A Novel Fog Computing Approach for Minimization of Latency in Healthcare using Machine Learning

Amit Kishor[1]*, Chinmay Chakraborty[2], Wilson Jeberson[1]

[1] Department of Computer Science & Information Technology, Sam Higginbottom University of Agriculture, Technology and Sciences (SHUATS), Allahabad, U.P. (India)

[2] Assistant Professor, Department of Electronics and Communication Engineering, Birla Institute of Technology, Mesra, Jharkhand (India)

Received 28 October 2020 | Accepted 23 December 2020 | Published 21 December 2020

Abstract

In the recent scenario, the most challenging requirements are to handle the massive generation of multimedia data from the Internet of Things (IoT) devices which becomes very difficult to handle only through the cloud. Fog computing technology emerges as an intelligent solution and uses a distributed environment to operate. The objective of the paper is latency minimization in e-healthcare through fog computing. Therefore, In IoT multimedia data transmission, the parameters such as transmission delay, network delay, and computation delay must be reduced as there is a high demand for healthcare multimedia analytics. Fog computing provides processing, storage, and analyze the data nearer to IoT and end-users to overcome the latency. In this paper, the novel Intelligent Multimedia Data Segregation (IMDS) scheme using Machine learning (k-fold random forest) is proposed in the fog computing environment that segregates the multimedia data and the model used to calculate total latency (transmission, computation, and network). With the simulated results, we achieved 92% as the classification accuracy of the model, an approximately 95% reduction in latency as compared with the pre-existing model, and improved the quality of services in e-healthcare.

Keywords

Cloud Computing, Data Segregation Scheme, Fog Computing, Latency, Machine Learning, Multimedia Healthcare Data Analytics, Multimedia Transmission, Quality Of Service.

DOI: 10.9781/ijimai.2020.12.004

I. Introduction

As per the International Data Corporation report, there will be 41.6 billion to 1 trillion IoT devices and that will generate a huge amount of data in zettabytes by 2025. There is a big demand of wireless communication due to many reasons such as the tremendous increase in the popularity of IoT devices, extensive use of social media, the dissemination of different mobile application, the population growth of the world, and the present lifestyle that is highly dependent on the latest technology in every aspect. A huge number of multimedia data is generated by IoT devices used in healthcare it is very important to process multimedia data in the healthcare sector, Cloud servers are mostly used world-wide to handle the immense data generated by these IoT devices. The extraction of information's about patient health from supplied analyzed multimedia data is plays a very important and crucial role. Analysis of data, storage of data, pre-processing of data is done by cloud servers. Mainly the cloud computing is the probably viable solution for establishing communication between IoT and healthcare [1]. The healthcare data generated by IoT devices is analyzed, filtered, pre-processed and aggregated only on the cloud. Cloud computing has its limitations. As the data transmission rate increases, due to the receiving of these excessive volumes of data, the response time is increasing in the cloud environment. A higher service delay has occurs to end-users. A high volume of data transmission over the network increases the probability of occurrence of an error and the delay. The loss of data packets and transmission latency is directly related to the quantity of data transmitted through IoT devices to the cloud. Due to this reason, it causes a low quality of service (QoS) produced to the end-user. Cloud computation and data storage are generally not desired in most of the time-sensitive applications of the IoT. Extreme time-bounded problems must be completed nearer to the IoT devices. As the healthcare infrastructures' main requirements are minimization in latency and reduction in network bandwidth, for this it requires data in real-time for a time-critical scenario [2]. The connection is established between end devices and the cloud through routers and gateways. Thus, a huge wide variety of routers are positioned among the cloud and the healthcare IoT's. Due to routers, computational delay increases. As the distance is larger, the large numbers of routers are connected between cloud and IoT devices, because of that a long route is travelled by data from source to destination and it consumes high bandwidth.

* Corresponding author.

E-mail address: amit_kishor@rediffmail.com

For the utilization of the complete advantages of the IoT with fog, it is essential to make available enough networking and infrastructure to produce minimum latency and rapid responding time for IoT applications. Fog computing is introduced as a prime catalyser for the execution and processing of the data generated by IoT devices. It is more effective to shift the applications, execution, and processing capabilities nearer to IoT devices that generated the data. The fog computing concept is properly well suited to resolve these issues.

Fog computing is an emerging concept that uses the processing and execution capabilities closer to the end-user to achieve an improved quality of service as previously used in the cloud paradigm [3]-

[4]. The fog computing layer is placed in between IoT and cloud; it brings low latency and low network usage. Fog computing provides storage, pre-processing, execution, networking, and computational services to their end-users at the edge of the network and closer to end-users. Despite the number of advantages of fog computing, the available research-work is still immature in this domain, and numbers of researchers are still working on the challenges of fog-computing and basic architecture [3]-[6]. The main challenges of fog devices are privacy, security, and consumption of energy. Fog computing is used to overcome the limitations of cloud computing. Fog Computing is used for such applications that require minimum latency and it works on geo-distribution, which is fast and transferable, and has a broad level distribution control system. It enables distributed and computation with low latency at the edge of the network to provides support to IoT applications. Sufficient amounts of available data can be stored, computed, and processed over the fog-networks and that can be controlled by end-users [7]. An open research aim is to improve the quality of services of fog computing by introducing a fog layer between cloud and IoT devices.

The motivation came from a study about how to generate minimum response time with a better quality of service for time-sensitive healthcare IoT based applications. The cloud alone is not able to satisfy the aforementioned requirements due to their limitations. The patient's physical status varies with time and needs rapid response as an action to monitor remote patients. This is possible when there is a very good network available. Otherwise, it will take more time to respond. In fact, due to unpredictable networks, there is high latency, the health data of patients are not considered as real-time data. This shows that the data become unreliable, worthless, and insufficient. The delay may increase for these IoT time-sensitive data from milli-seconds (ms) to seconds and then reaches to minutes [8]-[9], When the size of health data increases, therefore the situation become worsening in handling real-time operation [10]-[11]. The QoS requirements for medical health data [12]-[13] are shown in Table I and the QoS service requirements are shown in Table II for e-healthcare services.

TABLE I. QoS Requirement Parameter for Healthcare and Medical Data Transmission

Services for Healthcare	Data Rate (kbps)	Maximum Delay	Loss of Packet (%)
Audio	4.0–25.0	150.0–400.0 ms	3.0
Video	32.0–384.0	150.0–400.0 ms	1.0
Electro-cardio-gram (ECG)	1.0–20.0	Approx. 1 second	0.0
File -Transfer (FTP)	-	-	0.0

The main contributions of the research work-study are as follows:

1. An analytical model based on fog computing is proposed to transfer healthcare sensor data to end-users in real-time.
2. A random forest algorithm is implemented which reduces and avoids the "over-fitting" issues.
3. The proposed research scheme minimizes the total latency between healthcare sensors and cloud servers. A performance comparison is conducted for the proposed analytical model with existing models on different parameters.
4. To improve the quality of service for e-healthcare.

The remaining part of the paper is organized as follows; Section II describes related work. Section III, introduces a proposed system model for IoT-Fog-Cloud applications. Section IV of the paper is about the work done for the proposed system model. The analysis based on simulation results are provided in Section V. Section VI, comprises the conclusion and provides the future scope of the paper.

TABLE II. QoS Requirement Parameter Services for E-Healthcare

Types of e-healthcare services	e-healthcare system examples	Media type used	Maximum delay
Audio-based communication in real-time	Conferencing (audio) among patients/end-users or end-user/doctors	Audio	< 150.0 ms one way end to end
Video-based communication in real-time	Conferencing (video) among patients/end-users or end-user/doctors	Video	< 250.0 ms one way end to end
Robotic services in real-time	Remote based tele-surgery	Control of data, audio, video by robotics	< 300.0 milliseconds round trip time
Monitoring in real-time	Patient's essential sign transmission and video steaming in an urgent scenario	Sensors (to collect biomedical data)	< 300.0 milliseconds for real-time ECG
Real-time diagnosis	Transfer the medical images to remote areas in an urgent scenario	Images, text, data	-
Real-time messaging	Alarm based indication for urgency	Text, data, small images	No

II. Related Work

Silva et al. [14] used fog computing technology to manage patient records. Fog computing is used to overcome the problem of cloud computing for data management with challenges such as availability, performance, and secrecy. Alarm et al. [15] proposed a method to store the health data on the cloud. The data is generated by IoT-devices. They presented a system for data management in the cloud, based on the management of IoT. The collection of data is done in real-time and an alert system is there with a prior defined rule for notification. Nishtala et al. [16] used a combination of heuristic and reinforcement learning technology called Hipster used to control the latency-critical workloads in the cloud. Hipster aims to improve the efficiency of used resources concerning the quality of service. Latency for large computations is not discussed by the author for the cloud environment. Gia et al. [17] proposed research for continuous monitoring of time-sensitive health patient's through fog computing and concern cost is low. They provided automatic notification and analysis. The sensor-node (energy efficient) system is developed with a layer of fog. Medical practitioners access the data collected through sensors. Naas et al. [18] raised the major problem for IoT applications in time-sensitive cases which is resolved by the author with a technique proposed named iFogStor in fog environment. The author proposed a schema called GAP (Generalized Assignment Problem) for the placement of the data in fog computing. For the solution of GAP two methods are used, first is an accurate solution and the other one is the heuristics method. Rahmani et al. [19] discussed the different services such as real-time processing of local data, data-mining (embedded), and some higher-level services. They presented a prototype called UT-GATE for smart e-health gateway and through which they discussed the features. They have shown the enhancement in performance of overall systems. Wu et al. [20] proposed a schema as security services in fog computing in information-concentric social networks. The main contribution is the introduction of fog computing concepts with required parameters end-to-end communication, low latency, and computing resources at the network edge and also improving the security services by content-aware and matching. Although the network delays, as well as computation delay, are not being discussed by the author. Brogi et al. [21] presented a model for the deployment of QoS-aware in IoT used applications by the use of fog computing technology. The model is used to produce the latency and bandwidth of accessible resources but the author missed discussing network and computation latency.

Shahzad et al. [22] proposed a method to monitor the medical condition in real-time compare to the private cloud. A system is designed and known as BTS (bounded telemonitoring system) for monitoring of patients in real-time. The information for patients is captured in the boundary of the private cloud. They try to provide medical data of patient's with security. Kao et al. [23] introduced the time-critical data analysis in mobile computing for latency minimization the author presented a novel technique with the name of Hermes. The optimization technique based on NP-hard is used for the task data. Li et al. [24] introduced the SPSRP's (service popularity-based smart-resources partitioning) architecture for implementation in IoT and fog computing and also created a mathematical model for the popularity of service and cost of computation on Fog Nodes (FNs). The authors reduced fault tolerance, response-time, and delay time. The calculation for the cost of computing on FNs at the arrival of services from IoT by applying Zipf's law is provided.

Dinh et al. [25] used a service-oriented schema related to cost-effectiveness for providing the service of the IoT-Fog-Cloud network. The authors also used to measure VNF (Virtual Network Function) with development in the capabilities to enhance the availability of SFC (service function chaining) with the proposed metric. Mahmud et al. [26] discussed the problem that occurred in the use of healthcare due to the large volume of transmission of data and high latency. As a solution to these issues, the author presented an IoT-healthcare structure based on fog and explored the cloud-fog service over the traditional cloud. An improvement result is shown for network-traffic, power usage, and the cost. Ahsan et al. [27] highlighted the security, protection, and integrity of the data is a major concern in cloud computing. The author proposed a fog-centric scheme for the storage of data in the cloud. Data security issues had been discussed. Xor-combination is implemented to provide the protection and security of data in the cloud. The proposed method is used to prevent the attack of unauthorized access and malicious users. Hash technique was used in a new form to detect the data alteration with a high occurrence of probability. They also prevented a cyber attack. Rafique et al. [28] used a technique with modification and combination of the PSO (Particle Swarm Optimization) and CSO (Cat Swarm Optimization) to reduce the response time. With the combination of the above two algorithms, they produce NBIHA (Novel bio-inspired hybrid-algorithm) used to overcome the response-time in IoT-Fog-Cloud applications. Li et al. [29] introduced the factors of network delays and designed a framework based on IoT-Fog for estimation of latency. They can predict the delay occurred in the cloud-fog inter-node and proposed a GNP (Global Networking Position) a landmark-based algorithm to predict the latency with good accuracy. Thota et al. [30] presented sensor architecture by using a fog computing platform. Sensors were used to collect patient data and after that sensor send data to edge devices with security. They provided authentication and security of medical data, and unauthorized access was prevented by using asynchronous communication.

Tahani [31], used the scheduling algorithm MAX-MIN on medical data, and then the author used a new method for distribution of task to reduce the waiting time in queue, called TCVC (Task Classification and Virtual Machine Categorization). Raafat et al. [32], presented a model for resource allocation in fog and cloud environments when the data is generated by edge devices. They calculated the overall latency of the model in a fog environment using a genetic algorithm. Pan et al. [33], presented and discussed the current technologies summary report and the compatibility among the cloudlet, home cloud, nebula, fog computing, MEC (mobile-edge-computing). They discussed the different issues related to the aforementioned technologies. But no practical issue is discussed. Chakraborty et al. [36]-[37] measure QoS over heterogeneous networks. Nilashi et al.[38] presented a heart disease prediction model by using fuzzy-SVM. They improve accuracy and computation time. Tarik et al. [39] presented a model for diabetic patients. They analyzed the fasting blood sugar as attributes

TABLE III. Existing Literature Survey

Authors	Proposed Techniques	Advantages	Limitations
Alam et al. [15]	A real-time data collection in fog computing	Data collected in real-time. Transmission delay and computation delay is calculated	There is no calculation for network delay
Nishtala et al. [16]	Hipster : to control time-sensitive issues	Improved efficiency by using network and computation delay	Latency for large data is not discussed and also no method is designed for transmission delay
Naas et al. [18]	iFogStor : GAP for fog computing and heuristic approach	System efficiency is improved. Also they resolve the issue occurred with time-sensitive data.	The transmission delay is not calculated
Wu et al. [20]	Security services as well as content-aware filtering based on fog computing on the edge network	Shifts the task to edge end device from remote locations	There is no discussion about network and computation delay
Brogi et al. [21]	QoS-aware model deployment in IoT by fog computing	Deployed a QoS-aware model in IoT	No explanation for network and computation delay
Shahzad et al. [23]	Hermes: NP-hard technique	Task data is optimized by using NP-hard	There is no explanation for network and computation delay
Li et al. [24]	SPSRP for fog nodes (FNs) and IoT-device	Minimizes the response and delay time in the fog environment	Computation and network delay is not discussed
Dinh et al. [25]	Deployment of cost-effective schema through fog computing for IoT-application	Discusses the issues that occurred due to failure of software and hardware	
Mahmud et al. [26]	IoT-healthcare structure for cloud-fog	Discusses the issue of high volume data. Improved the performance of network traffic and power	Network delay is not being discussed
Ahsan et al. [27]	A fog-centric schema for data storage	Discusses the storage and security of data in the cloud	No discussion about transmission and network delay
Rafique et al. [28]	PSO and CSO techniques	Reduces the response time in the IoT-Fog-Cloud environment	There is no discussion about network and computation delay
Proposed scheme	**IMDS**	**Reduce the overall latency by using transmission, network, and computation delay**	-

for predicting of the diabetics. Mahmud et al. [40] highlighted the recent techniques to capture the different types of patient data in the research. They also captured the video data of the patients. Tarik et al. [41] presented a method for healthcare analysis of patients through meta-heuristic algorithms. This method is very useful for doctors and patients when the patients are suffering from different diseases. Jerry et al., [42] presented a model named BILU-NEMH for extraction and classification of data. They used hypergraph and deep learning concepts to enhance the performance of the designed model. Jerry et al. [43], highlighted the problem faced by sequence labeling and they proposed a model for enhancing the sequence labeling with latent variable conditional random fields. This model is very useful in the stage of the pre-processing of data. Machine understanding becomes strengthens through this model. Ahmed et al. [44], presented a machine learning-based classifier, and the method is mapped with OpenCL. The classification can be accelerated by the use of the proposed method for heterogeneous networks. They also highlighted the solution method for the data imbalance problem. Table III shows the survey on existing literature and Table IV shows the comparative analysis.

TABLE IV. The Comparative Analysis

Authors/Year	**Transmission Delay (T_D)**	**Network Delay(N_D)**	**Computation Delay(C_D)**
Alam et al. [15], 2016	Yes	No	Yes
Nishtala et al. [16], 2017	No	Yes	Yes
Naas et al. [18], 2017	No	Yes	Yes
Wu et al. [20], 2017	Yes	No	No
Brogi et al. [21], 2017	Yes	No	No
Shahzad et al. [23], 2017	Yes	No	No
Li et al. [24], 2018	Yes	No	No
Dinh et al. [25], 2018	Yes	No	No
Mahmud et al. [26], 2018	Yes	Yes	No
Ahsan et al. [27], 2019	No	No	Yes
Rafique et al. [28], 2019	Yes	No	No
Proposed IMDS Algorithm	**Yes**	**Yes**	**Yes**

After analyzing the available research and studying the comparative comprehensive analysis of the reduction in total latency (transmission delay, network delay, and computation delay) among IoT-Fog-Cloud networks, we found that there is a research gap and the available techniques for reducing the latency in healthcare used by the researcher are incomplete. Hence, a novel technique must be required to fill this research gap.

To achieve the imperative execution, the issue of minimization of latency in healthcare cloud and IoT was developed and for the aforementioned aim the system model is presented, the main aim is for the formation of the fog network, to effectively allocate, and distribute the task data. To create the network of fog and unload its task data, a fog node (FN) should search neighboring or adjacent FNs. The neighboring FNs in the system will dynamically appear and disappear. It is well known that, In the healthcare system for monitoring high-risk patients, regular monitoring of patients is required. To maintain the regular monitoring system by the human being is very difficult, tedious and it seems to be an unpractical approach. As a result, carelessness towards the high-risk patient occurs. To avoid such situations, the aim is to evaluate the patient health data to track the probability of any high risk, the system required an analysis of a large volume of healthcare data set and attributes. Random forest is applied for the detection, segregation, and analysis of data. Random forest is selected to avoid the over-fitting problem. The predicted data is sent to the end-user within minimum time. Here, to find the availability of adjacent FN for computation is very difficult. In addition to it, the total numbers of adjacent or neighboring FNs with their locations are unidentified and extremely unpredictable, so it is very challenging to manage the fog network creation and task data distribution process. So under such an unpredictable condition, also considering neighboring FNs is accountable for the appearance of new FNs, which also produces a higher data rate and increasing computational capabilities.

TABLE V. Description of Data Used in the Proposed Model

Sr. No.	**Variable Name**	**Description**
1	Age	Patient's age (in years)
2	Sex	Male/Female as 1/0
3	CP	Chest Pain type (result 1: Angina, result 2: A typical way of Angina, result 3: Not-angina, result 4: Angina symptom nil)
4	Trestbps	Blood Pressure values in resting in mm Hg
5	Chol	Cholesterol results in mg/dl
6	FBS	Blood Sugar results in fasting>120 mg/dl (1 as true; 0 as false)
7	Restecg	ECG resting results (result 0 for normal; result 1 and 2 for abnormal)
8	Thalach	Heart Rate (maximum) as recorded
9	Exang	Prompted Angina exercise(1 as yes; 0 as no)
10	Oldpeak	ST Depression prompted by Exercise as compare to rest
11	Slope	The slope respect to peak of exercise (result 1,2, and 3 for up sloping, flat, and downsloping)
12	CA	Major vessels number (total)
13	Thalrest	Values (at rest) of heart rate
14	NUM	Status of heart disease (result 0 = no heart disease; result greater than 0 = heart disease)

III. Proposed Model

The Fog computing environment based IoT- healthcare system model shown in Fig. 1. The proposed model collects the patient data as per table V. The data is transmitted from IoT or sensor devices and then data is classified into three categories such as low sensitive risk, normal, and high sensitive risk by applying random forest machine learning algorithm. Healthcare sensor data offload their task data to fog servers. After processing healthcare data the time-sensitive data are sent to the end-user in minimum time. FNs are used to distribute and allocate the task data packets in different available nodes and end-user. A principal FN manager is used and that maintains the topological details of task data packet distribution and allocation. Network topology is used to connect the nodes and every FNs are then linked with the principal FN. Here the study shows a continual task data packet allocation system using fog computing environment in machine learning. FNs transfer the task data packet to other FNs in the network to minimize the latency and reduce the network traffic. Here, the processing unit comprises the task data packets in a transmission queue, and then the task data is sent for computation in the computation queue, and these impacts the response time. The entire FN collects details, composes a decision, provides task data information and maintains the queue position. A network table is created by the principal FN and it considers all information that was distributed by the other nodes. The principal FN sends a request to other FNs to determine their availability to process the required data. After getting the availability, the task data will be sent to the nearest FN, where the allotment is performed based on requisite data and time. The work aims to reduce or minimize latency and traffic of network with the selection of time-sensitive data. The process of interacting is that IoT sends data to FN and received data is served by the same FN if data is small otherwise FN serves it partially and sends the remaining data to neighboring FNs to serve. The neighboring FNs compute the data if they are not currently processing any data otherwise data have to wait in the waiting queue for processing. After processing of data from neighboring nodes, these send back the data to the original FN (which transfers the task data first) then the FN sends it to the end-user or cloud. Therefore the selection of the best neighboring nodes is very important for task processing otherwise it increases the waiting time in the computation queue.

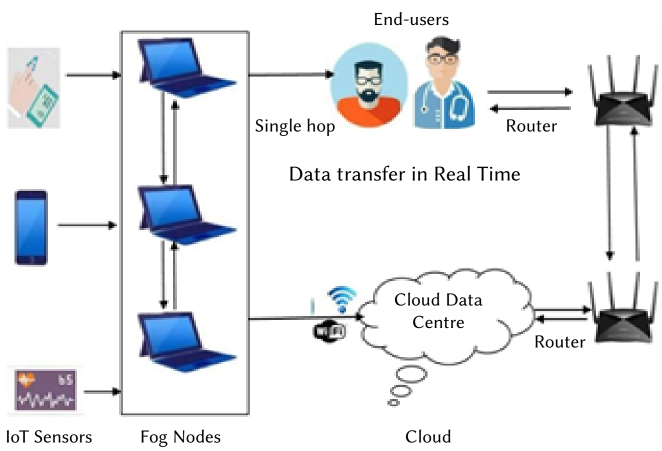

Fig. 1. IoT- HealthCare System Model.

The principal node will maintain an availability table in which all the detail of FNs will be available and maintain all the current updates of the status of FNs. The availability of nodes is shown in table VI, containing the approximate waiting time and the processing speeds of FNs. If any FN became free early then it updates the approximately waiting time in the availability nodes table. The principal node will regularly update the approximately waiting time. The selection of best neighboring FN can be performed by selecting the highest processing speed with a minimum of approximately waiting time.

TABLE VI. Availability of Nodes

Node Description	Approx Weighting Time (microseconds)	The processing speed of FNs (in G.Hz)
132.115.76.84	245	2.201
132.115.76.86	237	2.202
………	…..	….
………	…..	…..

The random forest machine learning algorithm [34]-[35] was used to achieve the aim of the minimization of latency. This aim can be achieved by reducing the used delay as transmission, network, and computation. The proposed model used IoT-Fog-Cloud application's dynamic behavior. A decision-oriented process has been performed using a random forest machine-learning algorithm to overcome the issue of task data demand at a distinct time interval among distinct users and the processing capabilities of FNs. In real-time, the random forest algorithms are used to monitor and care the health data. The main purpose is to minimize the delay that occurs in health monitoring. The FNs identify and select best neighboring FN for computation and process. The quality of service is also a major concern of the entire system. The FNs were used to select the task data communicated by the IoT application in the proposed research. Thereafter, it starts the processing of health task data, and the remaining part of data is transferred to the best neighboring FN and then these processed data is sent to either end-user or cloud in real-time. All the executions are required to be processed in minimum time.

IV. Materials and Methods

Healthcare heart disease data are taken from UC Irvine's machine learning repository [46]. In the simulation heart disease data set encompasses 303 instances and 14 attributes. Although, the UCI repository encompasses 76 attributes in the actual heart disease dataset. In total 14 attributes have been taken for simulation of the proposed algorithm. The testing of the algorithm is performed on these attributes. The attributes are categorized into qualitative and quantitative attributes as shown in the Table V, which shows the data description used in the proposed model. The selection of high-risk data is based on qualitative attributes.

To achieve the objective of the research we applied a k-fold random forest machine learning algorithm. The reason to apply random forest is having better contributions among other classifiers such as SVM (support vector machine), BN (Bayes Network), MP (Multilayer perception), etc. [33]. Feature selection becomes easier in a random forest based algorithm. Estimation of missing values is completed effectively. It avoids over-fitting problems despite that it is a collection of decision trees. Many of research work said that random forest has a quality for prediction of accuracy is excellent for both normal and abnormal data. In a random forest method, the optimization of features is governed by bootstrapped data and this can be performed by k-fold cross-validation (k=10). To avoid overfitting the other scheme such as early stopping and ensembling can also be used. Fig. 2 represents the flow chart of the intelligent multimedia data segregation (IMDS) scheme. The distance travel and the number of hops covered from the sensor to the cloud server is minimum for the high-risk data because it is processed near to sensor devices known as fog computing. By the use of this process, there is a reduction in transmission time due to the total latency time reduces.

In the proposed IMDS algorithm based on k-fold random forest machine learning techniques, the model collects the data at the initial level. Data is pre-processed after collection. Then data is divided into k-fold. Herein k-fold cross-validation is applied. The cross-validation process is evaluating the model by dividing the original sample into small k-chunks. The partition process of the original data in k-chunks used a random approach but the size is always equal. In k-fold, k-1 chunks are used for training the model, and the remaining single chunk is used to test the model. The Gini index is calculated for accurate measurement. Training and testing of data are completed with a ratio of 70 and 30. We can also train our proposed model by using meta-heuristic optimization techniques [45].

Proposed IMDS Algorithm:

```
1. from random import randrange
2. from csv import reader
3. from math import sqrt
4. def load_csv(filename):
5. def str_column_to_float(dataset, column):
6. def str_column_to_int(dataset, column):
7. def cross_validation_split(dataset, n_folds):
8. dataset_split = list()
9. dataset_copy = list(dataset)
10. fold_size = int(len(dataset) / n_folds)
11. def test_split(index, value, dataset):
12.     binwidth = int((max(df["survival_score"])-min(df["survival_score"]))/3)bins=range(min(df["survival_score"]),max(df["survival_score"],binwidth)
        group_name= ['normal','low_risk','high_risk']
13. def gini_index(groups, classes):
14. gini += (1.0 - score) * (size / n_instances)
15. return gini
16. def build_tree(train, max_depth, min_size, n_features):
17. root = get_split(train, n_features)
18. split(root, max_depth, min_size, n_features, 1)
19. return root
20. def predict(node, row):
21. if row[node['index']] < node['value']:
22.             if isinstance(node['left'], dict):
23.                     return predict(node['left'], row)
24.             else:   return node['left']
25. else:  if is instance(node['right'], dict):
26.                     return predict(node['right'], row)
27.             else:  return node['right']
28. def random_forest(train, test, max_depth, min_size, sample_size, n_trees, n_features):
29. trees = list()
30. for i in range(n_trees):
31.             sample = subsample(train, sample_size)
32.             tree = build_tree(sample, max_depth, min_size, n_features)
33.             trees.append(tree)
34. predictions = [bagging_predict(trees, row) for row in test]
35. return(predictions)
36. filename = 'sonar.all-data.csv'
37. dataset = load_csv(filename)
38. for i in range(0, len(dataset[0])-1):
39.             str_column_to_float(dataset, i)
        str_column_to_int(dataset, len(dataset[0])-1)
        n_folds = 10, max_depth = 10, min_size = 1, sample_size = 1.0
        n_features = int(sqrt(len(dataset[0])-1))
40. for n_trees in [1, 10, 10]:
41.             scores = evaluate_algorithm(dataset, random_forest, n_folds,
        max_depth, min_size, sample_size, n_trees, n_features)
42. print('Mean Accuracy: %.3f%%' % (sum(scores)/float(len(scores))))
```

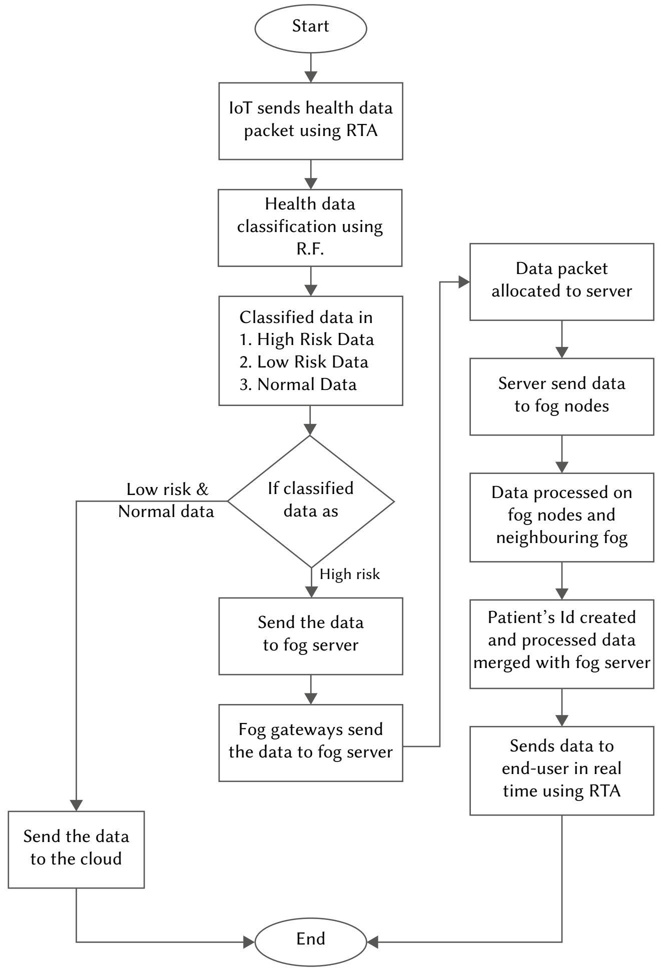

Fig. 2. Flow chart for IMDS scheme.

A. A mathematical Framework For Latency Calculation

Here we assume that there are different kinds of sensors used to forwarding their health data to an FN. In this research, for implementation of fog computing, we are concern with finding the low latency in health informatics. Considering a fog network, containing a cloud layer, a fog layer, and a sensor layer, here the data will be transferred regularly between all the tiers. The sensor layer is containing smart in nature and small in size IoT devices and they don't have enough capability of computations. The fog networks are placed closer to IoT devices to process the patient's data. It is considered that different kinds of sensor devices send their data to a FN (i) and the data size will be XP packets/second. FN (i) performs the task of controlling, storing, analyzing, and processing the health data received from sensors. Here the FNs (i) cooperate with other adjacent or neighboring nodes. After receiving XP packets of the task at FN (i) from the end-user node (e) then the FN distributes the task to adjacent or neighboring FNs (j) for computation. After computation the task is returned to the main FN. Here, the transmission delay for the request of FN and response time from neighboring FN is calculated as the transmission delay.

Transmission delay (T_D): Transmission delay (T_D) is the round trip time (RTT) in relaying of data fragment between end-users nodes (wearable sensors) to FNs can be calculated by M/D/1 system as follows

$$T_D = FN_{RT} + FN_{RPT} \tag{1}$$

Transmission delay between end-user node (e) to FN (i) is as follows

FN request (FN_{RT}) from e to i is,

$$\mathrm{T}_{ei} = \frac{\lambda_{ei}}{2\mu_{ei}(\mu_{ei}-\lambda_{ei})} + \frac{1}{\mu_{ei}} \tag{2}$$

FN response time (FN_{RPT}) from i to e is,

$$\mathrm{T}_{ie} = \frac{\lambda_{ie}}{2\mu_{ie}(\mu_{ie}-\lambda_{ie})} + \frac{1}{\mu_{ie}} \tag{3}$$

Where

$$\mu_{ei} = B_e \log_2\left(1 + \frac{g_{ei} P_{tx,e}}{B_e N_o^e}\right) \tag{4}$$

and $g_{ei} = \gamma_1 d_{ei}^{-\gamma_2}$

$$\mu_{ie} = B_i \log_2\left(1 + \frac{g_{ie} P_{tx,i}}{B_i N_o^i}\right) \tag{5}$$

and $g_{ie} = \gamma_3 d_{ie}^{-\gamma_4}$

Hence the transmission delay is as

$$\mathrm{T}_{D1} = \mathrm{T}_{ei} + \mathrm{T}_{ie}$$

$$= \frac{\lambda_{ei}}{2\mu_{ei}(\mu_{ei}-\lambda_{ei})} + \frac{1}{\mu_{ei}} + \frac{\lambda_{ie}}{2\mu_{ie}(\mu_{ie}-\lambda_{ie})} + \frac{1}{\mu_{ie}}$$

$$= \frac{\lambda_{ei}}{2\mu_{ei}(\mu_{ei}-\lambda_{ei})} + \frac{\lambda_{ie}}{2\mu_{ie}(\mu_{ie}-\lambda_{ie})} + \frac{1}{\mu_{ei}} + \frac{1}{\mu_{ie}} \tag{6}$$

T_D between FN (i) and neighboring node (j) can be expressed as follows

FN request (FN_{RT}) from i to j is,

$$\mathrm{T}_{ij} = \frac{\lambda_{ij}}{2\mu_{ij}(\mu_{ij}-\lambda_{ij})} + \frac{1}{\mu_{ij}} \tag{7}$$

FN response time (FN_{RPT}) from j to i is,

$$\mathrm{T}_{ji} = \frac{\lambda_{ji}}{2\mu_{ji}(\mu_{ji}-\lambda_{ji})} + \frac{1}{\mu_{ji}} \tag{8}$$

Where $\mu_{ij} = B_j \log_2\left(1 + \frac{g_{ji} P_{tx,j}}{B_j N_o^j}\right)$ (9)

And $g_{ij} = \gamma_5 d_{ij}^{-\gamma_6}$

$$\mu_{ji} = B_i \log_2\left(1 + \frac{g_{ij} P_{tx,i}}{B_i N_o^i}\right) \tag{10}$$

And $g_{ji} = \gamma_7 d_{ji}^{-\gamma_8}$

Hence the transmission delay is as

$$\mathrm{T}_{D2} = \frac{\lambda_{ij}}{2\mu_{ij}(\mu_{ij}-\lambda_{ij})} + \frac{\lambda_{ji}}{2\mu_{ji}(\mu_{ji}-\lambda_{ji})} + \frac{1}{\mu_{ji}} + \frac{1}{\mu_{ij}} \tag{11}$$

Transmission delay between end user node (e) and neighboring FN (j) is

FN request (FN_{RT}) from i to j is

$$\mathrm{T}_{ej} = \frac{\lambda_{ej}}{2\mu_{ej}(\mu_{ej}-\lambda_{ej})} + \frac{1}{\mu_{ej}} \tag{12}$$

FN response time (FN_{RPT}) from i to e is,

$$\mathrm{T}_{je} = \frac{\lambda_{je}}{2\mu_{je}(\mu_{je}-\lambda_{je})} + \frac{1}{\mu_{je}} \tag{13}$$

Where, $\mu_{je} = B_j \log_2\left(1 + \frac{g_{je} P_{tx,j}}{B_j N_o^j}\right)$ (14)

And $g_{ej} = \gamma_9 d_{ej}^{-\beta\gamma_{10}}$

$$\mu_{ej} = B_e \log_2\left(1 + \frac{g_{ej} P_{tx,e}}{B_e N_o^e}\right) \tag{15}$$

$$g_{je} = \gamma_{11} d_{je}^{-\gamma_{12}}$$

The transmission delay between the end node (e) and the neighboring FN (j) is

$$\mathrm{T}_{D3} = \frac{\lambda_{ej}}{2\mu_{ej}(\mu_{ej}-\lambda_{ej})} + \frac{\lambda_{je}}{2\mu_{je}(\mu_{je}-\lambda_{je})} + \frac{1}{\mu_{ej}} + \frac{1}{\mu_{je}} \tag{16}$$

Hence the total transmission delay will be

$$\mathrm{T}_D = \mathrm{T}_{D1} + \mathrm{T}_{D2} + \mathrm{T}_{D3}$$

$$= \frac{\lambda_{ei}}{2\mu_{ei}(\mu_{ei}-\lambda_{ei})} + \frac{\lambda_{ie}}{2\mu_{ie}(\mu_{ie}-\lambda_{ie})} + \frac{1}{\mu_{ei}} + \frac{1}{\mu_{ie}} + \frac{\lambda_{ij}}{2\mu_{ij}(\mu_{ij}-\lambda_{ij})} +$$
$$\frac{\lambda_{ji}}{2\mu_{ji}(\mu_{ji}-\lambda_{ji})} + \frac{1}{\mu_{ji}} + \frac{1}{\mu_{ij}} + \frac{\lambda_{ej}}{2\mu_{ej}(\mu_{ej}-\lambda_{ej})} +$$
$$\frac{\lambda_{je}}{2\mu_{je}(\mu_{je}-\lambda_{je})} + \frac{1}{\mu_{ej}} + \frac{1}{\mu_{je}}$$

$$= \frac{\lambda_{ei}}{2\mu_{ei}(\mu_{ei}-\lambda_{ei})} + \frac{\lambda_{ie}}{2\mu_{ie}(\mu_{ie}-\lambda_{ie})} + \frac{\lambda_{ij}}{2\mu_{ij}(\mu_{ij}-\lambda_{ij})} +$$
$$\frac{\lambda_{ji}}{2\mu_{ji}(\mu_{ji}-\lambda_{ji})} + \frac{\lambda_{ej}}{2\mu_{ej}(\mu_{ej}-\lambda_{ej})} + \frac{\lambda_{je}}{2\mu_{je}(\mu_{je}-\lambda_{je})} + \frac{1}{\mu_{ei}} + \frac{1}{\mu_{ie}}$$
$$+ \frac{1}{\mu_{ji}} + \frac{1}{\mu_{ij}} + \frac{1}{\mu_{ej}} + \frac{1}{\mu_{je}} \tag{17}$$

Here, μ_{ei} is the transmission service rate from the IoT device e to FN i, μ_{ie} is the transmission service rate from the FN i to the IoT device e, μ_{je} is the transmission service rate from neighbouring FN j to e, μ_{ej} is the transmission service rate from the IoT device e to node j, μ_{ij} is the transmission service rate from the FN i to the node j, T_{ei} is the T_D from IoT device e to the FN i, T_{ie} is the T_D from the FN i to IoT device e, T_{ej} is the T_D from the IoT device e to the neighbouring FN j, T_{je} is the T_D from the neighbouring FN j to the IoT device e, T_{ji} is the T_D from the neighbouring FN j to the FN i, T_{ij} is the T_D from the FN i to neighbouring FN j, g_{ei}, g_{ie}, g_{je}, g_{ej}, g_{ij} and g_{ji} are the channel gains for μ_{ei}, μ_{ie}, μ_{je}, μ_{ej}, μ_{ij}, and μ_{ji}, B_e, B_i, and B_j are the bandwidth for the IoT device e , for node i, and for node j, γ_1, γ_3, γ_5, γ_7, γ_9, and γ_{11} are the path loss exponent, γ_2, γ_4, γ_6, γ_8, γ_{10}, and γ_{12} are the Path loss constant, $P_{tx,e}$, $P_{tx,i}$ and $P_{tx,j}$ are the transmission power for node e , node i, and node j, d_{ei}, d_{ie}, d_{ej}, d_{je}, d_{ij}, and d_{ji} are the distance between e and i, i and e, e and j, j and e, i and j, and j and i, N_o^e, N_o^i, and N_o^j are the noise densities from nodes e to i and j, i to j and e, and j to i and e, λ_{ei} and λ_{ie} are arrival rates of task data from node e to node i, and from node i to node e, λ_{ej} and λ_{je} are the arrival rate of task data from node e to node j, and from node j to node e, λ_{ij} and λ_{ji} are the arrival rate of task data from node i to node j, and from node j to node i

Network Delay (N_D): Networks delay (N_D) incurred the delay which depends upon the total number of packets from the sensor network to fog network and fog network to sensor network. Network delay depends upon every hop delay as well as total packet sent from the end-user node e to FN i, FN i to neighboring node j, and from FN j to end-user node e and also assuming that there is equal latency for each hop delay. The network delay is calculated as:

N_D = N_D from node e to i + N_D from node i to j + N_D from node j to e

$$= \frac{d_\propto h_c\, e}{x_P} + \frac{d_\propto h_c\, i}{x_P} + \frac{d_\propto h_c\, j}{x_P} = \frac{d_\propto h_c}{x_P}(e + i + j) \tag{18}$$

Where h_c and $d_\propto$ are the hop count and hop delay.

Computation Delay (C_D): When task computation is done by FN, there exists a waiting queue in the task computation queue due to the prior task available in the queue for processing. The neighbouring FNs are not just receiving the task from a single source node they receive it from multiple nodes and also from end-users. Hence, the computation queue can be computed as an M/D/1 system, neglecting the loss of packets, with the task data arrival rate and the computation latency of

FNs that can be expressed as

$$C_{ei} = \frac{\lambda_{ei}}{2\mu_i(\mu_i - \lambda_{ei})} + \frac{1}{\mu_i} + \frac{\lambda_{ei}}{C_s} \tag{19}$$

$$C_{ij} = \frac{\lambda_{ij}}{2\mu_j(\mu_j - \lambda_{ij})} + \frac{1}{\mu_j} + \frac{\lambda_{ij}}{C_s'} \tag{20}$$

Total computation delay can be calculated as

$$C_D = C_{ei} + C_{ij} = \frac{\lambda_{ei}}{2\mu_i(\mu_i - \lambda_{ei})} + \frac{1}{\mu_i} + \frac{\lambda_{ei}}{C_s} + \frac{\lambda_{ij}}{2\mu_j(\mu_j - \lambda_{ij})} + \frac{1}{\mu_j} + \frac{\lambda_{ij}}{C_s'}$$

$$= \frac{\lambda_{ei}}{2\mu_i(\mu_i - \lambda_{ei})} + \frac{\lambda_{ij}}{2\mu_j(\mu_j - \lambda_{ij})} + \frac{1}{\mu_i} + \frac{1}{\mu_j} + \frac{\lambda_{ei}}{C_s} + \frac{\lambda_{ij}}{C_s'} \tag{21}$$

Where μ_i and μ_j are the hardware parameter at node i and node j, C_s and C_s' are the speeds of CPU at node i and node j.

Here, the first term used as a waiting time in the computation queue, and the second term is used as delay occurred for tracking the proper application used for task computation. The tracking delay depends upon the quality of the hardware used.

The total latency (T_L) or total delay time can be calculated as the sum of transmission delay, network delay, and computation delay

$$T_L = T_D + N_D + C_D$$

$$= \frac{\lambda_{ei}}{2\mu_{ei}(\mu_{ei} - \lambda_{ei})} + \frac{\lambda_{ie}}{2\mu_{ie}(\mu_{ie} - \lambda_{ie})} + \frac{\lambda_{ij}}{2\mu_{ij}(\mu_{ij} - \lambda_{ij})} + \frac{\lambda_{ji}}{2\mu_{ji}(\mu_{ji} - \lambda_{ji})} + \frac{\lambda_{ej}}{2\mu_{ej}(\mu_{ej} - \lambda_{ej})} + \frac{\lambda_{je}}{2\mu_{je}(\mu_{je} - \lambda_{je})} + \frac{1}{\mu_{ei}} + \frac{1}{\mu_{ie}} + \frac{1}{\mu_{ji}} + \frac{1}{\mu_{ij}} + \frac{1}{\mu_{ej}} + \frac{1}{\mu_{je}} + \frac{h_c}{X_P}(d_e e + d_i i + d_j j) + \frac{\lambda_{ei}}{2\mu_i(\mu_i - \lambda_{ei})} + \frac{\lambda_{ij}}{2\mu_j(\mu_j - \lambda_{ij})} + \frac{1}{\mu_i} + \frac{1}{\mu_j} + \frac{\lambda_{ei}}{C_s} + \frac{\lambda_{ij}}{C_s'} \tag{22}$$

V. Results and Discussion

In this section, we discussed the performance of the model. In this model, data is transferred from one layer to another layer started from IoT devices and reaches to cloud through a fog environment. The time consumed by data in travelling is calculated. As data is classified, it is processed and as per requirement, the data is sent to the end-user or cloud. To complete the research task, we use the tool of python editor. The result will be visualized after the completion of the simulation process. Here, the data set is divided into tenfold as we applied a k-fold random forest learning algorithm. 70% of the data set will use for training purposes whereas 30% of data used for testing purposes. Python 3.7 is used as a platform for implementing this work. Random forest algorithm classifies the data in high risk, low risk, and normal with the accuracy of 92% in the proposed work. It has taken 14 seconds as computation time.

For the simulation, we performed several tests for monitoring devices with five different configurations. The evaluation of latency, usage of the network, and consumption of RAM were performed by the simulations. The ifogsim [11] simulator is used to simulate the fog network and nodes. The evaluation of the transmission delay, computation delay, and network delay is simulated through the ifogsim [11] simulator. This simulator creates the physical topologies and they are programmed with Java API. JSON file format is used to store the updated and modified topologies. By varying the size of topology, the performance of simulation is evaluated.

The FNs are swapping the data packets among the system entities during the simulation. The wi-fi connection is established between Fog and IoT devices. In the process of testing the performance, different topologies parameters are used concerning the different number of fog and IoT devices. IoT-sensor, FNs, and cloud data centers as servers are used as physical topology parameters in the simulating tool. By varying the size of topology, the performance of simulation was evaluated.

All configurations (number five) such as configure.1, configure.2, configure.3, configure.4, and configure.5 are simulated with physical topologies on the simulated tool. This system is generated for the performance analysis of proposed work in the fog computing environment. The IoT_sensor device has 1200 million instructions as a CPU length, a network-length of 21000 bytes, and inter-arrival time (average) at data packet arrival of 20 ms.

The details of the used fog device, IoT-sensor, and link of the network are shown in Table VII and Table VIII.

TABLE VII. Details of Fog Device Parameters

Type of device	Processing speed (G.Hz)	Ram (MegaBytes)
Fog-device	2.60	2.0
Cloud-server	4.0	4.0

TABLE VIII. Details of Network Link Parameters

Source node	Destination node	Latency (ms)
IoT_sensor1	Fog-device	45.0
IoT_sensor2	Fog-device	45.0
IoT_sensor3	Fog-device	50.0
Fog-device	Cloud-server	75.0

A comparison in transmission delay between fog and cloud environment is shown in Fig. 3. First of all, a link (tuple) is generated by IoT-sensors, and the connection is established with available routers, gateways, and connected FNs. After getting the data packet on fog servers, processing and distributing to other neighbouring FNs, then data packets are received by the end-user.

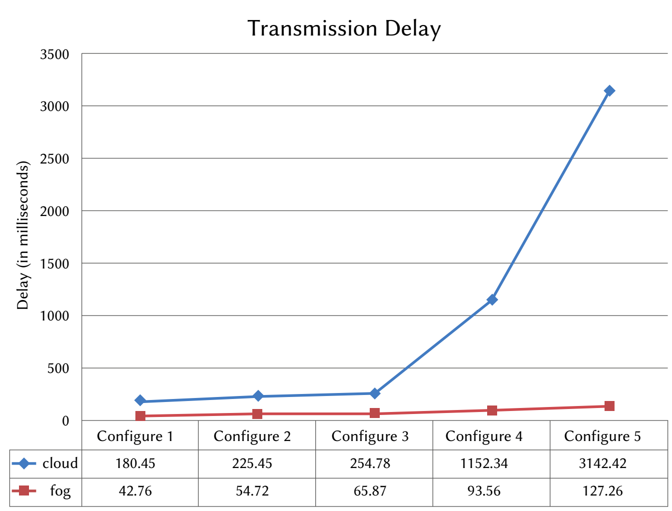

Fig. 3. Fog computing and cloud computing comparison for T_D.

The comparison of network delay between fog and cloud computing is shown in Fig. 4. When the transmission of data occurs between

IoT-sensors and fog servers, the hop counts decreases. Fig. 4. shows the reduction in network latency. When there is a large volume of data transmitted between IoT-sensors and cloud servers, there is an important increase in network latency for the cloud network while this is kept low for the fog network.

The comparison between fog and cloud computing computation delay is shown in Fig. 5. When task data reaches to FNs, it starts computing the data, and the computation depends upon the parameters such as the speed of the processing unit, hardware performance, and size of the data packet.

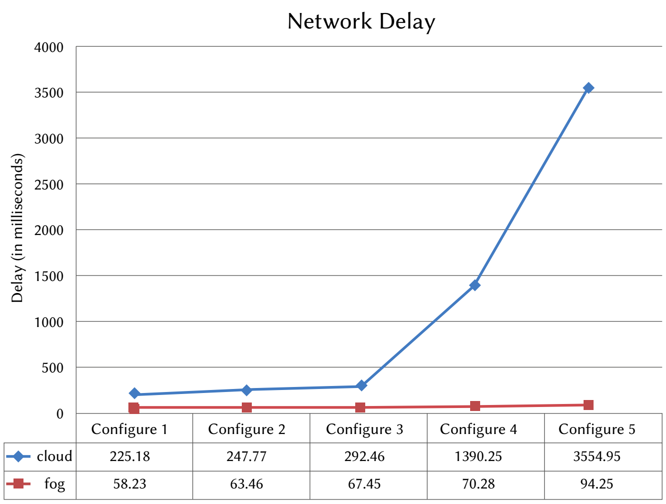

Fig. 4. Fog computing and cloud computing comparison for N_D.

Fig. 6 shows the consumption of usage of networks in fog and cloud computing environments. FNs are deployed over certain regions to overcome network congestion.

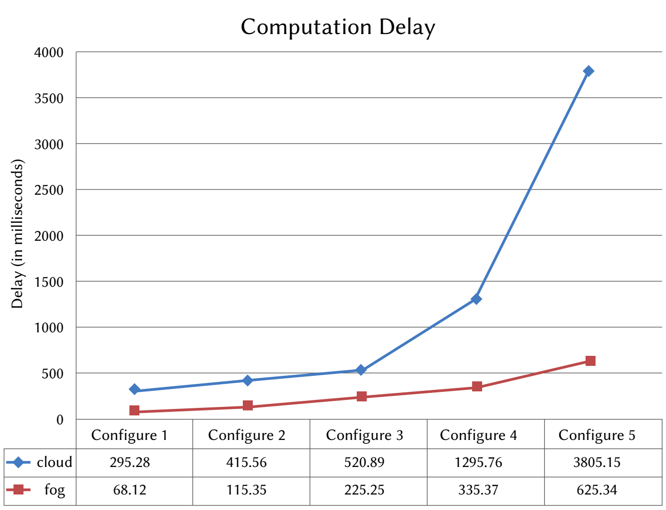

Fig. 5. Fog computing and cloud computing comparison for C_D.

In these simulation results, the various physical topology configurations are used in the fog computing environment. As a result, the average result of transmission delay is 76.834 ms, the average result of network delay is 70.734 ms, and 273.886 ms for average computation delay. The usage of the network is also minimized with the average result is kilo bytes. The existing state-of-art is compared with the proposed algorithm that minimized latency by 94-95%. We compared the proposed model by Hermes [23], iFogStor [18], and Hipster [16], where an improvement in the minimization of latency is by 16% with the model presented by Hermes, an 86% reduction in latency is demonstrated as a comparison to cloud computing by iFogStor, and Hipster improves the latency for web-searching by 80-90%. Raafat [32] shows the reduction in overall service latency by 21.9% to 46.6% in the fog environment.

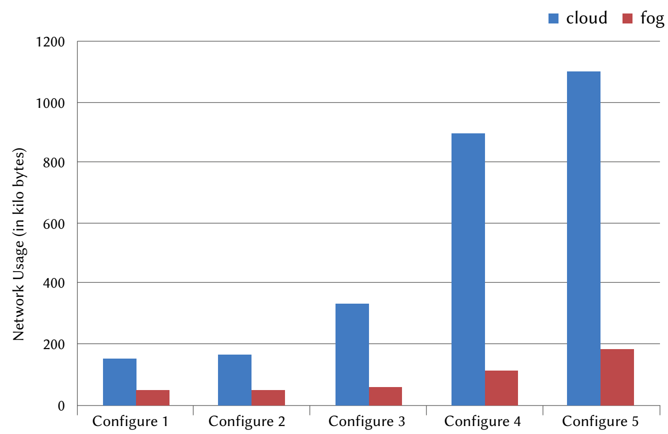

Fig. 6. Fog computing and cloud computing comparison for Network Usage.

VI. Conclusion

Classification of health data and minimization of latency is the most challenging task in e-healthcare, where the fog server is receiving a high volume of task data. Due to the complicated nature of data, fog computing technology becomes essential and important to minimize latency in e-healthcare. In this paper, we presented a novel intelligent multimedia data segregation (IMDS) scheme using machine learning (k-fold random forest) in the fog computing environment. The latency parameters such as transmission delay, network delay, and computation delay are evaluated and it shows the reduction in the high latency. The proposed model is improving the quality of service in e-healthcare and suitable for heterogeneous networks. The latency and usage of the network is a part of QoS. Hence, minimizing the latency and usage of network improves the QoS. In the future, the quality of services in e-healthcare and latency for high-risk data can be improved by using 5G as higher internet connectivity. A smart healthcare system can be implemented in a different hospital through the fog model.

Acknowledgment

The authors would like to thanks to Department of Computer Science & Information Technology, Sam Higginbottom University of Agriculture, Technology and Sciences (SHUATS), Allahabad, U.P., India to give this platform to work. Thanks to all my seniors in the department.

Conflicts of interest/Competing interests – There are no conflicts of interests.

References

[1] C. S. Nandyala and H. K. Kim, "From cloud to fog and IoT-based real-time U-healthcare monitoring for smart homes and hospitals", *International Journal of Smart Home*, vol. 10, no. 2, 2016, pp. 187-196.

[2] S. Cirani, G. Ferrari, N. Iotti, M. Picone M, "The IoT hub: A fog node for seamless management of heterogeneous connected smart objects", 12th Annual IEEE International Conference on Sensing, Communication, and Networking-Workshops (SECON Workshops), 2015.

[3] L.M. Vaquero, L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive deffinition of fog computing", ACM SIGCOMM

Computer Commun. Review, vol. 44, no. 5, 2014, pp. 27–32.

[4] F. Bonomi, R. Milito, J. Zhu, S. Addepalli S, "Fog computing and its role in the internet of things", in Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, 2012, pp. 13–16.

[5] S. Yi, C. Li C, Q. Li Q, "A survey of fog computing: Concepts, applications and issues", in *Proceedings of the 2 Workshop on Mobile Big Data, ACM,* 2015, pp. 37–42.

[6] P. G. Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, E. Riviere, "Edge-centric computing: Vision and challenges", SIGCOMM Computer Communication Review, vol. 45, no. 5, 2015, pp. 37–42.

[7] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities", *IEEE Internet of Things Journal*, vol. *3, no.* 6, 2016, pp. 854-864.

[8] T. N. Gia, M. Jiang, A. M. Rahmani, T. Westerlund, P. Liljeberg, H. Tenhunen, "Fog computing in healthcare internet of things: A case study on ecg feature extraction", in *2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing, IEEE,,* 2015, pp. 356-363.

[9] S.C. Hung, D. Liau, S.Y. Lien, K.C. Chen, "Low latency communication for Internet of Things", in *IEEE/CIC International Conference on Communications in China (ICCC), IEEE, 2015,* pp. 1-6.

[10] G. Lee, W. Saad, M. Bennis, "An online optimization framework for distributed fog network formation with minimal latency", *IEEE Transactions on Wireless Communications*, vol. *18, no.* 4, 2019, pp. 2244-2258.

[11] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments", *Practice and Experience*, vol. *47, no.* 9, 2017, pp.1275-1296.

[12] L. Skorin-Kapov and M. Matijasevic, "Analysis of QoS requirements for e-health services and mapping to evolved packet system QoS classes*", International journal of telemedicine and applications.* 2010.

[13] J. R. Gallego, A. Hernandez-Solana, M. Canales, J. Lafuente, A. Valdovinos, J. Fernandez-Navajas, "Performance analysis of multiplexed medical data transmission for mobile emergency care over the UMTS channel", *IEEE transactions on information technology in biomedicine. Vol. 9, no. 1,* 2005, pp.13–22.

[14] C. A. Silva, G. S. Aquino, S.R.M. Melo, D. J. B. Egidio, "A Fog Computing-Based Architecture for Medical Records Management", *Wireless Communications and Mobile Computing,* 2019.

[15] M. G. R. Alam, Y. K. Tun, C. S. Hong, "Multi-agent and reinforcement learning based code offloading in mobile fog", in *International Conference on Information Networking (ICOIN) , IEEE,* 2016, pp. 285-290.

[16] R. Nishtala, P. Carpenter, V. Petrucci, X. Martorell, "Hipster: Hybrid task manager for latency-critical cloud workloads", in *IEEE International Symposium on High Performance Computer Architecture (HPCA), IEEE* 2017, pp. 409-420.

[17] T. N. Gia, M. Jiang, V. K. Sarkar, A. M. Rahmani, T. Westerlund, P. Liljeberg, H. Tenhunen H, "Low-cost fog-assisted healthcare IoT system with energy-efficient sensor nodes" in Proceedings of *13th international wireless communications and mobile computing conference (IWCMC) , IEEE,* 2017, pp. 1765-1770.

[18] M. I. Naas, P. R. Parvedy, J. Boukhobza, L. Lemarchand, "iFogStor: an IoT data placement strategy for fog infrastructure", in *IEEE 1st International Conference on Fog and Edge Computing (ICFEC),* 2017, pp. 97-104.

[19] A. M. Rahmani, T. N. Gia, B. Negash, A. A. I. Azimi, M. Jiang, P. Liljeberg, " Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach", *Future Generation Computer Systems*, vol. *78*, 2018, pp. 641-658.

[20] J. Wu, M. Dong, K. Ota, J. Li, Z. Guan, "FCSS: Fog computing based content-aware filtering for security services in information centric social networks", *IEEE Transactions on Emerging Topics in computing, 2017.*

[21] A. Brogi and S. Forti, "QoS-aware deployment of IoT applications through the fog", IEEE Internet of Things Journal, vol. 4, no. 5, 2017, pp.1185–1192.

[22] A. Shahzad, Y. S. Lee, M. Lee, Y. G. Kim, N. Xiong, "Real-Time Cloud-Based Health Tracking and Monitoring System in Designed Boundary for Cardiology Patients", *Journal of Sensors*, 2018.

[23] Y. H. Kao, B. Krishnamachari, M. R. Ra, F. Bai, "Hermes: Latency optimal task assignment for resource constrained mobile computing". *IEEE Transactions on Mobile Computing*, vol. *16, no.* 11, 2017, pp. 3056-3069.

[24] G. Li, J. Wu, J. Li, K. Wang, T. Ye, "Service popularity-based smart resources partitioning for fog computing-enabled industrial Internet of Things", *IEEE Transactions on Industrial Informatics, vol. 14, no. 10,* 2018, pp.:4702–4711.

[25] R. Mahmud, F. L. Koch, R. Buyya, "Cloud-fog interoperability in IoT-enabled healthcare solutions", in *Proceedings of the 19th international conference on distributed computing and networking,* 2018, pp. 1-10.

[26] N. T. Dinh and Y. Kim, "An Efficient Availability Guaranteed Deployment Scheme for IoT Service Chains over Fog-Core Cloud Networks", *Sensors. 18(11):3970,* 2018.

[27] M. M. Ahsan, I. Ali, M. Imran, M. Y. I. Idris, S. Khan, A. Khan, "A Fog-centric Secure Cloud Storage Scheme", *IEEE Transactions on Sustainable Computing,* 2019.

[28] H. Rafique, M. A. Shah, S. U. Islam, T. Maqsood, S. Khan, C. Maple, "A Novel Bio-Inspired Hybrid Algorithm (NBIHA) for Efficient Resource Management in Fog Computing", *IEEE Access, vol. 7,* 2019, pp. 115760–115773.

[29] J. Li, T. Zhang, J. Jin, Y. Yang, D. Yuan, L. Gao, "Latency Estimation for Fog-based Internet of Things", in *27th International Telecommunication Networks and Applications Conference (ITNAC), IEEE,* 2017, pp. 1-6. 2017.

[30] C. Thota, R. Sundarasekar, G. Manogaran, R. Varatharajan, M. K. Priyan, " Centralized fog computing security platform for IoT and cloud in healthcare system", in *Fog computing: Breakthroughs in research and practice, IGI global,* 2018, pp. 365-378.

[31] T. Aladwani, "Scheduling IoT Healthcare Tasks in Fog Computing Based on their Importance", in *Proc of 16th International Learning & Technology Conference, Procedia Computer Science*, 2019, pp. 560-569.

[32] O. A. Raafat, A. K. Mazin, L. Taha, E. F. Khaled, "Scheduling Internet of Things Requests to Minimize Latency in Hybrid Fog-Cloud Computing", *Future Generation Computer Systems*, *111*, 2020, pp. 539-551.

[33] J. Pan J and J. McElhannon, "Future edge cloud and edge computing for internet of things applications", IEEE *Internet of Things Journal, vol. 5, no. 1,* 2017. pp. 439–49.

[34] Z. Alam, M. S. Rahman, M. S. Rahman, "A Random Forest based predictor for medical data classification using feature ranking", *Informatics in Medicine Unlocked*, *15*, 100180, 2019.

[35] C. Chakraborty, "Computational approach for chronic wound tissue characterization", *Informatics in Medicine Unlocked*, *17*, 100162, 2019.

[36] C. Chakraborty C and R. Roy, "Markov Decision Process based Optimal Gateway Selection Algorithm" International Journal of Systems, Algorithms & Applications (IJSAA), 2012, pp. 48-52.

[37] C. Chakraborty, R. Roy, S. Pathak, S. Chakrabarti, "An Optimal Probabilistic Traffic Engineering Scheme for Heterogeneous Networks", *Int. Journal of Fuzzy Systems, vol. 3, no. 2,* 2011, pp. 35-39.

[38] M. Nilashi, H. Ahmadi, A. A. Manaf, T. A. Rashid, S. Samad, L. Shahmoradi, A. Nahla, A. Elanz, "Coronary Heart Disease Diagnosis Through Self-Organizing Map and Fuzzy Support Vector Machine with Incremental Updates", *International Journal of Fuzzy Systems*, 2020, pp.1-13.

[39] T.A. Rashid, S. M. Abdullah, R. M. Abdullah, "An intelligent approach for diabetes classification, prediction and description", *Innovations in Bio-Inspired Computing and Applications,* Springer, Cham, 2016, pp. 323-335.

[40] H. Mahmud, M. Mohammadi,, N. Ali, T. A. R. Khan, K. A. S. Nawzad, R. M. D. Omer, L. Joan, " Technologies in medical information processing", *Advances in Telemedicine for Health Monitoring, Technologies, Design and Applications*, 31, 2020.

[41] T. A. Rashid, M. K. Hassan, M. Mohammadi, K. Fraser, "Improvement of variant adaptable LSTM trained with metaheuristic algorithms for healthcare analysis", *Advanced Classification Techniques for Healthcare Analysis*, IGI Global, 2019, pp. 111-131.

[42] J. C. W. Lin, Y. Shao, P. Fournier-Viger, F. Hamido, "BILU-NEMH: A BILU neural-encoded mention hypergraph for mention extraction", *Information Sciences*, *496*, 2020, pp. 53-64.

[43] J. C. W. Lin, Y. Shao, J. Zhang, U. Yun, "Enhanced sequence labeling based on latent variable conditional random fields", *Neurocomputing*, *403*, 2020, pp. 431-440.

[44] U. Ahmed, J. C. W. Lin, G. Srivastava, M. Aleem, "A load balance multi-scheduling model for OpenCL kernel tasks in an integrated cluster", *Soft Computing*, 2020, pp. 1-14.

[45] T. A. Rashid, P. Fattah, D. K. Awla, "Using accuracy measure for improving

the training of LSTM with metaheuristic algorithms", *Procedia Computer Science*, *140*, 2018, pp. 324-333.

[46] A. Janosi, M. Pfisterer, W. Steinbrunn, R. Detrano, J. Schmid, S. Sandhu, K. Gupta, S. Lee , V. Froelicher, UCI Machine Learning Repository 2019. https://archive.ics.uci.edu/ml/datasets/heart+disease

Amit Kishor

Er. Amit Kishor pursuing a Ph.D. in Computer Engineering from the Department of Computer Science and Information Technology, in the Faculty of Engineering and Technology, Sam Higginbottom University of Agriculture, Technology and Sciences, Naini, Allahabad, India. He is also working as an Assistant Professor in the Department of Computer Science and Engineering & I.T., Subharti Institute of Engineering and Technology, Swami Vivekanand Subharti University, Meerut. India His area of interest is cloud computing, Algorithms, Artificial Intelligence, and Data Structures. He has published more than 20 papers in reputed international journals. He is also a member of an International body of International Association of Engineers (IEANG).

Chinmay Chakraborty

Dr. Chinmay Chakraborty is working as an Assistant Professor (Sr.) in the Dept. of Electronics and Communication Engineering, Birla Institute of Technology, Mesra, India. He worked at the Faculty of Science and Technology, ICFAI University, Agartala, Tripura, India as a Sr. lecturer. He worked as a Research Consultant in the Coal India project at Industrial Engineering & Management, IIT Kharagpur. He worked as a project coordinator of the Telecommunication Convergence Switch project under the Indo-US joint initiative. He also worked as a Network Engineer in System Administration at MISPL, India. His main research interests include the Internet of Medical Things, Wireless Body Area Network, Wireless Networks, Telemedicine, m-Health/e-health, and Medical Imaging. Dr. Chakraborty has published sixty papers at reputed international journals, conferences, book chapters, and books. He is an Editorial Board Member in the different Journals and Conferences. He is serving as a Guest Editor of MDPI-Future Internet Journal, Internet Technology Letters, Wiley, and has conducted a session of SoCTA-19, ICICC – 2019, and also a reviewer for international journals including IEEE Access, Elsevier, Springer, Taylor & Francis, IGI, IET, TELKOMNIKA Telecommunication Computing Electronics and Control, and Wiley. Dr. Chakraborty is co-editing Eight books on Smart IoMT, Healthcare Technology, and Sensor Data Analytics with CRC Press, IET, Pan Stanford, and Springer. He has served as a Publicity Chair member at renowned international conferences including IEEE Healthcom, IEEE SP-DLT. Dr. Chakraborty is a member of Internet Society, Machine Intelligence Research Labs, and Institute for Engineering Research and Publication. He received a Young Research Excellence Award, Global Peer Review Award, Young Faculty Award, and Outstanding Researcher Award.

Wilson Jeberson

Prof.(Dr.) Wilson Jeberson was awarded Ph.D. degree in Computer Science and Communication, from Sam Higginbottom Institute of Agriculture, Technology & Sciences, University, Allahabad, India. He has received the MCA in computer Application and MBA in Management from Madurai Kamaraj University Tamilnadu, India. He had worked as Programmer at National Informatics Centre (NIC), Govt. of India, from 1999 to 2000. He also worked as Senior Software Engineer cum DBA at Quintessence Technologies Limited - Trivandrum, Kerala, India, from 2000 to 2002 and as Senior System Analyst at Netcare Technologies-Trivandrum, Kerala, India from 2002 to 2003.Currently he is working as Professor & Head, Department of Computer Science & Information Technology in Sam Higginbottom University of Agriculture, Technology & Sciences. Prayagraj, India from 2003. He has published more than 70 papers in reputed International journals and more than 15 Papers in National & International Proceedings. Some of the best papers are published in Open Computer Science (Formerly: Central European Journal of Computer Science) Published by De Gruyter Open (Formerly: Springer Verlag) titled "Survey of Context Information Fusion for Ubiquitous Internet-of- Things (IoT)", International Journal Elsevier, Science Direct titled "Covert Communication Using Arithmetic Division Operation", :International Journal Elsevier , Science Direct Titled "Fortune at the bottom of the Classifier Pyramid: A novel approach to Human Activity Recognition". A research paper was selected as best one among the top 20 papers presented in the Second GMSARN International Conference at Pattaya, Thailand and was published in International Journal of AIT , Thailand titled "An approach to ICT enabled solution architecture for critical Social Security issues and challenges for e-Governance". He has co-authored two books which are published in Narosa Publications and another in Springer International. He has presented research papers in international conferences in Singapore and Thailand and also attended a training program in Galilee International institute of Management, Israel. He is also member of many national and international bodies including Computer Society of India (CSI) and International Association of Engineers (IAENG), Hong Hong. He was the Principal investigator for two funded projects one funded by Computer Society of India Education Directorate, Chennai-India and another by SHIATS. He is one of the editorial board members of various journals including Scientific & Academic Publishing Co., Journal Name: Software Engineering, USA.