Computers, Materials & Continua
DOI: 10.32604/cmc.2022.023611
Article

Tech Science Press

Optimized Energy Efficient Strategy for Data Reduction Between Edge Devices in Cloud-IoT

Dibyendu Mukherjee[1], Shivnath Ghosh[1], Souvik Pal[2,*], D. Akila[3], N. Z. Jhanjhi[4], Mehedi Masud[5] and Mohammed A. AlZain[6]

[1]Department of Computer Science and Engineering, Brainware University, Barasat, 700125, India
[2]Department of Computer Science and Engineering, Global Institute of Management and Technology, Krishnanagar, 741102, India
[3]Department of Information Technology, School of Computing Sciences, Vels Institute of Science, Technology & Advanced Studies, Chennai, 600117, India
[4]School of Computer Science and Engineering, SCE Taylor's University, Subang Jaya, 47500, Selangor, Malaysia
[5]Department of Computer Science, College of Computers and Information Technology, Taif University, Taif, 21944, Saudi Arabia
[6]Department of Information Technology, College of Computers and Information Technology, Taif University, Taif, 21944, Saudi Arabia
*Corresponding Author: Souvik Pal. Email: souvikpal22@gmail.com

Received: 14 September 2021; Accepted: 03 December 2021

Abstract: Numerous Internet of Things (IoT) systems produce massive volumes of information that must be handled and answered in a quite short period. The growing energy usage related to the migration of data into the cloud is one of the biggest problems. Edge computation helps users unload the workload again from cloud near the source of the information that must be handled to save time, increase security, and reduce the congestion of networks. Therefore, in this paper, Optimized Energy Efficient Strategy (OEES) has been proposed for extracting, distributing, evaluating the data on the edge devices. In the initial stage of OEES, before the transmission state, the data gathered from edge devices are supported by a fast error like reduction that is regarded as the largest energy user of an IoT system. The initial stage is followed by the reconstructing and the processing state. The processed data is transmitted to the nodes through controlled deep learning techniques. The entire stage of data collection, transmission and data reduction between edge devices uses less energy. The experimental results indicate that the volume of data transferred decreases and does not impact the professional data performance and predictive accuracy. Energy consumption of 7.38 KJ and energy conservation of 55.57 kJ was found in the proposed OEES scheme. Predictive accuracy is 97.5 percent, data performance rate was 97.65 percent, and execution time is 14.49 ms.

Keywords: Energy; efficient; internet of things; transmission; performance; cloud computing; edge devices

1 Introduction

Energy efficient transmission of cloud data is a trending research area since it has wide range of applications [1]. The data generated using IoT devices is huge since there are wide range of IoT devices being used in the recent years. With the development of internet technology and modernization, wide ranges of devices are being used [2]. To avoid problems faced by smart energy techniques, edge computing is employed. Edge computing helps in reducing the amount of energy involved in cloud computing like encryption, transmission, storage etc. The edge computing techniques enables the reduction in the amount of data transferred [3]. The storage and processing of data in the cloud is done based on a distributed cloud computing infrastructure. This infrastructure is built using fog and edge computing [4]. These applications enable the deployment of network-accessible devices [5]. Optimization of latency is another important objective of cloud computing. Since cloud computing makes use of IoT devices that operate in real-time, these data must be processed with minimal latency. Hence, devising algorithms with minimal latency is a crucial task [6]. The production of smart products has increased at a rapid rate in the market. Big data analytics involve the processing of data generated using smart product [7]. Using the data processing, various actions are being automated. These automation techniques help in the reduction in human intervention [8]. The convergence of cloud computing is another important aspect involved in the pervasive service. Connection of heterogeneous devices is possible only if the devices are compatible with each other. The establishment of compatibility is done using effective communication technologies [9]. Indoor smart appliances are popularly being used in smart cities. To increase the reliability and efficiency of these devices, energy efficient algorithms are necessary. These smart devices are integrated based on seamless integration techniques. The wearable devices are used commonly by the elderly people [10]. It found that even as demand for cloud computing has raised, efficiency improvements have kept energy usage almost flat across the globe's data centers.

Furthermore, studies have repeatedly shown that hyper-scale (i.e., very large) data centers use far less energy than local servers. These devices are used for monitoring the motion of the elderly [11]. The data generated by these devices are processed using effective energy efficient algorithms, so that useful information like fall detection and abnormal action detection can be done in the case of emergencies [12]. It is estimated that by the end of the year 2025, more than 60 billion devices will be connected using internet around the world. Thus, the big data is estimated to get bigger and bulkier. Thus, energy efficiency is the main need of the hour [13]. To adapt to these changes mobile edge computing techniques are getting popular [14]. Encryption is another important task in cloud computing as the cloud data is easily accessible to the public. These algorithms must be energy efficiency and at the same time more secure. Since decryption is also done at the receiver side, the key transmission must be done in a secure manner [15]. Data processing in edge devices is done often to support the large number of IoT devices employed in smart buildings, cities and transportation. Gateways, cameras, and other IoT end devices benefit from Cloud IoT Edge's powerful data processing and machine learning capabilities, making these applications smarter, safer, and more dependable. These devices generate data at real-time. The main issue faced here is the transfer of huge amount of data. To avoid this issue, in this research, we propose a new technique for reduction of cloud data using data compression techniques. Section 2 represents the related works and Section 3 gives the paper contributions. The proposed work is discussed in Section 4 and the results are evaluated in the Section 5. The conclusion along with the future scope is given in the Section 6. Finally, the references are provided.

2 Related Works

This section discussed on related works on different strategies on energy efficiency in Cloud-IoT environment. Khan et al. [16] proposed a scheme for energy efficiency computation in edge computing systems. Deep reinforcement learning was employed for the energy conservation. The main advantage was the long-term power conservation due to the usage of deep reinforcement learning. However, this scheme did not provide higher performance gain. Wang et al. [17] did a survey on the relationship between deep learning schemes and edge computing. The applications of deep learning were discussed along with the practical implementation methodologies. Further, the fusion of edge intelligence was also presented. The usage of artificial intelligence was discussed in terms of cloud computing.

Jiang et al. [18] presented a scheme for edge computing based on energy aware schemes. The techniques employed for offload computation of cloud data sources were discussed along with the interactions between the edge servers. However, the methods to increase the security and encryption methodologies were not presented. Mekala et al. [19] presented a survey on the types of energy efficient sensors. The approaches used for green computing was presented along with the implementation techniques for the IoT devices. The difficulties involved in sensor selection and the impact of communication was analyzed. The impact rate of resource balancing was not provided in terms of mathematical modeling.

Shi et al. [20] designed a system for analyzing the energy-based artificial intelligence system involved in edge computing. The energy conservation was proposed to avoid the traffic congestion and energy requirements. Further, the privacy concerns involved in edge computing was discussed along with the future research directions. Kavitha et al. [21] presented a scheme for data management based on concurrent data. This scheme was designed for the mobile environment. The edge computing scheme was implemented in mobile platform to support delay aware systems. However, this system did not provide an optimized solution for data preservation.

Zahmatkesh et al. [22] presented the technologies involved in fog computing environment. An overview was provided on various technologies like machine learning, artificial intelligence and unmanned aerial vehicles employed in IoT based fog computing. However, the algorithms used for creation of energy efficient environment was not presented. Rimminen et al. [23] presented a virtualization model for the collaboration of edge computing. A system was designed that represented virtual nodes that were deployed over the physical devices. These devices formed the three-tier architectural framework. This paper failed to provide the methodologies for cost effective implementation.

Mohiuddin et al. [24] presented the security challenges faced by the cloud computing applications. The strategies used to overcome these challenges were also discussed in terms of cloud storage providers. The types of encryption algorithms employed for the security enhancement of cloud data was discussed along with their drawbacks. Authors in [25–27] proposed a scheme for the attainment of security in the low power IoT devices. The main focus was to ensure the transport layer security. This paper employed ultra-low power sensor nodes for the edge computing to increase the overall battery lifetime.

Jeyalaksshmi et al. [28] have discussed on mapping and allotment of VMs in cloud environment, and they have also presented reliability profile algorithm in VMs. Pal et al. [29,30] have presented resource migration algorithms in live migration concept and they have also shown the techniques to minimize execution and waiting time in cloud.

The above literature survey clearly shows that the main challenge of edge computing is energy conservation. Conservation of energy helps to increase the overall lifetime of the battery and performance of the sensor nodes. To overcome this challenge, this research provides a scheme for extracting, distributing, evaluating the data on the edge devices based on energy efficient techniques.

3 Contribution of the Paper

The contributions of this paper are as follows:

- A novel technique called Optimized Energy Efficient Strategy (OEES) is proposed for the energy efficient data reduction between edge nodes in cloud network.
- A new technique for data reduction between edge nodes is presented.
- The processing of cloud IoT data is discussed using mathematical models.

The performance of the proposed scheme is analyzed using metrics like energy consumption, energy conserved, predictive accuracy, data performance rate, execution delay, cloud accuracy, edge efficiency, compression ratio and reconstruction accuracy.

4 Proposed Methodology

4.1 Proposed Optimized Energy Efficient Strategy (OEES) Model

The proposed OESS model comprises of three main layers. The lower layer is the IoT device group, the second layer is the energy conservation group, and the third layer is the cloud data center.

Fig. 1 shows the proposed OESS model. The lower layer comprises of various IoT devices that include smart home, smart factory, smart vehicles, smart transportation, smart health care centres etc. The data obtained from these devices is huge. To reduce the amount of energy required for the energy transfer, this data is reduced using the proposed OESS model. The data reduction is done in the second layer. In this layer, the data of the sensor device is processed by the edge nodes. Further, the intermediate edge nodes are also responsible for the energy efficient data reduction. The reduced data is then transferred to the cloud data centre which the top layer.

4.2 Data Reduction Between Edge Nodes

The reduction of data between the edge nodes takes place using 4 main steps. These include data reduction, data reconstruction, data processing and deep learning.

Fig. 2 shows the block diagram of data reduction between edge nodes. The data reduction is done based on compressive sensing techniques. The compression is done by the multiplication with a measurement matrix. This matrix helps in the data compression with minimal reduction in data content. The compressed data is then reconstruction using inverse measurement matrix. This matrix is constructed based on the matrix used for data compression. Finally, the compressed data is processed to make it suitable for transmission. The processed data is then transmitted to the nodes through controlled deep learning techniques.

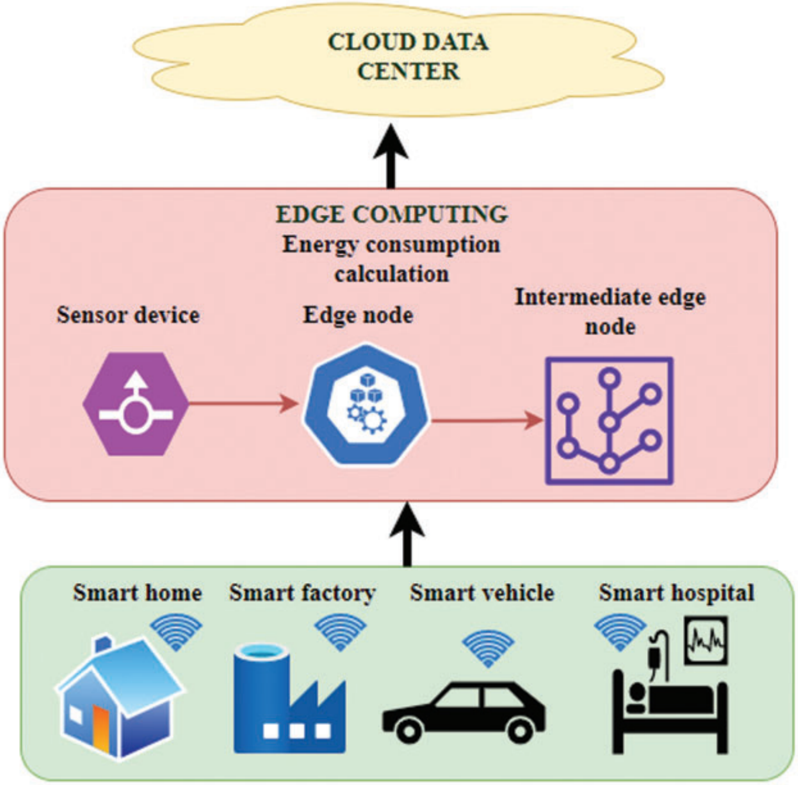

Figure 1: Optimized energy efficient strategy (OEES) model

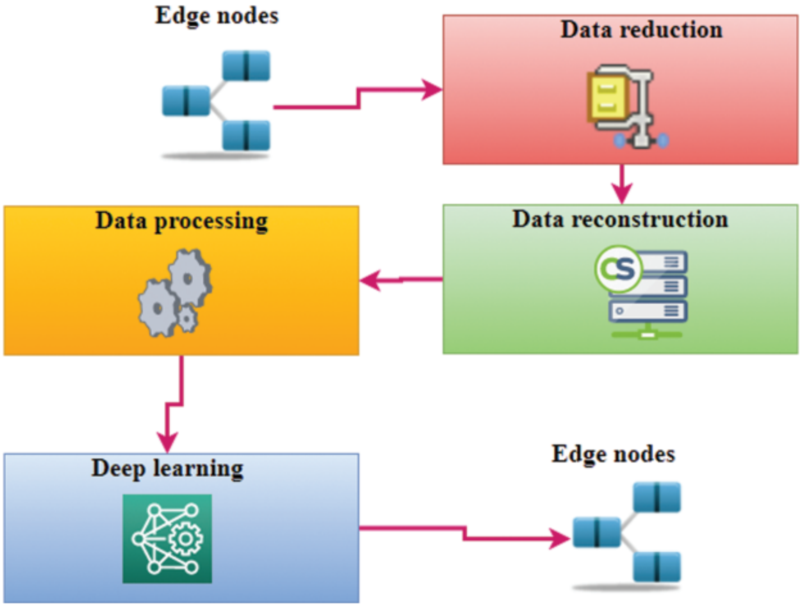

Figure 2: Data reduction between edge nodes

4.3 Optimized Energy Efficient Strategy (OEES) for Data Reduction

The proposed optimized energy efficient strategy comprises of steps like data acquisition, data display, data storage, data encryption, data analysis and data compression.

Fig. 3 shows the proposed OESS model used for data reduction. In this model, the first step is the acquisition of data from the cloud-based devices. This data is initially displayed using the display modules. The cloud data is then stored for temporary usage using data storage devices. Before transmission of this data through the cloud, this data is encrypted using encryption algorithms. The encrypted data is then analyzed and compressed using the proposed OESS algorithm.

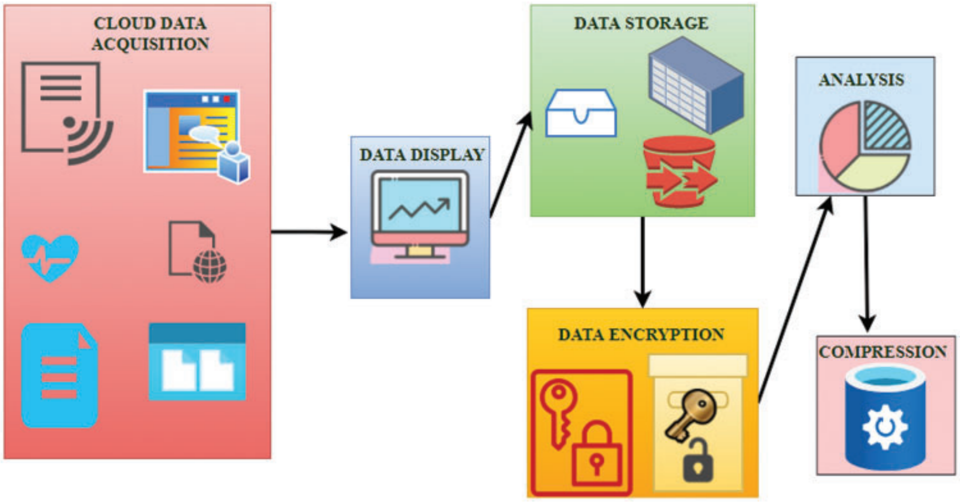

Figure 3: OEES model for data reduction

This algorithm makes use of input data like task $T_a = \{t_{ij}\}; i \leq 1 \leq U, j \leq 1 \leq V$, energy $E_n = \{e_{ij}\}; i \leq 1 \leq U, j \leq 1 \leq V$, execution time $E_t = \{x_{ij}\}; i \leq 1 \leq U, j \leq 1 \leq V$, virtual machine $V_m = \{v_{ij}\}; i \leq 1 \leq U, j \leq 1 \leq V$, time delay $T_d = \{d_{ij}\}; i \leq 1 \leq U, j \leq 1 \leq V$, node location $N_l = \{n_{ij}\}; i \leq 1 \leq U, j \leq 1 \leq V$, compression rate $C_r = \{c_{ij}\}; i \leq 1 \leq U, j \leq 1 \leq V$ and reconstruction rate $R_r = \{r_{ij}\}; i \leq 1 \leq U, j \leq 1 \leq V$.

Algorithm 1: Optimized Adaptive Energy Efficient Strategy (OEES) algorithm

Input:
Task $T_a = \{t_{ij}\}; i \leq 1 \leq U, j \leq 1 \leq V$
Energy $E_n = \{e_{ij}\}; i \leq 1 \leq U, j \leq 1 \leq V$
Execution time $E_t = \{x_{ij}\}; i \leq 1 \leq U, j \leq 1 \leq V$
Virtual machine $V_m = \{v_{ij}\}; i \leq 1 \leq U, j \leq 1 \leq V$
Time delay $T_d = \{d_{ij}\}; i \leq 1 \leq U, j \leq 1 \leq V$
Node location $N_l = \{n_{ij}\}; i \leq 1 \leq U, j \leq 1 \leq V$
Compression rate $C_r = \{c_{ij}\}; i \leq 1 \leq U, j \leq 1 \leq V$
Reconstruction rate $R_r = \{r_{ij}\}; i \leq 1 \leq U, j \leq 1 \leq V$

(Continued)

Algorithm 1: Continued

```
Output:
Energy efficiency optimization function EEO
Steps:
Initialize range $R \leftarrow 0$
Initialize iteration number $I \leftarrow 0$
Set Maximum iteration count $IT \leftarrow 500$
Set Total location $L \leftarrow 50$
    for each $L = I$
    Initialize L[IT]=0
Compute ECT(n), ECS(n) and PA(n)
        for n = 1:IT do
                if {ECT(n)<ECS(n)} then
                        Compute DP(n), ED(n), CA(n)
                end if
            end for
            L[IT]++
     Find EE(n)
     Compute min EE(n)
     Update CR(n)
     end for
     Compute RA(n)
 Calculate Energy efficiency optimization function EEO
 Return EEO
```

Energy consumption is computed as

$$ECT(n) = \prod_{i=1}^{U}\prod_{j=1}^{V}\left[\frac{t_{ij} + \partial(e_{ij})}{t_{ij} - e_{ij}}\right] * \left[\frac{x_{ij} + \partial(v_{ij})}{x_{ij} - v_{ij}}\right] \tag{1}$$

where, $ECT(n)$ is the energy consumed, t_{ij} is the task set, e_{ij} is the energy set, x_{ij} is the execution time set, v_{ij} is the virtual machine set, $\partial(e_{ij})$ is the rate of change of energy, $\partial(v_{ij})$ is the rate of change of virtual machine, $t_{ij} + \partial(e_{ij})$ is the sum of the task set and rate of change of energy and $x_{ij} + \partial(v_{ij})$ is the sum of execution time set and rate of change of virtual machine. This value gives the amount of energy consumed by the virtual machine.

Energy conserved is computed as

$$ECS(n) = \sum_{i=1}^{U}\sum_{j=1}^{V} ECT(n)/[e_{ij} + \exp(x_{ij}, v_{ij})^{ECT(n)-d_{ij}}] \tag{2}$$

where, $ECS(n)$ is the energy conserved, $ECT(n)$ is the energy consumed, e_{ij} is the energy set, x_{ij} is the execution time set, v_{ij} is the virtual machine set, d_{ij} is the time delay set, U represents the total x variants, and V represents the total y variants. This value gives the amount of energy conserved using the data reduction technique.

Predictive accuracy is given by

$$PA(n) = \underset{i,j}{arg\,max}\, \frac{ECT(n) * \exp(ECS(n))}{x_{ij} * v_{ij}} - d_{ij} * n_{ij} \tag{3}$$

where, $PA(n)$ is the predictive accuracy, $ECS(n)$ is the energy conserved, $ECT(n)$ is the energy consumed, x_{ij} is the execution time set, v_{ij} is the virtual machine set, d_{ij} is the time delay set and n_{ij} is the node location set. The predictive accuracy gives the rate at which the data is predicted correctly.

Data performance rate is defined as

$$DP(n) = \frac{[PA(n) + \mathrm{mod}(ECS(n))]}{v_{ij} * d_{ij}} * \frac{[ECS(n) + \mathrm{mod}(PA(n))]}{n_{ij} * c_{ij}} \tag{4}$$

where $DP(n)$ is the data performance rate, $PA(n)$ is the predictive accuracy, $ECS(n)$ is the energy conserved, v_{ij} is the virtual machine set, d_{ij} is the time delay set, n_{ij} is the node location set and c_{ij} is the compression rate. The data performance rate is the rate at which the data is reconstructed.

Execution delay is computed as

$$ED(n) = \left\{PA(n) * \sum\nolimits_{i=1}^{M} \frac{d_{ij} - n_{ij}}{d_{ij} + n_{ij}}\right\} / \left\{\sum\nolimits_{i=1}^{M} \frac{c_{ij} - r_{ij}}{c_{ij} + r_{ij}} * DP(n)\right\} \tag{5}$$

where $ED(n)$ is the execution delay, d_{ij} is the time delay set, n_{ij} is the node location set, c_{ij} is the compression rate, r_{ij} is the reconstruction rate, $DP(n)$ is the data performance rate and $PA(n)$ is the predictive accuracy. The execution delay is the time delay involved in the execution of virtual program.

Cloud accuracy is given by

$$CA(n) = \sqrt{\sum_{i=1}^{U} \sum_{i=1}^{V} \left\{\frac{\exp(DP(n) - n_{ij}) + \exp(ED(n) - c_{ij})}{r_{ij}}\right\}} \tag{6}$$

$CA(n)$ is the cloud accuracy, $ED(n)$ is the execution delay, $DP(n)$ is the data performance rate, n_{ij} is the node location set, c_{ij} is the compression rate and r_{ij} is the reconstruction rate. The cloud accuracy represents the rate at which the cloud data is processed correctly.

Edge efficiency is computed as

$$EE(n) = \frac{1}{UV} \sum_{i=1}^{U} \sum_{j=1}^{V} \left[\frac{||CA(n) - ED(n)||_2^2}{t_{ij} - e_{ij}} + \frac{||CA(n) - ED(n)||_2^2}{x_{ij} - v_{ij}}\right] \tag{7}$$

$EE(n)$ is the edge efficiency, $CA(n)$ is the cloud accuracy, $ED(n)$ is the execution delay, t_{ij} is the task set, e_{ij} is the energy set, x_{ij} is the execution time set, v_{ij} is the virtual machine set and the term $||CA(n) - ED(n)||_2^2$ is the difference between cloud accuracy and execution delay. The edge efficiency gives the efficiency of the edge network.

Compression ratio is calculated as

$$CR(n) = \frac{\int\limits_{i} \int\limits_{j} [(EE(n) - e_{ij}) + (x_{ij} * v_{ij})]}{CA(n) - d_{ij}} \tag{8}$$

where $CR(n)$ is the compression ratio, $EE(n)$ is the edge efficiency, $CA(n)$ is the cloud accuracy, e_{ij} is the energy set, x_{ij} is the execution time set, v_{ij} is the virtual machine set, d_{ij} is the time delay set and

$(x_{ij} * v_{ij})$ is the product of execution time set and the virtual machine set. Compression ratio gives the rate at which the data is compressed using the compression algorithm.

Reconstruction accuracy is given by

$$RA(n) = \oint_{i\geq 0} \max_{x=i}\{d_{ij}, 0\} - ||CR(n) - x_{ij}||_2^2 + v_{ij} - n_{ij} * EE(n) \tag{9}$$

where $RA(n)$ is the reconstruction accuracy, x_{ij} is the execution time set, v_{ij} is the virtual machine set, d_{ij} is the time delay set, $\max\{d_{ij}, 0\}$ selects the maximum among zero and the time delay set. Reconstruction accuracy gives the rate at which the data is reconstructed efficiently.

Energy efficiency optimization function EEO

$$EEO = Minimize \sum_{n=1}^{N} \frac{ECT(n) * ECS(n)}{DP(n)/ED(n)} + \sum_{f=1}^{N} \frac{CA(n) * EE(n)}{PA(n)/CR(n)} \; s.t. \; n\forall RA(n) \tag{10}$$

where EEO is the energy efficiency optimization function, $ECS(n)$ is the energy conserved, $ECT(n)$ is the energy consumed, $DP(n)$ is the data performance rate, $PA(n)$ is the predictive accuracy, $EE(n)$ is the edge efficiency, $CA(n)$ is the cloud accuracy, $ED(n)$ is the execution delay, $CR(n)$ is the compression ratio and $RA(n)$ is the reconstruction accuracy.

5 Results and Discussion

In this section, we have performed the performance analysis of the proposed scheme. The proposed OEES scheme was compared using various algorithms like dictionary compression (DC), prefix encoding compression (PEC), indirect encoding compression (IEC), sparse encoding compression (SEC), cluster compression (CC) and run-length encoding compression (REC). All the simulations were performed using Intel core i5 processor using Virtual box VM execution environment in a system with 6GB RAM.

5.1 Performance Analysis

For quantitative evaluation we have employed metrics like energy consumption, energy conserved, predictive accuracy, data performance rate, execution delay, cloud accuracy, edge efficiency, compression ratio and reconstruction accuracy.

Tab. 1 shows the comparison of energy consumption for variation in the number of virtual machines (VMs). It is clearly observed that, the overall energy consumption for dictionary compression is 21.44 KJ, prefix encoding compression is 27.78 KJ, indirect encoding compression is 28.76 KJ, sparse encoding compression is 22.24 KJ, cluster compression is 20.01 KJ and run-length encoding compression is 23.68 KJ. The proposed scheme attains a minimal energy consumption of 7.38 KJ. This is due to the effective data compression scheme implemented in the proposed system.

Table 1: Energy consumption (KJ)

Number of VMs in EDC	Energy consumption (KJ)						
	DC	PEC	IEC	SEC	CC	REC	OEES
1	24.35	24.13	23.95	38.63	29.28	23.1	7.44
2	17.7	22.12	34.42	26.28	10.04	23.1	6.1

(Continued)

Table 1: Continued

Number of VMs in EDC	Energy consumption (KJ)						
	DC	PEC	IEC	SEC	CC	REC	OEES
3	21.07	15.37	36.96	26.2	10.91	11.47	6.13
4	29.85	39.07	22.88	19.33	16.25	11.48	7.68
5	15.08	22.22	20.03	12.13	23.65	12.73	8.81
6	18.36	35.34	27.9	15.46	13.81	27.82	6.74
7	15.94	28.46	37.06	12.79	10.25	17.23	7.31
8	15.85	21.3	31.06	23.9	31.81	35.24	8.2
9	19.8	36.32	21.32	10.28	20.62	35.72	9.59
10	36.41	33.55	32.05	37.45	33.42	38.91	5.8
11	35.32	35.56	36.94	39.08	43.98	48.93	6.90

Tab. 2 shows the comparison of energy conserved for variation in the number of virtual machines (VMs). It is clearly observed that, the overall energy conserved for dictionary compression is 14.92 KJ, prefix encoding compression is 15.14 KJ, indirect encoding compression is 15.23 KJ, sparse encoding compression is 16.15 KJ, cluster compression is 16.74 KJ and run-length encoding compression is 13.59 KJ. The proposed scheme attains a highest energy conservation of 55.57 KJ. This is because of the usage of measurement matrix in the IoT data reduction.

Table 2: Energy conserved (KJ)

Number of VMs in EDC	Energy conserved (KJ)						
	DC	PEC	IEC	SEC	CC	REC	OEES
1	20	13.69	12.83	11.55	18.31	10.34	57.16
2	19.81	12.08	11.33	18.19	11.56	18.86	55.78
3	11.27	14.41	16.85	16.25	14.57	14.08	54.33
4	12.32	19.57	19.1	17.39	16.18	10.36	58.85
5	10.23	11.24	16.11	18.05	19.33	17.46	53.93
6	16.08	14.71	19	10.67	18.35	11.54	51.79
7	11.1	18.57	11.93	19.51	18.96	11.44	56.33
8	14.07	10.43	17.55	14.98	15.83	16.06	56.24
9	18.84	16.92	13.46	17.55	15.83	12.54	53.28
10	15.48	19.79	14.19	17.43	18.55	13.24	58.03
11	16.54	20.98	17.67	20.87	23.98	15.94	60.78

Fig. 4 shows the comparison of execution delays for variation in the number of virtual machines (VMs). It is clearly observed that, the overall execution delay for dictionary compression is 51.67 ms, prefix encoding compression is 48.32 ms, indirect encoding compression is 60.73 ms, sparse encoding compression is 56.22 ms, cluster compression is 50.54 ms and run-length encoding compression is 47.48 ms. The proposed scheme attains a lowest execution delay of 14.49 ms. The least value of execution delay in this scheme is due to the incorporation of time delay factor in the optimization. The least value of execution delay in this scheme is due to incorporating the time delay factor in the optimization. The Internet of Things (IoT) benefits from computing power being located closer to where a physical device or data source exists. Sensor and device data must be analyzed at the edge rather than traveling back to the central site for analysis before it can be used to react faster or mitigate issues. Edge computing reduces communication latency between IoT devices and the central IT networks to which they are connected by processing and storing data locally.

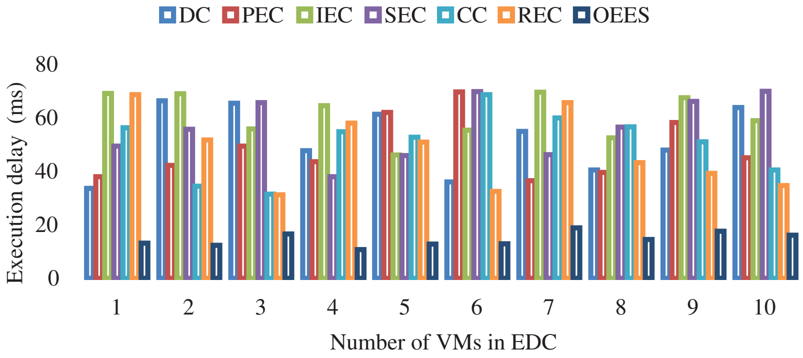

Figure 4: Variation of execution delay

Fig. 5 depict the variation of cloud accuracy. The cloud accuracy of the proposed OESS scheme is as high as 85.07%. However other schemes like dictionary compression, prefix encoding compression, indirect encoding compression, sparse encoding compression, cluster compression and run-length encoding compression have low cloud accuracy of 56.93%, 52.54%, 54.49%, 57.47%, 55.35% and 57.44% respectively. The cloud accuracy of the proposed scheme is high since it performs all the tasks in accurate manner using high reconstruction rate.

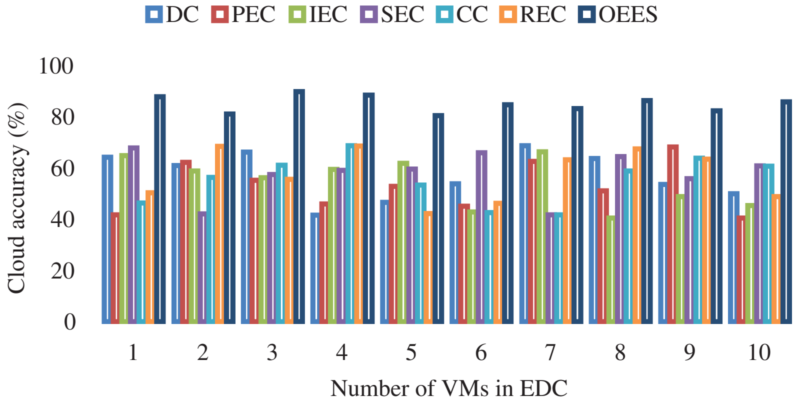

Figure 5: Variation of cloud accuracy

Fig. 6 shows the variation of edge efficiency. The edge efficiency of the proposed OESS scheme is as high as 85.07%. However other schemes like dictionary compression, prefix encoding compression, indirect encoding compression, sparse encoding compression, cluster compression and run-length encoding compression have low edge efficiency of 46.19%, 47.70%, 41.3%, 47.13%, 45.48% and 44.14% respectively. The edge efficiency of the proposed scheme is high since all the edge devices employed in this framework have good prediction rate.

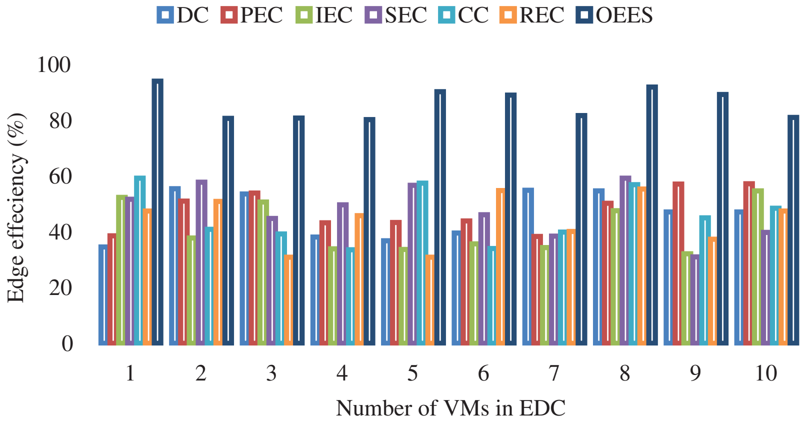

Figure 6: Variation of edge efficiency

Fig. 7 shows the variation of compression ratio. The compression ratio of the proposed OESS scheme is as high as 85.82%. However other schemes like dictionary compression, prefix encoding compression, indirect encoding compression, sparse encoding compression, cluster compression and run-length encoding compression have low compression ratio of 43.21%, 44.20%, 44.3%, 49.13%, 45.93% and 41.94% respectively. The compression ratio of the proposed scheme is the highest since OEES scheme utilizes compression based on compressive sensing theory.

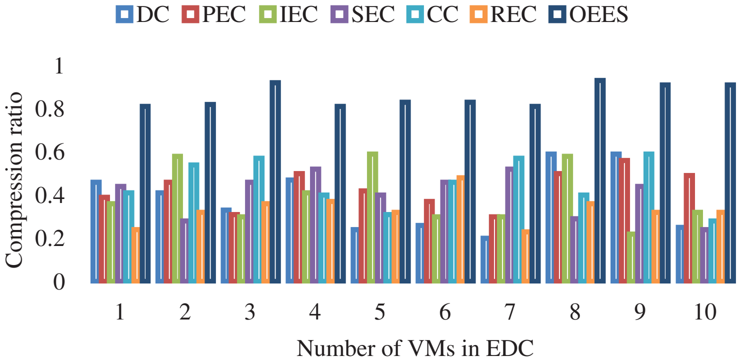

Figure 7: Variation of compression ratio

Tab. 3 shows the comparison of predictive accuracy for variation in the number of virtual machines (VMs). It is clearly observed that, the overall predictive accuracy for dictionary compression is 31.48%, prefix encoding compression is 47.07%, indirect encoding compression is 45.05%, sparse encoding compression is 42.15%, cluster compression is 37.59% and run-length encoding compression is 45.66%. The proposed scheme attains a maximum predictive accuracy of 85.11%. The proposed OESS scheme has a prediction rate greater than 85% since it employs virtual machine for the energy computation.

Table 3: Comparison of predictive accuracy (%)

Number of VMs in EDC	Predictive accuracy (%)						
	DC	PEC	IEC	SEC	CC	REC	OEES
1	34.11	54.42	43.42	24.08	42.45	43.15	88.23
2	34.38	51.57	25.78	41.71	22.46	67.46	80.25
3	33.17	59.45	50.84	13.08	23.15	63.57	84.14
4	46.59	59.68	24.02	55.42	29.55	31.39	87.32
5	20.01	27.6	37.39	46.12	15.75	42.78	87.82
6	21.28	28.56	33.07	61.43	54.85	30.8	83.67
7	15.67	41.38	42.32	69.3	54.91	47.37	87.45
8	29.39	29.52	69.51	65.77	42.6	57.8	88.93
9	56.18	59.91	55.32	34.57	30.29	54.75	82.42
10	24.04	58.62	68.83	10.02	59.94	17.53	81.29
11	25.98	59.89	70.87	11.09	60.87	20.89	82.76

Tab. 4 shows the variation of data performance rate. The data performance rate of the proposed OESS scheme is as high as 85.82%. However other schemes like dictionary compression, prefix encoding compression, indirect encoding compression, sparse encoding compression, cluster compression and run-length encoding compression have low data performance rate of 43.21%, 44.20%, 44.3%, 49.13%, 45.93% and 41.94% respectively. The data performance rate of the proposed scheme is as high as 85% since it uses minimal execution delay with highest energy conservation.

Table 4: Comparison of data performance rate (%)

Number of VMs in EDC	Data performance rate (%)						
	DC	PEC	IEC	SEC	CC	REC	OEES
1	49.98	27.52	41.71	49.65	42.86	48.26	83.37
2	24.21	47.62	23.16	18.14	43.65	32.92	85.25
3	11.88	10.23	18.94	43.09	24.18	43.99	84.3
4	18.54	34.41	22.49	37.04	27.2	21.05	93.92
5	25.91	42.05	33.38	19.96	32.89	34.89	80.77
6	23.35	19.32	43.2	29.03	38.03	33.54	88.89
7	19.18	47.3	21.62	25.96	39.7	48.54	82.44
8	47.45	40.53	26.1	33.98	40.32	13.43	92.58
9	37.33	43.06	44.49	42.02	25.56	30.02	82.51
10	48.49	32.94	34.59	14.2	27.17	30.86	87.53
11	50.87	33.89	35.67	15.78	28.54	31.89	88.76

6 Conclusion and Future Work

This research presented a new approach called Optimized Energy Efficient Strategy (OEES) for the optimal data transfer and data reduction between cloud devices. In this model, the acquisition of data from the cloud-based devices was done initially. This data was then displayed using the display modules. The displayed data was then stored for temporary usage using data storage devices. The stored data was encrypted using encryption algorithms. The encrypted data was then analyzed and compressed using the proposed OESS algorithm. The proposed OEES scheme was compared using various algorithms like dictionary compression, prefix encoding compression, indirect encoding compression, sparse encoding compression, cluster compression and run-length encoding compression. It was inferred that the proposed OESS scheme attained energy consumption of 7.38 KJ, energy conservation of 55.57 kJ, predictive accuracy of 97.5%, data performance rate of 97.65%, execution delay of 14.49 ms, cloud accuracy of 85.12%, edge efficiency of 85.07%, compression ratio of 85.82% and reconstruction accuracy of 80.04%. In future, we plan to develop a new application for the implementation of energy efficient data reduction in cloud data services. Edge computing reduces the amount of data flowing to and from the primary network by processing data closer to the source and prioritizing traffic. As a result, latency is reduced, and overall speed increases. Athletes need a lot of room to operate effectively. For industrial and enterprise-level businesses, edge computing is critical because it creates new and improved ways to maximize operational efficiency and safety while also improving performance and reliability.

Acknowledgement: The authors would like to thank for the support from Taif University Researchers Supporting Project number (TURSP-2020/98), Taif University, Taif, Saudi Arabia.

Funding Statement: Taif University Researchers Supporting Project number (TURSP-2020/98), Taif University, Taif, Saudi Arabia.

Conflicts of Interest: We declare that we have no conflicts of interest to report regarding the present study.

References

[1] Z. Xiong, Y. Zhang, N. C. Luong, D. Niyato, P. Wang *et al.,* "The best of both worlds: A general architecture for data management in blockchain-enabled internet-of-things," *IEEE Network*, vol. 34, no. 1, pp. 166–173, 2020.

[2] R. K. Singh, R. Berkvens and M. Weyn, "Energy efficient wireless communication for IoT enabled greenhouses," in *Proc. 2020 Int. Conf. on Communications Systems and Networks, COMSNETS 2020*, Bangalore, India, pp. 885–887, 2020.

[3] I. Sittón-Candanedo, R. S. Alonso, Ó. García, A. B. Gil and S. Rodríguez-González, "A review on edge computing in smart energy by means of a systematic mapping study," *Electron*, vol. 9, no. 1, pp. 1–33, 2020.

[4] P. P. Ray, D. Dash and D. De, "Edge computing for internet of things: A survey, e-healthcare case study and future direction," *Journal of Network & Computer Application*, vol. 140, no. May, pp. 1–22, 2019.

[5] M. Bendechache, S. Svorobej, P. T. Endo and T. Lynn, "Simulating resource management across the cloud-to-thing continuum: A survey and future directions," *Future Internet*, vol. 12, no. 6, pp. 1–25, 2020.

[6] W. Liu, G. Huang, A. Zheng and J. Liu, "Research on the optimization of IIoT data processing latency," *Computer Communication*, vol. 151, no. January, pp. 290–298, 2020.

[7] H. M. Al-Kadhim and H. S. Al-Raweshidy, "Energy efficient and reliable transport of data in cloud-based IoT," *IEEE Access*, vol. 7, no. 5, pp. 64641–64650, 2019.

[8] S. Aheleroff, "IoT-Enabled smart appliances under industry 4.0: A case study," *Advance Engineering Informatics*, vol. 43, no. May 2019, pp. 101043, 2020.

[9] A. Banijamali, O. P. Pakanen, P. Kuvaja and M. Oivo, "Software architectures of the convergence of cloud computing and the internet of things: A systematic literature review," *Informationand Software Technology*, vol. 122, no. November 2018, pp. 106271, 2020.

[10] M. Krichen, S. Mechti, R. Alroobaea, E. Said, P. Singh *et al.,* "A formal testing model for operating room control system using internet of things," *Computers*," *Materials & Continua*, vol. 66, no. 3, pp. 2997–3011, 2021.

[11] M. Chen, W. Li, G. Fortino, Y. Hao, L. Hu *et al.,* "A dynamic service migration mechanism in edge cognitive computing," *ACM Transaction on Internet Technology*, vol. 19, no. 2, pp. 1–15, 2019.

[12] M. Masud, G. S. Gaba and K. Choudhary,"A robust and lightweight secure access scheme for cloud based E-healthcare services,". *Peer-to-Peer Networking and Applications*, vol. 14, pp. 3043–3057, 2021.

[13] M. H. Shahid, A. R. Hameed, S. ul Islam, H. A. Khattak, I. U. Din *et al.,* "Energy and delay efficient fog computing using caching mechanism," *Computer Communications*, vol. 154, no. October 2019, pp. 534–541, 2020.

[14] Z. Zhou, S. Yu, W. Chen and X. Chen, "CE-IoT: Cost-effective cloud-edge resource provisioning for heterogeneous IoT applications," *IEEE Internet Things Journal*, vol. 7, no. 9, pp. 8600–8614, 2020.

[15] A. Javed, J. Robert, K. Heljanko and K. Främling, "IoTEF: A federated edge-cloud architecture for fault-tolerant IoT applications," *Journal of Grid Computing*, vol. 18, no. 1, pp. 57–80, 2020.

[16] I. Khan, X. Tao, G. M. Shafiqur Rahman, W. U. Rehman and T. Salam, "Advanced energy-efficient computation offloading using deep reinforcement learning in MTC edge computing," *IEEE Access*, vol. 8, no. 4, pp. 82867–82875, 2020.

[17] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan *et al.,* "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 869–904, 2020.

[18] C. Jiang, "Energy aware edge computing: A survey," *Computer Communications*, vol. 151, no. 2, pp. 556–580, 2020.

[19] M. S. Mekala and P. Viswanathan, "A survey: Energy-efficient sensor and VM selection approaches in green computing for X-IoT applications," *International Journal of Computers and Applications*, vol. 42, no. 3, pp. 290–305, 2020.

[20] Y. Shi, K. Yang, T. Jiang, J. Zhang and K. B. Letaief, "Communication-efficient edge AI: Algorithms and systems," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 4, pp. 2167–2191, 2020.

[21] B. C. Kavitha, R. Vallikannu and K. S. Sankaran, "Delay-aware concurrent data management method for IoT collaborative mobile edge computing environment," *Microprocessors and Microsystems*, vol. 74, no. 4, pp. 103021, 2020.

[22] H. Zahmatkesh and F. Al-Turjman, "Fog computing for sustainable smart cities in the IoT era: Caching techniques and enabling technologies - an overview," *Sustainable Cities and Socities*, vol. 59, no. January, pp. 102139, 2020.

[23] H. Rimminen, J. Lindström, M. Linnavuo and R. Sepponen, "Detection of falls among the elderly by a floor sensor using the electric near field," *IEEE Transaction on Information Technology in Biomedeicine*, vol. 14, no. 6, pp. 1475–1476, 2010.

[24] I. Mohiuddin and A. Almogren, "Security challenges and strategies for the IoT in cloud computing," in *Proc. Int. Conf. on Information and Communication Systems, (ICICS 2020)*, Irbid, Jordan, pp. 367–372, 2020.

[25] J. Mades, G. Ebelt, B. Janjic, F. Lauer, C. C. Rheinlander *et al.,* "TLS-Level security for Low power industrial IoT network infrastructures," in *Proc. 2020 Design, Automation & Test in Europe Conf. and Exhibiton, DATE 2020*, Grenoble, France, pp. 1720–1721, 2020.

[26] R. Kiruthiga and D. Akila, "Prediction-based cost-efficient resource allocation scheme for Big data streams in cloud systems," in *Proc. Int. Conf. on Mathematical Modeling and Computational Science: ICMMCS*, Pattaya, Thailand, pp. 233–242, 2020.

[27] M. Masud, G. S. Gaba, K. Choudhary, M. S. Hossain, M. F. Alhamid and G. Muhammad, "Lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare," in *IEEE Internet of Things Journal*, DOI 10.1109/JIOT.2021.3080461.

[28] S. Jeyalaksshmi, M. S. Nidhya, G. Suseendran, S. Pal and D. Akila, "Developing mapping and allotment in volunteer cloud systems using reliability profile algorithms in a virtual machine," in *Proc. 2nd IEEE Int. Conf. on Computation, Automation and Knowledge Management*, Dubai, UAE, pp. 97–101, 2021.

[29] S. Pal, and P. K. Pattnaik, "A Simulation-based approach to optimize the execution time and minimization of average waiting time using queuing model in cloud computing environment," *International Journal of Electrical and Computer Engineering*, vol. 6, no. 2, pp. 743–750, April 2016.

[30] S. Pal, R. Kumar, L. H. Son, K. Saravanan, M. Abdel-Basset *et al.,* "Novel probabilistic resource migration algorithm for cross-cloud live migration of virtual machines in public cloud," *Journal of Supercomputing*, vol. 75, no. May, pp. 5848–5865, 2019.