal{C}k_i^2 \mathcal{Q}_{i-1} \mathcal{Q}_i \tag{3}$$

In Eq. (3), the outcome attribute map $\mathcal{A}i^2$ is calculated using the input matrix dimension $\mathcal{A}s_i$, the convolution kernel dimension $\mathcal{C}k_i$, the aggregation dimension $\mathcal{P}o_i$ and the step range L_i represented as:

$$\mathcal{A}i^2 = \left(\frac{\mathcal{A}s_i - \mathcal{C}k_i + 2\mathcal{P}o_i}{\mathcal{L}_i} + 1\right)^2 \tag{4}$$

The storage footprint of the fuzzy DNN framework primarily involves two segments: The overall factor size and each layer's outcome attribute map. The factor size is the overall weight factor of every layer of the fuzzy DNN. The attribute map is the dimension of the attribute outcome by every layer of the network in the execution. The overall factor size pm_i of i^{th} the layer is associated with $\mathcal{C}k_i^2, \mathcal{Q}_{i-1}$ and Q_i as $Pm_i = \mathcal{C}k_i^2 \mathcal{Q}_{i-1} \mathcal{Q}_i$. Also, the attribute map dimension $Map_i = \mathcal{A}_i^2 \mathcal{Q}_i$. The storage footprint of the earlier j-layer fuzzy DNN is determined as:

$$\mathcal{D}t^j = \sum_{i=1}^{j} Pm_i + \sum_{i=1}^{j} Map_i \tag{5}$$

If $\mathcal{D}t^j$ of the earlier j-layer fuzzy DNN is lower than the storage space g of the edge device, the limits are reached. It is established that the storage space of the edge aggregator relates to the smallest amount of edge devices; thus, minor ranges of factor $\mathcal{G}_0 = \min\{\mathcal{G}_1,\ldots,\mathcal{G}_g\}$ where $\mathcal{G}$ is the number of edge devices is considered. Since the functioning network in the edge device considers a particular quantity of storage space, few memory limits of the edge nodes are defined. A threshold is assigned that $\mathcal{D}t^j$ of the earlier j-layer fuzzy DNN is similar to $\rho_0 = 80\%$ of the least storage space $\mathcal{G}_0$ i.e., the edge device storage is previously saturated.

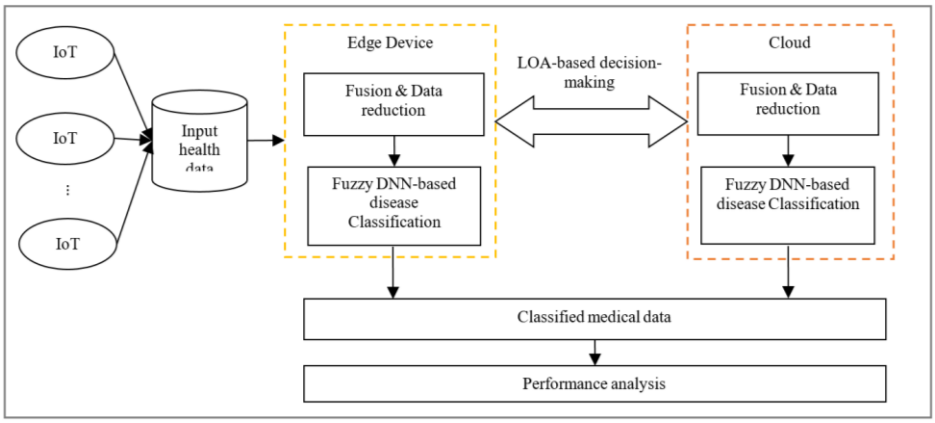

Fig. 1: Schematic representation of proposed cloud-edge computing framework

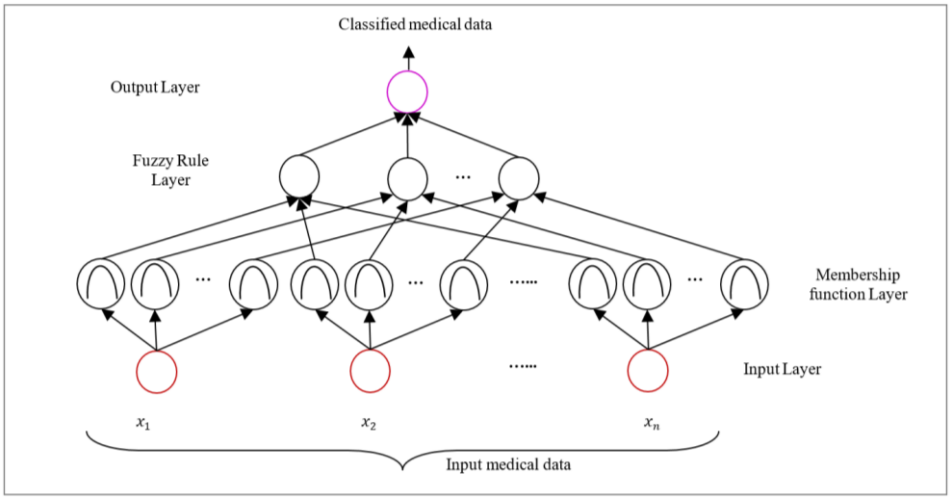

Fig. 2: Structure of fuzzy DNN classifier

Task Latency Limits

Regarding the limits of task latency, the highest latency $\mathcal{T}_{high}$ permitted by the task is mainly made up of the edge system latency $\mathcal{T}_{CE}$, the cloud processing latency $\mathcal{T}_{CP}$, the edge-to-cloud transfer latency $\mathcal{T}_{CE}$ and the cloud-to-edge uplink transfer latency $\mathcal{T}_{CE}$.

In the edge system, because the gap amid the edge devices is highly near, the transfer latency is small. Therefore, $\mathcal{T}_{ES}$ only comprises the task waiting and execution latency. Consider that there are $\mathcal{K}$ training tasks that appear in a particular time, represented as $\{x_1,\ldots, x_k\}$. The execution period is defined as:

$$t_{QP,Exe} = \int \frac{\mathcal{S}t^j}{\mathcal{F}_i} dx_k \tag{6}$$

In Eq. (6), F_i denotes the highest number of Floating-Spot Operations Per Second (FLOPS) of edge node j and $\mathcal{S}t^j$ denotes the computation capacity of the task x_k. Consider that the task arrives separately; so, the $M/M/N$ queuing framework is simulated the queuing in the edge node. The waiting period is determined as:

$$t_{QP,Wait} = \frac{a_k \mathcal{S}t^j}{\mathcal{F}_i\left(\mathcal{F}_i - a_k \mathcal{S}t^j\right)} \tag{7}$$

In Eq. (7), a_k refers to the arrival rate of x_k. The overall task queuing and processing latency is represented as:

$$\mathcal{T}_{QP} = t_{QP,Exe} + t_{QP,Wait} \tag{8}$$

Considering that the cloud computation rate is F_C, the cloud processing latency is represented as:

$$\mathcal{T}_{CP} = \int \frac{\left(\mathcal{S}t^n - \mathcal{S}t^j\right)}{\mathcal{F}_C} dx_k \tag{9}$$

For the edge-to-cloud transfer latency $\mathcal{T}_{EC}$, it is limited using the quantity of task information uploaded. In connection with an inadequate frequency band, if the edge-to-cloud forwards a huge amount of data, the task latency is critically influenced. Therefore, the quantity of information despatched via the task must be limited, which is equal to the transmission system bandwidth B_H and the efficiency of edge devices R. The highest information from the edge system to the cloud is defined as:

$$\mathcal{C}_0 = \min\min\left\{\mathcal{B}_H, \mathcal{R}_i\right\}\mathcal{T}_{EC} \tag{10}$$

Because the edge-to-cloud information transfer quantity is identical to the attribute map dimension of j layer of the fuzzy DNN Map_i, the quantity of information is definite. So, the edge-to-cloud information transfer latency $\mathcal{T}_{EC}$ is obtained. For the cloud-to-edge uplink transfer latency $\mathcal{T}_{CE}$, every task may choose the adjacent edge node for processing the task information represented as IE. The uplink latency of x_k is defined as:

$$\mathcal{T}_{CE} = \sum_{i=1}^{n} IE_k \int \frac{\mathrm{P}(vs_k)}{vs_k} dvs_k \tag{11}$$

In Eq. (11), $P(vs_k)$ is the probability density function of transfer rate vs_k and computed as:

$$\mathrm{P}(vs_k) = \frac{1}{\lambda_k x}\sum_{i=1}^{x} e^{\left(-\left(\frac{vs - vs_{k,i}}{\lambda_k}\right)^2\right)} \tag{12}$$

In Eq. (12), λ_k denotes the bandwidth parameter of the kernel utilized provided as $\lambda_k = \sigma_k\left(\frac{4}{3x}\right)^{0.2}$ where σ_k refers to the predicted standard variance of the edge node v_k.

Power Utilization Limits

Due to the resource-limited behaviour of edge computing, the power of every device is typically constrained. The power use of the edge device comprises stationary energy use and computational power use of the task. The fixed energy use is computed by the computational period of the study as:

$$E_{i,k}^{S} = e_{i,k}\int \frac{\mathcal{S}t_{i,k}^{j}}{\mathcal{F}_i} dx_k \tag{13}$$

In Eq. (13), $e_{i,k}$ is the unit period power use of v_j in $x_k, St_{(i,k)}^{j}$ is the computation capacity of v_j in x_k and F_i is the computation speed of v_j. Additionally, the computational power use is determined as:

$$E_{i,k}^{Com} = \omega_{i,k}\mathcal{S}t^{j}\mathcal{F}_i \tag{14}$$

In Eq. (14), $\omega_{i,k}$ is the power use of unit computation capacity and speed. The power use is determined as:

$$E_i = \sum E_{i,k}^{S} + E_{i,k}^{Com} \tag{15}$$

It can reach the limit if the task power use is less than the highest power use E permitted by the node. Briefly, the j^{th} the layer is the optimized framework that requests to reach the following limits:

$$\text{argmax } f(j), \text{ for } 0 \le j \le Z \tag{16}$$

$$\text{Subject to } g(j) = \left(\sum_{i=1}^{j} Pm_i + \sum_{i=1}^{j} Map_i\right) - \lambda_0\mathcal{G}_0 \le 0$$

$$h(j) = \sum_{i=1}^{k}\left(\mathcal{T}_{QP} + \mathcal{T}_{EC} + \mathcal{T}_{CE}\right) - \mathcal{T}_{high} \le 0$$

$$y(j) = \sum_{i=1}^{g}\sum_{k=1}^{n}\left(E_{i,k}^{S} + E_{i,k}^{Com}\right) - \left\{E_1, \ldots, E_g\right\} \le \tag{17}$$

In Eq. (16), Z is the total number of layers in the fuzzy DNN framework. This can be termed as the improved heuristic lion optimization.

Lion Optimization Algorithm (LOA)

A. Initialization

At first, the population is arbitrarily created over the solution space. Each result is known as a lion (edge devices). In a d-dimensional optimization dilemma, i.e., d set of determining factors $\left(T_{QP}, T_{EC}, T_{CE}, E_{(i,k)}^{S}, E_{(i,k)}^{Com} \text{ and } T_{high}\right)$, a lion (edge devices) is denoted as:

$$Lion(\text{edge devices}) = \left[l_1, \ldots, l_d\right] \tag{18}$$

The fitness range of every edge device (lion) is determined by assessing the objective function given in (16) as:

$$f(\text{edge device}) = f\left(l_1, \ldots, l_d\right) \tag{19}$$

In the primary stage, d_{pop} solutions are created arbitrarily in exploring space, % d of completed results are arbitrarily selected as migrant edge devices. The remaining population is randomly split into φ pride. Each solution comprised a specified gender and stayed stable in the optimization task. During the searching task, each lion observes its most excellent entered site. Depending on such observed sites, every pride's region is created. Therefore, for every pride, observed sites (most excellent entered sites) generate that pride's region through its members.

B. Hunting

In all pride, few females focus on a prey optimal set of $\left(T_{QP}, T_{EC}, T_{CE}, E^{S}_{(i,k)}, E^{Com}_{(i,k)} \text{ and } T_{high}\right)$ in a crowd for offering food for their pride. Such seekers include particular policies for encircling the prey and holding it. Normally, edge devices pursue roughly similar models while hunting. During hunting, every lion adjusts its site depending on its site and group members' locations. Because of this concept that during hunting, few such seekers encircle prey and attack from the opposite side, opponent-based training is utilized.

Based on this concept, the seekers are randomly split into three subgroups. The group with the maximum collective members' fitness is termed a centroid and the other two groups are termed "two wings". A fake prey (*prey*) is taken in the centroid of seekers as:

$$\left(prey = \sum hunters\left(l_1, \ldots, l_d\right) / \, number \ of \ \ hunters\right)$$

During hunting, seekers are suddenly chosen one after the other. Every elected seeker attacks fake prey, following the crowd that the elected lion belongs to. If a seeker enhances its fitness, $prey$ can escape from the seeker and fresh site of *prey*(*prey'*) is found as:

$$prey^{'} = prey + rand\left(0,1\right) \times PI \times \left(prey - hunter\right) \tag{20}$$

In Eq. (20), *prey* denotes the current site of prey, $hunter$ indicates the current site seeker who hit to target and *PI* represents the rate of increase in the objective of the seeker. To mimic encircling prey by selected hunter groups, the fresh locations of hunters (*hunters'*) which are belonging to both left and right wings are created as:

$$hunter^{'} = \begin{cases} rand\left(\left(2 \times prey - hunter\right), prey\right), & \left(2 \times prey - hunter\right) < prey \\ rand\left(prey, \left(2 \times prey - hunter\right)\right), & \left(2 \times prey - hunter\right) > prey \end{cases} \tag{21}$$

Additionally, the new locations of centroid hunters are created as:

$$hunter^{'} = \begin{cases} rand\left(hunter, prey\right), & hunter < prey \\ rand\left(prey, hunter\right), & hunter > prey \end{cases} \tag{22}$$

In Eq. (21) and (22), *rand*(*a*, *b*) produces a random number between *a* and *b* which are upper and lower limits, accordingly.

C. Shifting Towards Secure Site

The new site for a female lion is defined as:

$$\begin{aligned} &femalelion^{'} = female \ lion + 2Dist \times \\ &rand\left(0,1\right)\left\{R1\right\} + U\left(-1,1\right) \times \tan\tan\theta \times Dist \times \\ &\left\{R2\right\}\left\{R1\right\}.\left\{R2\right\} = 0, \left\{R2\right\} = 1 \end{aligned} \tag{23}$$

In Eq. (23), *female lion* denotes the current site of the *female lion*, *Dist* indicates the gap between the female lion's site and the decided spot selected by event choice among the pride's region, $\{R1\}$ denotes the vector which its starting spot is the past position of the female lion and its direction is toward the elected site, $\{R2\}$ denotes the perpendicular to $\{R1\}$.

The success of a lion is defined when it enhances its most excellent site at the final iteration of the lion optimizer. In crowd χ, the success of lion *i* at iteration *t* is described as:

$$uccess\left(i,t,\chi\right) = \begin{cases} 1, & most \ excellent^{t}_{i,\chi} < most \ excellent^{t-1}_{i,\chi} \\ 0, & most \ excellent^{t}_{i,\chi} = most \ excellent^{t-1}_{i,\chi} \end{cases} \tag{24}$$

In Eq. (24), $^{most\,excellent^{t}_{i,\chi}}$ indicates the most excellent site obtained by lion *i* at *t*. The maximum number of successes means that the lions have converged on a remote spot from the optimal spot. Likewise, the minimum amount of success indicates that the lions are swinging around the best result with no considerable enhancement. Therefore, this factor is applied as a valuable element for the event range. By utilizing the success ranges, $K_j(sc)$ is determined as:

$$K_j\left(sc\right) = \sum_{i=1}^{n} Success\left(i,t,\chi\right), j = 1, \ldots, \chi \tag{25}$$

In Eq. (25), *n* denotes the quantity of lion in pride and $K_j(sc)$ indicates the quantity of lion in pride *j* which achieves an enhancement in their objective in the final iteration. Therefore, the event range for every pride is dynamic in each iteration. It indicates that while the success range is reduced, the event range is maximized, resulting in high diversity. So, the tournament range is determined as:

$$T_j^{range} = \max\max\left(2, ceil\left(\frac{K_j\left(sc\right)}{2}\right)\right), j = 1, \ldots, \chi \tag{26}$$

D. Roaming

Every male lion in a pride roams in that pride's region because of a few causes. To imitate this nature of inhabitant males, %*R* of pride region are chosen arbitrarily and are entered by that lion. Including roaming, if the inhabitant male enters a new site that is healthier than its present most excellent site, his most excellent entered result is updated. This roaming is a robust local exploration and supports the lion optimizer for exploring an approximate solution to enhance it. To randomly explore solution space and prevent trapping in local optima, nomad lions and their dynamic roaming are considered.

So, the new site of nomad lions is created as:

$$lion_{ij}^{'} = \begin{cases} lion_{ij}, & if rand_j > P_i \\ RAND_j, & Or\ else \end{cases} \quad (27)$$

In Eq. (27), $lion_{ij}$ denotes the current site of i^{th} nomad lion and j^{th} range, $rand_j$ refers to an arbitrary standard number between 0 and 1, $RAND$ denotes the randomly created vector in search space and P_i stands for the probability which is determined for every nomad lion separately as:

$$P_i = 0.1 + \min\min\left(0.5, \frac{(nomad_i - most\ excellent_{nomad})}{most\ excellent_{nomad}}\right), \quad (28)$$
$$i = 1, \ldots, no.of\ nomad\ lions$$

In Eq. (28), $nomad_i$ and $mostexcellent_{nomad}$ are the objective functions of the current site of the i^{th} lion in nomads and the most excellent objective function of the nomad lion, accordingly.

E. Mating

It is a crucial task that guarantees the lion's survival and offers a chance for data transfer among members. In each pride, %*Ma* of female lions is mating with single or various inhabitant males. For generating offspring, such males are chosen at random from the same pride as the females. For nomad lions, it is varied that a nomad female only mates with one of the males, which are chosen arbitrarily.

The mating function is a linear mixture of parents to generate two fresh offspring. So, the new cubs are developed after choosing the female lion and males for mating as:

$$off\ spring_j 1 = \beta \times female\ lion_j + \sum \frac{(1-\beta)}{\sum_{i=1}^{NR} Success_i} \times male\ lion_j^i \times Success_i \quad (29)$$

$$offspring_j 2 = (1-\beta) \times female\ lion_j + \sum \frac{(\beta)}{\sum_{i=1}^{NR} Success_i} \times male\ lion_j^i \times Success_i \quad (30)$$

In Eq. (29) and (30), $Success_i$ is 1 when male i is chosen for mating; or else, it is 0, NR denotes the number of inhabitant males in a pride, β represents the arbitrarily created integer with uniform distribution with average value 0.5 and standard variance 0.1. One of 2 new offspring is chosen as male and another as a female in a random manner. A mutation is performed on every chromosome of one of the created offspring with the possibility (%*Mu*). A random integer swaps the chromosome range. Through mating, the lion optimizer distributes data between genders if fresh cubs inherit behaviour from both genders.

F. Defence

Nomad male lions attack prides randomly for fighting with other males in their pride. If the nomad lion is sufficiently powerful, the weakest male lion is rejected from the pride and is termed as a nomad by both genders.

G. Migration

In every pride, the highest quantity of females is computed by α% of pride populace. For migration function, a few females decided arbitrarily and turned into nomads. The number of migrated females in every pride is identical to the sum of excess females in every pride and % I of the highest quantity number of females in a pride. If decided, females migrate from pride and become nomads. Fresh nomad females and previous nomad females are ranked based on their objectives. After, the most excellent females are chosen arbitrarily and circulated to pride to satisfy the unfilled site of migrated females. It preserves the diversity of the entire populace and distributes data among pride.

H. Lion's Population Equilibrium

Because there is constant equilibrium in the lion's populace, the number of lions must be controlled at the end of every iteration. So, the nomad lions with the minimum objective value are rejected depending on the highest allowed number of every gender in nomads. So, this process is continued until the stopping criterion is reached, i.e., the optimal set of $\mathcal{T}_{QP}, \mathcal{T}_{EC}, \mathcal{T}_{CE}, E_{i,k}^{S}, E_{i,k}^{Com}$ and $\mathcal{T}_{high}$ is obtained for a set of edge nodes to execute the fuzzy DNN framework. Based on this set of factors obtained for edge nodes, the cloud server decides which task will be executed at the edge device or cloud nodes. If the optimal set of factors satisfies the edge node's constraints for task execution, the cloud executes the task at the particular edge node. Otherwise, the task will be performed at the cloud node itself:

Algorithm:

Input: Different set of factors $\left(\{\mathcal{A}s_i, \mathcal{C}k_i, \mathcal{P}o_i, \mathcal{L}_i, \mathcal{Q}_i\}, \{a_k, \sigma_k, \mathcal{R}, \mathcal{B}_H, \mathcal{D}t, vs\}, \{e_{i,k}, \omega_{i,k}\}, \{n, k, i, \lambda_0, \mathcal{F}, \mathcal{G}, E\}\right)$;

Output: Optimal set of factors for edge devices

Begin

$i \leftarrow 1$; $j \leftarrow 1$; $\mathcal{S}t^j \leftarrow 0$; $\mathcal{D}t^j \leftarrow 0$;

while (i <*N*)

Compute $\mathcal{A}_i^2 Pm_i$, Map_i, $\mathcal{S}_i$ and $\mathcal{D}_I$;

$i \leftarrow i + 1$;

while ($j \leq N$)

$\mathcal{D}t^j \leftarrow \mathcal{D}t^{j-1} + \mathcal{D}_I$;

$\mathcal{S}t^j \leftarrow \mathcal{S}t^{j-1} + \mathcal{S}_i$;

Compute $\mathcal{T}_{QP}, \mathcal{T}_{EC}, \mathcal{T}_{CE}, E_{i,k}^{S}, E_{i,k}^{Com}$ and $\mathcal{T}_{high}$;

Initialize the number of lions (edge devices) and

iterations;
Create an arbitrary result for every lion;
Assign pride and nomad lions;
while (*t*< max *iteration*)
for (*every pride*)
A few females are chosen at random to hunt;
The rest of the females travel toward the most excellent sites in the region;
Every male roam in %*R* of the region;
%*Ma* of female's mate with only one inhabitant males;
The weakest male is rejected by pride and turns into a nomad;
for (*every nomad lion*)
Males and females travel arbitrary distances in exploring space;
%*Ma* of female's mate with only one male;
Nomad males hit pride;
for (*every pride*)
%*I* of females immigrate from pride and turn into nomads;
Each nomad lion gender is ranked based on its objective range;
Most excellent females are chosen and circulated to pride, satisfying empty sites;
Nomad lions with the minimum objective content will be rejected depending on the highest allowed quantity of every gender;
end for
end for
end for
end while
Obtain the optimal set of $\mathcal{T}_{QP}, \mathcal{T}_{EC}, \mathcal{T}_{CE}, E_{i,k}^{S}, E_{i,k}^{Com}$ and $\mathcal{T}_{high}$ for edge devices;
Cloud decides which tasks will be carried out at the edge devices;
end while
end while
End

Experimental Results

In this section, the performance of task execution decisions using LOA for load-balancing in the cloud-edge computing paradigm is compared with the ACO (Xian-Jia, 2015) in terms of energy use and latency. An experiment is implemented in JAVA to analyze the efficiency of the cooperative cloud-edge computing paradigm. To simulate the complex scenario in the IoT, 25 edge devices are considered. The experiment is performed using the Melanoma image dataset which includes both images and metadata. The metadata for all images includes image name, patient id, gender, approximate patient age at the period of imaging, location of the imaged site, diagnosis information, an indicator of malignancy of imaged lesion and binarized version of the target variable (either benign or malignant). From this dataset, 5680 data are considered for training and 1078 data are considered for testing.

Also, classifying medical data using a fuzzy DNN classifier is analyzed in precision, recall and accuracy, compared with the ML (Ghosh and Grolinger, 2019) on disease classification.

Dataset Format

The considered dataset is accessible in 2 formats. The first is the file format defined in the DICOM standard. The DICOM file format is a mixture of the metadata and pixel data in a single file. The pixel data is encoded in JPEG format. The second format is where the images are in JPEG format and the metadata is added in a linked CSV file.

Energy Use

It is defined as the amount of energy consumed by the edge devices for task execution.

Figure 3 demonstrates the comparative efficiency of ACO and LOA-based task execution optimization in terms of energy use. It is identified that the proposed LOA-based task execution optimization for load-balancing in a cloud-edge computing system outperforms the ACO algorithm because it addresses a reduction of 12.7% in energy use for 25 edge devices in a network compared to the ACO algorithm.

Latency

It is the time taken by the edge nodes to successfully execute the tasks.

Fig. 4 depicts the comparative efficiency of ACO and LOA-based task execution optimization in terms of latency. The LOA-based task execution optimization for load-balancing in a cloud-edge computing system achieves less latency than the ACO algorithm since it views a decrease of 32.1% in latency for 25 edge devices compared to the ACO algorithm.

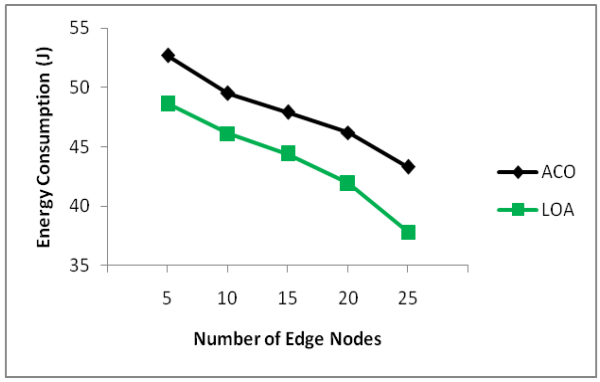

Fig. 3: Comparison of energy consumption

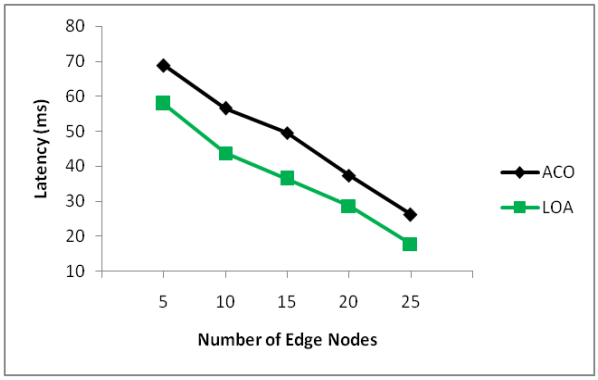

Fig. 4: Comparison of energy latency

Analysis of Classification Performance

Accuracy

The fraction of proper classification of diseased patients over the overall number of trails executed is called accuracy:

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

True Positive (TP) gives a result where the fuzzy DNN/ML properly classifies the diseased patients as itself.

False Positive (FP) results in the fuzzy DNN/ML improperly classifying the diseased patients as healthy patients.

False Negative (FN) results in the fuzzy DNN/ML improperly classifying healthy patients as diseased patients.

True Negative (TN) gives a result where the fuzzy DNN/ML correctly classifies the healthy patients as itself.

Precision

The number of exactly classified diseased patients at TP and FP rates is known as precision:

$$Precision = \frac{TP}{TP + FP}$$

Recall

The number of correctly classified diseased patients at TP and FN rates is known as recall:

$$Recall = \frac{TP}{TP + FN}$$

F-measure

The harmonic average of precision and recall is called f-measure:

$$F - measure = 2 \times \frac{Precision \, CRecall}{Precision + Recall}$$

Fig. 5 - 8 displays the comparative efficiencies of ML and fuzzy DNN classifiers in terms of precision, recall, f-measure and accuracy when increasing the number of images. It observes that the fuzzy DNN achieves 85.8% accuracy, 87.2% precision, 84.7% recall and 85.9% f-measure whereas the ML has 76.6% accuracy, 79.1% precision, 78.5% recall and 78.8% f-measure when considering 200 images. Thus, it concludes that the fuzzy DNN can effectively classify the medical data compared to the ML.

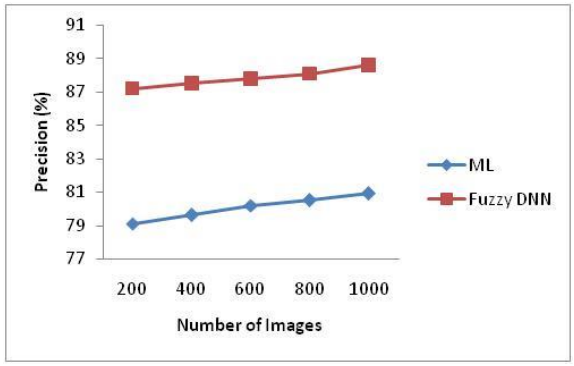

Fig. 5: Comparison of precision for ML and fuzzy DNN classifier

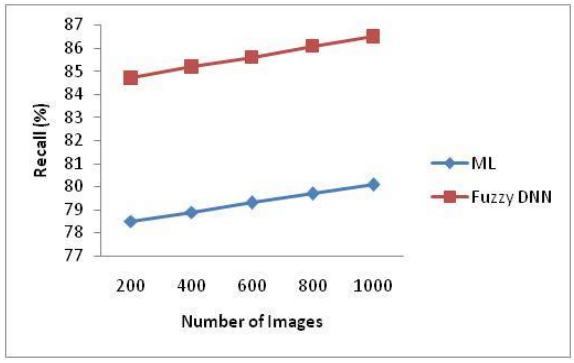

Fig. 6: Comparison of Recall for ML and Fuzzy DNN Classifier

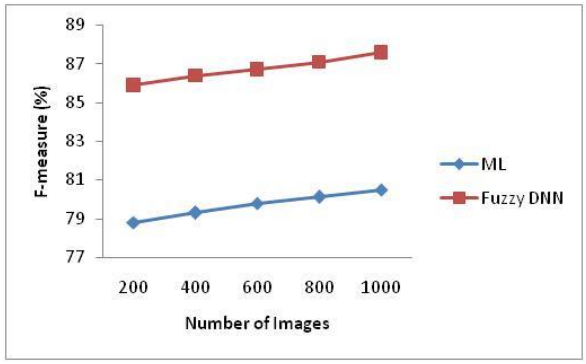

Fig. 7: Comparison of f-measure for ML and fuzzy DNN classifier

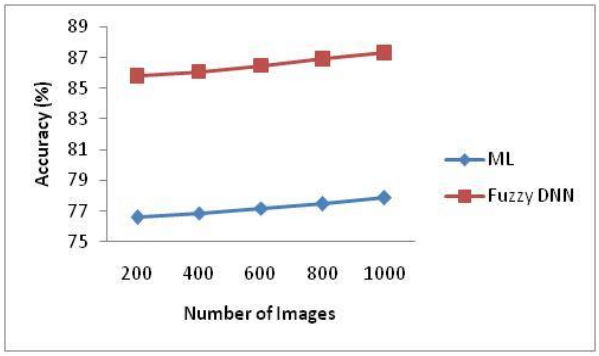

Fig. 8: Comparison of accuracy for ML and fuzzy DNN classifier

Conclusion

In this article, a cooperative cloud-edge computing structure was presented by allowing how to successfully execute the fuzzy DNN classifier into the edge devices to control the computationally complex tasks of DNNs. Primarily, the edge devices were built with fuzzy DNNs via assessing the computational intensity and latency. Afterwards, the cloud servers collaborated with the edge devices to produce the cooperative cloud-edge computing model. Besides, an adaptive deployment technique was adopted to guarantee the load-balancing of edge devices with the help of LOA. According to this optimization solution, the cloud server can choose which task will be performed at the edge devices. To conclude, the findings demonstrated that the proposed algorithm achieves a better efficiency than the conventional algorithms. It realizes that the fuzzy DNN classifies the medical data accurately than the ML since it notices an improvement of 12% accuracy, 10.2% precision, 7.9% in recall and 9% F-measure while compared to the ML classifier. However, it considers only traffic-associated attributes which are not able to improve the load-balancing effectively. Hence, future work will focus on considering network structure characteristics to enhance load-balancing in cloud-edge computing.

Acknowledgement

We thank everybody who supported us to improve this research.

Author's Contributions

S. S. Saranya: The inportant contribution are experimental design data analysis, intapertation and witing of the paper.

N. Sabiyath Fatima: Verifies the contributed, data also verifies flow of the manuscript.

Ethics

We testify that this research paper submitted to the Journal of Science Publication has not been published in entire or in part elsewhere.

This research project was conducted in full compliance with the research ethics norms of SRM Institute of Science and Technology, SRM University - Tamilnadu.

References

Adi, E., Anwar, A., Baig, Z., & Zeadally, S. (2020). Machine learning and data analytics for the IoT. Neural Computing and Applications, 32(20), 16205-16233. https://link.springer.com/article/10.1007/s00521-020-04874-y

Alabdulatif, A., Khalil, I., Kumarage, H., Zomaya, A. Y., & Yi, X. (2019). Privacy-preserving anomaly detection in the cloud for quality assured decision-making in smart cities. Journal of Parallel and Distributed Computing, 127, 209-223. doi.org/10.1016/j.jpdc.2017.12.011

Ali, F., El-Sappagh, S., Islam, S. R., Kwak, D., Ali, A., Imran, M., & Kwak, K. S. (2020). A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. Information Fusion, 63, 208-222. doi.org/10.1016/j.inffus.2020.06.008

Asadi, S., Nilashi, M., Husin, A. R. C., & Yadegaridehkordi, E. (2017). Customers perspectives on adoption of cloud computing in banking sector. Information Technology and Management, 18(4), 305-330. https://link.springer.com/article/10.1007/s10799-016-0270-8

Bhatia, M., Kaur, S., Sood, S. K., & Behal, V. (2020). Internet of things-inspired healthcare system for urine-based diabetes prediction. Artificial Intelligence in Medicine, 107, 101913. doi.org/10.1016/j.artmed.2020.101913

Botta, A., De Donato, W., Persico, V., & Pescapé, A. (2016). Integration of cloud computing and internet of things: a survey. Future Generation Computer Systems, 56, 684-700. https://doi.org/10.1016/j.future.2015.09.021

Chauhan, D., Kumar, A., Bedi, P., Athavale, V. A., Veeraiah, D., & Pratap, B. R. (2021). An effective face recognition system based on cloud based IoT with a deep learning model. Microprocessors and Microsystems, 81, 103726. doi.org/10.1016/j.micpro.2020.103726

Dang, L. M., Piran, M., Han, D., Min, K., & Moon, H. (2019). A survey on internet of things and cloud computing for healthcare. Electronics, 8(7), 768. doi.org/10.3390/electronics8070768

Ding, S., Li, L., Li, Z., Wang, H., & Zhang, Y. (2019). Smart electronic gastroscope system using a cloud–edge collaborative framework. Future Generation Computer Systems, 100, 395-407. doi.org/10.1016/j.future.2019.04.031

Fraga-Lamas, P., Fernández-Caramés, T. M., Suárez-Albela, M., Castedo, L., & González-López, M. (2016). A review on internet of things for defense and public safety. Sensors, 16(10), 1644. doi.org/10.3390/s16101644

Ghosh, A. M., & Grolinger, K. (2019, May). Deep learning: edge-cloud data analytics for iot. In 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE) (pp. 1-7). IEEE. doi.org/10.1109/CCECE.2019.8861806

Gokhale, P., Bhat, O., & Bhat, S. (2018). Introduction to IOT. International Advanced Research Journal in Science, Engineering and Technology, 5(1), 41-44. https://www.researchgate.net/profile/Omkar-Bhat/publication/330114646_Introduction_to_IOT/links/5c2e31cf299bf12be3ab21eb/Introduction-to-IOT.pdf

Hassan, R., Qamar, F., Hasan, M. K., Aman, A. H. M., & Ahmed, A. S. (2020). Internet of things and its applications: a comprehensive survey. Symmetry, 12(10), 1674. doi.org/10.3390/sym12101674

Jayaram, R., & Prabakaran, S. (2020). Onboard disease prediction and rehabilitation monitoring on secure edge-cloud integrated privacy preserving healthcare system. Egyptian Informatics Journal. doi.org/10.1016/j.eij.2020.12.003

Kaur, C. (2020). The cloud computing and internet of things (IoT). Int. J. Sci. Res. Sci. Eng. Technol, 19-22. doi.org/10.32628/IJSRSET196657

Lakshmanaprabu, S. K., Mohanty, S. N., Krishnamoorthy, S., Uthayakumar, J., & Shankar, K. (2019). Online clinical decision support system using optimal deep neural networks. Applied Soft Computing, 81, 105487. doi.org/10.1016/j.asoc.2019.105487

Lei, J., Liu, J., & Li, W. (2021). Hospital information systems in developing countries: a state-of-the-art systematic review. Kybernetes. https://www.emerald.com/insight/content/doi/10.1108/K-09-2020-0590/full/html

Liu, Y., Zhang, L., Yang, Y., Zhou, L., Ren, L., Wang, F., ... & Deen, M. J. (2019). A novel cloud-based framework for the elderly healthcare services using digital twin. IEEE Access, 7, 49088-49101. doi.org/10.1109/ACCESS.2019.2909828

Mahdavinejad, M. S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P., & Sheth, A. P. (2018). Machine learning for internet of things data analysis: a survey. Digital Communications and Networks, 4(3), 161-175. doi.org/10.1016/j.dcan.2017.10.002

Samuel, O. W., Asogbon, G. M., Sangaiah, A. K., Fang, P., & Li, G. (2017). An integrated decision support system based on ANN and Fuzzy_AHP for heart failure risk prediction. Expert Systems with Applications, 68, 163-172. doi.org/10.1016/j.eswa.2016.10.020

Saranya, S. S., & Fatima, N. S. (2020). Context aware data fusion on massive IOT data in dynamic IOT analytics. Webology, 17(2). doi.org/:10.14704/WEB/V17I2/WEB17080

Shah, S. A., Seker, D. Z., Rathore, M. M., Hameed, S., Yahia, S. B., & Draheim, D. (2019). Towards disaster resilient smart cities: can internet of things and big data analytics be the game changers?. IEEE Access, 7, 91885-91903. doi.org/10.1109/ACCESS.2019.2928233

Tuli, S., Basumatary, N., Gill, S. S., Kahani, M., Arya, R. C., Wander, G. S., & Buyya, R. (2020). HealthFog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments. Future Generation Computer Systems, 104, 187-200. doi.org/10.1016/j.future.2019.10.043

Xian-Jia, R. (2015, July). Research on hybrid task scheduling algorithm simulation of ant colony algorithm and simulated annealing algorithm in virtual environment. In 2015 10th International Conference on Computer Science & Education (ICCSE) (pp. 562-565). IEEE. doi.org/10.1109/ICCSE.2015.7250310

Yazdani, M., & Jolai, F. (2016). Lion optimization algorithm (LOA): A nature-inspired metaheuristic algorithm. Journal of Computational Design and Engineering, 3(1), 24-36.

Zhang, C., Zhu, L., Xu, C., & Lu, R. (2018). PPDP: An efficient and privacy-preserving disease prediction scheme in cloud-based e-Healthcare system. Future Generation Computer Systems, 79, 16-25. doi.org/10.1016/j.future.2017.09.002