Machine Learning for Cloud, Fog, Edge and Serverless Computing Environments: Comparisons, Performance Evaluation Benchmark and Future Directions

Parminder Singh

School of Computer Science and Engineering, Lovely Professional University, Phagwara, India;
E-mail: parminder.16479@lpu.co.in

Avinash Kaur

School of Computer Science and Engineering, Lovely Professional University, Phagwara, India;
E-mail: avinash.14557@lpu.co.in

Sukhpal Singh Gill*

School of Electronic Engineering and Computer Science, Queen Mary University of London, Mile End Rd, Bethnal Green, London E1 4NS, United Kingdom;
E-mail: s.s.gill@qmul.ac.uk
*Corresponding author

Abstract: The compute-intensive and latency-sensitive Internet of Things (IoT) applications needs to utilize the services from various computing paradigms, but they are facing many challenges such as large value of latency, energy and network bandwidth. To analyse and understand these challenges, we designed a performance evaluation benchmark which integrates Cloud, Fog, Edge and Serverless computing to conduct a comparative study for IoT-based healthcare application. It gives the platform for the developers to design IoT applications based on user guidelines to run various applications concurrently on different paradigms. Furthermore, we used recent machine learning techniques for the optimization of resources, energy, cost and overheads to identify the best technique based on important Quality of Service parameters. Experimental results show that serverless computing performs better than non-serverless in terms of energy, latency, bandwidth, response time and scalability by 3.8%, 3.2%, 4.3%, 1.5% and 2.7%, respectively. Finally, various promising future directions are highlighted.

Keywords: Artificial Intelligence; Fog Computing; Edge Computing; Internet-of-Things; Machine Learning; Serverless Computing; Cloud Computing

Reference P. Singh, A. Kaur and S. S. Gill, Machine Learning for Cloud, Fog, Edge and Serverless Computing Environments: Comparisons, Performance Evaluation Benchmark and Future Directions, Int. J. Grid and Utility Computing, Vol. x, No. x, 2021

Biographical notes: Parminder Singh is working as an Associate Professor with the School of Computer Science and Engineering, Lovely Professional University. His research interests include machine learning, deep learning, blockchain, cloud/fog/edge computing, network security, and web services. He is an Active Member of IEEE and ACM. He has been the Session Chair and an Advisory Member for various international conferences.

Avinash Kaur is an Associate Professor in the Department of Computer Science and Engineering of Lovely Professional University with specialization in cloud computing. She has more than 30 publications in reputed journals, conferences and book chapter. She is currently supervising 5 Ph.D. candidates in the area of cloud computing. More than 5 master's thesis has been completed so far under her supervision.

Sukhpal Singh Gill is a Lecturer (Assistant Professor) in Cloud Computing at School of Electronic Engineering and Computer Science, Queen Mary University of London, UK. Prior to this, Dr. Gill has held positions as a Research Associate at the School of Computing and Communications, Lancaster University, UK and also as a Postdoctoral Research Fellow at CLOUDS Laboratory, The University of Melbourne, Australia. Dr. Gill is serving as an Associate Editor in IET Networks Journal. His research interests include Cloud Computing, Fog Computing, Software Engineering, Internet of Things and Healthcare. For further information, please visit http://www.ssgill.me

1 Introduction

Internet of Things (IoT) enriches the digital machines, sensors and objects to observe the given infrastructure and share the data to globe through Internet services. It also consist of integration and analytical services for produced data to take adequate action physically with actuators. Therefore, it provides the platform to design the smart devices with minimum interventions of human (Gill et al. (2019a)). It is predicted that by 2025, IoT devices will cross the 1 trillion devices (Zhu et al. (2020)). Previously, cloud paradigm is considered to be sufficient to deploy the IoT applications and deliver the services to geographically distributed edge devices. However, the distance between cloud datacenters and IoT devices is quite large which hikes the delay in services and data transfer (Chen et al. (2019)). The IoT applications such as smart city, healthcare are the latency sensitive and requires interaction at minimum-latency among cloud datacenters and IoT devices. The large amount of data is generated by IoT devices. This data when transfer to the Internet at same time, produces a network congestion. To solve these fundamental issues of cloud computing for IoT paradigm, the edge and fog computing concepts have emerged (Gill et al. (2019b)). Fog and edge paradigms give priority to the edge devices to run the IoT applications. The potential edge devices are mobile phones, personal computers, Raspberry Pi, micro-datacenters, routers, etc (Tuli et al. (2019)). Some get confused between fog and edge due to the similarity of edge devices, whether few considers the edge computing inherited form fog computing (Tuli et al. (2020a)).

The intermediate layer of cloud and IoT-based system is managed by fog computing. Fog's computing instances are called fog nodes and distributively deployed in the edge network. These fog nodes are providing services similar to cloud such as software-as-a-service (SaaS), platform-as-a-service (PaaS) and infrastructure-as-a-service (IaaS) near to the IoT/edge devices. This helps to improve the Quality of Experience (QoE) for the users by minimizing the network congestion and delay in services (Naeem et al. (2019)).

There are numerous advantages of the fog nodes, besides, they are limited and heterogeneous, due to this it is difficult to process every fine-grained task on these nodes. Therefore, the cloud and fog environment must be work in collaboration with IoT enabled infrastructure to manage the flash requirements of the applications (Mukherjee et al. (2017)). Usually, the integration of these paradigms are top-down approach in cloud-centric system. It is difficult to manage the fog devices in such situation when higher processing power is required for flash crowd (Mouradian et al. (2017)). Here, the central decision of resource management may ineffective. Therefore, the bottom-up approach is more useful. Furthermore, the heterogeneous resources cause the blockage in internal and external services during the deployment or processing of the applications (Ni et al. (2017)). In this situation, a generic approach can remove the obstruction between node-to-node data transfer and execution of the applications. It is challenging task to design the integrated environment due to diverse resource management policies, platform and infrastructure (Mukherjee et al. (2018)). Further, this complication increased with the horizontal/vertical scaling, which could be solved by leveraging serverless functions to provide cost-effective autoscaling (Gill; Aslanpour et al. (2021)). Further, the utilization of latest machine learning or Artificial Intelligence (AI) models can optimize the system performance with an effective resource management.

1.1 Motivation and Our Contributions

There is a need of system model which integrates various paradigms such as cloud, fog, edge and serverless to conduct a comparative study for the identification of best paradigm in different scenarios (Jonas et al. (2019); McGrath and Brenner (2017); Baldini et al. (2017)). In this paper, we designed an integrated system model which integrates cloud, fog, edge and serverless together and used to test the performance of various machine learning models through QoS parameters. The **main contributions** of this article are:

1. To integrate computing paradigms with IoT applications to manage the cloud, fog, edge, serverless resources as per application requirements.
2. To solve the heterogeneity issue in integration by enabling the platform independence for nodes interaction and application processing.
3. To design a system model for performance evaluation of cloud, fog, edge, servelress computing for IoT based healthcare application using latest Machine Learning (ML) models.
4. To deploy the system model on FogBus (Tuli et al. (2019)) & iFaaSBuS (Golec et al. (2021)) and test the performance in terms of QoS parameters such as energy consumption, latency, network bandwidth, response time, scalability to find out the best ML model.
5. To test the best machine learning model in cloud, fog, edge, serverless computing environments for performance comparison.

1.2 Article Structure

The rest of the article is structured as follows: Section 2 presents discusses the background technologies. Section 3 presents the related work. Section 4 presents the an integrated system model. Section 5 discusses the case study on smart healthcare system. Section 6 presents the performance evaluation. Finally, Section 7 concludes the article and highlights various future directions.

2 Background Technologies

This section discuss the background technologies used in this work.

2.1 Internet of Things (IoT)

IoT devices such as actuators, software, sensors and computer components to collect and exchange data between IoT devices and system for further processing (Gill et al. (2019a)). The examples of IoT devices are medical sensors, smart watches, fitness trackers etc.

2.2 Machine Learning

It is a branch of AI to automate the process of data analysis to predict the trends, which would be helpful to make effective decisions without the involvement of humans (Gill et al. (2019a)).

2.3 Cloud Computing

It is a on-demand service, which is available over the Internet for many cloud users to access the compute intensive and data intensive resources using a given user interface (Gill et al. (2019a)). Cloud computing offers three different types of services such as software, platform and infrastructure (Aslanpour et al. (2020)).

2.4 Fog Computing

It is a decentralized model which is designed to offer assistance to cloud by locating between IoT devices and cloud data center to reduce response time and latency for deadline-oriented IoT applications (Singh et al. (2021a)).

2.5 Edge Computing

It is a distributed computing paradigm which improves the response time and latency by moving the data storage and computation service closed to the edge/IoT devices (Aslanpour et al. (2021)).

2.6 Serverless Computing

It is cloud computing execution model that permits cloud users to design and execute their services and IoT applications without taking care of servers (Golec et al. (2021)). In serverless computing, service is provided per function instead of pay per use time model (Aslanpour et al. (2021)).

3 Related Work

In literature (Chen et al. (2019) Mukherjee et al. (2017) Mouradian et al. (2017) Ni et al. (2017) Mukherjee et al. (2018) Bittencourt et al. (2018)), many authors have been designed cloud, fog and IoT-enabled system's integration software framework. Most of the frameworks are designed to support only platform independence and parallel execution of applications (McGrath and Brenner (2017)). This minimizes the developers scope and users tailoring of services as per their requirements (Baldini et al. (2017)). Moreover, these frameworks force the IoT devices for raw data processing and excess storage for cloud instances (Singh et al. (2021a)). The centralized approach of existing frameworks gives poor QoS to the users and open the integration system to certain vulnerabilities (Golec et al. (2021)). In this section, we discuss some similar studies briefly.

Abbasi et al. (2021) proposed workload scheduling architecture using fog–cloud paradigm to optimize the energy consumption for IoT applications. This work used Genetic Algorithm (GA) for handling user requests to improve quality of service. Further, the trade-off between delay and energy consumption has been identified. Mahmud et al. (2018) designed an IoT-based healthcare solution using fog computing paradigm to optimize the network delay. Further, iFogSim simulator is used to evaluate the performance as compared to cloud and results show that this work gives better performance in terms of network delay and energy usage. Peña and Fernández (2019) proposed architecture to improve manage computation nodes dynamically in edge-cloud environments for IoT applications. This work optimizes the performance in terms of latency. Munir et al. (2017) proposed fog centric architecture to for the optimization of IoT-based smart transportation. Further, consumer applications use case is designed to test the performance of proposed architecture in terms of latency and energy.

3.1 Critical Analysis

Table 1 shows the comparison of our work with existing frameworks. In (Abbasi et al. (2021) Mahmud et al. (2018) Munir et al. (2017)), fog and cloud is integrated to improve the QoS for IoT applications in terms of energy. Further, latency is optimized only in (Munir et al. (2017)). In (Peña and Fernández (2019)), edge and cloud is integrated to improve in terms of latency for IoT applications. None of existing studies have compared the performance of IoT application in cloud, fog, edge and serverless computing environment. In our article, we designed a system model which integrates cloud, fog, edge and serverless computing for IoT applications. Further, our work uses machine learning models for the optimization of performance in terms of network bandwidth, latency, scalability, response time and energy consumption.

4 System Model: Performance Evaluation Benchmark

This integrated system model is a combination of various software and hardware components to provide the platform independence and structured communication.

Table 1 Comparison of our work with existing frameworks*

Work	Model				IoT	Performance Parameters					Machine Learning Models				
	Cloud	Fog	Edge	Serverless		NB	L	RT	E	S	EV	LR	RNN	GBT	AutoML
Abbasi et al. (2021)	✓	✓			✓				✓						
Mahmud et al. (2018)	✓	✓			✓				✓						
Peña and Fernández (2019)	✓		✓		✓		✓								
Munir et al. (2017)	✓	✓			✓		✓		✓						
Our work	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

****Abbreviations for Table 1*** - NB: Network Bandwidth, L: Latency, E: Energy, RT: Response Time, S: Scalability (Unsuccessful Response Rate), LR: Linear Regression, RNN: Recurrent Neural Network, GBT: Gradient Boosting Trees and AutoML: Automated Machine Learning

Figure 1 shows the system model, which integrates cloud, fog, edge and serverless computing. The main components of the system model are:

4.1 Cloud Datacenter

IoT back-end applications are executed on cloud when there are insufficient resources with Fog infrastructure to process the application or latency-sensitive applications are in execution. This way the system model explore the computational resources for IoT applications. It integrates with serverless platform. The main components of cloud datacenter are resource scheduler (for scheduling of physical and virtual resources), Virtual Machine (VM) manager (for the management for VMs), computing (to perform the computations) and storage (to store the data for processing).

4.2 Servelress Platform

Servelress platform is an interface between cloud datacenter and fog Infrastructure, which permits cloud users to design and execute their services and IoT applications without taking care of servers. It offers the dynamic scalability and executes the IoT applications in a cost-effective manner. The main components of Servelress platform are storage (to store the data for processing), computing (to perform the computations), monitoring (to observe the execution of the system) and provisioning (to provision the requested resources for the execution of user requests).The other components of Servelress platform are data manager, resource manager, machine leaning model and security manager. Data manager handles the gathered data from various IoT devices for further processing. Resource manager provisions and schedules the resources for the execution of workloads. Machine leaning model is using dataset to train and deploying for the predicting or forecasting of trends as per the requirement of an IoT application. Security manager provides the required level of security using various security protocols.

4.3 Fog Infrastructure

The main components of fog infrastructure are fog gateway nodes and fog computational nodes.

4.3.1 Fog Gateway Nodes (FGNs)

The FGNs are the entry point in the distributed computing environment. In the proposed framework, the IoT devices get assistance from the FGNs regarding placement of jobs and processing of applications. The interface of other application are also provided through FGN such as backed program access, credentials authentication, manage IoT devices, resource demand for application processing and express the service expectation. Moreover, FGN cleans the data and prepare in a common format. The aggregation of data is also performed after collection from the different sources. The integrated environment used to transfer the data to other computing nodes for large scale processing. Simple network Management Protocol (SNMP) or Contained Application Protocol (CoAP) are used to perform these operations for fast communication.

4.3.2 Fog Computational Nodes (FCNs)

The proposed framework is devised to manage the massive amount of FCNs in parallel. FCNs comes up with different resource architecture and storage capacity. FCNs are the consist of processing cores, storage, memory and bandwidth for processing the tasks. The roles of FCNs are:

1. Repository Nodes (RNs): The RNs are used to manage the distributed database to perform replication, data sharing, recovery and storage security. RNs help in historical data analysis and current data access. The meta-data is prepared and managed for the applications such as dependencies, processing requirements and model. Although, these nodes are used for deciding the stopping-point for anomaly driven applications where run-time data is generated.

2. General Computing Nodes (GCNs): FGNs are not directly approachable for every FCN. The Broker Node (BNs) act as intermediater for FGN and FCN. These BNs jobs is resource management and, pass the application for processing along with required data. A GCNs are capable to serve various broker nodes simultaneously with consistence performance. An automatic cluster of GCNs is built under broker node during the processing of distributed applications.

3. Broker Nodes (BNs): IoT applications back-end execution is facilitates through available FCNs with coordination of FGNs. FCNs starts the back-end execution of the application with sufficient

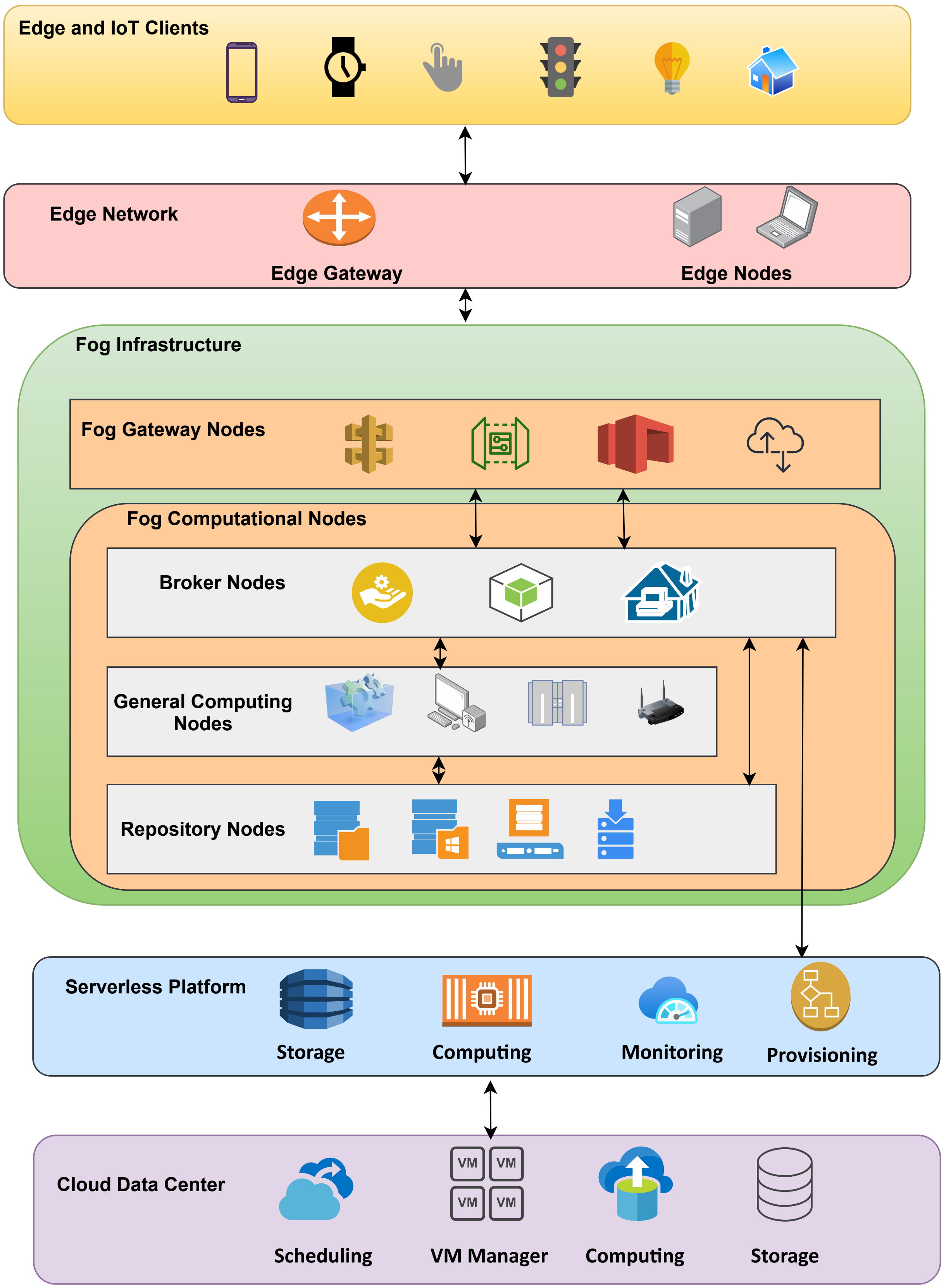

Figure 1 System Model

number of resources. In case, the FCN face a overflow condition itself in term of resources required, it act like a BN to provision the resource on the behalf of FGN to process the back-end application. It communicates with cloud datacenter and other FCNs. Thus, it dispense the task to various FCNs and synchronize, monitor and coordinates its operations. The proposed framework is designed to facilitate these borker noes with deep learning for anomaly detection, Blockchain for security features and replication for fault tolerance. This is a robust framework which provides a secure communication between FCNs, FGNs and cloud datacenters.

4.4 Edge Network

The main components of the edge network are edge gateway and edge nodes. Edge gateway is a mian entry point for network to making a connection with cloud/serverless and the examples of edge gateway are multiplexers, routing switches, routers etc. Edge nodes perform the required computations for execution of user requests near the edge/IoT device.

4.5 Edge and IoT devices

Edge/IoT devices are used to spread the Internet connection from normal devices such as smartphones, laptops, tablets and desktop to non-Internet everyday objects. Using this technology, these objects can interact with Internet and, controlled and monitored from remote locations.

5 Case Study: IoT based Smart Healthcare System

In this work, we have considered smart healthcare system called HealthFog (Tuli et al. (2020a)) as a case study, which uses integrated IoT and fog computing environments to collect the patient's data and diagnose the health status of heart patients automatically (Gill et al. (2018)). The main components of the HealthFog are:

5.1 IoT Devices

HealthFog collects the data from patients using IoT devices of three different types such as environmental sensors, activity sensors and medical sensors. Medical sensors include glucose level sensor, respiration rate sensor, temperature sensor, oxygen level sensor, Electro Myo Graphy (EMG) sensor, Electro Encephalo Gram (EEG) sensor and Electro Cardio Gram (ECG) sensor, which forwarded the collected data to connected gateway devices. We have considered tablets, laptop and mobile phones as Gateway devices to gather data from sensors and transfer to Broker nodes for data processing.

5.2 Machine Learning Module

This module uses dataset to train machine learning models for the classification of data-points which are feature vectors acquired after pre-processing the data acquired from IoT devices. In previous work, we used ensemble voting technique (Tuli et al. (2020a), Atallah and Al-Mousa (2019)) to predict the status of heart patients using Graphical User Interface (GUI). In this work, we used latest machine learning algorithms such as Linear Regression (Yao and Li (2014)), Recurrent Neural Network (RNN) (Cho et al. (2014)), Gradient Boosting Trees (Guelman (2012)) and AutoML (He et al. (2021)) for prediction of health status of heart patients.

5.3 Resource Manager in Cloud/Fog/Edge

Resource manager contains two main sub-components: arbitration module and workload manager (Tuli et al. (2019)). Workload manager processes the incoming job requests and perform data processing while queuing tasks. Arbitration module provisions and schedules the fog/cloud resources for the execution of various workloads based on their QoS requirements. Broker will decide whether the job will be processed at fog or cloud node, it depends on the user requirements.
In this system, credential archive is maintained to preserve the authentication credentials of users. Further, credential archive distributes the security keys and description of every data block created by the broker service to others (Tuli et al. (2019)). For cloud integration, Transport Layer Security (TLS) and Secure Socket Layer (SSL) certificates will be provided for data encryption and decryption.

Authors can read HealthFog (Tuli et al. (2020a)) and FogBus (Tuli et al. (2019)) for more details.

6 Performance Evaluation

This section presents the experimental setup, dataset and results.

6.1 Experimental Setup for Fog-Edge-Cloud

We used real testbed i.e. FogBus (Tuli et al. (2019)) to test the performance of proposed framework by using various machine learning algorithms. FogBus (Tuli et al. (2019)) is a real testbed to validate the IoT application in an integrated Fog-Edge-Cloud environment. It helps to connect various IoT devices to gateways sensors for transfer of data among cloud servers, edge devices and fog nodes. Broker layer performs resource management and start execution of tasks using various fog nodes. Further, HTTP RESTful APIs is used by FogBus to perform various REST operations for collaboration. FogBus uses encryption approaches and authentication blockchain to offer data integrity, privacy and security, which helps to improve the reliability, robustness and

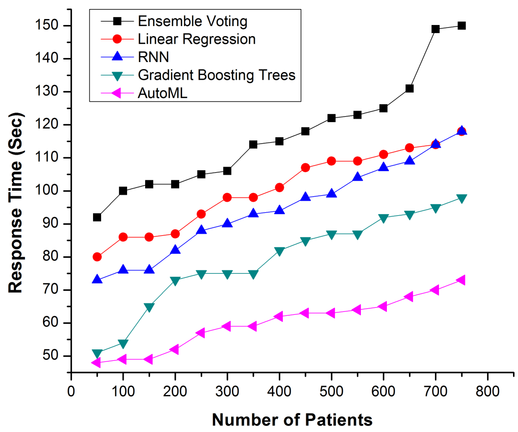

Figure 2 Comparison of Machine Learning algorithms based on Response Time

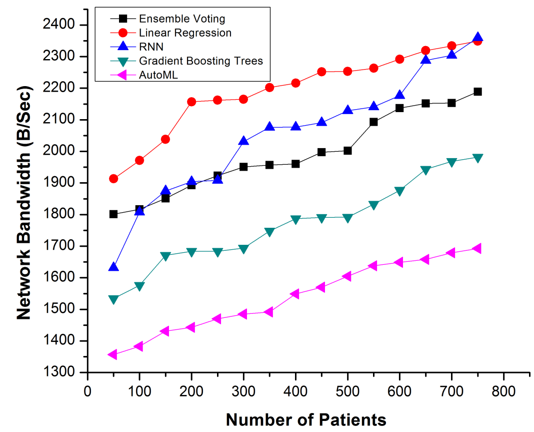

Figure 4 Comparison of Machine Learning algorithms based on Network Bandwidth

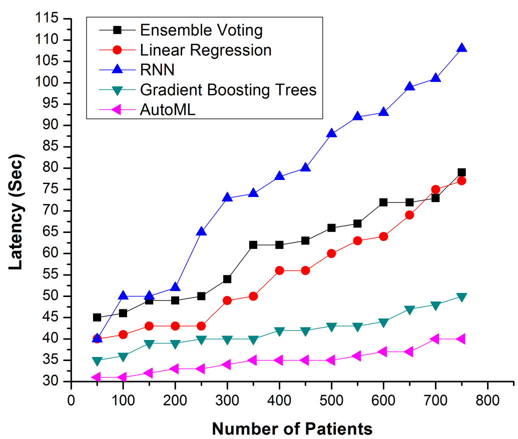

Figure 3 Comparison of Machine Learning algorithms based on Latency

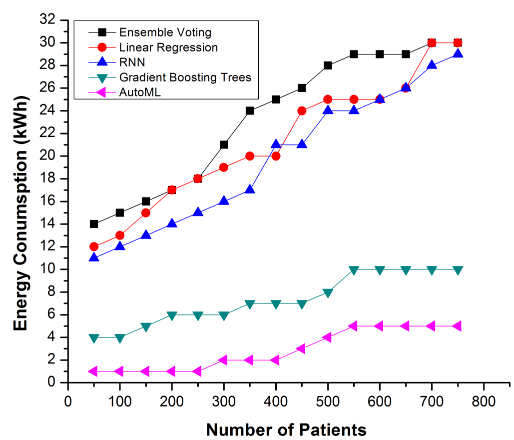

Figure 5 Comparison of Machine Learning algorithms based on Energy Consumption

consistency. Master node is controlling the computing nodes and brokers using LAN. Every broker retains its own Blockchain to ensure data confidentiality and privacy. In this work, FogBus is used for healthcare smart system by implementing different machine learning models on various brokers and handles the shared data on Blockchain platform. Authors can read FogBus (Tuli et al. (2019)) for more details about experimental setup.

6.2 Experimental Setup for Serverless

We used real testbed i.e. iFaaSBus (Golec et al. (2021)) which uses Heroku (her (2021)) for the implementation of serverless computing and Apache JMeter (Apa (2021)) for the measurements of scalability feature with changing number of user requests. iFaaSBus is a real testbed to validate the IoT application in Serverless computing environment. We have done experiments for Serverless computing with two important assumptions: 1) The performance has been measured with reliable Internet connection to evaluate the performance more accurately. 2) Heroku service gives a free service up to a certain number of users and processing power.

6.3 Dataset

To perform the experiments, we used dataset of heart patients to identify the existence of symptoms related to heart disease, which is a binary value 0 (no existence of hear disease) or 1 (existence of hear disease) (Kato et al. (2015), Malik et al. (2018), Dua and Graff (2019)). The Cleveland database (Dua and Graff (2019)) is used to conduct the experiments which was created by Andras Janosi (M.D.) at the Gottsegen Hungarian Institute of Cardiology, Hungary and others. In this work, we are keeping the patient's personal confidential. In this dataset, 14 key attributes (target (num): diagnosis of heart disease, thalassemia, the slope of the peak exercise, depression induced by exercise relative to rest, exercise induced angina, maximum heart rate, resting

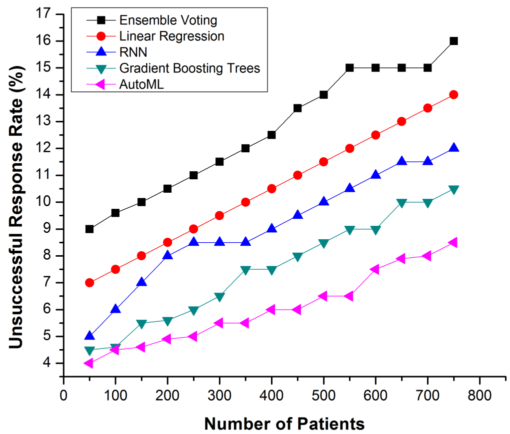

Figure 6 Comparison of Machine Learning algorithms based on Unsuccessful Response Rate

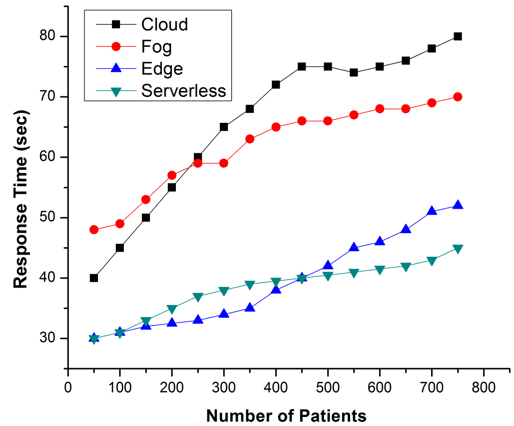

Figure 7 Comparison of Cloud, Fog, Edge and Serverless computing based on Response Time

electrocardiographic results, fasting blood sugar (greater than 120 mg/dl), serum cholesterol (mg/dl), resting blood pressure, chest pain type, sex, age) are given, which we have considered to diagnose the status of patient. Authors can read HealthFog (Tuli et al. (2020a)) for more details about dataset.

6.4 Results and Discussions

We have tested the system model in two different scenarios: 1) Compares the performance of ML models using Fog-Edge-Cloud environment and 2) Evaluates the performance of Fog, Edge, Cloud and Serverless computing.

6.4.1 Performance Comparison of ML models using Fog-Edge-Cloud Environment

We have compared the performance of various machine learning algorithms such as Linear Regression, RNN, Gradient Boosting Trees, AutoML and Ensemble Voting in terms of response time, network bandwidth, latency, energy consumption and unsuccessful response rate. The detailed description of these metrics is given in previous work (Aslanpour et al. (2020)). For these experimental results, we have considered 750 patients.

Figure 2 shows the response time for different machine learning algorithms. In terms of response time, AutoML is performing better than other ML algorithms followed by Gradient Boosting Trees. The average value of response time for AutoML is 4.5%, 5.5%, 6% and 8.5% less than Gradient Boosting Trees, RNN, Linear Regression and Ensemble Voting respectively.

Figure 3 shows the latency for different machine learning algorithms. In terms latency, AutoML and Gradient Boosting Trees are giving almost same results but AutoML outperforms. The average value of latency for AutoML is 2.5%, 5.2%, 5.9% and 9% less than Gradient Boosting Trees, RNN, Linear Regression and Ensemble Voting respectively.

Figure 4 shows the network bandwidth for different machine learning algorithms. In this experiment, AutoML gives better results as compared to other ML models. The average value of network bandwidth for AutoML is 3.5%, 7.5%, 8% and 9.1% less than Gradient Boosting Trees, RNN, Linear Regression and Ensemble Voting respectively.

Figure 5 shows the energy consumption for different machine learning algorithms. AutoML and Gradient Boosting Trees are consuming less energy for processing different the requests of different number of patients. The average value of energy consumption for AutoML is 6%, 9%, 9.8% and 10.5% less than Gradient Boosting Trees, RNN, Linear Regression and Ensemble Voting respectively.

Figure 6 shows the unsuccessful response rate for different machine learning algorithms. AutoML and Gradient Boosting Trees have less value of unsuccessful response rate for processing different the requests of different number of patients. The average value of unsuccessful response rate for AutoML is 4%, 6.5%, 8.9% and 11.2% less than Gradient Boosting Trees, RNN, Linear Regression and Ensemble Voting respectively.

Experimental results are showing that AutoML is performing better than other ML algorithms because AutoML is very efficient in training and predicting. Further, it shows that AutoML is utilizing the cloud, fog, edge resources effectively for managing data coming from various IoT applications.

6.4.2 Performance Comparison of Fog, Edge, Cloud and Serverless Computing

We have conducted an experiment to test the performance of best machine learning model (AutoML) on both serverless and non-serverless computing to identify the impact of number of user requests on the QoS parameters. The experimental setup for Non-Serverless (Fog, Edge, Cloud) is given in Section 6.1 and Serverless is given in Section 6.2. For these experimental

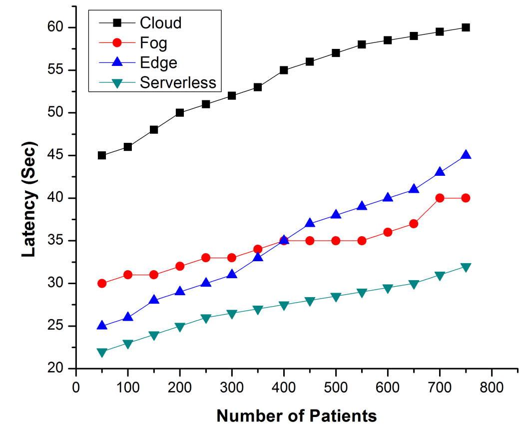

Figure 8 Comparison of Cloud, Fog, Edge and Serverless computing based on Latency

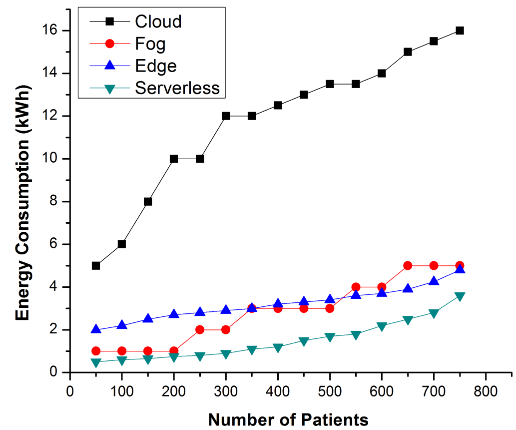

Figure 10 Comparison of Cloud, Fog, Edge and Serverless computing based on Energy Consumption

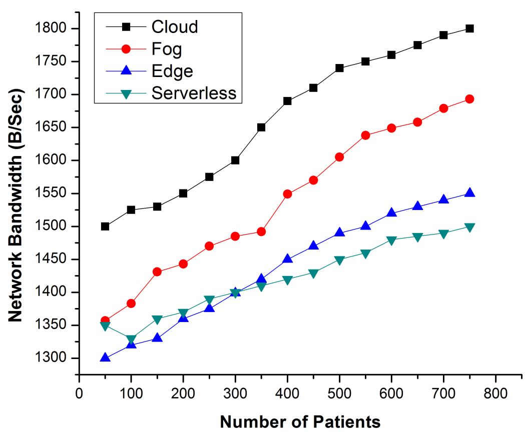

Figure 9 Comparison of Cloud, Fog, Edge and Serverless computing based on Network Bandwidth

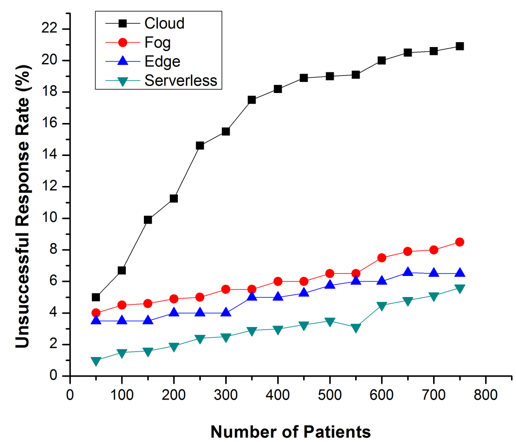

Figure 11 Comparison of Cloud, Fog, Edge and Serverless computing based on Unsuccessful Response Rate

results, we have considered 750 patients. Figure 7 shows the performance comparison of servleress and non-serverless computing based on response time for AutoML machine learning model. Edge gives better results as compared to serverless till 450 user requests but servelress gives better performance for 450+ job requests. The average value of response time in serverless computing is 1.5%, 8%, and 9.5% less than fog, edge and cloud respectively. Figure 8 shows the performance comparison of servleress and non-serverless computing based on latency for AutoML machine learning model. The average value of latency in serverless computing is 3.2%, 4.5%, and 14% less than fog, edge and cloud respectively. Figure 9 shows the performance comparison of servleress and non-serverless computing based on network bandwidth for AutoML machine learning model. The average value of network bandwidth in serverless computing is 4.3%, 6.1%, and 10.5% less than fog, edge and cloud respectively. Figure 10 shows the performance comparison of servleress and non-serverless computing based on energy consumption for AutoML machine learning model. The average value of energy consumption in serverless computing is 3.8%, 3.85%, and 17.45% less than fog, edge and cloud respectively. Figure 11 shows the performance comparison of servleress and non-serverless computing based on unsuccessful response rate for AutoML machine learning model. The average value of unsuccessful response rate in serverless computing is 2.7%, 3.3%, and 15.75% less than fog, edge and cloud respectively. Results clearly show that servleress computing gives better performance as compared to non-serverless computing for 750 user requests.

6.4.3 Analysis of Results

The integrated system model is scalable, easy to deploy and cost-efficient. With proposed system model, the application services providers can save the cost with proper utilization of computing resources. The main

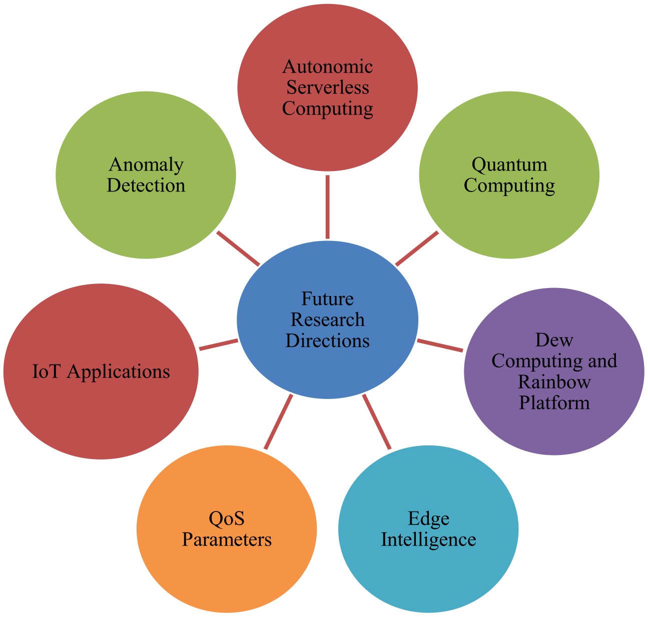

Figure 12 Future Research Directions

reason of better performance in serverless computing is dynamic scalability at runtime, which improves the autoscaling and saves the money.

7 Conclusions and Future Scope

Machine learning plays a vital role for the optimization of various parameters for effective management of computing resources. In this paper, we have designed an integrated system model for Cloud, Fog, Edge and Serverless computing using machine learning models. This system model is scalable and open to test various IoT applications under the existing and custom performance metrics. We implemented the Linear regression, RNN, Ensemble Voting, Gradient Boosting Trees and AutoML for response time, latency, network bandwidth, energy consumption and unsuccessful response rate. The comparative study shows that the AutoML performance is superior as compare to other machine learning algorithms. Finally, the performance comparison of Cloud, Fog, Edge, Serverless is presented which clearly shows the superiority of serverless computing because serverless computing is very effective in providing dynamic scalablilty. The proposed integrated system model can be considered as performance evaluation framework for future IoT applications.

7.1 Future Research Directions

Figure 12 shows the possible future research directions.

1. Autonomic Serverless Computing: It is a technique to provide the services on the usage basis. The existing system charges based on number of servers or bandwidth usage, whereas serverless computing ensure the services solely on basis of functionalities usage through the serverless vendors (Gill, Aslanpour et al. (2021)). The serverless computing have the capacity to shrink the operational cost of cloud applications.

2. Quantum Computing: Our lives are already revolutionized by the quantum physics by giving us great products like transistor and laser. Similar way, the quantum communication and quantum computing has the potential to empower the current systems such as finance, healthcare, security, etc. Recent researches predict the million-billion dollar quantum industry in next 5 to 10 years. The real-world implementation challenges of quantum computing must be examined to make this technology reliable (Gill, Gill et al. (2020)).

3. Dew Computing and Rainbow Platform: The end-devices got the capacity of cloud with dew computing. Rainbow computing ensure the low cost for cross-cloud micro-services in open fog computing platform (Ray (2017), Singh et al. (2021b), Tuli et al. (2021)). In future, we will develop the scalable framework for Dew computing using Rainbow platform. In this paper, we used Blockchain for security and privacy. In future, the benchmark framework could be designed to use various security mechanism as plug and play manner.

4. Edge Intelligence: Proposed framework can utilize the concept of edge intelligence to locate the most effective edge device to improve network performance by reducing latency, which can further improve the resource utilization and save energy consumption using various AI techniques (Zhou et al. (2019)).

5. QoS Parameters: Further, other QoS parameters such as reliability, availability, cost can be incorporated in this framework to improve its performance (Gill et al. (2019a)). Moreover, trade-off among various QoS parameters can help to identify the inter-dependency among various QoS parameters. In this future, cost benefit analysis can be done in terms of various overhead such as training cost, training time and complexity.

6. IoT Applications: We have done experiments using healthcare application related to heart patients, but this framework can be extended for other healthcare domains such as diabetes, cancer, COVID-19 (Tuli et al. (2020b)). Further, this framework Can be used for other IoT applications such as agriculture, smart home and weather forecasting (Gill et al. (2019a)).

7. Anomaly Detection: This framework can be extended for anomaly detection by monitoring any data sources such as servers, networks, devices and

user logs (Benedict (2020)). Further, there is a need to identify the zero-day attacks as well as unknown security threats for more secure and reliable service (Himeur et al. (2021)).

References

S. S. Gill, S. Tuli, M. Xu, I. Singh, K. V. Singh, D. Lindsay, S. Tuli, D. Smirnova, M. Singh, U. Jain *et al.*, "Transformative effects of iot, blockchain and artificial intelligence on cloud computing: Evolution, vision, trends and open challenges," *Internet of Things*, vol. 8, p. 100118, 2019.

G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Toward an intelligent edge: Wireless communication meets machine learning," *IEEE communications magazine*, vol. 58, no. 1, pp. 19–25, 2020.

X. Chen, S. Tang, Z. Lu, J. Wu, Y. Duan, S.-C. Huang, and Q. Tang, "idisc: A new approach to iot-data-intensive service components deployment in edge-cloud-hybrid system," *IEEE Access*, vol. 7, pp. 59 172–59 184, 2019.

S. S. Gill, P. Garraghan, and R. Buyya, "Router: Fog enabled cloud based intelligent resource management approach for smart home iot devices," *Journal of Systems and Software*, vol. 154, pp. 125–138, 2019.

S. Tuli, R. Mahmud, S. Tuli, and R. Buyya, "Fogbus: A blockchain-based lightweight framework for edge and fog computing," *Journal of Systems and Software*, vol. 154, pp. 22–36, 2019.

S. Tuli, N. Basumatary, S. S. Gill, M. Kahani, R. C. Arya, G. S. Wander, and R. Buyya, "Healthfog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated iot and fog computing environments," *Future Generation Computer Systems*, vol. 104, pp. 187–200, 2020.

R. Z. Naeem, S. Bashir, M. F. Amjad, H. Abbas, and H. Afzal, "Fog computing in internet of things: Practical applications and future directions," *Peer-to-Peer Networking and Applications*, vol. 12, no. 5, pp. 1236–1262, 2019.

M. Mukherjee, R. Matam, L. Shu, L. Maglaras, M. A. Ferrag, N. Choudhury, and V. Kumar, "Security and privacy in fog computing: Challenges," *IEEE Access*, vol. 5, pp. 19 293–19 304, 2017.

C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE communications surveys & tutorials*, vol. 20, no. 1, pp. 416–464, 2017.

J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing fog computing for internet of things applications: Challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 601–628, 2017.

M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018.

S. S. Gill, "Quantum and blockchain based serverless edge computing: A vision, model, new trends and future directions," *Internet Technology Letters*.

M. S. Aslanpour, A. N. Toosi, C. Cicconetti, B. Javadi, P. Sbarski, D. Taibi, M. Assuncao, S. S. Gill, R. Gaire, and S. Dustdar, "Serverless edge computing: vision and challenges," in *2021 Australasian Computer Science Week Multiconference*, 2021, pp. 1–10.

E. Jonas, J. Schleier-Smith, V. Sreekanti, C.-C. Tsai, A. Khandelwal, Q. Pu, V. Shankar, J. Carreira, K. Krauth, N. Yadwadkar *et al.*, "Cloud programming simplified: A berkeley view on serverless computing," *arXiv preprint arXiv:1902.03383*, 2019.

G. McGrath and P. R. Brenner, "Serverless computing: Design, implementation, and performance," in *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)*. IEEE, 2017, pp. 405–410.

I. Baldini, P. Castro, K. Chang, P. Cheng, S. Fink, V. Ishakian, N. Mitchell, V. Muthusamy, R. Rabbah, A. Slominski *et al.*, "Serverless computing: Current trends and open problems," in *Research advances in cloud computing*. Springer, 2017, pp. 1–20.

M. Golec, R. Ozturac, Z. Pooranian, S. S. Gill, and R. Buyya, "ifaasbus: A security and privacy based lightweight framework for serverless computing using iot and machine learning," *IEEE Transactions on Industrial Informatics*, 2021.

M. S. Aslanpour, S. S. Gill, and A. N. Toosi, "Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research," *Internet of Things*, p. 100273, 2020.

J. Singh, P. Singh, and S. S. Gill, "Fog computing: A taxonomy, systematic review, current trends and research challenges," *Journal of Parallel and Distributed Computing*, 2021.

L. Bittencourt, R. Immich, R. Sakellariou, N. Fonseca, E. Madeira, M. Curado, L. Villas, L. DaSilva, C. Lee, and O. Rana, "The internet of things, fog and cloud continuum: Integration and challenges," *Internet of Things*, vol. 3, pp. 134–155, 2018.

M. Abbasi, E. Mohammadi-Pasand, and M. R. Khosravi, "Intelligent workload allocation in iot–fog–cloud architecture towards mobile edge computing," *Computer Communications*, vol. 169, pp. 71–80, 2021.

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

M. A. L. Peña and I. M. Fernández, "Sat-iot: An architectural model for a high-performance fog/edge/cloud iot platform," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. IEEE, 2019, pp. 633–638.

A. Munir, P. Kansakar, and S. U. Khan, "Ifciot: Integrated fog cloud iot: A novel architectural paradigm for the future internet of things." *IEEE Consumer Electronics Magazine*, vol. 6, no. 3, pp. 74–82, 2017.

S. S. Gill, R. C. Arya, G. S. Wander, and R. Buyya, "Fog-based smart healthcare as a big data and cloud service for heart patients using iot," in *International Conference on Intelligent Data Communication Technologies and Internet of Things*. Springer, 2018, pp. 1376–1383.

R. Atallah and A. Al-Mousa, "Heart disease detection using machine learning majority voting ensemble method," in *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*. IEEE, 2019, pp. 1–6.

W. Yao and L. Li, "A new regression model: modal linear regression," *Scandinavian Journal of Statistics*, vol. 41, no. 3, pp. 656–671, 2014.

K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

L. Guelman, "Gradient boosting trees for auto insurance loss cost modeling and prediction," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3659–3667, 2012.

X. He, K. Zhao, and X. Chu, "Automl: A survey of the state-of-the-art," *Knowledge-Based Systems*, vol. 212, p. 106622, 2021.

"Heroku," 2021, [Online; accessed 09-Aug-2021]. [Online]. Available: https://www.heroku.com/

"Apache jmeter," 2021, [Online; accessed 09-Aug-2021]. [Online]. Available: https://jmeter.apache.org//

N. Kato, M. Loh, F. Takeuchi, N. Verweij, X. Wang, W. Zhang, T. N. Kelly, D. Saleheen, B. Lehne, I. M. Leach *et al.*, "Trans-ancestry genome-wide association study identifies 12 genetic loci influencing blood pressure and implicates a role for dna methylation," *Nature genetics*, vol. 47, no. 11, pp. 1282–1293, 2015.

R. Malik, G. Chauhan, M. Traylor, M. Sargurupremraj, Y. Okada, A. Mishra, L. Rutten-Jacobs, A.-K. Giese, S. W. Van Der Laan, S. Gretarsdottir *et al.*, "Multiancestry genome-wide association study of 520,000 subjects identifies 32 loci associated with stroke and stroke subtypes," *Nature genetics*, vol. 50, no. 4, pp. 524–537, 2018.

D. Dua and C. Graff, "Uci machine learning repository," 2019. [Online]. Available: http://archive.ics.uci.edu/ml

S. S. Gill, A. Kumar, H. Singh, M. Singh, K. Kaur, M. Usman, and R. Buyya, "Quantum computing: A taxonomy, systematic review and future directions," *arXiv preprint arXiv:2010.15559*, 2020.

P. P. Ray, "An introduction to dew computing: definition, concept and implications," *IEEE Access*, vol. 6, pp. 723–737, 2017.

M. Singh, S. Tuli, R. J. Butcher, R. Kaur, and S. S. Gill, "Dynamic shift from cloud computing to industry 4.0: Eco-friendly choice or climate change threat," pp. 275–293, 2021.

S. Tuli, R. Bansal, R. Paul *et al.*, "Tango: Commonsense generalization in predicting tool interactions for mobile manipulators," *International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

S. Tuli, S. Tuli, R. Tuli, and S. S. Gill, "Predicting the growth and trend of covid-19 pandemic using machine learning and cloud computing," *Internet of Things*, vol. 11, p. 100222, 2020.

S. Benedict, "Serverless blockchain-enabled architecture for iot societal applications," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 5, pp. 1146–1158, 2020.

Y. Himeur, K. Ghanem, A. Alsalemi, F. Bensaali, and A. Amira, "Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives," *Applied Energy*, vol. 287, p. 116601, 2021.