electronics

MDPI

Review

Intelligent Load-Balancing Framework for Fog-Enabled Communication in Healthcare

Swati Malik [1], Kamali Gupta [1], Deepali Gupta [1], Aman Singh [2], Muhammad Ibrahim [3,4], Arturo Ortega-Mansilla [2,5], Nitin Goyal [1,*] and Habib Hamam [6,7,8]

1 Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab 140401, India; swati.malik@chitkara.edu.in (S.M.); kamali.singla@chitkara.edu.in (K.G.); deepali.gupta@chitkara.edu.in (D.G.)
2 Higher Polytechnic School, Universidad Europea del Atlántico, C/Isabel Torres 21, 39011 Santander, Spain; aman.singh@uneatlantico.es (A.S.); arturo.ortega@uneatlantico.es (A.O.-M.)
3 Department of Information Technology, University of Haripur, Haripur 22620, Pakistan; ibrahimmayar@uoh.edu.pk
4 Department of Computer Engineering and Research Center of Advance Technology, Jeju National University, Jeju-si 63243, Korea
5 Department of Project Management, Universidad Internacional Iberoamericana, Campeche 24560, Mexico
6 Faculty of Engineering, Université de Moncton, Moncton, NB E1A3E9, Canada; habib.hamam@umoncton.ca
7 International Institute of Technology and Management, Commune d'Akanda, Libreville BP1989, Gabon
8 School of Electrical Engineering, Dept of Electrical and Electronic Engineering Science, University of Johannesburg, Johannesburg 2006, South Africa
* Correspondence: nitin.goyal@chitkara.edu.in

Abstract: The present technological era significantly makes use of Internet-of-Things (IoT) devices for offering and implementing healthcare services. Post COVID-19, the future of the healthcare system is highly reliant upon the inculcation of Artificial-Intelligence (AI) mechanisms in its day-to-day procedures, and this is realized in its implementation using sensor-enabled smart and intelligent IoT devices for providing extensive care to patients relative to the symmetric concept. The offerings of such AI-enabled services include handling the huge amount of data processed and sensed by smart medical sensors without compromising the performance parameters, such as the response time, latency, availability, cost and processing time. This has resulted in a need to balance the load of the smart operational devices to avoid any failure of responsiveness. Thus, in this paper, a fog-based framework is proposed that can balance the load among fog nodes for handling the challenging communication and processing requirements of intelligent real-time applications.

Keywords: fog computing; load balancing; healthcare; cloud computing

Citation: Malik, S.; Gupta, K.; Gupta, D.; Singh, A.; Ibrahim, M.; Ortega-Mansilla, A.; Goyal, N.; Hamam, H. Intelligent Load-Balancing Framework for Fog-Enabled Communication in Healthcare. *Electronics* **2022**, *11*, 566. https://doi.org/10.3390/electronics11040566

Academic Editor: Juan M. Corchado

Received: 12 January 2022
Accepted: 9 February 2022
Published: 13 February 2022

1. Introduction

Visualization of the future connected world is incomplete without including the Internet of Things (IoT). IoT devices are artificially intelligent and are available to be utilized by various applications that can provide efficient performance for end-users. The data of various applications are collected by these devices [1]. Various smart devices produce a large amount of data, which is utilized by various applications for better decision making and better services for end-users [2].

The interconnections between various devices and application helps in learning the existing system and improving it for efficient functionality. IoT is utilized in various fields for day-to-day activities, such as smart offices, buildings, healthcare, grid system, traffic management, healthcare, agriculture and many more fields. Medical services are an important concern in everyone's life. The contribution to health services can reduce the cost and provide better services to people.

Various routine activities can be monitored with IoT devices, such as incorrect posture, unhealthy eating habits, and a prolonged sedentary schedule. These activities can cause multiple diseases when repeated frequently. If these activities are monitored well by IoT devices, many drastic diseases can be prevented. Some other routine activities that can be monitored include nutritional habits, sleep duration, workout duration etc. Patient health routine information, health emergencies, temperature and blood information can also be handled by IoT devices effectively. Additionally, the use of IoT has increased the quality of various healthcare environments, such as continuous real-time tracking, management of patient information, health emergency management, the management of blood information and health management [3,4].

Various medical sensors and healthcare devices produce data at high speed as patients are monitored in real-time; the information thus produced is stored, processed and analyzed. Generally, devices equipped with sensors have less power, limited battery, less storage and limited networking capabilities [5]. Thus, the data collected depend on another framework that can perform computation, storage and analysis. The main concern in implementing smart healthcare is the storage and security of huge data produced by smart healthcare devices. One possible solution for this is the cloud.

Cloud computing has massive storage and processing capability. The major requirement of healthcare resources is sharing of data and information [6,7]. Cloud computing can share and maximize resources by utilizing virtualization. Location-independent services are provided by cloud computing [8,9]. Cloud services can be utilized by end-users from anywhere and through any device. This integration of IoT and cloud computing can reduce the overall cost [10–12].

The major advantage of cloud computing in the healthcare scenario is that services are moved towards the home of the patient, which plays a major role in reducing the overall cost of healthcare. Some other advantages include a healthy patient environment and healthcare resources that can be immediately provided to patients in an emergency with a minimum delay [13,14]. Early identification and diagnosis of health issues can be done effectively by utilizing the services of the cloud. Sometimes cloud computing does not perform well with latency-sensitive applications. As the number of requests and computation requests increases it becomes difficult for cloud computing to process all the requests with minimum delay.

To handle the difficulties faced by cloud computing, fog computing has been introduced [15]. Some other computing paradigms have also been proposed by researchers, such as edge computing and mist computing. In mist computing, micro-controllers and sensors are equipped on end devices, which are utilized for performing computations; whereas, in edge computing, computation is done at the end device, and in fog computing, the computations are performed at fog nodes that lies between cloud and end devices.

The major goal of this work is to improve health monitoring systems based on IoT devices such that data collected from Wireless Sensor Networks (WSNs) are processed quickly and context-sensitive data that is relevant to the patient is taken into account [16]. We do this by implementing a fog layer that reduces the latency in health monitoring systems and allows for real-time monitoring. In doing so, we hope to secure patient information security, ensuring that patient privacy is protected and that data tampering by third parties is prevented.

The main contributions of this paper are as follows:

- A study of load-balancing algorithms of fog computing along with the upcoming challenges are presented. The role of AI techniques in healthcare for upcoming researchers is depicted.
- A framework is proposed for healthcare so that critical patients requiring immediate help can obtain it immediately from the hospital.
- Fog computing is utilized in the framework to minimize service latency.

The paper organization is as follows: Section 2 presents a literature survey on the load-balancing approaches used in fog-based architecture. Section 3 depicts the role of AI in healthcare. Section 4 presents the research methodology. Section 5 includes a proposed framework, and Section 6 concludes our findings and presents the future scope. When fog computing architecture is utilized for healthcare services, the most crucial parameter that can affect the overall performance is the latency, which can be better handled if no node is over-utilized or under-utilized. Therefore, the next section briefly discusses the taxonomy of load balancing.

2. Taxonomy of Load Balancing in Fog Computing

This section covers the literature survey related to fog load balancing. Although there are multiple ways of classifying the load-balancing algorithms, broadly these can be classified as approximate, exact, fundamental and hybrid methods (Figure 1). However, some other classifications are also possible, such as centralized, distributed, semi-distributed and by who initiated the process or system state. The literature survey highlights the differences between these techniques, the evaluation tools used, parametric evaluation, the evaluation methods and the pros and cons.

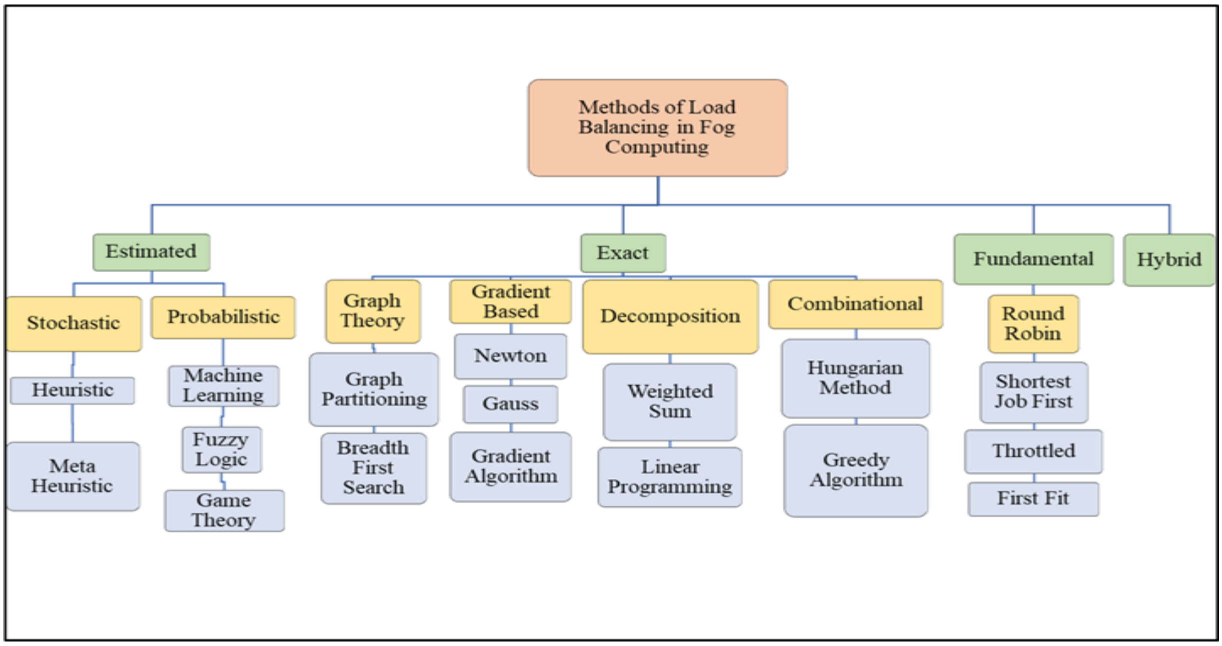

Figure 1. Methods of load balancing in fog computing.

2.1. Estimated Methods

In this method, techniques, such as stochastic, probabilistic and statistics are studied. Stochastic methods are also known as random methods. These are the mathematical model for such processes, which vary randomly.

Stochastic Methods and Probabilistic Methods

Probabilistic methods tend to combine logic and reasoning to handle uncertainty with deductive logic. Statistical methods are concerned with the collection, organization, interpretation and analysis of parameters and factors associated with a phenomenon. This deals with various aspects of data, such as planning for collecting data, creating a survey design and experiments.

- Heuristic Methods

Heuristic methods are based upon experience for applications that seek to attain optimization for finding the best possible solution to a problem. This "trial and error" method is used for finding the best possible solution in the most favorable amount of time. In this approach, the solution can be better than the optimal solution. Sometimes they can outperform the guess. Hill climbing [17], Min-conflicts [18] and Analytic Hierarchy Process [19] are examples of heuristic methods.

Researchers have used different architectures for implementing load balancing in fog computing: some have used single-layer architectures, whereas some have used three-layer or multi-layer architectures. Zahid and co-authors [17] presented a three-layer scenario that consists of a consumer layer, distributed fog layer and centralized cloud layer. This algorithm reduces the response time and processing time of fog nodes to end-users.

Kamal et al. [18] used Min conflict scheduling for solving constant satisfaction problems. Three-layer architecture was used by the author. This method uses a heuristic technique for proposing the load-balancing mechanism. To model the performance of a fog system, Banaie et al. [19] developed various vacation-based queuing systems. They used multi-gateways architecture and a resource caching policy in the IoT domain to speed up user access to sensor data.

To provide global load fairness across network entities, a load-balancing strategy based on the Analytic Hierarchy Process (AHP) method was also used. Furthermore, by referring to load balancing in a fog environment, Oueis et al. [20] focused on increasing the user experience. They took into account many users who required compute offloading, and the requests had to be completed entirely by local computing cluster resources.

A Virtual-Machine (VM) scheduling technique as also suggested by Xu et al. [21] to balance the load in the cloud–fog system.

- Meta-Heuristic Methods

As a higher-level heuristic method, a meta-heuristic technique is problem-independent and can be used to solve a wide range of issues. All recent higher-level approaches are referred to as "meta-heuristics" today [17]. Diversification and intensification are the two major components of current meta-heuristics [22]. To develop an influential and effective meta-heuristic method, it is necessary to strike a balance between diversification and intensification. A metaheuristic method investigates the entire solution space; a new set of solutions should be generated, and the search should be intensified around the optimal or near-optimal solutions. Some meta-heuristic methods have been studied in the literature, including Particle Swarm Optimization [23–25], the Fireworks Algorithm [26] and the Bat Algorithm [27].

To address the problems, He et al. [23] combined fog and Software-Defined Networking (SDN). They presented an SDN-based modified constrained optimization particle swarm optimization method for the proper utilization of SDN and cloud/fog architecture on the Internet of Vehicles. Wan et al. [24] proposed an energy-aware load balancing and scheduling solution based on the fog network.

To balance the manufacturing cluster load, an energy consumption model was proposed on the fog node that was related to the workload, and then an optimization function was established. The modified Particle Swarm Optimization (PSO) method was then used to obtain a good solution, and the manufacturing cluster was given precedence to complete jobs. Shi et al. [25] solved the delay problem by incorporating fog and SDN into a cloud-based mobile face recognition architecture. They also defined load balancing in the SDN and fog/cloud systems as an optimization issue and proposed solving the problem with a Fireworks Algorithm (FWA) based on SDN centralized control.

Yang [26] proposed a three-layered architecture based on fog/cloud networks and big medical data, including cloud, fog, and medical devices as components. The bat algorithm in the suggested architecture used a load balancing method to complete the first setup of bat population data, which increased the quality of the initial sample solution.

Machine learning, fuzzy logic [27] and game theory [28] are examples of probabilistic/statistic load-balancing mechanisms mentioned in this section. Singh et al. [27] also presented a load balancer based on fuzzy logic in fog networks, with several levels of tuning and design of fuzzy controls. The proposed fuzzy logic model was utilized to analyze links as interconnects for traffic management. Abedin et al. [28] proposed a fog-load-balancing problem to reduce the cost of fog-load-balancing in a Narrow-Band Internet of Things environment (NB-IoT). To begin, the NB-temporal IoT's resource scheduling challenge was modelled as a bankruptcy game. The transportation problem was then solved using Vogel's approximation methodology, which finds a feasible load balancing solution.

2.2. Exact Methods

Exact approaches can tackle optimization problems in the most efficient way possible. Each optimization problem can be handled by using an exact search; however, the larger the number of instances, the longer it takes to find the best answer. The exhaustive search takes a long time compared to the exact approaches [29]. Graph theory [30,31], gradient-based [32–34] decomposition [35–38] and combinatorial [39,40] are some of the exact approaches in the literature examined.

Ningning et al. [30] employed graph partitioning theory to develop a fog computing load-balancing approach based on dynamic graph partitioning. The authors demonstrated that the fog computing framework could flexibly design the system network after cloud atomization and that the dynamic load-balancing mechanism was capable of customizing the system and decreasing the node migration caused by system modifications. Puthal et al. [31] also proposed a load balancing methodology to assess Edge Data Centers (EDCs) and discover less loaded EDCs for work distribution. When locating less loaded EDCs for work allocation, this strategy is more useful than previous ways. It not only improves the load balancing efficiency but also enhances security by requiring destination EDCs to authenticate.

Furthermore, Fan and Ansari [32] established a workload balancing model in a fog network for decreasing the data flow delay in processing operations and communications by connecting IoT devices to appropriate base stations. Fog computing was employed by Barros et al. [33] to reduce the logical distance between consumption and central distribution. The power flow information as managed more effectively and at a lesser cost by IoT devices at the network edge. They compared the performance of the Gauss–Seidel and Newton–Raphson methods to produce real-time computations of the load flow problem with the help of fog.

Beraldi and Alnuweiri [34] investigated load balancing among fog nodes and handled the fog system's unique problems. They used randomized-based load balancing methods, which took advantage of the power of random choice. They developed sequential probing as an alternative to traditional randomization procedures based on parallel exploration.

Chen and Kuehn [35] looked at the downlink of a cache-Enabled Fog-Radio Access Network (FRAN) and examined how to communicate while consuming the least amount of power. An effective load-balancing technique was proposed based on channel states. The proposed algorithm considered increasing the cache memory for a higher content hitting rate to be a cost-effective technique of achieving greener networks.

Maswood et al. [36] presented a Mixed-Integer Linear Programming (MILP) model for improving the bandwidth cost in routing, network link utilization, and server resource usage in the fog/cloud environment. They looked at load balancing strategies at both the server and network levels. Sthapit et al. [37] also provided remedies for circumstances where the cloud or fog is not available. The sensor network was first represented using a network of queues, and then a linear programming methodology was used to determine scheduling decisions while considering load balancing.

By employing connected automobile systems as an illustrative example, Chen et al. [38] demonstrated that vehicle mobility patterns may be used to execute periodic load balancing in fog servers. They suggested a task model for tackling the scheduling problem at the server level, rather than at the device level.

In addition, Dao et al. [39] proposed an Adaptive Resource Balancing (ARB) model to maximize serviceability in FRANs, in which Resource Block (RB) utilization within Remote Radio Heads (RRHs) is balanced using the Hungarian method and backpressure technique while taking into account a time-varying network topology caused by potential

RRH mobility. Mukherjee et al. [40] presented a load-balancing approach to specify the trade-off between computing delays and transmission in FRANs.

2.3. Fundamental Methods

Studies on load-balancing techniques in fog computing in the literature are based on simple methods without sophisticated computations, which are characterized as fundamental approaches. Shortest Job First [41], Throttled, Round Robin (RR) and First Fit [42,43] are examples. The selected fundamental methods are reviewed in this section. Nazar et al. [41] proposed a load-balancing algorithm based on the Modified Shortest Job First (MSJF) to manage user request load amongst VMs at the fog level to optimize the cloud and fog performance.

Ahmad et al. [42] presented an integrated cloud and fog-based platform for successfully managing energy in smart buildings. For load balancing, the First Fit (FF) method was used, which selects VMs based on memory block partitioning. Smart buildings with several flats contain IoT devices that were considered in the cloud/fog-based paradigm. Tariq et al. [43] created a fog-based system to cover a huge area of six global regions, each treated as a distinct region with many customers sending fog requests for access to the required resources.

Chekired et al. [44] also provided a decentralized scheduling architecture for the energy management of Electric Vehicles (EVs) based on the fog system paradigm, where an optimal Load Balancing Algorithm (LBA) was achieved using a priority-queuing model. Regarding specific multi-tenancy concerns, such as latency and priority, Neto et al. [45] suggested a Multi-tenant Load Distribution technique for Fog networks (MtLDF). They also gave case examples to demonstrate the applicability of the suggested strategy in comparison to a latency-driven load distribution mechanism.

Batista et al. [46] offered an approach based on performing load balancing needs for the Fog of Things (FoT) platforms through programmability for distributed IoT settings using SDN. The authors dealt with the issues by using a FoT load-balancing technique and evaluating response time and lost samples as two measures. A load-balancing mechanism was proposed for efficiently selecting VMs within a fog system so that customers obtain a quick response with minimal latency.

Verma et al. [47] proposed an Efficient Load Balancing (ELB) methodology in addition to fog-cloud-based architecture. It used the information replication methodology to keep those data on fog networks, reducing the overall reliance on big data centers. Talaat et al. [48] provided an influential load-balancing approach for fog systems that is suitable for healthcare applications. ELBS used caching methods and real-time scheduling to achieve essential load balancing in a fog environment. The authors presented ELBS for fog environments that are suitable for healthcare applications.

2.4. Hybrid Methods

Hybrid approaches use a combination of approximate, accurate, and basic methods to achieve load balancing in fog networks [48–53]. This section looks at studies that used hybrid methodologies. Naqvi et al. [49] introduced a fog computing concept to boost cloud computing processing speed as a cloud computing supplement.

For request processing, fog nodes with four to nine VMs employ service broker policies. The Ant Colony Optimization (ACO) load-balancing algorithm, throttle, and RR are used to balance the load on virtual machines. Abbasi et al. [50] focused on the application of fog computing to a Smart Grid (SG) that consists of a distributed generation environment known as a microgrid.

The study's goal was to enhance the reaction time, delay time, and resource use. The proposed virtual machine load-balancing technique outperformed the other strategies in the study. To increase the communication between consumers and the electrical supplier, Ali et al. [51] presented a four-layered SG-based architecture, which covered a large area of residents. For VM allocation, three load balancing strategies were used, with the service broker policies used for simulations being the most dynamically reconfigurable and closest to data centers.

By employing bin pack approaches, Zubair et al. [51] employed a Genetic Algorithm (GA), throttle and Round Ronin (RR) for load-balancing mechanisms. In this study, an SG was combined with fog, as well as a cloud-based model and three locations with some buildings. Waseem et al. [52] proposed a Software-Defined Network (SDN)-based technique. Furthermore, Talaat et al. [53] introduced a load-balancing strategy in a fog-based healthcare setting employing the dynamic resource allocation methodology based on Q-learning and GA.

The load-balancing strategy continuously monitors network traffic, collects information about each server's load and manages incoming requests using a dynamic resource allocation mechanism to spread them across the available servers. Talaat et al. [53] provided an influential load-balancing approach for fog systems that is suitable for healthcare applications. The approach uses caching methods and real-time scheduling to achieve essential load balancing in a fog environment. The authors presented ELBS for fog environments that are suitable for healthcare applications.

In the literature survey, various load-balancing mechanisms were studied. The most explored parameters by various researchers were the response time, latency, energy, processing time and resource utilization. Other parameters, such as the scalability, reliability, security and throughput still require attention from researchers. Most researchers are assuming that virtual machines are already clustered and pay almost negligible attention to cluster-based techniques for improving the load balancing process.

In this paper, a framework is presented that considers clustering as a pre-process for load balancing. Artificial intelligence applications when combined with fog computing framework can immensely improve the end-user experience by minimizing latency for the time-sensitive scenarios. In the next section, the importance of AI in healthcare scenarios is explicated.

3. AI in Health Care

Artificial intelligence (AI) in healthcare employs a large number of complicated algorithms that automate the completion of specific tasks. AI is a term that refers to technology that can mimic and recreate human-like activities. These processes are related to learning, adapting and understanding. The simplest way to describe this technology is that it can "act like a person." Artificial intelligence can take numerous forms and is based on tools or ideas, such as biology, logic and mathematics. When data is injected into computers by researchers, doctors and scientists, the newly developed algorithms can review, understand and even suggest remedies to difficult medical problems [54,55]. Here are some of the most recent technological applications of AI in healthcare [55–59]:

- Diagnostics in medicine

Artificial Intelligence is used in medical diagnostics to diagnose patients with specific disorders. AI can predict the disease with accuracy by analyzing the symptoms in much earlier stages of the disease. Early detection of the diseases plays a critical role in the treatment and recovery of the patient. AI can help in diagnosis by decision making and automating the workflow within the hospital and home patients as well.

- Drug discovery

Drug discovery is the process of finding a new medicine. Artificial Intelligence is being used by dozens of health and pharmaceutical businesses to assist with drug development and to improve the lengthy timelines and processes associated with developing and bringing pharmaceuticals to market [60].

- Clinical trials

AI technologies are helpful in making sense out of complex data and solving difficult problems. They tend to provide accurate answers for difficult problems by utilizing machine learning approaches. Clinical trials are experiments, tests done on humans to study and evaluate the effects of drugs or clinical processes don on human. Incorporating AI approaches with wearable technology allows for efficient, real-time and tailored patient monitoring that can be done automatically and constantly throughout the trial. This can help ensure that protocol requirements are followed and that the endpoint assessment is accurate.

- Pain management

Patients can use AI-enabled tools to navigate the maze of chronic pain, map out their symptoms, discover and correct trends, find alternative treatments and obtain assistance for enhancing the patients' quality of life. Pain management is still a recent phenomenon of focus in healthcare. One approach for managing pain could be to simulate realities that can distract patients from their current source of pain and even aid with the opioid by combining virtual reality with artificial intelligence.

- Improving the treatment outcomes

Artificial-intelligence-driven techniques can improve patient care in a variety of ways by embedding it with smart home architecture. Amazon Alexa is a great example of such applications. It can help in various ways, such as providing diabetes patients with the tools they need to effectively manage their solutions. The assistance of elder people with medication administration interaction at the hospital can be improved. Alexa can assist with blood pressure management. Alexa can be used to purchase and manage insurance claims. Live first aid tutorials can be accessed. Obtaining real-time information about the hospital before arrival and receiving health advice to help live a healthier lifestyle. Alexa is providing diagnostic suggestions. It can provide hands-free assistance to emergency medical technicians.

The applications of AI discussed in this section can perform better if these scenarios are combined with the fog computing framework, which is further supported by the cloud. AI plays a significant role in healthcare not only for real-time applications but also for research in healthcare. Artificial environments provided by AI can help in exploring the new techniques for handling devices and inventing new gadgets for handling patients easily. The next section discusses the methods of the literature used in this paper for studying load-balancing mechanisms in fog computing.

4. Research Methodology

In this section, a detailed review is presented for classifying the load-balancing approaches used in fog-based systems. To discover important synonyms and keywords for the techniques, the following string terms were used: ("Load" OR "Load Balancing" AND "Fog Computing" "OR "Load balancing in fog computing").

A total of 963 documents were found in the search to determine the research papers related to loading balancing in fog computing. The documents were filtered based on the following exclusion and inclusion criteria: (1) non-English articles were excluded. (2) Conference papers were also taken into account. (3) Book and book chapters were excluded. (4) Only peer-reviewed articles were included. (5) Documents having a length of fewer than six pages were not taken into account.

Various renowned publishers, such as IEEE, Elsevier, Springer, ACM, Wiley and Hindawi, have published a good number of papers. Figure 2 depicts the count of research documents publisher wise according to year. Since 2016, IEEE has published the highest number of papers. Elsevier and Springer have an almost similar number of papers and are at second and third position respectively.

Load-balancing techniques are divided into the following categories, namely estimated, precise, primary and hybrid. Figure 3 presents a percentage-wise distribution of load-balancing techniques in fog computing. Estimating techniques have a maximum share in the techniques utilized for balancing the load in a fog environment. Precise techniques are nearly equal to estimated techniques in terms of percentage. The hybrid techniques have the least share in the percentage-wise distribution of techniques.

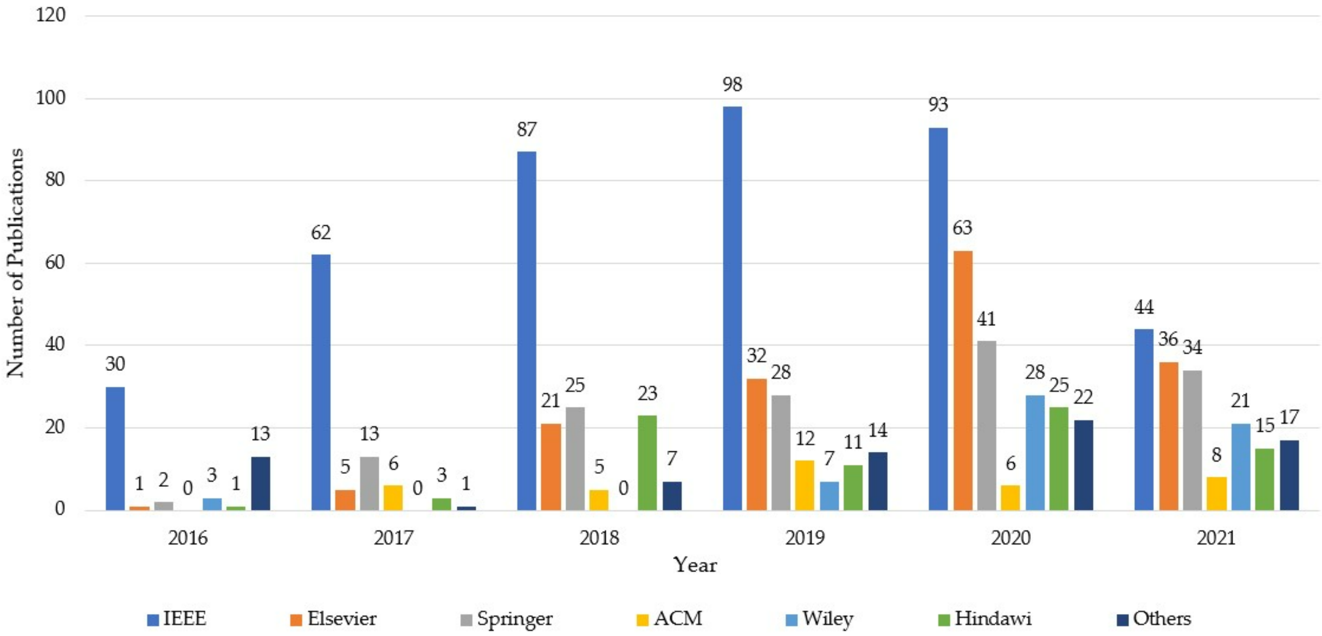

Figure 2. Year-wise research paper publications by publisher.

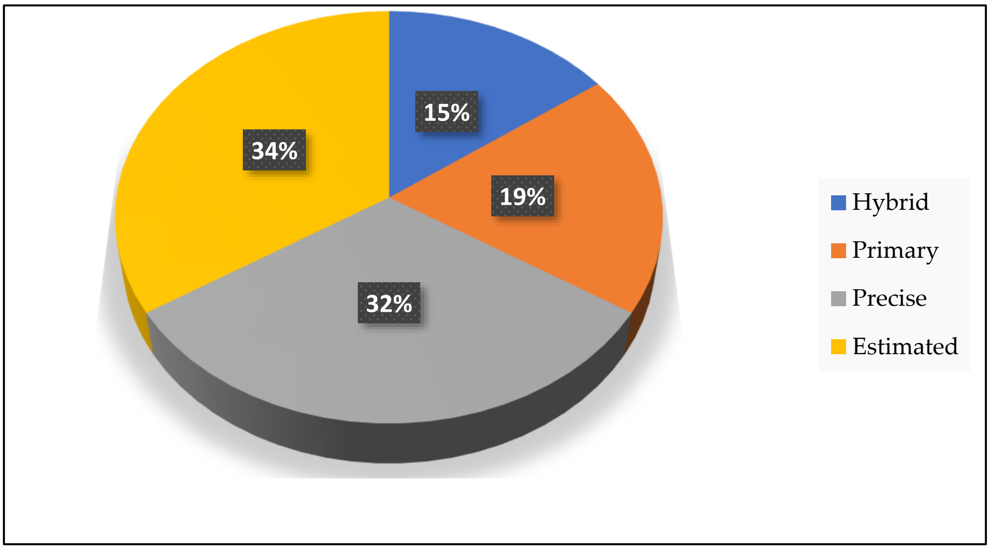

Figure 3. Percentage-wise load-balancing techniques used in fog computing.

Evaluation parameters for measuring the performance of load-balancing algorithms implemented in a fog environment are depicted percentage-wise in Figure 4. The maximum share of articles has focused on the response time of fog nodes. After that, 18% focused on cost minimization. The energy parameter also has a significant share among all i.e., 15%. Processing time was focused on in 13% of the research documents considered in the literature review.

The resource utilization and scalability were 13% and 9%, respectively. Security, throughput and reliability were given less preference by researchers as a few documents are associated with improving these parameters in comparison to other parameters considered in this study. From time to time, various tools have been introduced to simulate and implement the techniques for load-balancing mechanisms. As fog computing has been derived from cloud computing, some of the tools utilized for implementing cloud-based techniques have been utilized in fog computing also.

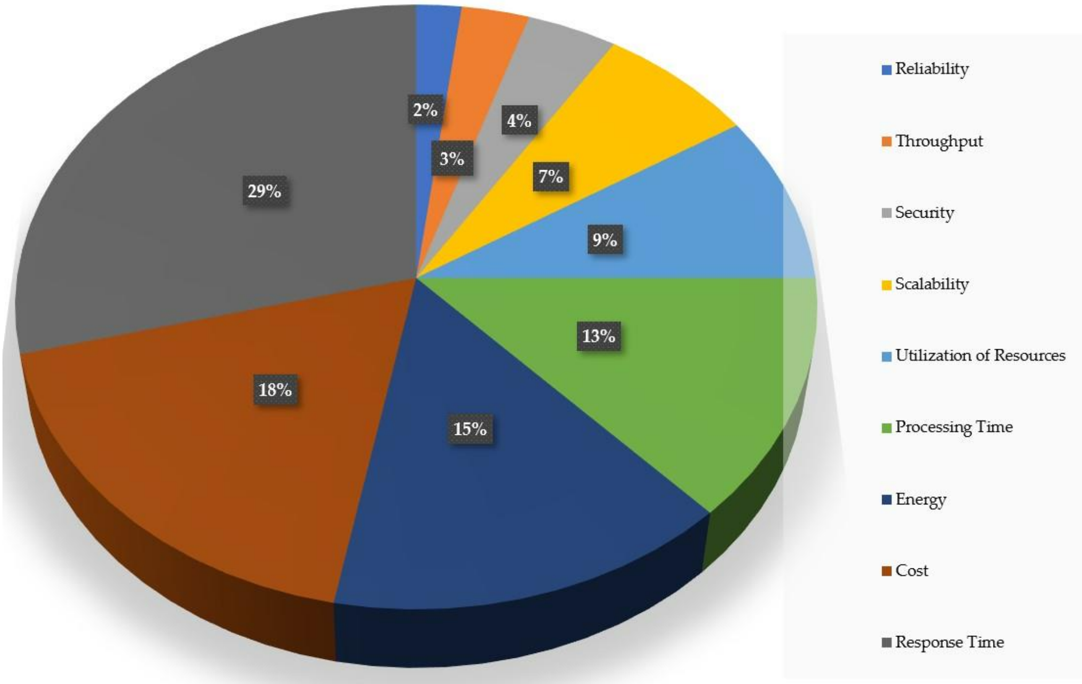

Figure 4. Percentage-wise evaluation parameters for load balancing in fog computing.

Figure 5 shows a percentage-wise tool used by researchers for implementing and simulating the load balancing fog environment for improving the evaluation parameters. The highest percentage of researchers did not mention the tools for implementation. Cloud Analyst and MATLAB had similar utilization percentages, i.e., 19% and 16%. Cloud Sim, java, ifogsim, Ns-2/NS-3 and work Robots had 9%, 7%, 6%, 5% and 4% shares in a tool pie chart. Schter, custom simulator, Jmeter and Mininet had 3%, 3%, 2%and 2% shares in the tool chart.

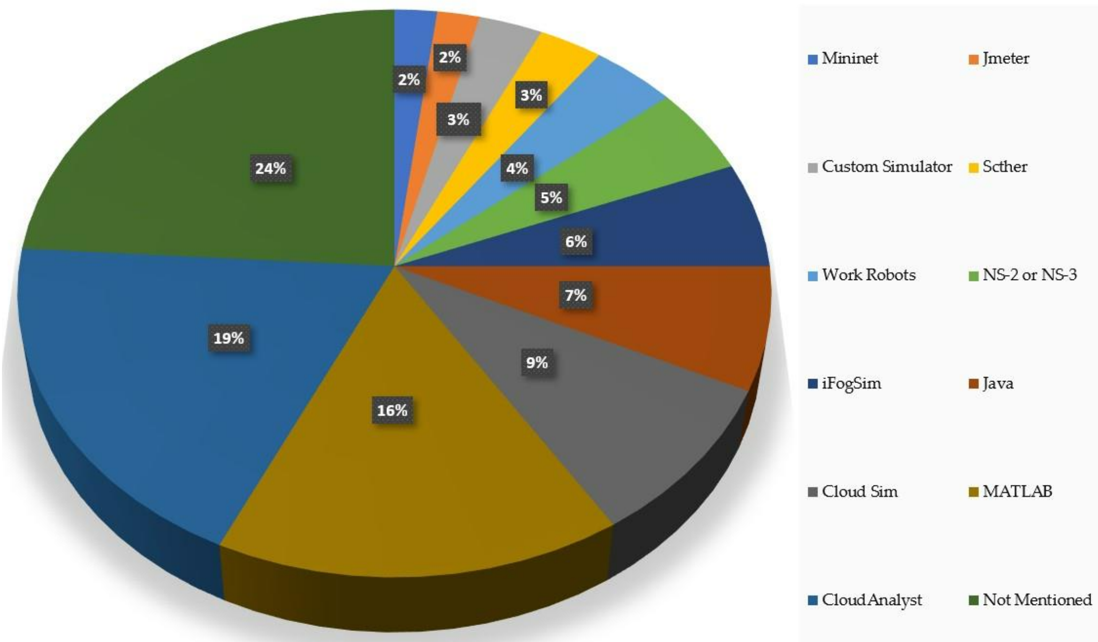

Figure 5. Percentage-wise tools for implementing load balancing in fog computing.

5. Proposed Framework

In the proposed framework presented in Figure 6, there are additions to the basic fog architecture. AI-enabled smart devices are the edge devices that can sense the various parameters of the human body, such as the heart rate, oxygen level, blood pressure, calories burned and the activity performed. These end devices have monitoring systems and wearable devices, which are heterogeneous and can have different specifications.

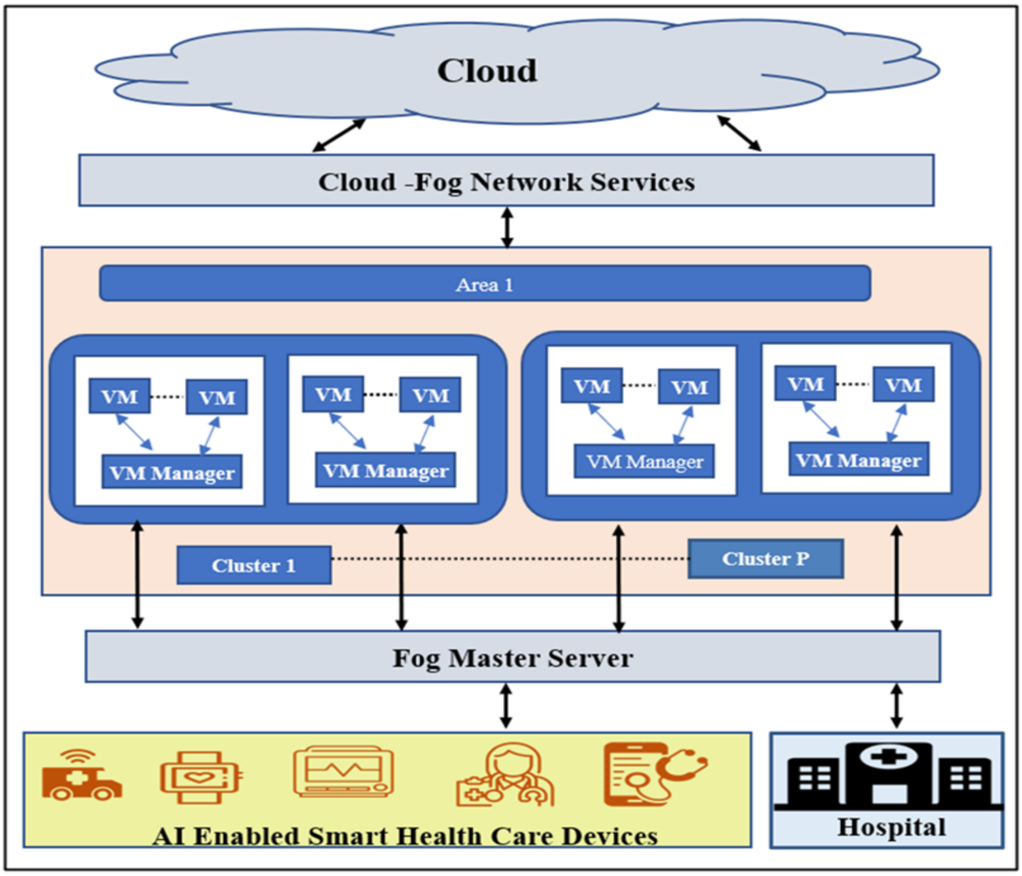

Figure 6. Framework for fog-enabled communication in healthcare.

Intelligent end-user devices coordinate and monitor the user's or patient's health remotely. End-user health care devices have limited computing capabilities. They can rely on the fog layer for advanced computation capability. Most of the responses can be automatically generated for the patient but some situations still require intervention from the hospital in the case of emergency or the situation for which no automated response works well.

To minimize the latency, response time and bandwidth requirements, the fog layer can effectively handle such situations and respond to patients in minimal time. However, if all the requests from various applications are handled by the cloud directly, it could be possible that latency-sensitive applications may suffer as the cloud is having enormous computation and large storage. It deals with multiple applications and their data at a time. If the fog layer is introduced in-between the cloud and end-user, latency-sensitive applications will be served better. As request processing is done at the fog layer, it can work with a limited amount of resources, such as the bandwidth, cost and time. The processing is done closer to the end clients.

Some of the assumptions used in this framework are:

1. All the end-user devices are AI-enabled smart devices that are capable of sensing the parameters related to the human body.

2. The fog master server is assumed to be self-adaptive.

The fog layer consists of fog nodes with various virtual machines. To provide better services to patients, an efficient load-balancing mechanism is required at the fog layer. These VMs are grouped according to their storage, functionality, computation capability and specifications. This group of VMs along with the virtual machine manager is known as a cluster. This grouping is expected to help in the speedy allocation of tasks and reduce latency time. Each Virtual Machine Manager (VM Manager) is further connected with the Fog Master Server (FMS) of that area.

FMS is near to the end-user and hospital. The fog master server is responsible for allocating the tasks cluster-wise and balancing the load among clusters considering the availability of resources. Within a cluster, the fog master server (as presented in Figure 7) consists of three modules: Data Routing, Task Allocation and Cluster Formation. The task allocation module further consists of three modules, namely Discovery, Benchmarking and Load Balancing. Each VM manager is further connected with the FMS of that area. The elaboration of the modules of the FMS is presented as follows.

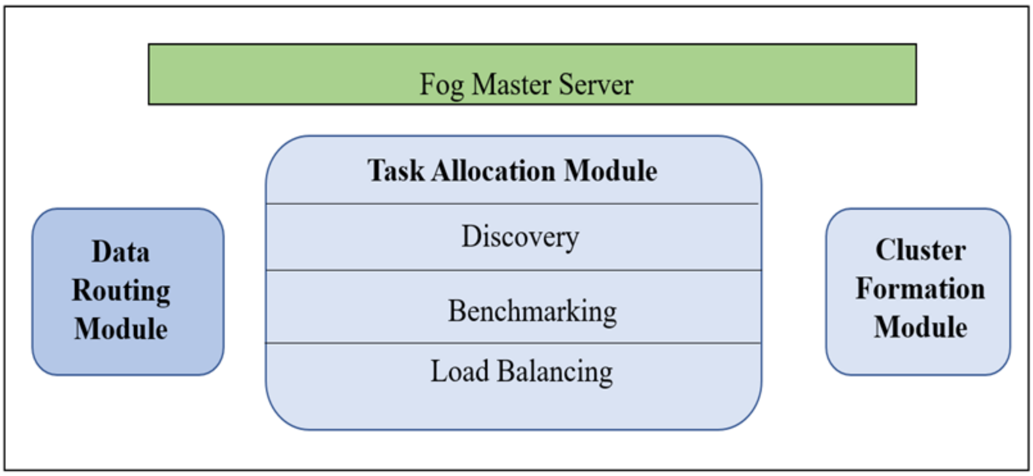

Figure 7. Modules in the Fog Master Server.

Data Routing Module: This module is responsible for routing the data between fog nodes, end devices and hospitals.

Task Allocation Module: Task Allocation is done in three parts, including the discovery of the best suitable VM according to the request generated by end devices and a benchmarking process that involves computing the success count after completion of a job/task assigned to it. The load-balancing procedure ensures that neither of the VM is underloaded or overloaded. Algorithmic steps are required for the functioning of the task allocation module. A cluster-based algorithm for a load-balancing algorithm is presented in Algorithm 1.

Discovery: Discovery module is responsible for finding the suitable cluster and VM for allocating the task.

Benchmarking: This module collects the feedback after allocation of the task, such as the delay, response time and QoS parameters related to task allocation. This information is further utilized at the time of the next task allocation.

Load Balancing: Generally, the load balancing module ensures that none of the virtual machines present within the network are overloaded and that the load is almost equally distributed.

Here, the uniqueness in the load balancing module is because of the concept of two-level load balancing used. In this proposed scenario, load balancing will be done at two levels. Once it will be done by the virtual machine manager within the clusters, another load balancing will be done at the fog master server for equal load distribution among clusters. Thus, neither virtual machines should be overloaded nor clusters should be overloaded.

Cluster Formation Module: This module is responsible for the grouping of VMs, which have similar specifications, storage and computation capabilities.

The virtual machine manager communicates with the fog master server for auditing and reporting the information about the cluster. The virtual machine manager at each cluster is responsible for task allocation and balancing the load within the cluster. By using the efficient clustering mechanism, load balancing can be achieved in the proposed scenario. The hospital can monitor and communicate with the patient in case of additional help and monitoring for diagnosis and prevention of disease. Hospitals can provide appropriate guidelines to patients either through an automated system or manually depending upon the circumstances and parameters collected by end devices and the fog layer. These guidelines can be given by video conferencing or calling.

Algorithm 1: Cluster-Based Algorithm for Load Balancing in Fog Computing.

1. Set up the N number of fog devices $FN1, FN2, \ldots, FNN$
2. Estimate the R number of incoming requests $RQ1, RQ2, \ldots, RQR$
3. Estimate the total number of clusters C as $C1, C2, \ldots, CC$
4. Assign fog devices to each cluster (Ci), and the cluster size (CS) is computed as follows:
$$Cluster\ size\ = Total\ number\ of\ fog\ devices/Total\ No.of\ clusters$$
$$CS\ = N/\mathrm{C}$$
5. Assign each cluster with CS number of Virtual Machines as $VM1, VM2\ \ldots, VMS$.
6. **For every incoming request $\boldsymbol{RQi\ = RQ1, RQ2, \ldots, RQ_R}$ do:**
7. **For each cluster $\boldsymbol{Cj\ = C1, C2, \ldots, CZ}$, do**
8. Find out the locally optimal virtual machine having better efficiency (MIPS), least loaded in Cj and high value for success count. Success count is computed by each VM as follows:
$$Success\ Count = Total\ number\ of\ requests\ successfully\ fulfilled\ by\ the\ VM$$
$$/Total\ number\ of\ requests\ assigned\ to\ a\ VM$$
9. Store the index of the best virtual machine of Cj in the array,
$$Local - Best[j]$$
10. **End of For loop of step 7.**
11. **For each cluster indexed $\boldsymbol{j\ = 1, \ldots, Z}$, do**
12. Find out the Global Best VM, GVM for Ri having better efficiency (MIPS), least loaded among the local best machines and having higher value of success count for selected VM in each cluster, from the array:
$$Local - Best\ [j].$$
13. **End of For loop of step 11.**
14. Assign the task Ri to the Global Best VM, GVM.
15. Repeat step 5 until all requests/tasks have been completed.
16. **End of For loop of step 6.**

Figure 8 shows the workflow of the proposed framework. This module is responsible for the grouping of VMs, which have similar specifications, storage and computation capabilities. The step-by-step working is presented below:

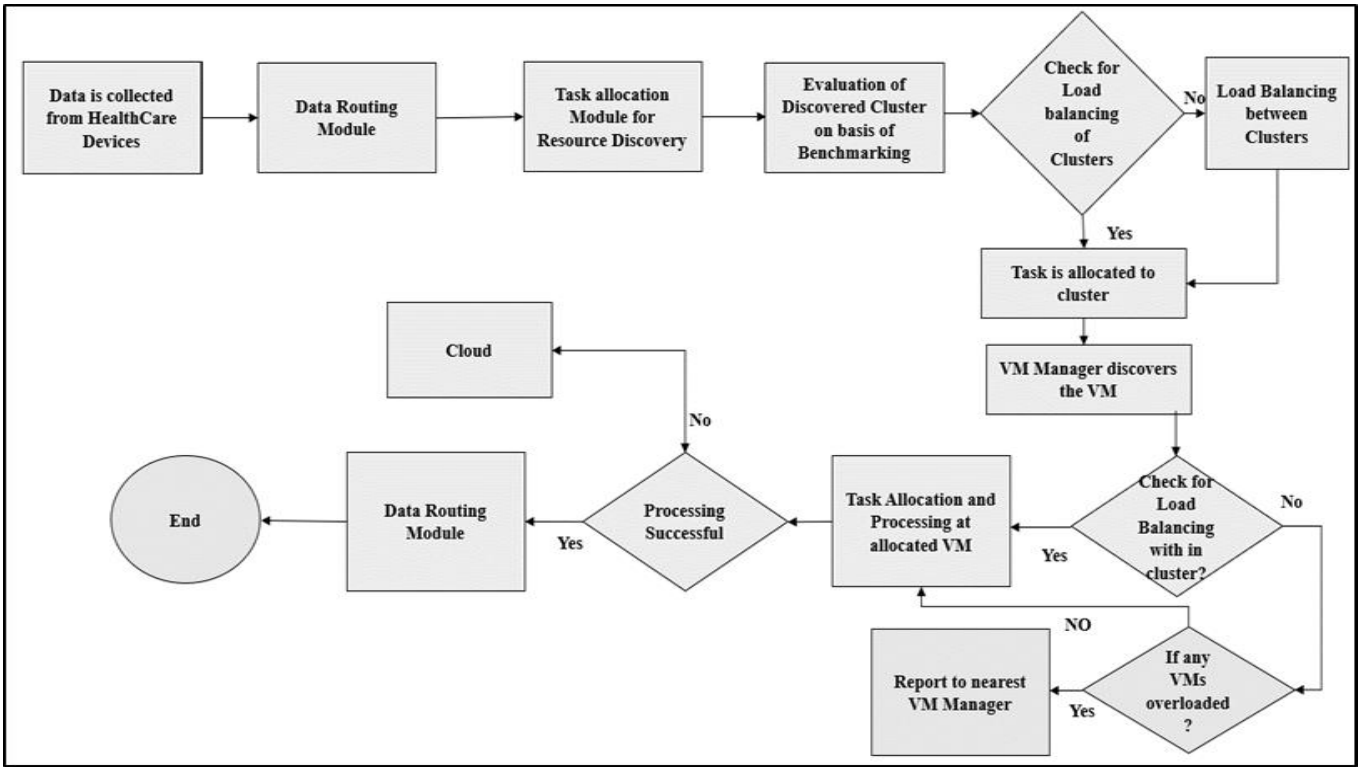

Figure 8. Work flow of the proposed framework.

Step 1. Fog computing layers consist of multiple virtual machines. These VMs are grouped based on similar parameters to form a cluster. Each cluster in the fog layer has a virtual machine manager.

Step 2. The fog layer has a Fog Master Server (FMS), which is further connected with the Virtual Machine manager of each cluster.

Step 3. Whenever any health care device or end-user device needs to utilize services of fog layer, it sends a request to the nearest Fog Master Server present in the geographical region.

Step 4. The Fog Master Server will perform allocation of resources within the region, this allocation is done with the help of modules present at the FMS. Modules present in FMS are explained previously. One unique idea in this work is the two-layer load balancing.

Step 5. Consider the case when the demand of Virtual Machine increases while execution of the task in real-time, then the Fog Master Server will have communication with next nearest Fog Master present in different geographical region and completes the execution of the real-time task.

Step 6. Another possibility while executing Step 5 is that if the data needed to perform a task are not available within the area covered by that FMS, then, in this case, the FMS will send the data request to the cloud layer with the help of cloud–fog network services.

Step 7. Finally, the updated data and task execution details are sent to the cloud layer, so that, in the case of non-availability of data and details at the fog layer, the cloud layer can be utilized.

The importance of the fog layer in this scenario is that patients can be provided with appropriate guidelines with minimum delay and also that they can rely on the fog layer for complex computations to be done at the fog layer. The fog nodes have greater computation capability then intelligent and smart end devices but are lower than the cloud layer.

If the fog layer is incapable of handling the requests from patients or users, they can be sent to the cloud layer. The cloud has enormous storage and computation capability. Although the delay will be more when the requests are handled at the cloud layer, they can be handled well as the computation capability does not have any bounds at the cloud layer.

Advantages of Proposed Framework

The proposed framework is expected to perform efficiently in terms of latency. Latency is expected to be less than in the cloud-only scenario when the end devices will be directly connected to the cloud. The cloud will handle the requests in a centralized scenario, whereas fog computing provides a decentralized scenario for handling the requests from users. The cloud has a large number of requests whereas in the fog scenario, requests are distributed among fog nodes.

Other advantages of the proposed framework include that, in critical conditions, when the nurse or doctor is not available in-person with the patient, an automated reply or steps for guiding the patient according to the problems faced can be dictated to the patient or the family member present nearby.

6. Conclusions and Future Directions

The proposed framework is capable of enhancing the service quality of intelligent healthcare scenarios in terms of establishing communication between the devices as load balancing among fog nodes can effectively reduce the latency time for providing services to patients. The clustering of fog nodes can help in reducing the offloading overhead of a task. It is subtle as, in the case that one virtual machine fails, another virtual machine within the cluster can be assigned that task as virtual machines within a cluster have similar capabilities. The novelty of the present work is the implementation of two-level load balancing for improved access to the service machines. This work, when combined with various artificial intelligence-based technological applications, can enhance the various parameters associated with health care services.

In future, this work will be simulated, implemented, and evaluated. Additionally, consultation sessions can be included in this framework. The patient can request teleconsultation or video consultation with a specialist health expert. As the records and clinical parameters are available either at the fog layer or cloud layer, the specialist can access the previously sensed and processed data and plan a consultation session accordingly. This framework can be utilized for providing immediate attention to critical patients that are in a state of emergency.

This framework is anticipated to help in determining the severity of the condition and can provide timely responses to patients as the fog layer is capable of handling the requests with a minimum delay. One aspect of the proposed work that still requires improvement is the security and trustworthiness of the healthcare data. The overhead incurred in the framework is that no additional layer has been added to filter the data that is envisaged towards routing the critical data.

Author Contributions: Conceptualization, S.M. and K.G.; Methodology, S.M. and K.G.; Validation, D.G. and A.S.; Formal Analysis, D.G. and A.S.; Investigation, M.I.; Resources, H.H.; Data Curation, A.O.-M.; Writing—Original Draft, S.M. and N.G.; Writing—Review Editing, N.G., Supervision, K.G.; Project Administration, H.H. and M.I. All authors have read and agreed to the published version of the manuscript.

Funding: The authors thank Natural Sciences and Engineering Research Council of Canada (NSERC) and New Brunswick Innovation Foundation (NBIF) for the financial support of the global project. These granting agencies did not contribute in the design of the study and collection, analysis, and interpretation of data.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Khan, S.; Nazir, S.; García-Magariño, I.; Hussain, A. Deep learning-based urban big data fusion in smart cities: Towards traffic monitoring and flow-preserving fusion. *Comput. Electr. Eng.* **2020**, *89*, 106906. [CrossRef]
2. Malik, S.; Gupta, K. Resource Scheduling in Fog: Taxonomy and Related Aspects. *J. Comput. Theor. Nanosci.* **2019**, *16*, 4313–4319. [CrossRef]
3. Mamata, R.; Solanki, V.K. Performance improvement in contemporary health care using IoT allied with big data. In *Handbook of Data Science Approaches for Biomedical Engineering*; Academic Press: Cambridge, MA, USA, 2020; pp. 103–119.
4. Zaguia, A.; Ameyed, D.; Haddar, M.; Cheikhrouhou, O.; Hamam, H. Cognitive IoT-Based e-Learning System: Enabling Context-Aware Remote Schooling during the Pandemic. *J. Heal. Eng.* **2021**, *2021*, 1–12. [CrossRef]
5. Kamali, G.; Katiyar, V. Deadline-credit aware heuristic for dynamic resource provisioning in virtualised cloud environment. *Int. J. Internet Technol. Secur. Trans.* **2019**, *9*, 390–403.
6. Farahani, B.; Barzegari, M.; Aliee, F.S.; Shaik, K.A. Towards collaborative intelligent IoT eHealth: From device to fog, and cloud. *Microprocess. Microsystems* **2019**, *72*, 102938. [CrossRef]
7. Muhammad, S.; Safdar, M.T.; Shahzada, K.; Abid, G.A.; Fawaz, A.; Habib, H.; Omar, C.; Muhammad, M.S. Efficient Joint Key Authentication Model in E-Healthcare. *CMC-Comput. Mater. Contin.* **2022**, *71*, 2739–2753.
8. Dave, M.; Rastogi, V.; Miglani, M.; Saharan, P.; Goyal, N. Smart Fog-Based Video Surveillance with Privacy Preservation based on Blockchain. *Wirel. Pers. Commun.* **2021**, 1–18. [CrossRef]
9. Mostafa, N. Cooperative fog communications using a multi-level load balancing. In Proceedings of the 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC), Rome, Italy, 10–13 June 2019; pp. 45–51.
10. Bali, M.S.; Gupta, K.; Malik, S. Integrating IoT with Blockchain: A Systematic Review. *IoT Anal. Sens. Netw.* **2022**, *244*, 355–369.
11. Malik, S.; Gupta, K. Smart City: A new phase of sustainable development using fog computing and IoT. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Sanya, China, 12–14 November 2021; IOP Publishing: Bristol, UK, 2021; Volume 1022, p. 012093.
12. Ijaz, M.; Li, G.; Lin, L.; Cheikhrouhou, O.; Hamam, H.; Noor, A. Integration and Applications of Fog Computing and Cloud Computing Based on the Internet of Things for Provision of Healthcare Services at Home. *Electronics* **2022**, *10*, 1077. [CrossRef]
13. Kumar, M.; Raju, K.S.; Kumar, D.; Goyal, N.; Verma, S.; Singh, A. An efficient framework using visual recognition for IoT based smart city surveillance. *Multimed. Tools Appl.* **2021**, *80*, 31277–31295. [CrossRef]
14. Malik, S.; Gupta, K.; Singh, M. Resource Management in Fog Computing Using Clustering Techniques: A Systematic Study. *Ann. Rom. Soc. Cell Biol.* **2020**, *24*, 77–92.
15. Mahmud, R.; Kotagiri, R.; Buyya, R. Fog computing: A taxonomy, survey and future directions. In *Internet of Everything*; Springer: Singapore, 2018; pp. 103–130.
16. Abro, A.; Deng, Z.; Memon, K.A.; Laghari, A.A.; Mohammadani, K.H.; Ain, N.U. A Dynamic Application-Partitioning Algorithm with Improved Offloading Mechanism for Fog Cloud Networks. *Future Internet* **2019**, *11*, 141. [CrossRef]
17. Zahid, M.; Javaid, N.; Ansar, K.; Hassan, K.; Khan, M.K.; Waqas, M. Hill climbing load balancing algorithm on fog computing. In Proceedings of the International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Taichung, Taiwan, 27–29 October 2018; Springer: Cham, Germany, 2018; pp. 238–251. [CrossRef]
18. Kamal, M.B.; Javaid, N.; Naqvi, S.A.A.; Butt, H.; Saif, T.; Muhammad, D.K. Heuristic min-conflicts optimizing technique for load balancing on fog computing. In Proceedings of the International Conference on Intelligent Networking and Collaborative Systems, Bratislava, Slovakia, 5–7 September 2018; Springer: Cham, Germany, 2018; pp. 207–219.
19. Banaie, F.; Yaghmaee, M.H.; Hosseini, S.A.; Tashtarian, F. Load-Balancing Algorithm for Multiple Gateways in Fog-Based Internet of Things. *IEEE Internet Things J.* **2020**, *7*, 7043–7053. [CrossRef]
20. Oueis, J.; Strinati, E.C.; Barbarossa, S. The Fog Balancing: Load Distribution for Small Cell Cloud Computing. In Proceedings of the 2015 IEEE 81st Vehicular Technology Conference (VTC Spring), Glasgow, Scotland, 11–14 May 2015; pp. 1–6.
21. Xu, X.; Liu, Q.; Qi, L.; Yuan, Y.; Dou, W.; Liu, A.X. A heuristic virtual machine scheduling method for load balancing in fog-cloud computing. In Proceedings of the 2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS), Omaha, NE, USA, 3–5 May 2018; pp. 83–88.
22. Nanda, S.J.; Panda, G. A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm Evol. Comput.* **2014**, *16*, 1–18. [CrossRef]
23. He, X.; Ren, Z.; Shi, C.; Fang, J. A novel load balancing strategy of software-defined cloud/fog networking in the Internet of Vehicles. *China Commun.* **2016**, *13*, 140–149. [CrossRef]
24. Wan, J.; Chen, B.; Wang, S.; Xia, M.; Li, D.; Liu, C. Fog computing for energy-aware load balancing and scheduling in smart factory. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4548–4556. [CrossRef]
25. Shi, C.; Ren, Z.; He, X. Research on load balancing for software defined cloud-fog network in real-time mobile face recognition. In *International Conference on Communicatins and Networking in China*; Springer: Cham, Germany, 2016; pp. 121–131.
26. Yang, J. Low-latency cloud-fog network architecture and its load balancing strategy for medical big data. *J. Ambient Intell. Humaniz. Comput.* **2020**, 1–10. [CrossRef]
27. Singh, S.P.; Sharma, A.; Kumar, R. Design and exploration of load balancers for fog computing using fuzzy logic. *Simul. Model. Pr. Theory* **2019**, *101*, 102017. [CrossRef]

28. Abedin, S.F.; Bairagi, A.K.; Munir, S.; Tran, N.H.; Hong, C.S. Fog Load Balancing for Massive Machine Type Communications: A Game and Transport Theoretic Approach. *IEEE Access* **2018**, *7*, 4204–4218. [CrossRef]
29. Woeginger, G.J. Exact algorithms for NP-hard problems: A survey. In *Combinatorial Optimization—Eureka, You Shrink!* Springer: Berlin/Heidelberg, Germany, 2003; pp. 185–207.
30. Ningning, S.; Chao, G.; Xingshuo, A.; Qiang, Z. Fog computing dynamic load balancing mechanism based on graph repartitioning. *China Commun.* **2016**, *13*, 156–164. [CrossRef]
31. Puthal, D.; Obaidat, M.S.; Nanda, P.; Prasad, M.; Mohanty, S.P.; Zomaya, A.Y. Secure and sustainable load balancing of edge data centers in fog computing. *IEEE Commun. Mag.* **2018**, *56*, 60–65. [CrossRef]
32. Fan, Q.; Ansari, N. Towards Workload Balancing in Fog Computing Empowered IoT. *IEEE Trans. Netw. Sci. Eng.* **2018**, *7*, 253–262. [CrossRef]
33. Barros, E.; Peixoto, M.; Leite, D.; Batista, B.; Kuehne, B. A fog model for dynamic load flow analysis in smart grids. In Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC), Natal, Brazil, 25–28 June 2018; pp. 1–6.
34. Beraldi, R.; Alnuweiri, H. Sequential randomization load balancing for fog computing. In Proceedings of the 2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Supetar, Croatia, 13–15 September 2018; pp. 1–6.
35. Chen, D.; Kuehn, V. Adaptive radio unit selection and load balancing in the downlink of fog radio access network. In Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 4–8 December 2016; pp. 1–7.
36. Maswood, M.M.S.; Rahman, R.; Alharbi, A.G.; Medhi, D. A Novel Strategy to Achieve Bandwidth Cost Reduction and Load Balancing in a Cooperative Three-Layer Fog-Cloud Computing Environment. *IEEE Access* **2020**, *8*, 113737–113750. [CrossRef]
37. Sthapit, S.; Thompson, J.; Robertson, N.M.; Hopgood, J.R. Computational Load Balancing on the Edge in Absence of Cloud and Fog. *IEEE Trans. Mob. Comput.* **2018**, *18*, 1499–1512. [CrossRef]
38. Chen, Y.-A.; Walters, J.P.; Crago, S.P. Load balancing for minimizing deadline misses and total runtime for connected car systems in fog computing. In Proceedings of the 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), Guangzhou, China, 12–15 December 2017; pp. 683–690.
39. Dao, N.-N.; Lee, J.; Vu, D.-N.; Paek, J.; Kim, J.; Cho, S.; Chung, K.-S.; Keum, C. Adaptive Resource Balancing for Serviceability Maximization in Fog Radio Access Networks. *IEEE Access* **2017**, *5*, 14548–14559. [CrossRef]
40. Mukherjee, M.; Liu, Y.; Lloret, J.; Guo, L.; Matam, R.; Aazam, M. Transmission and latency-aware load balancing for fog radio access networks. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, UAE, 9–13 December 2018; pp. 1–6.
41. Nazar, T.; Javaid, N.; Waheed, M.; Fatima, A.; Bano, H.; Ahmed, N. Modified shortest job first for load balancing in cloud-fog computing. In Proceedings of the International Conference on Broadband and Wireless Computing, Communication and Applications, Fukuoka, Japan, 28–30 October 2018; Springer: Cham, Germany, 2018; pp. 63–76.
42. Ahmad, N.; Javaid, N.; Mehmood, M.; Hayat, M.; Ullah, A.; Khan, H.A. Fog-cloud based platform for utilization of resources using load balancing technique. In Proceedings of the International Conference on Network-Based Information Systems, Bratislava, Slovakia, 5–7 September 2018; Springer: Cham, Germany, 2018; pp. 554–567.
43. Tariq, S.; Javaid, N.; Majeed, M.; Ahmed, F.; Nazir, S. Priority Based Load Balancing in Cloud and Fog Based Systems. In Proceedings of the International Conference on Broadband and Wireless Computing, Communication and Applications, Taichung, Taiwan, 27–29 October 2018; Springer: Cham, Germany, 2018; pp. 725–736.
44. Chekired, D.A.; Khoukhi, L.; Mouftah, H.T. Queuing model for EVs energy management: Load balancing algorithms based on decentralized fog architecture. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6.
45. Neto, E.C.P.; Callou, G.; Aires, F. An algorithm to optimise the load distribution of fog environments. In Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 5–8 October 2017; pp. 1292–1297.
46. Batista, E.; Figueiredo, G.; Peixoto, M.; Serrano, M.; Prazeres, C. Load balancing in the fog of things platforms through software-defined networking. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 30 July–3 August 2018; pp. 1785–1791.
47. Verma, S.; Yadav, A.K.; Motwani, D.; Raw, R.S.; Singh, H.K. An efficient data replication and load balancing technique for fog computing environment. In Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 16–18 May 2016; pp. 2888–2895.
48. Talaat, F.M.; Ali, S.H.; Saleh, A.I.; Ali, H. Effective Load Balancing Strategy (ELBS) for Real-Time Fog Computing Environment Using Fuzzy and Probabilistic Neural Networks. *J. Netw. Syst. Manag.* **2019**, *27*, 883–929. [CrossRef]
49. Naqvi, S.A.A.; Javaid, N.; Butt, H.; Kamal, M.B.; Hamza, A.; Kashif, M. Metaheuristic optimization technique for load balancing in cloud-fog environment integrated with smart grid. In Proceedings of the International Conference on Network-Based Information Systems, Bratislava, Slovakia, 5–7 September 2018; Springer: Cham, Germany, 2018; pp. 700–711.
50. Abbasi, S.H.; Javaid, N.; Ashraf, M.H.; Mehmood, M.; Naeem, M.; Rehman, M. Load stabilizing in fog computing environment using load balancing algorithm. In Proceedings of the International Conference on Broadband and Wireless Computing, Communication and Applications, Fukuoka, Japan, 28–30 October 2018; Springer: Cham, Germany, 2018; pp. 737–750.

51. Zubair, M.; Javaid, N.; Ismail, M.; Zakria, M.; Zaheer, M.A.; Saeed, F. Integration of cloud-fog based platform for load balancing using hybrid genetic algorithm using bin packing technique. In Proceedings of the International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Tunghai University, Taichung, Taiwan, 27–29 October 2018; pp. 279–292.
52. Waseem, Q.; Alshamrani, S.; Nisar, K.; Din, W.W.; Alghamdi, A. Future Technology: Software-Defined Network (SDN) Forensic. *Symmetry* **2021**, *13*, 767. [CrossRef]
53. Talaat, F.M.; Saraya, M.S.; Saleh, A.I.; Ali, H.A.; Ali, S.H. A load balancing and optimization strategy (LBOS) using reinforcement learning in fog computing environment. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 1–16. [CrossRef]
54. Shah, R.; Chircu, A. IoT and AI in healthcare: A systematic literature review. *Issues Inf. Syst.* **2018**, *19*.
55. Shaheen, M.Y. AI in Healthcare: Medical and socio-economic benefits and challenges. *ScienceOpen Preprints* **2021**. [CrossRef]
56. Panesar, A. *Machine Learning and AI for Healthcare*; Apress: Coventry, UK, 2019.
57. Nisar, K. Smart Home: Multisensor Information Fusion towards Better Healthcare. *Adv. Sci. Lett.* **2018**, *24*, 1896–1901. [CrossRef]
58. Haider, I.; Khan, K.B.; Haider, M.A.; Saeed, A.; Nisar, K. Automated Robotic System for Assistance of Isolated Patients of Coronavirus (COVID-19). In Proceedings of the 2020 IEEE 23rd International Multitopic Conference (INMIC), Bahawalpur, Pakistan, 5–7 November 2020; pp. 1–6.
59. Sodhro, A.H.; Al-Rakhami, M.S.; Wang, L.; Magsi, H.; Zahid, N.; Pirbhulal, S.; Nisar, K.; Ahmad, A. Decentralized Energy Efficient Model for Data Transmission in IoT-based Healthcare System. In Proceedings of the 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring), Helsinki, Finland, 25–28 April 2021; pp. 1–5.
60. Sarkar, N.I.; Kuang, A.X.; Nisar, K.; Amphawan, A. Hospital environment scenarios using WLAN over OPNET simulation tool. *Int. J. Inf. Commun. Technol. Hum. Dev. (IJICTHD)* **2014**, *6*, 69–90. [CrossRef]