Dynamic Replica Management in Fog-enabled IoT using Enhanced Data Mining Technique

MSc Research Project
Cloud Computing

Rahul Sudhakar Dhande
Student ID:18182852

School of Computing
National College of Ireland

Supervisor: Prof. Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Rahul Dhande
Student ID:	18182852
Programme:	Cloud Computing
Year:	2020
Module:	MSc Research Project
Supervisor:	Prof. Vikas Sahni
Submission Due Date:	23/08/2020
Project Title:	Dynamic Replica Management in Fog- enabled IoT using Enhanced Data Mining Technique
Word Count:	4837
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	15th August 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Dynamic Replica Management in Fog- enabled IoT using Enhanced Data Mining Technique

Rahul Dhande
18182852

Abstract

With the fast growth of IoT devices and Sensors, producing a massive amount of data. It causes various problems such as storage overhead, high latency, and network congestions. Fog computing includes fog nodes which are heterogeneous and provide solutions on latency-sensitive data services to place data on fog nodes closer to data generators. Unfortunately, existing cloud technology cannot meet minimum latency and data management for IoT devices and end-users. Simultaneously, related work focuses only on data placement on a fog node. In this research paper, a novel solution is presented for dynamic replica management in Fog enabled IoT by considering data mining called Enhanced data mining dynamic replication (EDMDR). Using maximal frequent pattern mining not only improves replication but help to diminish data management cost. In the end, the results of simulation reduced total latency issues by 62%,83% and 79% as compared to FC analytical model,iFogStorM and DMDR. Simultaneously, the response time of EDMDR is less than existing methods, which proves the better optimization for Fog infrastructure.

Keywords: Data Mining,Fog Computing,Dynamic Replication,Latency,iFogSim

1 Introduction

A recent development in the Internet of Things referred to as Internet of Everything (IoE) introduces the next generation of networks and real-time data management over the world, as well as it is set to reach $1.7 trillion by 2025.Alarifi et al. (2019) and Sharma and Saini (2019) The cost of the IoT sensors that are required to provide autonomous driving capabilities is upward of $75,000 per vehicle, which is more than double the price of the typical model car by 2026.[1] Moreover, US factories, houses, cars, wearable devices and healthcare market will connect trillions of sensors by 2023, which results in a massive amount of data and exhaust network bandwidth. Traditionally, cloud technology provides solutions and services like dynamic replication for the data with various limitations such as data access latency issues and failure of data centres. Besides, there is a need for new computing model over cloud close to IoT sensors referred to as fog Computing helps to overcome and handle above crises.Shukla et al. (2019)

Fog computing by Cisco links to cloud data centres specifically bring cloud features at the edge of the network and beneficial for IoT Systems over cloud computing.Dhande (2020) It boosts the performance and efficiency of cloud storage systems by minimizing

[1] `https://www.gartner.com/en/newsroom/press-releases/2019-11-14-gartner-forecasts-more-than-740000-autonomous-ready-vehicles-to-be-added-to-global-market-in-2023`

data transfer rate and latency issues in IoT systems. Reliability in fog will sometimes be increased by periodic checkpoints to resume once a failure, rescheduling failing tasks or replication to exploit execution in parallel. Yi et al. (2015)

The massive data from IoT systems have different storage capacity, which placed on fog nodes requires new techniques such as data mining, analytics to proper data placement improve the reliability of fog enabled IoT devices. Kumari et al. (2019)Fog Data Analytics(FDA) helps to overcome challenges like quality of service, heterogeneity and latency. There are various replica placement solutions in section 2 available in the market in cloud, edge and fog environment. However, the maximal frequent pattern mining replication strategy provide a proper replica placement and storage, network usage solution with the low response time. Mansouri et al. (2019)

1.1 Background and Motivations

Dynamic replica management aims to focus on the optimizing replicas dynamically as per user data accessing needs as well as measures replication factor based on the popularity of data.John and Mirnalinee (2019) Fog network topologies change over time due to IoT service migrations that's why it referred to as Dynamic.Naas et al. (2018) Dynamic replication techniques help to boost data reliability, availability and access efficiency of the fog node in Fog, Edge and Cloud environment.Chunlin et al. (2019) and Li, Wang, Tang, Zhang, Xin and Luo (2019)

However, data mining techniques play a significant role in proper management and placement of replica as well as summarise different data that beneficial for Fog.Mansouri et al. (2019) The Hamrouni et al. (2016) is shown a summary data mining based dynamic replication strategies in the data grid and cloud environment.

1.2 Research Project Specification

It includes a detailed information of goals and contributions of this research.

1.2.1 Research Question

This research paper deal with the following research question by Dhande (2020):
RQ:*"Can reduce response time and latency issues using data mining dynamic replica allocation strategy in cloud data storage of fog enabled IoT?"*

1.2.2 Research Objectives and Contributions

- Under E-DMDR architecture, design and develop maximal frequent correlated pattern mining algorithm and apply data mining based dynamic replication strategy in Fog enabled IoT environment.
- Examine and test the strategy to reduce latency issues and proper management of IoT sensor data in a fog storage.
- Analyze research work and compare with actual work and successful in the test.

The contributions includes

- Critical analysis and review of the gathered results of the previous related work.

- Evaluation on iFogSim [2] by considering historical real-time sensor data
- Comparing results with previous work as well as develop and implement project work in the configuration manual.

1.3 Conclusion

The results of this research paper are provide a reliable solution in proper data management in fog-IoT infrastructure benefits to efficient access data as per user need.
The reminder of the paper organised as follows: **Section 2** described previous work and summary of comparison based on evaluation factors. After that,**Section 3** shown tools and proper method Additionally, sketch of method by **Section 4** and **Section 5** provided detailed view of Implementation **Section 6** various tests cases with analyse and Compare the results. Finally, **Section 7** Conclusion and future work of research paper.

2 Critical analysis of related work

Proper Data replica placement aims to improve fault tolerance,reliability,availability and efficiency of data storage in fog-IoT environment. Various Replica placement strategies and algorithms help to reduce latency issues of data and Cost in Fog,edge and cloud Environments.Dhande (2020)

2.1 Replica Management in Fog-IoT Environment

Huang et al. (2019) presented a latency aware model called iFogStorM helps the placement of multiple data replicas in fog environment. To minimise the overall latency of data storage, the author proposed a MultiCopyStorage, a greedy algorithm based data replica placement technique. As compared to,Naas et al. (2018) showed a heuristic approach called divide and conquer (iFogStorG) for data placement in fog infrastructure as well as they used linear programming to minimise solving time. In that, the author used graph modelling and partitioning methods to split the original data placement problem into many balanced subproblems. Shukla et al. (2019) introduced a novel analytical approach aim to diminish latency from healthcare IoT sensors and cloud servers as well as proposed a fuzzy-based reinforcement algorithm in a fog. ECG sensor dataset used for performance evaluation analysis between latency, network usage, and RAM consumption. Sharma and Saini (2019) introduced a four-tier architecture and proposed the Earliest Deadline First algorithm for energy consumption and service delay fog environment. They aim to focus on reducing the high latency in task scheduling.

Alarifi et al. (2019) describes a novel three-layer architecture for allocating requests for services to the most appropriate devices in IoT-based fog cloud environments called fault-tolerant scheduling as well as a proposed check-pointing algorithm for Normal variance of the running time between failures in the fog-IoT context. Maiti et al. (2018) introduced a four-layer Qos aware fog node architecture for IoT devices aims to focus on reducing the overall latency arising from traffic aggregation and transformation by suitable mapping between gateways and fog nodes. Maniglia et al. (2019) Describes an analysis of three different algorithms for data placement in Fog computing environment called Edge-ward, Mapping and Cloud only.

[2]`https://github.com/Cloudslab/iFogSimTutorials`

Taneja and Davy (2017) proposed a Module Mapping Algorithm for effective use of network infrastructure resources by efficiently deploying application modules for IoT related applications in the Fog-Cloud infrastructure, and it helps to reduce applications latency as well as It returns an application's efficient mapping of its modules to network infrastructure. Also, Mahmud, Ramamohanarao and Buyya (2018) Focused on Heuristic resource optimization by latency-aware Software module transmission works within clusters as well as Considers different aspects of latency in a body of decentralized management of distributed applications. The proposed strategy shows improvement in resource optimization, module placement and forwarding time. Mahmud, Srirama, Ramamohanarao and Buyya (2018) provided a Quality-aware application placement policy for fog environments, and IoT devices include separate Fuzzy logic-based approaches that prioritize specific application placement requests and identify Fog computational instances based on user preferences and current instance status.

The Vales et al. (2019) introduced a novel hybrid based fog storage system for bringing data in a system with relevant functions includes User data usage, energy use and the distance between nodes for mobile devices and fog nodes. Finally, the results reduce the file transfer time that end users undergo. In Verma et al. (2016) Present an efficient algorithm for reducing data dependency in big data centres called Edge load balancing for fog computing platform and IoT using a data replication strategy. As compared to,Ozeer et al. (2018) enlight on first-fit heuristic and genetic algorithm to solve fog service placement issues in fog environment as well as a fault-tolerance approach based on reliability issue. Successful simulation improves the usage of fog assets and response time of IoT applications.

2.2 Solutions of replica placement in edge environment

Fang et al. (2019) presented a unique model to reduce task response time between edge servers and the central cloud server. Moreover, the proposed algorithm focused on deciding to send tasks to other edge servers or cloud if fully loaded to make the most use of edge servers in the same area. Aral and Ovatman (2018) introduces a decentralised replica placement algorithm in edge environment aims to focus on dynamic replica creation/removal guided by continuous monitoring of data requests from underlying network edge nodes. In Shao et al. (2018) formulate a novel scheme for 0-1 integer programming problem of data-intensive IoT workflows in collaborative edge and cloud environments with Intelligent swarm optimization (ITO) algorithm consist of coding and fitness functions for data replica scheduling and distributions solution.

The Li, Wang, Tang and Luo (2019) proposed the multi-objective replica placement algorithm as well as consider the non-dominated genetic sorting algorithm includes replica placement rules and optimal replica placement strategy. Simulation results show improvement in average response time, utilization of network and a load balancing guarantee for data nodes. The Li, Wang, Tang, Zhang, Xin and Luo (2019) Provided a dynamic replica creation algorithm for Takes data blocks as a replication granularity, examines the real-time server environment, and dynamically changes the number of replicas to user needs as well as a dynamic replica creation model to improve prediction accuracy. Additionally,Chunlin et al. (2019) Proposed a replica consistency maintenance strategy based on the fast Paxos algorithm to solve the problem of data inconsistency induced by multi-user competitiveness solve the problem of data inconsistency induced by multi-user competitiveness.

2.3 Traditional replica placement in cloud

Mansouri et al. (2019) presented dynamic replica placement strategy using data mining (DMDR) and Maximum frequent pattern mining algorithm which finds Frequently associated patterns of size two by a regularly associated pattern of size one in a cloud environment. Pattern mining simulation helps to reduce average response time and latency. Hamrouni et al. (2016) summarise the dynamic replication and replica placement algorithms based on data mining in a data grid environment. The survey evidently focuses on data mining or knowledge discovery enables or plays a significant role to enhance dynamic replication in data grids environment like a k-nearest neighbor, Association rules and Regression as well as works with historical data.

John and Mirnalinee (2019) proposed a swarm intelligence based optimisation method focused on user request to a selection of optimizing replica based on bandwidth and removal of the unwanted replica to manage storage in the cloud. Also, Chang and Wang (2019) introduces modified algorithms named as leader selection, and replica placement aims to focus on reducing latency among replicas in various datacenters called write aware replica placement. The final benchmark results show improved performance in write operations with excellent reliability and low latency.

2.4 Data Mining in Fog-IoT environment

Data mining plays a significant role in Fog and IoT environments. Kumari et al. (2019) describes a detailed taxonomy of Data Analytics techniques over cloud computing to deal with a massive amount of big data from IoT sensors and their latency called Fog-Data analytics. Simultaneously, Ma et al. (2018) introduced Swarm search-driven feature selection is used as a preprocessing tool to improve the accuracy and speed of local Fog data analytics. In data mining, Fog computing is a virtualized infrastructure that provides processing, storage, and network connectivity between the data centre and end-users for cloud computing. For example, it brings many advantages to promote low latency, high performance, position awareness, support for mobility, broad geographical distribution and wireless connectivity.

In the Fog world, data analytics must be real-time with low energy consumption. The data mining model is relatively small in size so that the data mining system can fit into limited memory so that data analytics can be implemented directly into endpoints such as network nodes, data sinks, controllers or even sensors themselves. Data mining algorithms in Fog computing based on the decision tree. Fong (2017) Also, Braun et al. (2019) introduced two different approaches in fog based mining based on networking services In that, Local data is transmitted from IoT devices to their local networking infrastructure between IoT devices and the normal data centres to mine patterns from IoT devices with fog computing.

The following table describes detail comparative analysis of previous work. It provides the necessary information required for replica placement problems and shows how such schemes are attempting to improve placement cycle using various methods of forecasting. Moreover, it categorizes the schemes analyzed concerning their methods of prediction and describes their key goals, advantages and drawbacks.Also, it helps to analyze the effectiveness of the outlined replica placement strategies in cloud, edge and Fog Environment.

Authors	**Algorithm**	**Evaluation Factors**	**Simulator**	**Limitations**
Huang et al. (2019)	MultiCopyStorage	Latency and solving time	iFogSim	less active to provide reliable solution
Naas et al. (2018)	Edge Weighting	Latency and Required time	iFogSim	Unable to focus on storage and network usage
Shukla et al. (2019)	fuzzy-based reinforcement	Network and RAM consumption	iFogSim and Spyder Python	Not to best fit for replica placement
Verma et al. (2016)	load balancing algorithm	Responce time and Total cost consumption	CloudSim	It doesn't perform well in replication and analysis not shown properly
Aral and Ovatman (2018)	D-ReP	latency,Cost Ratio and Network Overhead	CloudSim	Simulation takes more time to execute
Shao et al. (2018)	Intelligent swarm optimization	Total data access cost,transmission and movements	Steady-State Method	Ineffective methods and random results
Li, Wang, Tang, Zhang, Xin and Luo (2019)	Dynamic replica creation	Responce time and storage and network Utilization	Apache Jmeter	Total access frequency of data is low
Chunlin et al. (2019)	Fast Paxos algorithm	Average consistency maintenance time and throughput	HDFS strategy algorithm	Difficult to calculate obtained results
John and Mirnalinee (2019)	Intelligent water-drop algorithm	Storage space and traffc	CloudSim	Neglecting the effect of file access pattern
Mansouri et al. (2019)	Maximal frequent correlated pattern mining	Responce time,Access frequency,Storage and Network usage	CloudSim	Proposed method less effective for replica placement
Chang and Wang (2019)	leader selection, and replica placement	Responce delay and bandwidth Consumption	CloudSim and YCSB workload	Data availability takes more time
Dhande (2020)	Maximal frequent correlated pattern mining(FPMax)	Latency,Storage and Network usage	iFogSim	Execution with limited real time traces

Table 1: Comparative analysis of replication strategies in fog,edge and cloud

2.5 Conclusion of Literature

According to all previous literary works and solutions, it is evident that,many authors aim to focus on proper replica placement in data storage of Fog, IoT Edge and cloud environment to diminish latency, response time. However, there is a requirement of a reliable technique in fog. Besides,there is an open challenge called reliability in fog and IoT environment.(Ozeer et al. (2018),Mansouri et al. (2019)) This research not only focuses on reducing latency issues but provide a reliable solution for efficient storage management in fog enabled IoT Environment.

3 Methodology

3.1 Enhanced Data Mining Dynamic Replication in Fog-IoT

The dynamic replication method of this research paper primary focus on imitate work introduced by the Mansouri et al. (2019) in Fog enabled IoT Environment. The technique primarily works based on centrality factor of last access data. However, to boost the replication frequency and reduce data access latency issues needs to take advantage of maximal frequent pattern mining replication. The following dynamic replication using data mining method in the figure 1 depicts working scenario of maximal frequent item correlated pattern mining technique in fog-enabled IoT.

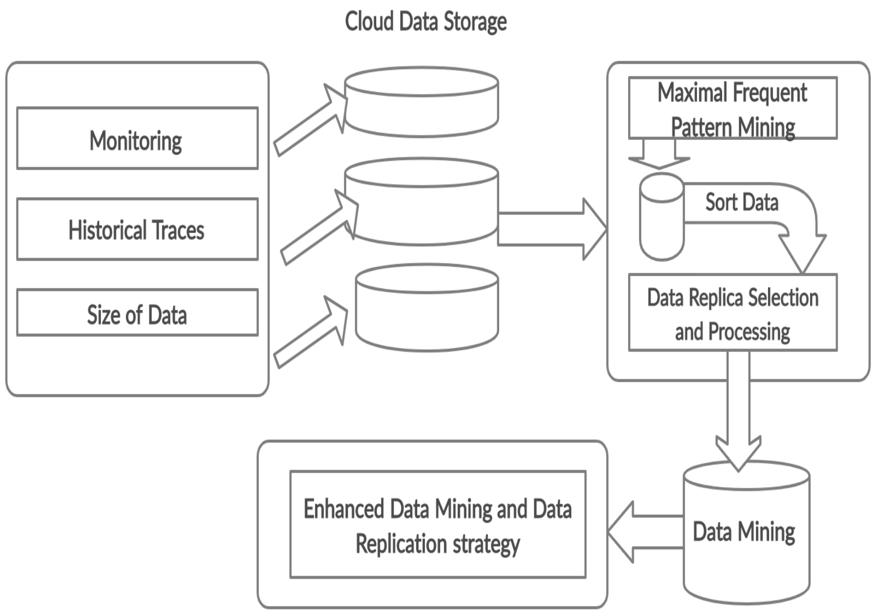


Figure 1: Dynamic replication using data mining technique by Dhande (2020)

3.2 Maximal Frequent Correlated Pattern Mining Strategy

This technique based on data mining aims to focus on the replication of data. The primary function of this technique is to convert history file access into binary context based on correlation measure.Mansouri et al. (2019)

$$Centrality(R) = \frac{G-1}{\sum_{a \neq b} v(a,b)} \quad (1)$$

Where G for total No of Fog Nodes and v(d,b) indicates the distance between fog node a and fog node b.Centrality is the central value of the fog node N.

Moreover, this algorithm takes three inputs such as historical dataset(real-time traces), output and minsupport value. The steps include firstly, Analyse access history of data and assigns the job. Secondly, Convert data into logical value referred to as a binary context and make a group. Finally, investigate the best fog node for replication and apply dynamic Replication.The benefits of this technique boost effective network usage, response time and replication frequency.Hamrouni et al. (2016)

3.3 Experimental Tools

3.3.1 iFogSim

iFogSim is an open-source Java-based tool used to simulate the algorithm by creating the virtual environment of dynamic replication for Fog-enabled IoT. This assessment framework works explicitly for Fog-IoT environments. Also, It measures the impact on latency, network congestion, energy use and cost of resource management. Besides, It supports some cloud characteristics, but it is near to the IoT sensors network.Gupta et al. (2016) The figure (**2**) depicts a detailed architecture for iFogSim 3.0 tool. It includes multiple layers of functionally responsible for different tasks.

The major components of the architecture are IoT data streams, Monitoring layer, Resource Management and application models.

1. IoT Data Streams : Known as a tuple of iFogSim generated from fog devices transmitted from application module to actuators.
2. Monitoring Services : Gives a real-time record of resource usage, power consumption and system availability (i.e., sensors, actuators and Fog devices)
3. Resource Management: It minimizes resource wastage and gives a proper plan for Quality of services.
4. Application models: It based on distributed data flow (DDF) and responsible for different applications deployments in Fog.

3.3.2 Eclipse IDE

For the evaluation of this research paper algorithm, configured iFogSim on Eclipse IDE. [3] Simultaneously, used JDK version 8 because iFogSim has coded in java. [4] Also, used Java Language and APIs for both Mining Algorithm and Dynamic Replication algorithm. The maximal frequent correlated itemset algorithm is based on SPMF Open-Source Data mining Libraries to carry out various data mining activities .[5] The SPMF Jars added as External Jars into ifogsim to generate frequent itemsets from historical IoT healthcare sensors traces for dynamic replication.

[3] `https://www.eclipse.org/downloads/`
[4] `https://www.oracle.com/in/java/technologies/javase/javase-jdk8-downloads.html`
[5] `http://www.philippe-fournier-viger.com/spmf/FPMax.php`

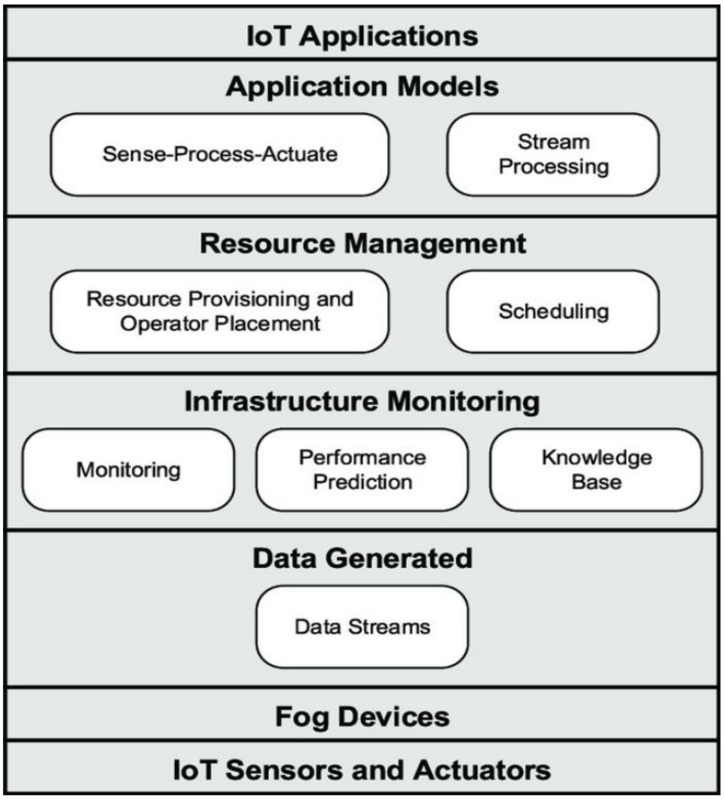


Figure 2: Architecture of iFogSim by Perez Abreu et al. (2019)

3.4 Dataset Collection

For this research paper simulation, used synthesized realtime dataset includes historical traces provided as an input. Also, to know which node is best for replication, used cleveland healthcare information collect from UCI Machine Learning Repository.[6] It includes data from a patient with heart disease due to high blood pressure and high cholesterol with seven columns. It considered for replica frequency threshold to make replica decisions.

3.5 Conclusion

To summarise, this section gives a detailed structure of the tools and methods used in this research paper of dynamic replication.

4 Design Specification

4.1 Workflow of E-DMDR

As shown in figure **(3)**, iFogSim use the step by step implementation process of E-DMDR technique in fog-enabled IoT.It helps to understand the proper plan of this experiment. It includes replica placement procedure, and functions start by taking different inputs such as historical trace, support value, Fog network and replica frequency threshold value. Also, it explains the effectiveness of the dynamic replication algorithm.

[6]`https://archive.ics.uci.edu/ml/datasets/heart+Disease`

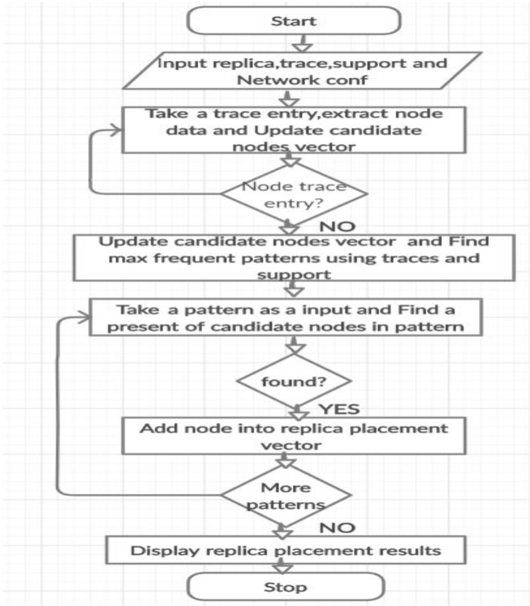


Figure 3: E-DMDR WorkFlow by Dhande (2020)

4.2 Architecture

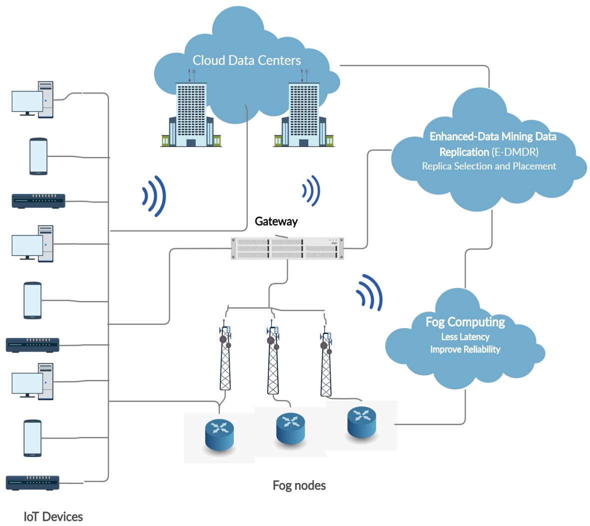


Figure 4: Functional architecture of E-DMDR by Dhande (2020)

The research design includes an functional architecture of E-DMDR in Fog-IoT Environment. It contains four main stages the first phase referred to as IoT Sensors contains enormous data and a second phase called Fog Nodes for saving data as well as third phase indicates the data mining replication strategy to reduce latency. Lastly, the cloud Data centre phase for proper data management using E-DMDR.

4.3 Network Topology

The physical network topology indicated individually fog gateways, devices and IoT sensors and actuators connected as per different latencies in the table **3** in iFogSim graph-

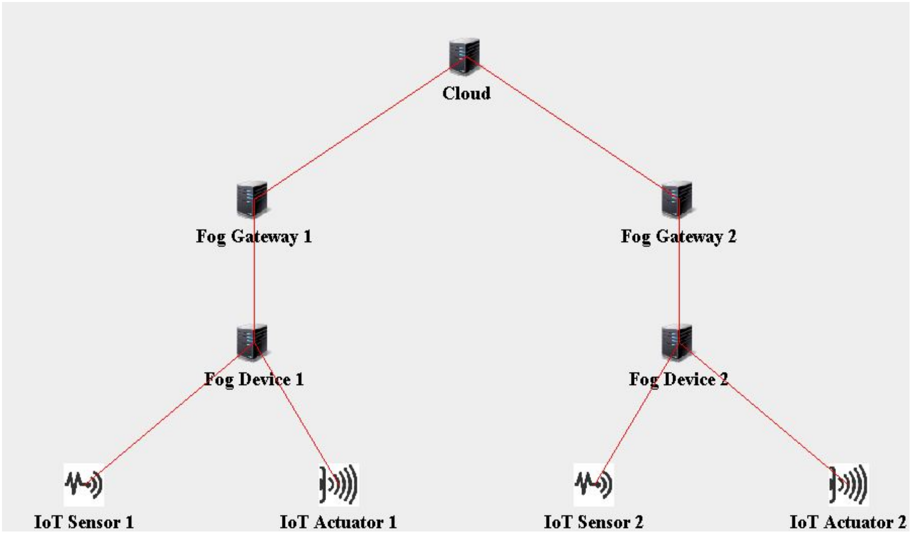


Figure 5: Network topology arrangements in iFogSim GUI

ical user interface.The GUI developed by Gupta et al. (2016). The GUI simulation for evaluation of storage and network overhead, latency and response time with real-time traces from IoT processed on fog nodes and servers. Besides,iFogSim allows multiple program execution and supports program module migration. Also,used SPMF libraries in iFogSim to execute the simulation. The dynamic replication algorithm programmed into the available libraries of the iFogSim simulator.

5 Implementation

5.1 E-DMDR Algorithm

This section presents a pseudocode of the replication algorithm and E-DMDR functionality. As per the functional architecture as shown in figure **4** and Flow of Implementation in the figure (**3**),it is evident that frequent patterns supports replica management in fog environment. The replica placement pseudocode follows significant steps.

- Firstly, the historical file access data (traces) coming from IoT sensors on Fog Node N, Support value and replica f (frequency threshold) provided as an Input.
- Secondly, the Maximal frequent pattern Mining(MFCP) find and filter out the candidate nodes from a trace file using runMFCP() function.
- Thirdly, validateNodes() function checks if any different type nodes present or not in trace file.
- Finally, removeDuplicates() function removes repeated nodes from the trace file. Afterwards, proper replica placed based on decisions by using a frequency threshold.

The E-DMDR applies function runDMDRPAlgorithm() is to reduce overall latency and response time by finding and requesting subsequent file in fog node. Additionally, the runDMDRPAlgorithm() reduces overall latency by taking the fog network N selects historical trace T from real-time traces and d is the total size of the pattern. Thus, the **Time Complexity** of function runDMDRPAlgorithm() is d $\cdot T^2 \cdot N^2$.

Algorithm 1 E-DMDR Replication by Dhande (2020)

```
Input: Network N, replica f, historical trace T, support sup
Output: Proper placement of replica                                  ▷ final result
Initialize: Candidate nodes vector N',frequent patterns vector P and replica place-
   ment nodes vector R
for each trace t in T                                    ▷ finding the candidate nodes
Extract node
Add node to N'
end for
N'=removeDuplicates(N')
N'=validateNodes(N)
P=runMFCP(T, sup)
for each node n' in N'                                        ▷ Filter candidate nodes
for each pattern p in P
if p has n' Then
Add n' to R
end if
end for
end for
for each replica placement node r in R                           ▷ Replica Placement
Place replica file f in node r
end for
```

5.2 Experimental Setup

In this work, to verify the dynamic replication E-DMDR strategy, extended iFogSim used for the simulation experiments and setup in Eclipse IDE. iFogSim simulator (3.3) is beneficial for performing tests, simulation and implementations of Fog computing and IoT based algorithms. The significant components of this experiment are Eclipse IDE and iFogSim. For coding, consideredd Eclipse IDE for both data mining algorithm (MFCP) and Replication algorithm(E-DMDR).

Also,integrated data mining algorithm to iFogSim by using API Invocation. The network topology of this experiment **5** with a configuration of communication latency's is in the table **3**. The full assessment performed by using a machine having Windows 10 Home Single Language OS with Intel (R) Core(TM) i5-8250U CPU and 8GB of RAM.

5.3 Dataset Design and Description

For the evaluation of replica placement results, a synthetic realtime IoT traces considered. It contains candidate nodes.The MFCP replication algorithm works only on historical real-time traces to make an appropriate replica placement. The synthetic dataset includes cloud, fog gateway node 1, fog gateway node2, fog device 1 and fog device 2 as per physical network topology.

Furthermore,assigned values to the nodes like 1 for cloud,2 for fog gateway node 1,3 for fog gateway node 2 ,4 for fog device 1 and 5 for fog node 2 helps to understand replica placed on which node frequently. Also, considered Cleveland healthcare information of healthcare application works for replica frequency threshold to make proper replica decisions and to understand which is best node and which replica placed on which node.

6 Evaluation of Results and Experiments

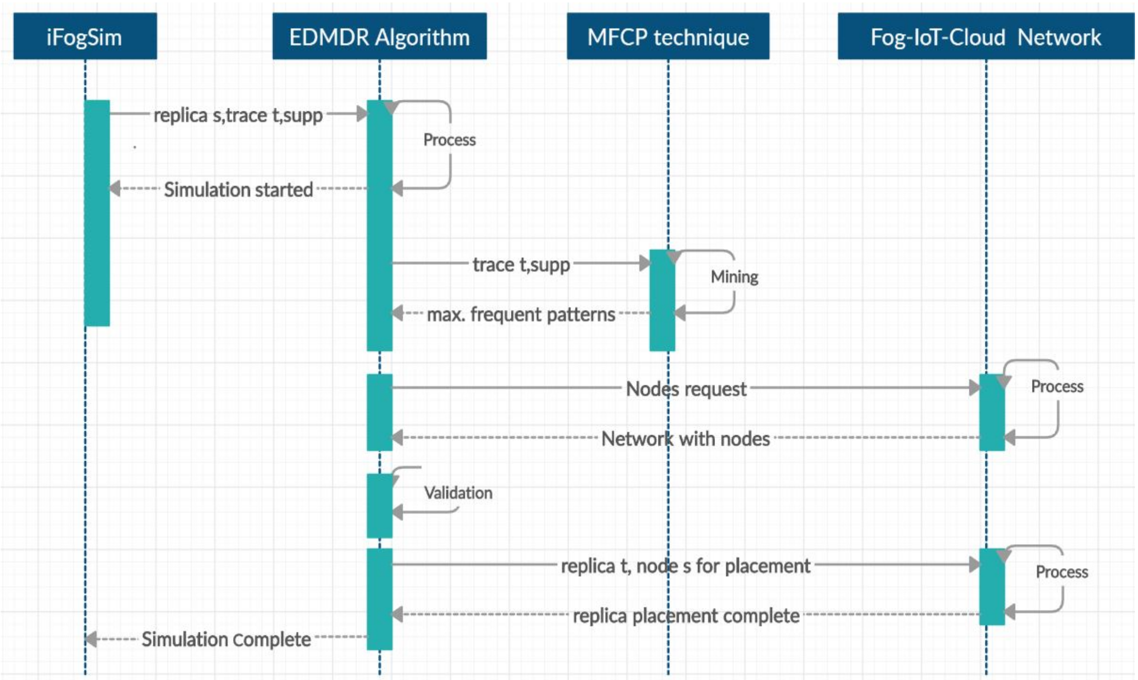


Figure 6: Simulation sequence diagram of replica placement

A sequence diagram indicated frequent pattern count generation and replica placement execution, as shown in the figure 6. Frequent patterns count generate on the bases of real-time fog node traces, replica frequency and support value using MFCP in section 3.2. If no more traces entries then remove duplicates using removeDuplicates() function and validate candidate Nodes. Also, if found any traces then add into replica decisions vector and display the final replica placement results.

The fog network attributes shown in table 2. It includes Cloud,fog gateways, fog devices, sensors and actuators. The efficiency of the CPU power, RAM, upload and download bandwidth (UpBW and DownBW in KBytes) characterizes each node, the hierarchical level of the rate per MIPS (cost rate per MIPS used).

The table 3 describes a detailed configurations of different communication latencies between Fog and Cloud data center in iFogSim network topology in figure 5. In that, IoT sensors and Actuators generate data collect by individual Fog Gateways and process on individual Fog Nodes (Data Hosts). Furthermore, the fog node sends the data packet to end-users.

	Cloud	**Gateway**	**Device**	**Sensor**	**Actuator**
Level	0	1	2	3	4
UpBW	100	10000	10000	0	0
DownBW	10000	10000	270	0	0
CPU (Mips)	44800	2800	3200	0	0
RAM (GB)	40000	4000	1000	0	0
Rate Per MIPS	0.01	0.0	0	0	0

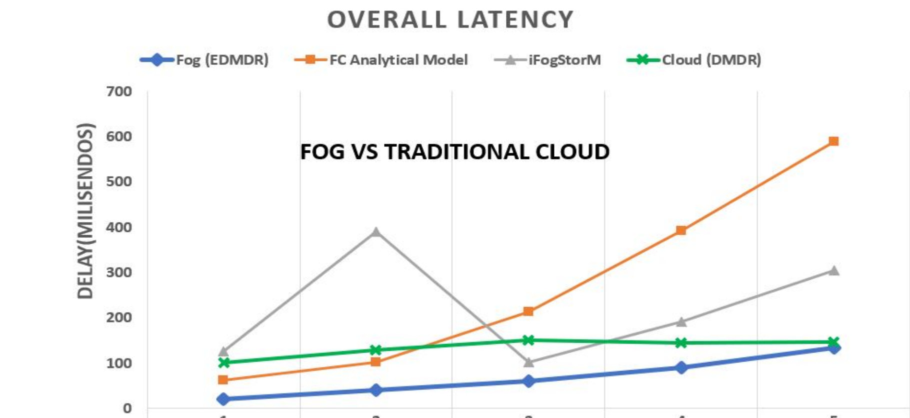
Table 2: Fog network attributes

Source	Destination	Latency(ms)
IoT Sensor1	Fog Device 1	50
IoT Actuator1	Fog Device 1	55
IoT Sensor2	Fog Device 2	25
IoT Actuator2	Fog Device 2	30
Fog Device 1	Fog Gateway 1	200
Fog Device 2	Fog Gateway 2	250
Fog Gateway 1	Cloud	500
Fog Gateway 2	Cloud	550

Table 3: Description of network links in iFogSim

6.1 Experiment 1: Execution Delay(Overall Latency)

In this experiment 1, aim to focus on diminishing the cpu execution delay (latency) in IoT sensors and actuators using fog network. Also, performed this experiment by using number of files includes candidate nodes in section 5.3 as an input to measure latency.



	1	2	3	4	5
Fog (EDMDR)	21.094	40.034	60.142	89.471	132.995
FC Analytical Model	62.12	101.37	212.19	391.34	587.47
iFogStorM	125.627	389.383	102.257	190.979	304.022
Cloud (DMDR)	100.019	129.539	149.942	145.444	145.923

Figure 7: Overall latency comparison with related strategies

The figure 7 is shown that overall latency(milliseconds) achieved in different replica placement strategies. The average execution delay of EDMDR 79% lower than the traditional cloud DMDR strategy. Also, EDMDR latency drop by 62%,83% compared to FC Analytical model and iFogStorM respectively. Overall, EDMDR reduces latency significantly because of MFCP keeps replica in fog nodes are close to the IoT sensors and actuators and improve the effectiveness as compared to the different replica placement strategies.

6.2 Experiment 2: Response Time

In experiment 2, evaluate the total average response time. For the same, consider a mathematical equation by the Mansouri et al. (2019)

$$ResponseTime = \frac{\sum_{h=1}^{e}\sum_{y=1}^{e_j}(H_{he}(rt)-H_{he}(st)}{\sum_{h=1}^{e} E_j} \tag{2}$$

Where H_he(rt) and H_he(st) indicates the send and receive times of job e in user h besides, e_j the number of the jobs for the user h. Moreover,performed this experiment by using different size of files (70 KB) includes candidate nodes in section 5.3 an input for response time.

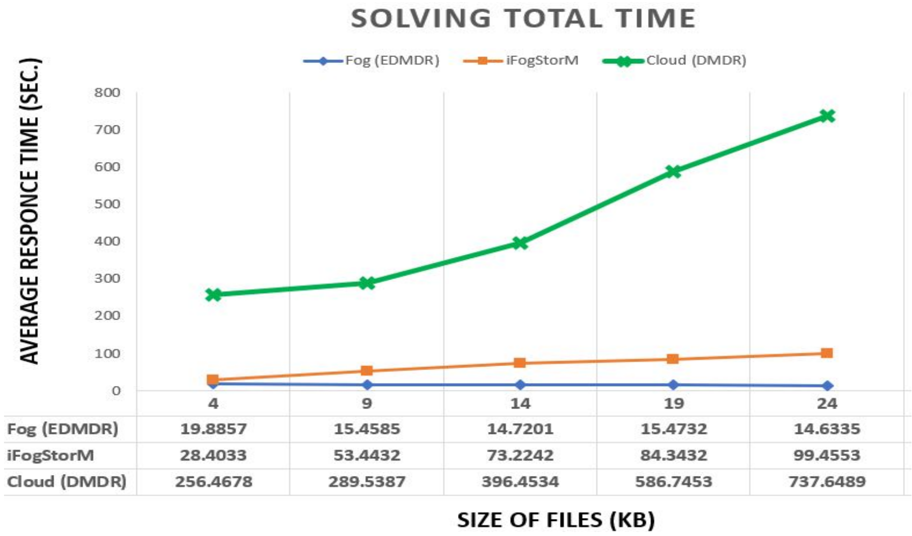


Figure 8: Response Time comparison with related strategies

As shown in the figure 8, compare with EDMDR, the response time of iFogStorM and Cloud(DMDR) are higher. Subsequently, to EDMDR total response time with the minimum support value 0.4 decreases by 71% and 92% respectively. Also, if the minimum support value is above 0.5, then EDMDR takes more time to respond. It uses pattern mining technique to find the frequent itemset(data) from the real-time traces. Thus, it reduces latency as well as average response time as per size of files(70 KB).Additionally, the lower response time indicates the effectiveness in data respond bandwidth.

6.3 Experiment 3: Total Network Usage (TNU)

In experiment 3, the evaluation of effective network usage includes the total ratio of transferred data as per user data requested from fog network. The low value of network usage indicates the proper placement of a replica. For the same, considered a mathematical equation by the Mansouri et al. (2019).

$$T_{tnu} = \frac{S_{rfa} + S_{fa}}{S_{lfa}} \tag{3}$$

Where S_rfa the number of access times the fog node receives a data(access patterns) from IoT sensors, S_fa indicates the total number operation of historical data replication, and S_lfa the number of times the fog node uses the data over the network.

Implementation of the EDMDR algorithm on iFogSim reduces the network usage by the distribution of the fog nodes at the network edges. The figure 9 is shown the network usage of EDMDR is better than other replication algorithms. It considers centrality and historical access log for replica placement. The network bandwidth utilization depends on effective network usage. The FC analytical model and Traditional cloud (DMDR) network usage are significantly increased by up to 85% and 93% compared to EDMDR respectively. Additionally,network usage increases if a large amount of data is shared between IoTs and cloud servers.

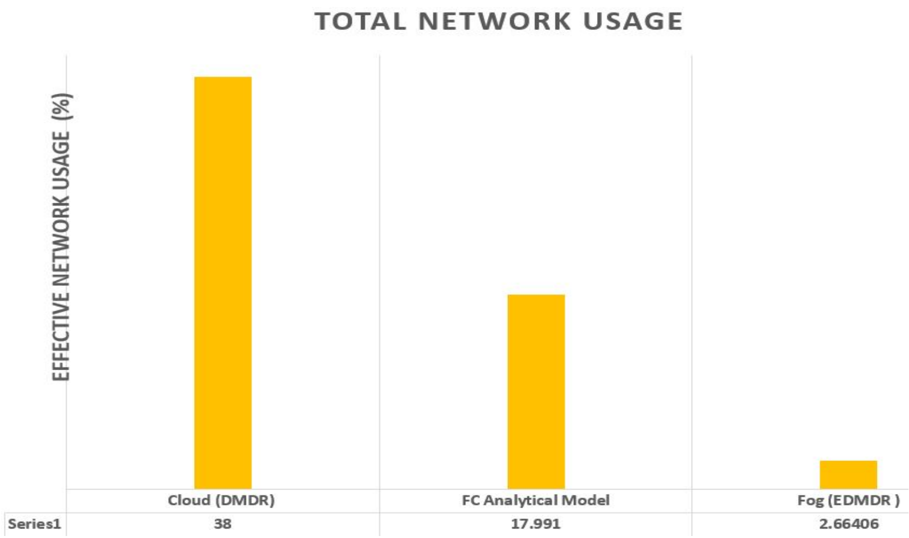


Figure 9: Total network usage comparison with related strategies

6.4 Experiment 4: Total Storage Usage

In experiment 4, aim to focus on minimizing the storage overhead, which proportional to reduce the cost of the resources. On the other hand, another objective is it would maximize the use of storage space in the cloud. The storage usage of dynamic replication is calculated by (Mansouri et al. (2019)). Also, evaluate this experiment by using different size of files (70 KB) as input and measure heap memory allocation.

$$Storage = \frac{Sufficient\_Storage\_Available}{Storage} \tag{4}$$

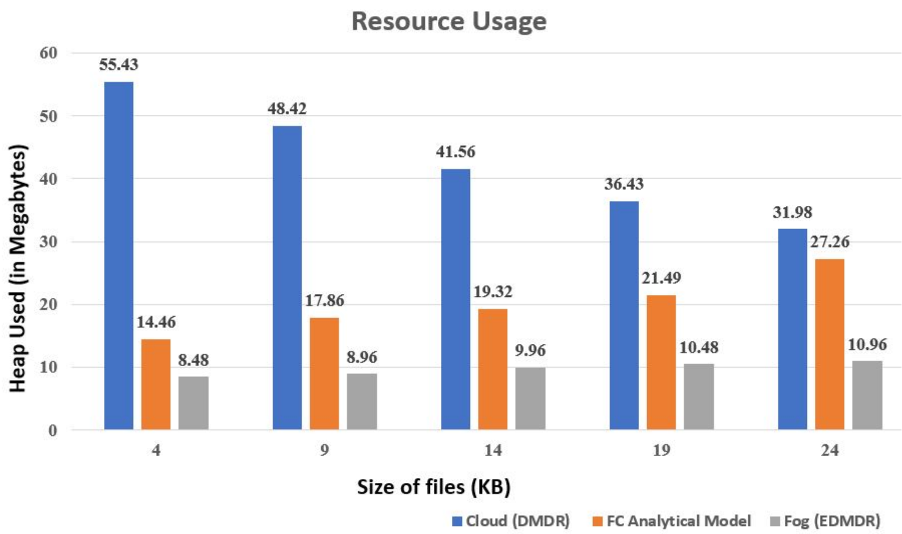


Figure 10: Resource usage comparison with related strategies

The implementation of EDMDR in iFogSim reduces storage overhead and usage. The figure 10 is shown the EDMDR storage usage is better on the basis of heap memory used in the java class. The FC analytical model and cloud(DMDR) storage consumption is diminished up to 41% and 85%, respectively.

6.5 Experiment 5: Total Energy Consumption

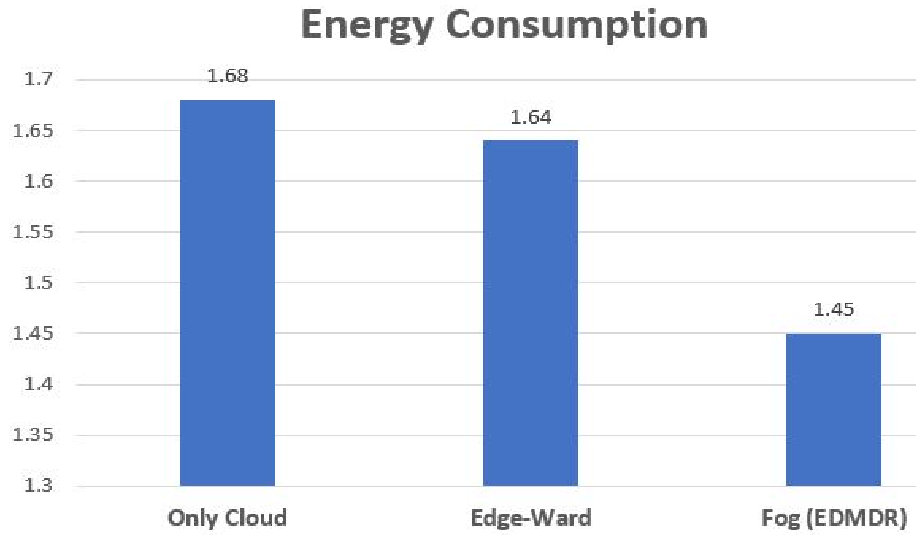


Figure 11: Cloud Energy consumption comparison with related strategies

The replica placement algorithms play a major role in reducing energy consumption. The figure 11 is shown the comparison of cloud energy consumption in different algorithms calculated in joules as per previous literature Maniglia et al. (2019). For the evaluation of this experiment, used existing class PowerDataCenter invocation in iFogSim package. Overall, the results indicates EDMDR energy consumption(KW) of power datacenter in the cloud is better as compared to cloud-only(14%) and Edge ward(12%) algorithm.

6.6 Discussion

In this simulation, the Overall latency , Response time , Network usage ,storage usage and Energy consumption in figures 7,8,9,10,11 for real-time fog traces in IoT infrastructure using EDMDR model in different physical network topology configurations are 550,250 and 55 ms demonstrated individually. Moreover, the total average values of network usage,Storage usage and energy consumption are reduced to 2.667%, 8.57 megabytes and 1.45 KW respectively. The EDMDR algorithm is shown and proved the reduction of overall latency as compared to existing literary works. The evaluation results from iFogSim simulation prove better reliability and enhancement in latency compared to iFogStorM 83%, FC analytical model 66% and 79% of cloud(DMDR). Additionally, the algorithm execution with the limited synthetic real-time dataset.Also, other important configuration parameters used in simulation is summarised in table 2,3 and figure 5 Overall, the experimental results of latency reduction using EDMDR is successfully examined,demonstrated and achieves the main goal in fog enabled IoT environment.

7 Conclusion and Future Work

In this research paper, EDMDR model introduced in fog enabled IoT environment to reduce high latency and cost of IoT applications. Moreover, implemented maximal frequent pattern mining(MFCP) method in fog environment works on real-time historical traces. Besides, upgraded ifogSim and implemented real-time scenario of EDMDR approach.

The overall results show significant improvement and enhancements in a reduction of latency as compared to related replica placement algorithms such as ifogStorM, FC Analytical Model and cloud(DMDR). EDMDR offers good data management solutions in fog infrastructure.

As proper replica placement is an active issue in fog enabled IoT infrastructure, this paper specifies limited real-time IoT traces.Moreover, the whole approach evaluated and examined results by considering a number of files and size of files. It consists of a cloud data centre node, fog gateway nodes(fog controller) and fog device nodes. As the size of the file increases, the response time is diminished. The high latency issue was investigated using evaluation metrics like Total latency(ms), the Response time(sec), Network Usage(%) and storage usage(MB). Also, it proves the maximal frequent correlated pattern mining supports fog enabled IoT environment.

In the future, plan to simulate the same approach in a real-world fog node using testbed. Also, the interesting part is to see how overall latency, response time , storage usage and network usage will be reduce using real-time testbed environment.

Acknowledgement

Special thanks to my project mentor and collaborator Prof.Vikas Sahni for constant support, inspiration and encouragement of actual direction throughout the MSc research project. Also, I like to thank the National College of Ireland for excellent online study resources such as library services as well as all teachers for my research.

References

Alarifi, A., Abdelsamie, F. and Amoon, M. (2019). A fault-tolerant aware scheduling method for fog-cloud environments, *PLOS ONE* **14**(10): 1–24. JCR Impact Factor:2.766:(2019).
URL: *https://doi.org/10.1371/journal.pone.0223902*

Aral, A. and Ovatman, T. (2018). A decentralized replica placement algorithm for edge computing, *IEEE Transactions on Network and Service Management* **15**(2): 516–529. JCR Impact Factor:3.286:(2018).

Braun, P., Cuzzocrea, A., Leung, C., Pazdor, A., Souza, J. and Tanbeer, S. (2019). Pattern mining from big iot data with fog computing: Models, issues, and research perspectives, City:Larnaca,2019, pp. 584–591. CORE Ranking: A.

Chang, W.-C. and Wang, P.-C. (2019). Write-aware replica placement for cloud computing, *IEEE Journal on Selected Areas in Communications* **PP**: 1–1. JCR Impact Factor:7.172:(2019).

Chunlin, L., Tang, H. and Luo, Y. (2019). Scalable and dynamic replica consistency maintenance for edge-cloud system, *Future Generation Computer Systems* **101**. JCR Impact Factor:4.639:(2019).

Dhande, R. (2020). Dynamic replica management for fog-enabled iot using enhanced data mining technique.

Fang, J., Li, K. and Ma, A. (2019). Latency aware online tasks scheduling policy for edge computing system, *Journal of Physics: Conference Series* **1325**: 012152. JCR Impact Factor:0.540:(2019).

Fong, S. (2017). Big data mining algorithms for fog computing, City:New York,2017, pp. 57–61. CORE Ranking: A.

Gupta, H., Dastjerdi, A., Ghosh, S. and Buyya, R. (2016). ifogsim: A toolkit for modeling and simulation of resource management techniques in internet of things, edge and fog computing environments, *Software: Practice and Experience* . JCR Impact Factor:1.931:(2017).

Hamrouni, T., Slimani, S. and Charrada, F. B. (2016). A survey of dynamic replication and replica selection strategies based on data mining techniques in data grids, *Eng. Appl. Artif. Intell.* **48**: 140–158. JCR Impact Factor:2.819:(2016).
URL: *https://doi.org/10.1016/j.engappai.2015.11.002*

Huang, T., Lin, W., Li, Y., He, L. and Peng, S. (2019). A latency-aware multiple data replicas placement strategy for fog computing, *Journal of Signal Processing Systems* pp. 1–14. JCR Impact Factor:1.088:(2019).

John, S. and Mirnalinee, T. (2019). A novel dynamic data replication strategy to improve access efficiency of cloud storage, *Information Systems and e-Business Management* . JCR Impact Factor:1.032:(2019).

Kumari, A., Tanwar, S., Tyagi, S., Kumar, N., Parizi, R. M. and Choo, K.-K. R. (2019). Fog data analytics: A taxonomy and process model, *Journal of Network and Computer Applications* **128**: 90 – 104. JCR Impact Factor:3.991:(2019).
URL: *http://www.sciencedirect.com/science/article/pii/S1084804518304016*

Li, C., Wang, Y., Tang, H. and Luo, Y. (2019). Dynamic multi-objective optimized replica placement and migration strategies for saas applications in edge cloud, *Future Generation Computer Systems* **100**. JCR Impact Factor:4.639:(2019).

Li, C., Wang, Y., Tang, H., Zhang, Y., Xin, Y. and Luo, Y. (2019). Flexible replica placement for enhancing the availability in edge computing environment, *Computer Communications* **146**. JCR Impact Factor:2.610:(2019).

Ma, B., Fong, S. and Millham, R. (2018). Data stream mining in fog computing environment with feature selection using ensemble of swarm search algorithms, City:durban,2018, pp. 1–6. CORE Ranking: C.

Mahmud, M., Ramamohanarao, K. and Buyya, R. (2018). Latency-aware application module management for fog computing environments, *ACM Transactions on Internet Technology* **19**. JCR Impact Factor:1.727:(2019).

Mahmud, M., Srirama, S., Ramamohanarao, K. and Buyya, R. (2018). Quality of experience (qoe)-aware placement of applications in fog computing environments, *Journal of Parallel and Distributed Computing* . JCR Impact Factor:1.185:(2018).

Maiti, P., Shukla, J., Sahoo, B. and Turuk, A. K. (2018). Qos-aware fog nodes placement, *2018 4th International Conference on Recent Advances in Information Technology (RAIT)*, City:Dhanbad,2018, pp. 1–6. CORE Ranking: C.

Maniglia, D., Silva, A., Asaamoning, G., Orrillo, H. and Mendes, P. (2019). An analysis of fog computing data placement algorithms, City:Houston,2019. CORE Ranking: A.

Mansouri, N., Javidi, M. and Mohammad Hasani Zade, B. (2019). Using data mining techniques to improve replica management in cloud environment, *Soft Computing* . JCR Impact Factor:2.237:(2019).

Naas, M., Lemarchand, L., Boukhobza, J. and Raipin Parvedy, P. (2018). A graph partitioning-based heuristic for runtime iot data placement strategies in a fog infrastructure, City:Brno,2018. CORE Ranking: B.

Ozeer, U., Etchevers, X., Letondeur, L., Ottogalli, F.-G., Salaün, G. and Vincent, J.-M. (2018). Resilience of stateful iot applications in a dynamic fog environment, City:New York,2018, pp. 332–341. CORE Ranking: A.

Perez Abreu, D., Velasquez, K., Curado, M. and Monteiro, E. (2019). A comparative analysis of simulators for the cloud to fog continuum, *Simulation Modelling Practice and Theory* **101**: 102029. JCR Impact Factor:2.092:(2019).

Shao, Y., Li, C. and Tang, H. (2018). A data replica placement strategy for iot workflows in collaborative edge and cloud environments, *Computer Networks* **148**. JCR Impact Factor:2.522:(2019).

Sharma, S. and Saini, H. (2019). A novel four-tier architecture for delay aware scheduling and load balancing in fog environment, *Sustainable Computing: Informatics and Systems* **24**: 100355. JCR Impact Factor:1.196:(2019).

Shukla, S., Hassan, M. F., Khan, M. K., Jung, L. T. and Awang, A. (2019). An analytical model to minimize the latency in healthcare internet-of-things in fog computing environment, *PLOS ONE* **14**(11): 1–31. JCR Impact Factor:2.766:(2019).
URL: *https://doi.org/10.1371/journal.pone.0224934*

Taneja, M. and Davy, A. (2017). Resource aware placement of iot application modules in fog-cloud computing paradigm, City:Lisbon,2017, pp. 1222–1228. CORE Ranking: A.

Vales, R., Moura, J. and Marinheiro, R. (2019). Energy-aware and adaptive fog storage mechanism with data replication ruled by spatio-temporal content popularity, *Journal of Network and Computer Applications* **135**. JCR Impact Factor:3.991:(2019).

Verma, S., Yadav, A. K., Motwani, D., Raw, R. S. and Singh, H. K. (2016). An efficient data replication and load balancing technique for fog computing environment, *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, City:Delhi,2017, pp. 2888–2895. CORE Ranking: B.

Yi, S., Li, C. and Li, Q. (2015). A survey of fog computing: Concepts, applications, and issues, City:Hangzhou,2015, pp. 37–42. CORE Ranking: B.