Copyright ⓒ 2024 KSII

Enhancement of Semantic Interoperability in Healthcare Systems Using IFCIoT Architecture

Sony P[1,2], Siva Shanmugam G[1*], and Sureshkumar Nagarajan[3]
[1] School of Computer Science and Engineering,
Vellore Institute of Technology, Vellore
Tamil Nādu, India
[2]Department of Computer Science,
Govt. Model Engineering College (Managed by IHRD), Thrikkakara,
Kerala, India
[email :sony@mec.ac.in]
[3]Department of CSE, School of Computing,
Kalasalingam Academy of Research and Education, Krishnankovil,
Tamil Nādu, India
[*]Corresponding author: Siva Shanmugam G

Received December 15, 2022; revised November 19, 2023; accepted March 27, 2024; published April 30, 2024

Abstract

Fast decision support systems and accurate diagnosis have become significant in the rapidly growing healthcare sector. As the number of disparate medical IoT devices connected to the human body rises, fast and interrelated healthcare data retrieval gets harder and harder. One of the most important requirements for the Healthcare Internet of Things (HIoT) is semantic interoperability. The state-of-the-art HIoT systems have problems with bandwidth and latency. An extension of cloud computing called fog computing not only solves the latency problem but also provides other benefits including resource mobility and on-demand scalability. The recommended approach helps to lower latency and network bandwidth consumption in a system that provides semantic interoperability in healthcare organizations. To evaluate the system's language processing performance, we simulated it in three different contexts. 1. Polysemy resolution system 2. System for hyponymy -hypernymy resolution with polysemy 3. System for resolving polysemy, hypernymy, hyponymy, meronymy, and holonymy. In comparison to the other two systems, the third system has lower latency and network usage. The proposed framework can reduce the computation overhead of heterogeneous healthcare data. The simulation results show that fog computing can reduce delay, network usage, and energy consumption.

Keywords: Fog Computing, Interoperability, Internet of Medical Things, Ontology, UMLS

1. Introduction

In the healthcare sector, semantic interoperability refers to a healthcare system's ability to read and process medical data generated or controlled by another healthcare system. In the Internet of Things age, healthcare data must be monitored in real-time, relevant decisions made, and alerts issued in seconds. Furthermore, the data generated by IoT devices is massive, primarily in big data. Medical data from IoT devices and unstructured electronic health records involve complex processing to provide timely health warnings. Developing all the solutions in the cloud causes a lot of network congestion and computational delays.

When a patient receives an infusion, necessary treatments, surgery, intensive care, or in ambulatory services require real-time monitoring. Real-time monitoring may occasionally save lives in critical cardiac emergencies, strokes, and to name a few. Real-time monitoring in healthcare necessitates an immediate response. When data is collected from IoT devices and delivered to the cloud for processing, the real-time monitoring system is hampered by its inability to receive a timely response from the cloud. In most circumstances, IoT Healthcare devices do not have enough computing power, and Fog Computing was conceived to meet these requirements.

Fog computing is used in many sectors, including agriculture, clinical decision support systems[1], logistics, transportation, etc. The literature review on fog computing is summarized in **Table 1**. Fog computing methodology can be used in healthcare for providing health monitoring[2] and real-time notifications and can play an important role in decision support systems [3]. In healthcare, fog computing may be used to monitor conditions such as high blood pressure[4] , arthritis [5] and diabetes[3] to name a few, and is an excellent resource for geriatric care [6].

Cloud services are extended to IoT devices through the Fog layer [7], the middle layer between the cloud and the device. Fog Computing is applicable in time-critical applications such as data analysis of road traffic to predict accidents, medical emergency analysis, etc. Fog computing is applicable if the data needs to be analyzed within a fraction of a second and is used when the number of devices is huge. They are separated by a considerable geographical distance.

The edge layer, located very close to the device layer, can handle the data obtained from the physical layer and an intermediary layer between the fog and device layers. Processing data in real time at the source is made possible by edge layers. As a result, the computing operations are effectively moved to the network's edge. In other words, this process takes place much closer to the source of the data rather than piping all the data back up to the cloud for analysis and action. Routers, switches, and even IoT sensors collecting the data can all be considered edge devices if they have adequate processing power.

Researchers conducted both simulated and real-time fog computing trials in healthcare[8]. As real-time healthcare datasets are very few and the volume of the accessible datasets is small, most researchers relied on simulation. For modeling many types of applications, simulation systems should include capabilities for physical or virtual resources, network architecture, control methods, and data management. There are several simulators available, including iFogSim[9] edge-fogcloud [10], iFogSim++, iFogSim2[11], and others. Some simulators like edge-fog-cloud [12] were written in Python, while others iFogSim2.0[11]was written in Java. Many researchers stated that iFogSim 2.0 is a reasonable alternative for simulating fog environments since it has lower latency and less network congestion than other simulators.

The phrase "fog computing" is not new in the field of healthcare research. Prediction, classification, and entity recognition are just a few of the applications that use fog computing.

Using the edge and fog computing paradigms, researchers are currently attempting to resolve healthcare's semantic interoperability problems. Most solutions to the semantic interoperability issue did not rely on real-time processing, but relied on archived data.

In 2020. Jaleel et.al offered a technique to address the issue of data interoperability by leveraging collaborative healthcare equipment for real-time processing. The suggested approach could be able to shorten the latency. The fact that the repeated data was not taken into account is one of the suggested model's limitations. For example, a patient's temperature was recorded by an IoTMD device as 99.7F, 99.7F, 99.7F, and 98.4F at 10 A.M., 11 A.M., 12 P.M., and 1 P.M., respectively. The computing system has to handle each temperature independently even if the data is the same from 10 a.m. to 1 p.m. In order to process each healthcare solution, previous data is required. The historical records may be structured or unstructured. Therefore, while creating an IoT healthcare solution, we should take of all kinds of data.

1.1 Semantic Interoperability in Healthcare IoT

Interoperability refers to the ability of computer systems or software to share and utilize information. Suppose Healthcare System A indicates that a patient has been confirmed to have Cancer, and healthcare System B reveals that the same patient has been detected with Carcinoma. In that case, the two systems should understand that these two sets of terms have the same meaning[13]. It is difficult to convey and comprehend clinical meaning in healthcare, especially when utilizing a healthcare information system (HIS). Healthcare practitioners devote significant time to investigating and analyzing the complexity and relationships between clinical data. Single-word phrases like Cancer, thyroid, multiword phrases like very high temperature, thyroid glands, and abbreviations like I.T. (Intrathecal) and IV(Intravenous) are all examples of medical terms found in the Electronic Health Record (E.H.R.). These words are written in uppercase or lowercase characters, or a combination of both, resulting in distinct strings for the same term. (Cancer, CANCER, and Cancer all refer to the same thing.) Polysemous words are used in the medical field with different meanings in different content. In one sense, the name "ALL" stands for acute lymphoblastic leukemia, while in another, it stands for "face all." Some medical words are synonymous (e.g.: Cancer and Carcinoma). There are specific unique associations between words, such as is-a and part-of. Cancer is a disease that affects the liver. The thyroid illness is hypothyroidism. Leg discomfort includes left leg pain. Another big issue with medical documentation is Named Entity Recognition. In medical terminology, nouns can be a person’s name, disease’s name, a symptom’s name, a procedure’s name, and so on.

Synonyms and co-synonyms can exist within a clinical environment. The heterogeneity of electronic health records is increased by the presence of hyponym and hypernym relationships and co-hyponyms. Another significant issue with medical documents is the use of holonymy and meronymy. Polysemy problems also exist in medical documents, which causes ambiguity in healthcare concepts. Each EHR uses a separate medical document format, notations, and terminology. Similarly, the terminology used by one IoT-Medical Device and its medical acronyms, data format, and measurement units vary. Organizations must use uniform terminology systems that reflect the more significant healthcare industry to deliver safe, high-quality treatment and achieve semantic interoperability in the future.

1.2 Why We Need Fog Computing in Healthcare

Fog computing is a networking system that involves edge devices to do much of the computation (edge computing), storage, and communication, locally before routing it over the

Internet backbone. The term "fog" refers to the cloud-like qualities of IoT devices closer to the ground. Rather than transmitting all of this data to cloud-based servers to be analyzed, fog computing tries to accomplish as much processing as possible, leveraging computing units co-located with the data-generating devices such as sensors and smart healthcare devices.

As sensor-layer IoT devices are limited in processing and storage capacities, healthcare IoT may rely on supporting technologies such as cloud computing. However, healthcare may rely on supporting technologies such as fog computing to overcome bandwidth, latency overhead, and security issues. According to [14], fog computing is an appropriate platform for addressing latency challenges in healthcare.

Contributions of the paper is listed as follows

1. Healthcare semantic interoperability is envisaged using fog computing.
2. A data granularity architecture is used to integrate the IoTMD data.
3. Four algorithms have been developed to address lexical, morphological, and linguistic issues with the medical documents.
4. Utilizing simulation studies with iFogSim2.0, which meets the requirements for reduced latency, reduced energy consumption, reduced network usage, reduced simulation time, and reduced migration time, the system's performance is assessed.

The article is organized as follows. Section 2 presents the overview of the literature review, and section 3 depicts the proposed system architecture. Implementation details are represented in section 4. section 5 discusses the obtained result and section 6 concludes the article.

2. Related Works

Healthcare IoT devices' large volume of data necessitates latency-sensitive processing, which is impossible when the applications are deployed in remote cloud data centers [15]. A fog node is placed between the device and application layer to resolve the latency issues. A fog node can be any device that includes processing, storage, and network connectivity element [16] . In most healthcare fog solutions, a three-layer architecture with the bottom layer as the device layer, the middle layer as Fog, and the upper layer as the Cloud layer was proposed[17].

Even before 2010, several researchers pointed out that cloud computing was vulnerable to latency and bandwidth difficulties. At an ACM conference in 2014, CISCO presented Fog computing[14] to the world. A solution proposed by Malik et.al [17]consisted of three-layer architecture. The bottom layer was the Device layer, the middle layer was Fog, and the upper layer was the Cloud layer. When the number of sensing devices increases considerably, problems such as the processing time of time-critical IoT applications will increase due to network congestion caused by offloading data to the cloud and uploading data from many IoT generators [12].

Table 1. Related Works

Article	Implementation	Tools / Technique used	Application area	Evaluation Performed
[4]	Real Time, Archived	OMNET++ ,Bayesian Classifier, Thread Protocol	Arthritis analysis in human beings	Packet delivery rate, Packet delivery ratio, and Response Time
[2]	Real Time, Archived	Type-2 neutrosophic and VIKOR technique	Type 2 diabetes diagnosis	Comparative analysis with TOPSIS and SAW Execution time between

[5]	Real time, Archived	MySignals, HW V2, Arduino Uno, ESP 8266	Elderly care	optimization and non-optimization SUS(System Usability Scale
[14]	Not Implemented		Diabetic device data processing	
[3]	Real time using datasets, Archived	KNN, MLP,LR, ANN	Hypertension attack	Classification Accuracy, Classification response time, delay time

Numerous simulators were created to mimic the fog computing environment. Some of these include iFogSim, Edge-Cloud Fog, etc. A simulator titled Edge Cloud Fog [12] was created in 2017 using a three-layer, node-oriented methodology. The Fog was positioned between the data store and edge layer, with an edge as the outermost layer and a datastore as the innermost.

Rajkumar Buyya and his team developed iFogSim[18] simulator based on two processing models, the sense-process-actuate and stream processing models. The developed simulator was based on Java language. It supports the mobility of fog nodes.

Some authors have proposed a Fog computing-based paradigm to diagnose diseases. Sood et.al [4] proposed a fog-based healthcare framework to identify and monitor hypertension attacks in human beings, and they used an artificial neural network to predict hypertension. Internet of Things and Fog Computing presented a real-time deployment of an E-health system for monitoring the health of older people [6]. The My Signals H.W. V2 platform and an Android app that functions as a fog server were used to collect pharmacological and overall health parameters from the elderly regularly. The parameters were collected, evaluated, and cached before being sent to the cloud via a specific REST API using the JSON data model. Data distribution, communication, and management layers were the three layers in the suggested design. The data distribution layer makes use of MongoDB. A decision support system for healthcare IoT was created using soft computing and fog computing principles. The six levels of the Fog computing architecture in the suggested research [3] were the physical and virtualization layer, monitoring layer, pre-processing layer, temporary storage layer, security layer, and transport layer.

Ahmed et. al[19] have suggested another fog computing-based healthcare monitoring system. The suggested system consisted of three layers: a sensor network, a fog, and a cloud layer. The sensor network layer contains a variety of sensors, such as E.C.G. sensors, B.P. sensors, and temperature sensors. The fog layer handled data security, compression, notification service, data analysis, and local storage. The cloud layer handled big data processing, ample data storage, and disease prediction. One of the primary drawbacks of the suggested technique was that it was only applicable to images.

For handling diabetic healthcare IoT data, David et al. [20] compare cloud computing with edge/fog computing environments. The bottom layer comprises diabetic instruments with sensors, while the intermediate layer includes intelligent devices, hubs, routers, and gateways. The top layer, referred to as the Cloud layer, contains all computing components. Hassen et al. Ben [6]created a real-time home hospitalization system using the healthcare and environmental parameters of five persons aged 56 to 61. The data was gathered using My signal hw V2 in their proposed system, while Android phones and tablets are used as fog servers. The system's main drawback was that the dataset's size was too small. Using the Fog computing paradigm, a medical decision support system based on fuzzy set theory was

presented for healthcare IoT by Abdel[3]. Applying type 2 neutrosophic and the V.I.K.O.R. technique, a type 2 diabetes diagnosis method was developed.

Bayesian classifier was used to analyze human arthritis in a fog computing environment [5], while the THREAD Protocol was the communication protocol applied in the proposed system. A total of 431 arthritis patients were included in the case study. OMNet++ was used to design the application. Three evaluation metrics, namely packet delivery rate, packet delivery ratio, and response time, were tested on the proposed system with fog, fog, and thread environment, and without fog and thread environment to validate the results. Thread and fog environments had the shortest response times, but their packet delivery ratios were the highest.

Sood et al.[4] suggested a healthcare architecture based on IoT fog for identifying human hypertension. The proposed system consists of a cloud system, a fog system, and an IoT-focused user subsystem. Internet-of-things user subsystems deal with a wide variety of data. Data granulation, risk evaluation, stage categorization, and warnings were all handled by the fog subsystem. Within the cloud subsystem, data was kept. Sarkar S et.al [21] compared fog computing against cloud computing and claimed that at least one-fourth of IoT applications require real-time, low-latency services.

Several authors, Markus A et.al [22], Ashouri et.al [23], and Naha R K et.al [24] to mention a few, conducted various analyses on the quality of simulators. They claimed iFogSim was one of the best simulators accessible, due to its event-driven and open-source nature, as well as its implementation language. According to Markus et.al [22], 63% of simulators are event-driven and 70% of them are written in the language Java. According to [23], iFogSim was the optimal tool for modeling environments that demand faster response durations and higher processing usage, as well as lower bandwidth and energy consumption.

3. Proposed Architecture

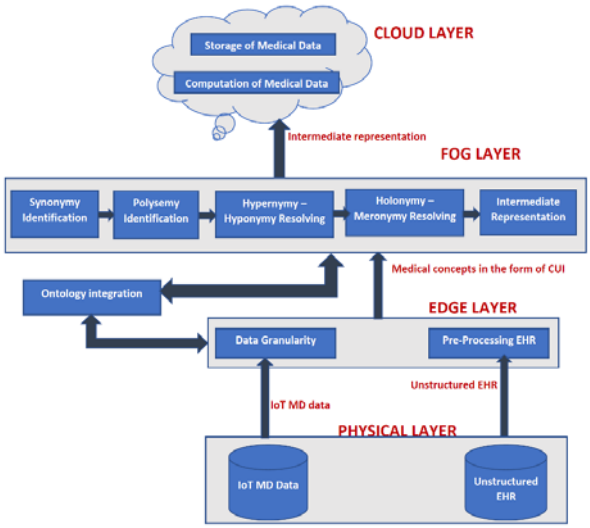

Fig. 1. The Architecture of the Proposed System

Fig. 1 shows the proposed four-layered Edge Fog-based semantic interoperability approach. In this architecture, physical layer devices such as wearable sensors, implanted sensors, wireless and wired IoT devices, environmental sensors, sensors attached to the human body,

and other intelligent healthcare IoT devices can transmit sensed medical data to edge nodes along with their geospatial locations. The physical layer includes unstructured E.H.R. also.

The layer nearer to the physical layer is called the edge layer. The edge layer often uses resources like mobile phones, tablets, laptops, workstations, etc., as computing facilities.
The two computing modules in the edge layer are modules for data granularity and modules for ontology integration. The medical IoT data received from intelligent devices require extra processing with the help of the UMLS ontology for storing and indexing the IoT medical data. The Data Granularity module is in charge of processing IoT medical data. The processed data is transferred to the fog layer, comprised of fog nodes designated as fog gateways. These gateways served as a router, switches, and other computational services, as well as providing storage. Some fog gateways have been renamed fog proxy, providing communication with the third layer, the cloud layer.

After being transformed into an intermediate format comprehensible by all healthcare professionals, it can be readily translated to the target formats, where the processed data is stored in the cloud. In this case, the fog node serves as a link between the cloud and the end devices.

Smartphones, tablets, or desktops P.C. can act as fog gateways. The computational and storage capacities of these devices are limited. The sensed data from the physical layer is transmitted to the cloud for additional processing and calculation after the initial essential computation. Every hospital has one or more fog nodes, which sends the analyzed data to the cloud. If the caregiver needs access to previous data, the fog node may get it from the cloud and make it available. Suppose a caregiver needs medical information from another hospital. In that case, the fog node can retrieve the intermediate from the cloud and send it to the caregiver, who can decode it into the necessary format. All data packets received are not transmitted to the cloud computing module. Only applications with time flexibility are routed to the top levels.

The cloud has the capacity to translate medical data collected from numerous fog nodes into an intermediate format in order to maintain interoperability, and it may store the processed data for a longer period of time.

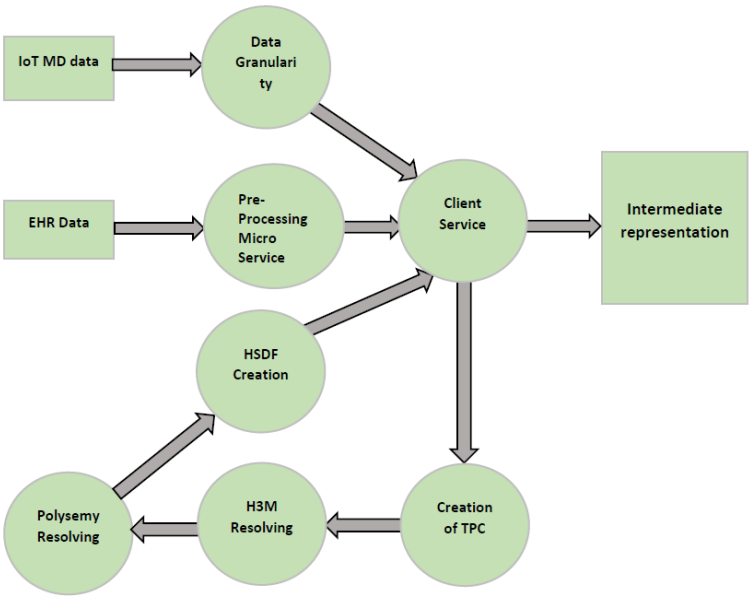

Fig. 2. Micro Service Architecture

Fig. 2 depicts the semantic Interoperability model's whole architecture. Structured health records, unstructured E.H.R. information, and IoT MD data are fed into the model. Unstructured medical records are pre-processed initially. The TPC(Term Phrase Concept) generation module accepts preprocessed data, structured data, and IoT MD data in the next module. Using an ontology called UMLS[25], the TPC generation module retrieves terms, phrases, and concepts from medical data. H3M module receives the extracted concepts. H3M stands for holonymy, hypernymy, hyponymy, and meronymy. The repetition of concepts in this module can be minimized. The H3M module follows, which identifies various linguistic relations. These are sent to the polysemy module, which identifies various polysemous words. Anaphora resolution is the next module. Finally, there is the intermediate representation module.

3.1 iFogSim and Its Components

Rajkumar Buyya and his team created iFogSim [18], an IoT-enabled fog computing simulator, as an expansion of CloudSim Software. Physical components, logical components, and managerial components are the three basic types of components found in iFogSim. Sensors, fog devices, and actuators are all physical components. Logical components include the application edges and modules. The controller and mapping are management components. iFogSim 2.0[11] was created in 2019 as an expansion of iFogSim to accommodate the mobility feature of IoT devices.

The salient features of iFogSim2 are:

1. Data parser module for receiving and processing data from external files.
2. Introduction to module migration mobility management logic for random and directed mobility methods
3. Cloud-centric, non-hierarchical, intra- and inter-cluster implementation options are available.

3.2 Application model

One or many hospitals might be portrayed in the fog environment. The fog computing environment is configured for numerous branches if a hospital has several sister branches. In iFogSim, there is a hierarchical placement approach. If a Fog node cannot handle a module allocated to it, the controller object moves the module to an upper-level fog node. In iFogSim, application models describe the simulated system's underlying workflow. Distinct application models show connections between different modules in different ways. Two types of application models are employed in our suggested work: master salve and sequential.

EHR data acquisition, Pre-processing the E.H.R. data, IoT MD data acquisition, Creation of Terms, phrases, and Concepts, Hypernymy-hyponymy, holonymy-meronymy resolving, polysemy resolving, UMLS integration are all part of the S.I. application model.

The E.H.R. data acquisition microservice is loaded on the hospital's desktop, laptop, or workstation computer. The medical records that have previously been stored are analyzed and transferred to the pre-processing microservice module, which is likewise installed on the hospital's desktop, laptop, or workstation computer. To extract relevant medical terms from EHR data, various pre-processing techniques such as stemming, chunking, and parsing are used. TPC (Words, Phrases, Concepts) generation microservice module is installed on Fog nodes and receives these medical terms. In order to retrieve the Concept Unique Identifier, all relevant words supplied from the pre-processing microservice are validated in the UMLS repository. An SQL database is used to store the UMLS repository. External database UMLS

may be accessed via TPC generation microservice, h3m resolving microservice, and polysemy resolving microservice.

3.3 Mobility of the Fog Node

Both human and intelligent devices' mobility can impact the functioning of any Internet of Things system. We used two types of mobility patterns in this experiment: directed mobility and random mobility [11]. Individuals and IoT devices can move quickly in directed mobility models. To put it into practice, we first identified a set of evenly spaced locations across India. Later, the class Simevents of CloudSim 5.0 is used to simulate the movement of persons and IoT devices. During the direct mobility simulations, a constant speed is maintained.

Random mobility patterns are created using the RandomMobilityGenerator class in the Random mobility model. The user's direction, stopping criteria, and speed are all elements that influence mobility.

Table 2. Sample Real Time IoTMD Monitoring data

Age (ADT) 12yrs	**Allergies-vancomycin**			
27-06-2021	**7 AM**	**8.00 A.M**	**8 A.M**	**9 A.M**
Fantanyl 15mcg		40		
Midazolan 0.9mg				
Paracetamol 0.5mg				
O2				
SPo2		98	96	
EtCO2 4	3.8			
Temperature(F)	99.3	99.3	100.3	
Pulse	80	80	80	93

4. Implementation Details

We import all necessary libraries before creating a primary class. In the main class, we start by building actual physical objects like actuators, sensors, and fog devices. The number of areas, sensors per area, and other variables are declared, and the number of areas and hospitals is the same. If the "private static Boolean cloud" is false, all processing is carried out in the fog nodes. Otherwise, the processing is done on the cloud. The module mapping module assigns each module to the devices.

In iFogSim, developing an application is similar to creating a Directed Acyclic Graph (D.A.G.). The function *"AddAPPMODULE,"* which is part of the application class in iFogSim, is used to create vertices for the D.A.G. We must describe the edges connecting the vertices once we have created all of the vertices (modules). The edges are specified using the *"ADDAPP EDGE"* function. The "*ADDTUPLE MAPPING*" function specifies all input output relationships. Each Task in iFogSim 2.0 can be

$$TS = \{TS1, TS2, TS3 \ldots\ldots\ldots\ldots TSn\} \quad (1)$$

Each TSi has attributes

$$TS_i = \{ST_{Si}, LT_{Si}, TT_{si}, idT_{si}\} \quad (2)$$

STsi,LTsi,TTsi,idTsi , where STsi represents the state of the task, LTsi represents the Length of the task, TTsi represents the Type of the task and idTsi represents the identifier of the task TSi. Each edge node is represented as

$$E = \{EN1, EN2, EN3 \dots \dots \dots ENn\} \quad (3)$$

ENi represents edge node i. Each edge node is characterized by the features

$$ENi = \{SENi, CENi, BENi\} \quad (4)$$

SENi represents the storage capacity of Edge node ENi. represents the computing capacity of Edge node ENi. BENi represents the battery energy of Edge node ENi.
Each fog node is represented as

$$F = \{FN1, FN2, FN3 \dots \dots FNn\} \quad (5)$$

FNi represents Fog node i. Each Fog node is characterised by the features

$$FNi = \{SFNi, CFNi, BFNi\} \quad (6)$$

SFNi represents the storage capacity of Fog node FNi. CFNi represents the computing capacity of Fog node FNi. BFNi represents the battery energy of Fog node EFi.

4.1 Data granularity

The data format for physical device is represented as the following categories.
Physical device as a vital Sensing machine
{PD $_{id}$, PD_S, PD_{VN} ,CV,VR,$UHID_P$, PL,$UHID_C$,CL,TS,S_{PD}}
(a) PD_{id} —Identifier of the physical device
(b) PD_S —Status of the physical device
(c) $PD_{V\,N}$ —Name of the vital that physical device read
(d) CV —Count of the vital
(e) VR —vital reading
(f)$UHID_P$ —Unique hospital identifier of the patient P
(g) PL —Location of the patient
(h)$UHID_C$ —Unique hospital identifier of the Caregiver C
(i) CL —Location of the caregiver.
(j) TS —Time stamp
(k)S_{PD} —Specification of the physical device

Table 3. IoTMD Matrix

UHID	TimeStamp	Item	Value
453657	27/06/2021 7 A.M	Temperature	99.3
453657	27/06/2021 8 A.M	Temperature	100.3
453657	27/06/2021 7 A.M	Pulse	80
453657	27/06/2021 9 A.M	Pulse	93
747589	27/06/2021 7A.M	Temperature	102
6357426	27/06/2021 7 A.M	Temperature	100.4
353657	27/06/2021 7. A.M	Food intake	Idly 20gm
447589	27/06/2021 8 A.M	Urine Output	0.5litre

Physical device as a smart medication box
{PD_{id}, PD_S,PD_{MB},$UHID_P$, PL,UHIDC , TS,S_{PD},D_n,D_s,R_m,MS,Balance}
(a) MS —Status of Medication
(b) Balance —Balance amount of medication in smart medicine box.

The device(machine) identifier is represented as PD_{id}. Unique healthcare identifier of the patient is represented in Pid. Category (Type) of the vital is represented as TV. Number of vital recorded is represented as NV

We use typical medical data processing cycles to deal with unstructured EHR, such as parsing, stemming, chunking, phrase and concepts identification, and so on. A novel approach is used to deal with IoT MD data. A sample IoT MD data is shown in the **Table 2**. As smart healthcare devices generate a large amount of data, some of which is duplicated, an automata-based data representation strategy is employed to decrease data redundancy and consequently space.

Sony P et al [26] suggested a finite automata-based indexing strategy for healthcare IoT data. The requirement for separate automata modelling for each patient is the disadvantage of research method proposed in [26]. Here we have developed three separate automata for three different kinds of medical data.

1.Vital Automata
2. Symptom Automata
3. Medication Automata

The vital automata VA(Q,Σ,δ,q0,F, Edgein,Edgeloop) Q represents set of states in triplet format(TV:NV:VV) TV represents the type or name of the vital, NV is the no of Vital values of TV, VV is the Value of the Vital. Σ is the input in triplet format (Pid:Mid:TS). The patient identity is Pid, and Mid represents the machine identifier (identifier of the sensor device or the medical device) and TS is the Time stamp. δ is the transition function from Q × Σ to Q (TVi:NVVi:VVi) × (pid,Midk:TS) → (TVi:NVVi:VVj). A special state Φ∈ F is the set of final states represented as double sided oval. Edgein is the incoming edge of a state (TVi:NVVi:VVi). Self loop of a state (TVi:NVVi:VVi) is represented as Edgeloop. Advantage of automata modelling is the easy retrieval of data and reduction in redundancy.

Symptom automata MA(Q,Σ,δ,q0, Edgein,Edgeloop). Q represents set of states with five elements in each state of the form Pid,Sid,Ts,Severity. Pid represents patient identifier. Sid represents the unique identifier(CUI) of symptom. Ts is timestamp. Severity is the level of symptom. Σ is the input in triplet format (Pid:TS). δ is the transition function from Q × Σ to Q. Pid,Sid,Ts,Severity × (Pid:TS) to Pid,Sid,Ts,Severity.F is the set of final states represented as double sided oval. Edgein is the incoming edge of a state (Pid,Sid,Ts,Severity). Self loop of the state is represented as Edgeloop.

Medication automata MA(Q,Σ,δ,q0,F, Edgein,Edgeloop). Q represents set of states with five elements in each state of the form Medication Name, dose, Time, and Route of administration, and Status. Σ is the input in triplet format (Pid:Did:TS). Pid represents the patient identifier, Did represents the doctor id, and TS represents the timestamp. δ is the transition function from Q × Σ to Medication Name, dose, Time, Route of administration, Status × (Pid:Did:TS) → Medication Name, dose, Time, Route of administration, Status.F is the set of final states represented as double-sided oval. Edgein is the incoming edge of a state (TVi:NVVi:VVi). Self loop of the state is represented as Edgeloop. When the machine reaches the final state, the patient has finished a particular course of medication. A two-dimensional matrix(a sample 2-D matrix is shown in **Table 3** is generated by using the previously mentioned automata.

4.2 UMLS Integration and Linguistic Processing

The National Library of Medicine (NLM)'s Unified Medical Language System (UMLS) has three knowledge sources: Metathesarus, Semantic network, and Specialist lexicon and lexical tools. CPT, ICD-10-CM, LOINC, MeSH, RxNorm, and SNOMED CT medical terminology and codes, as well as their hierarchies, meanings, and other relationships and attributes, are all included in Metathesarus. A Semantic Network is a representation of semantic categories and their relationships. There is a large syntactic lexicon of biomedical and general English, as well as tools for normalizing texts, generating lexical variants, and creating indexes.
Using the iFogSim tools, we can connect to any external database we wish. For example, depending on the events created in the *Controller class*, we might create an object in the Controller class and then save the data in the database on a regular basis. We could also read the database in place. Another scenario is reading data from the database in the program's main function while building the environment. We can also store all of the data obtained after the simulation to the database using the main function. Here we created an external database UMLS in the main function of the application. The following steps are used in linguistic processing.

1. Medical Concept Identification
2. Synonymous words Identification
3. Polysemous words Identification
4.Holonymy-Meronymy Identification
5. Hypernymy -Hyponymy Identification

The medical concepts are identified by the concept processing algorithm represented in [27]. Algorithm 1,2,3 and 4 represent the synonymy, polysemy, holonymy and hypernymy identification of medical documents. After resolving the linguistic issues, HSDF (Health Sign Description Framework) creation microservice is called. HSDF creation algorithms specified in the article [28] are used for the creation of healthcare signs.

Algorithm 1. Synonymy Resolution

Input : Text String X
Output: Synonymous Strings

```
While X € Medical Concept Γ
    Do find CUI(X) from MRCONSO Table of UMLS
    For each i, γi € Γ  do
        If CUI(X) ==CUI(γi )
            Add synonym set of (X) to STR(γi )
            Else
            ADD CUI(X) to Synonym Set
        Endif
    End For
End While
```

Algorithm 2. Polysemy Resolution

Input : Text String X
Output: Polysemous Table ψ

```
1  While X € Medical Concept Γ
2      Do find CUI(X) from MRCONSO Table of UMLS
3      For each CUI(X) do
4          Find STR(CUI(X) from MRCONSO Table
           Add STR(CUI(X) to Polysemous Table ψ
5      End For
6  End While
```

Algorithm 3. Holonomy Meronymy Resolution

Input : Text String X1,X2
Output: Holonomy and Meronymy Table

```
1  While X1, X2 € Medical Concept Γ
2      Do find CUI(X1) from MRCONSO Table of UMLS
3      Do find CUI(X2) from MRCONSO Table of UMLS
4      If RELA(X1,X2)= "part-of" then
5          Holonymy (X1)=X2
6          Meronymy (X2)=X1
7      End If
8  End While
```

Algorithm 4. Hypernymy Hyponymy Resolution

Input : Text String X1,X2
Output: Holonomy and Meronymy Table

```
1  While X1, X2 € Medical Concept Γ
2      Do find CUI(X1) from MRCONSO Table of UMLS
3      Do find CUI(X2) from MRCONSO Table of UMLS
4      If RELA(X1,X2)= "is-a" then
5          Hypernymy (X1)=X2
6          Hyponymy(X2)=X1
7      End If
8  End While
```

4.3 Simulation Environment

1. Duration of the experiment: 35000seconds
2. No of location change events: 98
3.Network architecture: heterogeneous
4. Deployment: Grid and Uniform
5. No of patients in one cluster: 10
6. No of biosensors in one cluster: 48

Table 4. Configuration Parameters

Configuration Parameters	Cloud VM	Proxy Server	Fog Gateway	Smart Phone
Count	8	4	16	52
Speed (MIPS)	2000-2500	2000-2500	2000-2500	400
RAM(GB)	16	16	8	2
Uplink Bandwidth (Mbps)	80	20	80	150
Downlink Bandwidth (Mbps)	80	40	130	200
Busy Power (MW)	106.223	106.223	106.223	67.56
Idle Power (MW)	82.34	82.34	82.34	61.78

An Intel Core 2 Duo CPU operating at 2.33 GHz, 16 G.B. of RAM, and iFogSim 2.0 are used to execute the simulations. **Tables 4 and 5** provide the configuration and simulation parameters, respectively. Healthcare sensors can operate in three modes: sustained, sleep-awake, and periodic. When a patient is in a critical care unit, in an emergency, or has surgeries or procedures, the patient's operating mode is set to Sustained. Sleep Awake mode is set when the person is in a normal situation. Sensors are set to Periodic mode when health data is required on demand.

For the model simulation, two separate datasets are used. The first is structured EHR data, while the second is data from IoT medical devices. IoT MD data set is generated from the patients residing in 25 different states of India. For each user, we create smartphones as a gateway. Through tier-1 nodes in the Fog layer, the gateway may connect with tier-0 nodes in the cloud layer. Cloud-centric, nonhierarchical, and intra/inter-cluster migrations are the types of migration methodologies employed.

The Clustering class implemented in iFogSim can be used to cluster Fog nodes in order to improve storage and computing capacity. The IoT MD data set comprises the locations of each fog node; locations are parsed using the data parser class. Each node has access to information about its parent, children, communication range, and bandwidth.

Table 5. Simulation Parameters

Microservice Module	RAM(GB)	Input (MB)	Output (MB)	CPU Length (MIPS)
EHR Data Acquisition	2		3.5	10000
IoTMD Data Acquisition	2		1	8000
Pre-Processing	2	3.5	2.5	12000
TPC Creation	2	3.5	5	14000
H3M Resolving	2	5	2.5	14000
Polysemy Resolving	2	3	2.5	12000
Intermediate Representation Module	4	2.5	3.5	10000

5. Results and Discussion

The results obtained from the comprehensive simulation and performance analysis shed light on the efficacy of the proposed semantic interoperability solution within a fog computing environment for healthcare IoT. In this section, we present a detailed discussion of the findings,

examining key metrics such as simulation time, mobility scheme performance, microservice migration, placement algorithms, computation time, task delay, network usage, and effect of linguistic processing. These results provide valuable insights into the system's behavior under various scenarios, allowing for a nuanced understanding of its strengths, limitations, and potential avenues for improvement. Through a thorough exploration of each aspect, we aim to contribute to the ongoing discourse on optimizing healthcare data management and processing in the era of IoT, fog computing, and semantic interoperability.

5.1 Simulation Time Analysis

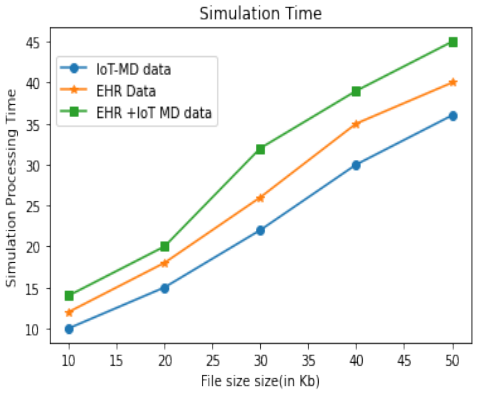

Fig. 3. Simulation Time

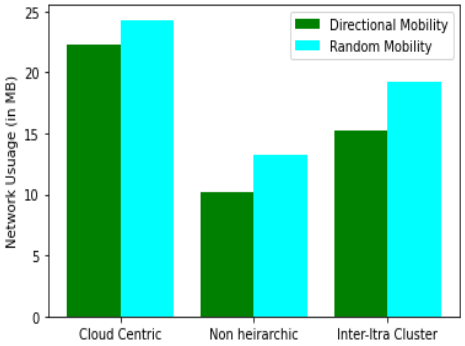

Fig. 4. Network Usage during Migration

In the simulation, the time taken from submitting IoT Medical Device (IoTMD) data to the creation of an intermediate form is compared for different types of data. Initially, we sent IoTMD data to the simulator, then unstructured EHR data, and ultimately both to the simulator for analysis. Simulation time is observed to be shorter for documents containing only IoTMD data, longer for documents containing only Electronic Health Record (EHR) data, and intermediate for documents containing both. This variation is visually represented in **Fig. 3**.

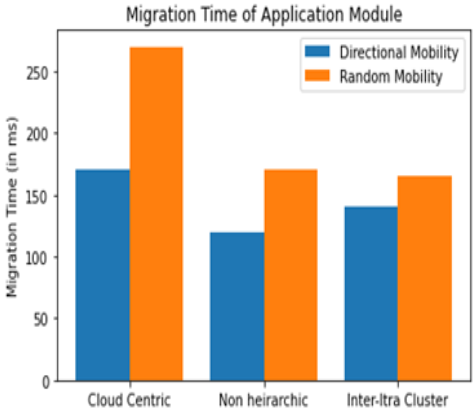

Fig. 5. Migration Time of Application Module

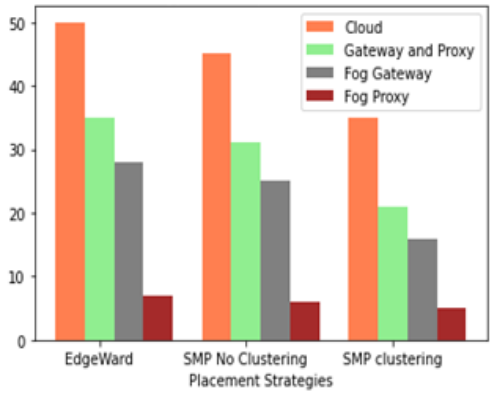

Fig. 6. Network Usage during Migration

To evaluate the performance of mobility schemes in diverse clustering environments, migration time and network usage are analyzed. Across cloud-centric, non-hierarchic, and inter-cluster methods, the random mobility scheme exhibits higher migration time and network

usage compared to the directional mobility scheme. **Fig. 5** illustrates the migration time of microservice modules using both random and directed mobility strategies.

Fig. 6 demonstrates network bandwidth usage during the migration of microservice modules. In the cloud centric approach, migrating from a fog gateway to the cloud and back increases latency due to the involvement of numerous fog nodes. The non-hierarchical approach, leveraging mesh connections, reduce network usage, particularly when compared to random mobility in all three systems. The loop responsible for translating unstructured EHR into intermediate representation and the loop responsible for transforming IoT MD data into intermediate representation are key control loops in the proposed paradigm. Both loops need communication between many components. These methods of communication, as well as the processing latency, contribute to the delay. The increased usage of networks for cloud environments might cause network congestion and application performance deterioration. The use of fog nodes can help to minimize network traffic.

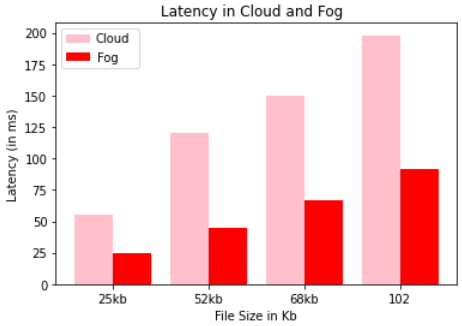

Fig. 7. Latency

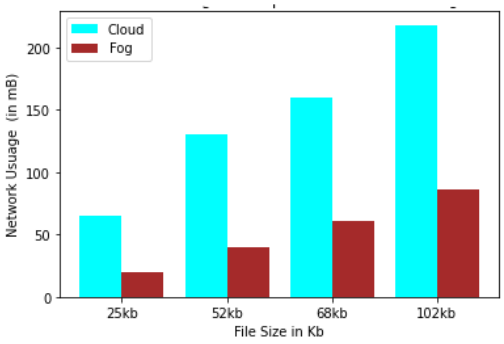

Fig. 8. Network Usage

Computation time of a task in Edge node ENi is represented as

$$CT(ENi) = \frac{Task\ Length}{Computing\ capacity} \tag{7}$$

$$CT(ENi) = \frac{LTSi}{CENi} \tag{8}$$

Computation time of a task in Fog node FNi is represented as

$$CT(FNi) = \frac{LTSi}{CFNi} \tag{9}$$

When the initial piece of healthcare data is sent out from the source, the latency or delay measures how long it takes for the full batch of healthcare data to arrive at the destination[29]

$$Delay\ L = \psi_T + \varphi_T + \Phi_T + \lambda_T \tag{10}$$

ψ_T represents propagation time, φ_T represents transmission time, Φ_T represents queuing delay and λ_T represents processing time. Propagation time of data is the amount of time it takes to get from its source to its destination.

$$\psi_T = \frac{D}{\psi_s} \tag{11}$$

Distance is represented as D, ζS represents Propagation Speed. Transmission time is the amount of time needed to transmit a message depends on its size and the channel's bandwidth.

$$\varphi_{T=\frac{Data\ Size}{B}} \tag{12}$$

The amount of time require d for each intermediate or end device to retain the message before processing it is known as the queuing time. The queue time is a variable component fluctuates according to the network demand.
Latency cloud layer and fog layer are shown in **Fig. 7**. Since most of the processing is done in the fog layer in the proposed system, latency and network usage in the fog layer is much higher than that of cloud layer. Network Usage of cloud and fog layer is shown in **Fig. 8**.

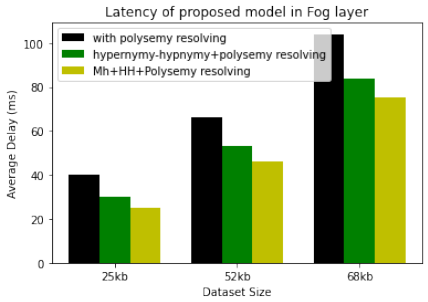

Fig. 9. Latency of Linguistic Processing

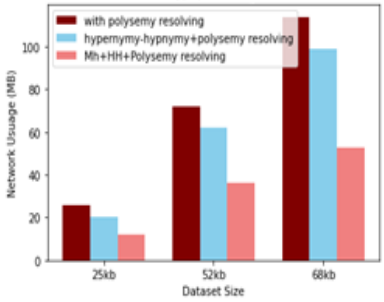

Fig. 10. Network Usage of Linguistic Processing

We simulated the system in three environments to assess its performance in linguistic processing. 1. Polysemy resolution system 2. System for hyponymy -hypernymy resolution with polysemy 3. System for resolving polysemy, hypernymy, hyponymy, meronymy, and holonymy. In comparison to the other two systems, the third system has lower latency and network usage. **Fig. 9** and **10** depict the latency and network usage of linguistic processing, respectively. **Fig. 11** depicts the average delay of different placement strategies.

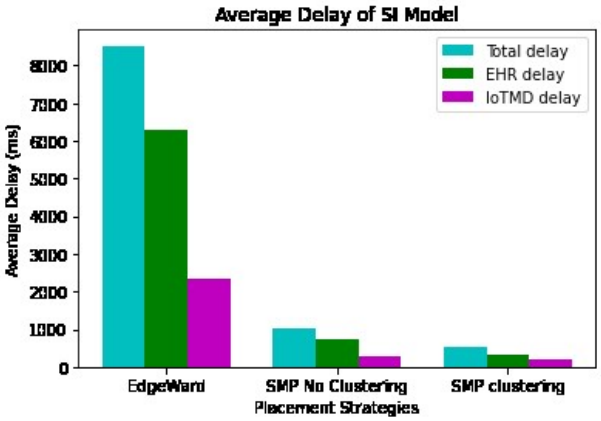

Fig. 11. Average delay of different placement strategies

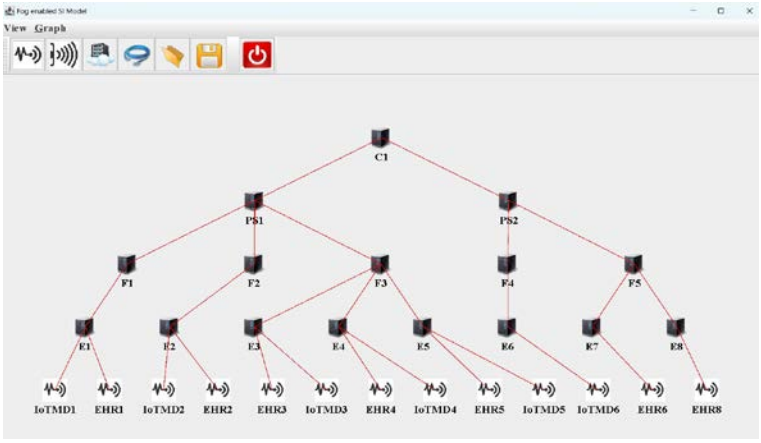

Fig. 12. A Sample Topology of the Proposed System

The semantic interoperability solution proposed by Gupta et.al is not based on fog computing. There is no cloud fog interoperability with the approach suggested by Rahmani et al [30], Ahmad et al [1] and Gupta et al[31]. A sample topology and comparison with existing work is shown in **Fig. 12 and Table 6** respectively. Other than our technique, none of the solutions are appropriate for use with unstructured EHRs and do not offer document level interoperability.

If the simulation framework does not comprehensively model various types of edge devices commonly found in healthcare, such as wearable health monitors, smart medical devices, or point-of-care diagnostics tools, the simulation results may not accurately reflect the interactions and dynamics at the edge of the network. Ifogsim 2.0 might not fully capture the diverse characteristics of fog nodes that are crucial in healthcare environments. For instance, fog nodes in healthcare may include local servers, gateways, or edge computing devices with specific processing capabilities, storage capacities, and communication protocols. Inadequate representation of these characteristics can lead to a lack of fidelity in the simulation. Edge and fog computing elements in healthcare may communicate using various protocols depending on the devices and technologies involved. If the simulation framework oversimplifies or overlooks this heterogeneity, it may not capture the intricacies of communication challenges and opportunities in a healthcare setting.

Table 6. Comparison with similar works

Article	Applicable to Healthcare Domain	Applicable to Unstructured Data	Document Level Interoperability satisfied	Cloud Fog Interoperability	Fog based
Rahmani et.al[30]	YES	NO	NO	NO	YES
Ahmad et.al [1]	YES	NO	NO	NO	YES
Gupta et.al[31]	YES	NO	NO	NO	YES
Mahmud et.al[32]	YES	NO	NO	YES	YES
Our Method	YES	YES	YES	YES	YES

6. Conclusion

Fog computing (FC) is emerging as a novel paradigm for Small and Medium-Sized Enterprises [29]. Research on fog computing is currently in an exploratory phase, with several pressing issues requiring in-depth investigation. These challenges span across platform, infrastructure, and application domains, with critical issues including the exploration of task partitioning, scheduling methodologies, and resource allocation strategies [30]. Interfog communication and latency are one of the key disadvantages of Fog Computing [31].

Depending on the device, healthcare sensors produce data in a variety of forms and units. Each healthcare organization use disparate format, language, and representation of healthcare data [33]. One of the most important research topics in the healthcare industry is semantic interoperability. Most hospitals keep medical data on the cloud because it is so massive. But in a cloud setting, there is general performance delay. In the Fog environment, this article suggests a semantic interoperability solution for IoT in healthcare. Both structured and unstructured medical documents are supplied into the suggested system.

The suggested technique makes use of an ontology Unified Medical Language System (UMLS) created by the National Library of Medicine (NLM). With the use of the UMLS ontology, four algorithms are suggested to resolve lexical, morphological, and other linguistic problems in the medical documents. The data granularity microservice for the edge layer introduces an automata-based storage format to drastically reduce on redundancies in IoT-MD data

The suggested model-based fog environment decreases latency and network congestion as a result of its observations. The suggested technique uses linguistic processing to minimize network latency and congestion. In three separate scenarios, we evaluated the system's linguistic processing capabilities. 1. Polysemy resolution system 2. A method for dealing with polysemy hypernymy and hyponymy simultaneously. 3. A technique for resolving polysemy, hypernymy, hyponymy, meronymy, and holonymy. In comparison to the other two approaches, the third one has less latency and network delay. As a future scope we would like to implement the microservices presented in the fog layer using any deep learning techniques for getting better accuracy.

In addition to implementing microservices with deep learning techniques for enhanced accuracy, further emphasis should be placed on fortifying the system's security aspects. This includes the integration of advanced encryption methods, access controls, blockchain technology, and privacy-preserving techniques to ensure data integrity, protect patient information, and comply with relevant regulations. Continuous security audits, user education initiatives, and collaboration with cybersecurity experts will be pivotal in maintaining a robust and secure healthcare data environment.

References

[1] N. Ahmad, S. Du, F. Ahmed, N. ul Amin, and X. Yi, “Healthcare professionals satisfaction and AI-based clinical decision support system in public sector hospitals during health crises: a cross-sectional study,” *Information Technology and Management*, Aug. 2023. Article (CrossRef Link)

[2] W.-H. H. H. C. S. S. R. Anand Paul, Hameed Pinjari, “Fog Computing-Based IoT for Health Monitoring System,” *Journal of Systems and Software*, vol. 2018, pp. 1687–725X, 2018. Article (CrossRef Link).

[3] M. Abdel-Basset, G. Manogaran, A. Gamal, and V. Chang, “A Novel Intelligent Medical Decision Support Model Based on Soft Computing and IoT,” *IEEE Internet Things J*, vol. 7, no. 5, pp. 4160–4170, 2020. Article (CrossRef Link).

[4] S. K. Sood and I. Mahajan, "IoT-Fog-Based Healthcare Framework to Identify and Control Hypertension Attack," *IEEE Internet Things J*, vol. 6, no. 2, pp. 1920–1927, 2019. Article (CrossRef Link).
[5] S. Tanwar et al., "Human Arthritis Analysis in Fog Computing Environment Using Bayesian Network Classifier and Thread Protocol," *IEEE Consumer Electronics Magazine*, vol. 9, no. 1, pp. 88–94, 2020. Article (CrossRef Link).
[6] H. ben Hassen, N. Ayari, and B. Hamdi, "A home hospitalization system based on the Internet of things, Fog computing and cloud computing," *Inform Med Unlocked*, vol. 20, p. 100368, 2020. Article (CrossRef Link).
[7] S. K. G. R. B. Harshit Gupta, Amir Vahid Dastjerdi, "iiFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments," *Distributed, Parallel, and Cluster Computing*, 2016. Article (CrossRef Link).
[8] F. Firouzi, B. Farahani, and A. Marinšek, "The convergence and interplay of edge, fog, and cloud in the AI-driven Internet of Things (IoT)," *Inf Syst*, vol. 107, p. 101840, Jul. 2022. Article (CrossRef Link).
[9] R. Mahmud, F. L. Koch, and R. Buyya, "Cloud-fog interoperability in IoT-enabled healthcare solutions," in *Proc. of the 19th international conference on distributed computing and networking*, pp. 1–10, 2018. Article (CrossRef Link).
[10] N. Mohan and J. Kangasharju, "Edge-Fog cloud: A distributed cloud for Internet of Things computations," in *Proc. of 2016 Cloudification of the Internet of Things (CIoT)*, pp. 1–6, 2016. Article (CrossRef Link).
[11] R. Mahmud, S. Pallewatta, M. Goudarzi, and R. Buyya, "iFogSim2: An extended iFogSim simulator for mobility, clustering, and microservice management in edge and fog computing environments," *Journal of Systems and Software*, vol. 190, p. 111351, 2022. Article (CrossRef Link).
[12] N. Mohan and J. Kangasharju, "Edge-Fog cloud: A distributed cloud for Internet of Things computations," in *Proc. of 2016 Cloudification of the Internet of Things (CIoT)*, pp. 1–6, 2016. Article (CrossRef Link).
[13] P. Sony, G. S. Shanmugam, and S. Nagarajan, "An intuitionistic fuzzy-based intelligent system for semantic interoperability and privacy preservation in healthcare systems," *Soft comput*, Jul. 2023. Article (CrossRef Link).
[14] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM computer communication Review*, vol. 44, no. 5, pp. 27–32, 2014. Article (CrossRef Link).
[15] N. Ahmad, Z. Liu, J. Wu, F. Alam, M. Waqas, and X. Yi, "Examining the Employees Behavior Control in Cloud Computing Performance Through the Moderating Lenses of Transformational Leadership," *E-Business. Digital Empowerment for an Intelligent Future*, pp. 285–297, 2023. Article (CrossRef Link)
[16] S. Patel and R. Patel, "A Comprehensive Analysis of Computing Paradigms Leading to Fog Computing: Simulation Tools, Applications, and Use Cases," *Journal of Computer Information Systems*, vol. 63, no. 6, pp. 1495–1516, Nov. 2023. Article (CrossRef Link)
[17] S. Malik and K. Gupta, "Smart City: A new phase of sustainable development using fog computing and IoT," *IOP Conf Ser Mater Sci Eng*, vol. 1022, no. 1, p. 12093, Jan. 2021. Article (CrossRef Link)
[18] M. Mahmud and R. Buyya, "Modeling and Simulation of Fog and Edge Computing Environments Using iFogSim Toolkit: Principles and Paradigms," in *Fog and Edge Computing: Principles and Paradigms*, 2019, pp. 433–465. Article (CrossRef Link) .
[19] A. I. T. A. A. Ahmed Elhadad Fulaylan Alanazi, "Fog Computing Service in the Healthcare Monitoring System for Managing the Real-Time Notification," *Journal of Healthcare Engineering*, vol. 2022. pp. 2040–2295, 2022. Article (CrossRef Link)

[20] D. C. Klonoff, "Fog Computing and Edge Computing Architectures for Processing Data From Diabetes Devices Connected to the Medical Internet of Things," *J Diabetes Sci Technol*, vol. 11, no. 4, pp. 647–652, 2017. Article (CrossRef Link) .
[21] S. Sarkar and S. Misra, "Theoretical modelling of fog computing: a green computing paradigm to support IoT applications," *Iet Networks*, vol. 5, no. 2, pp. 23–29, 2016. Article (CrossRef Link)
[22] A. Markus and A. Kertesz, "A survey and taxonomy of simulation environments modelling fog computing," *Simul Model Pract Theory*, vol. 101, p. 102042, 2020. Article (CrossRef Link).
[23] M. Ashouri, F. Lorig, P. Davidsson, and R. Spalazzese, "Edge computing simulators for iot system design: An analysis of qualities and metrics," *Future Internet*, vol. 11, no. 11, p. 235, 2019. Article (CrossRef Link)
[24] R. K. Naha et al., "Fog Computing: Survey of Trends, Architectures, Requirements, and Research Directions," *IEEE Access*, vol. 6, pp. 47980–48009, 2018. Article (CrossRef Link).
[25] O. Bodenreider, "The unified medical language system (UMLS): integrating biomedical terminology," *Nucleic Acids Res*, vol. 32, no. suppl_1, pp. D267–D270, 2004. Article (CrossRef Link).
[26] P. Sony and N. Sureshkumar, "A new indexing technique for healthcare IoT," *Smart Intelligent Computing and Applications*, vol. 105, pp. 345-353, 2019. Article (CrossRef Link).
[27] P. Sony and N. Sureshkumar, "Concept-based electron ic health record retrieval system in healthcare IoT," in *Cognitive Informatics and Soft Computing*, Springer, 2019, pp. 175–188, Article (CrossRef Link).
[28] P. Sony and S. Nagarajan, "Semantic Interoperability Model in Healthcare Internet of Things Using Healthcare Sign Description Framework," *International Arab Journal of Information Technology*, vol. 19, no. 4, pp. 589-596, 2022. Article (CrossRef Link).
[29] B. A. Forouzan, *Data Communications and Networking*, 2000. Article (CrossRef Link).
[30] A. M. Rahmani et al., "Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641–658, Jan. 2018. Article (CrossRef Link).
[31] D. Gupta and A. Khamparia, *Fog, Edge, and Pervasive Computing in Intelligent IoT Driven Applications*, John Wiley & Sons, 2020. Article (CrossRef Link)
[32] R. Mahmud, F. L. Koch, and R. Buyya, "Cloud-Fog Interoperability in IoT-enabled Healthcare Solutions," in *Proc. of the 19th International Conference on Distributed Computing and Networking*, pp. 1–10, Jan. 2018. Article (CrossRef Link).
[33] S. Purushothaman, G. S. Shanmugam, and S. Nagarajan, "Achieving Seamless Semantic Interoperability and Enhancing Text Embedding in Healthcare IoT: A Deep Learning Approach with Survey," *SN Comput Sci*, vol. 5, no. 1, p. 99, 2024. Article (CrossRef Link).

Dr. Sony P is currently working as an assistant professor in Govt. Model Engineering College, Kochi, Kerala, India. She had completed her Ph.D. degree from Vellore Institute of Technology, Vellore, Tamilnadu. Her research interest includes Natural Language Processing, Machine Learning and Internet of Things.

Dr. Siva Shanmugam G is a Senior Assistant Professor in the School of Computer Science at the VIT University. His doctoral research focused on cloud resource optimization, and he has 15 years of teaching experience. His research contributes multiple techniques for load balancing, an alternate model for VM modelling, and Optimized Internet infrastructure setup. His teaching interest include Distributed Computing, Cognitive Computing, and Artificial Intelligence.

Dr. Sureshkumar Nagarajan is currently working as Professor and Head of the Department in the School of Computing at Kalasalingam Academy of Research and Education KrishnanKovil, Tamandu, India. His research interest includes Image Processing, Computational Intelligence and Big Data. He has got more than 60 publications in various reputed journals and conferences.