Learning Automata and Agent based Architecture for Processing Data Locallyin Internet of Things

J. Mahalakshmi[a], P. Venkata Krishna[b]

[a] Department of Computer Science, Bharathiar University, Tamilnadu, India
[b] Senior Member IEEE, Department of Computing Science, Sri Padmavati Mahila Visvavidyalayam, Tirupati, India
[a] Mahalakshmi1203@gmail.com, [b] pvk@spmvv.ac.in

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: As the Internet of Things (IoT) facilitates the communication among the things/nodes in the network, it created its remarkable place in the current research. The sensors are the main nodes in IoT which generates data continuously and hence huge amount of data is generated in IoT. This paper presents a Learning Automata (LA)and agent based architecture for processing data locally inIoT. As the data being generated in IoT is huge, mostly the data will be offloaded to cloud or any servers which are capable of performing huge and complex computations. But in the case of proposed architecture, the data is made to process locally. The data processing is done by the master M-agent of a particular node and slave m-agent from other nodes. This agent is capable of migrating among different nodes in the network. Learning Automata is used to determine the number of replications of agentis to be done. The cost function is used to test the performance of proposed system. The proposed architecture is proved to be performing effectively when compared to the system without LA..

Keywords: Learning automata,M-Agent, IoT, MapReduce

1. Introduction

The things in the real life can be made connected with each other and enable them to communicate among them. This network of things is referred as Internet of Things (IoT). The nodes in the IoT are the sensor nodes. These nodes sense the environment and generates the information that is sensed.This generated information is very huge. This data need to be analysed to make proper use of this information. But it is very difficult to analyse such huge amount of data in IoT. To do this task efficiently, the resources need to be effectively used where the techniques of Big Data Analytics helps IoT to a large extent. Hence, Big Data Analytics [2] can be used for processing the huge data that is produced in IoT.To process the huge data generated by sensors, the data may need to be offloaded to the servers like which have high computational capacity. This process require additional energy. The power of the nodes is very less and bandwidth is limited in IoT. So, the life time can be enhanced if the offloading is made least [3-5].

The most complicated job in IoT is processing the data generated by nodes in the network. Some research is carried out in this field and proposed various procedures to perform computations to resolve this problem. MapReduce exemplary is developed using LISP by Google for processing data [6] and Yahoo used Hadoop to perform the same [7, 8]. The data is replicated to the servers with more computational power earlier to the data is processed. Large amount of stowage is required to store data and this process is less effective related to cost as the cost increases as more data is replicated or more offloading is performed from the nodes in the network to the cloud [9].

In view of resolving the high cost, a learning automata and agent based architecture for processing data locally within the nodes in IoT itself. The key objective of the proposed architecture, MapReduce is developed to be operated in the nodes itself, i.e, locally in the network only instead of cloud servers which is exterior of the network.

Introduction to Learning Automata

In, learning automata concept, there are two units referred as Learning Automaton (LA) and the Random Environment. LA selects the optimal value with the help of the feedback given by random environment and enhances its behaviour. The Figure. 1 shows the communication amid LA and the random environment.

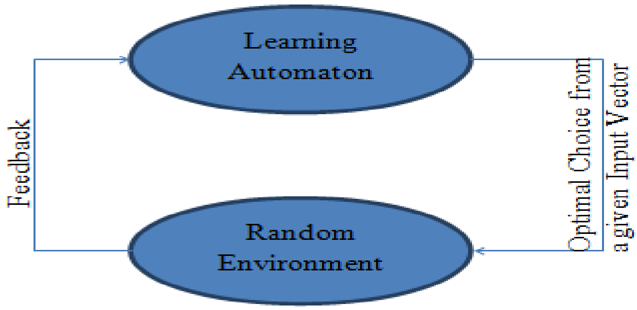

Figure. 1 Communication between Automaton and Environment

Initially, the LA will be choosing one of the inputs from the input vector and the predefined processing is carried out in the random environment. After the process is completed, the random environment provides feedback to LA. Based on this feedback the next input is chosen by LA from the input vector. This process is repeated until the LA is able to choose the optimal value from the input vector.

In general, Learning automaton can be represented using a Quintuple: {Φ, α, β, F(•,•), H(•,•)}, where, Φ represents the set of states, α is the input vector or set of input actions, β is the set of responses from the random environment and it contains more than two elements, F(•,•) is the function which defines the output given an input state and the response and H(•,•) is the function which defines the action to be performed for a given state and response.

The organization of the rest of the sections in this paper is: literature survey carried out in this related field is presented in Section 2, the proposed architecture is presented and discussed in section 3, performance of the proposed architecture is evaluated in section 4 and section 5 concludes the paper.

2. Related Work

In general, Big Data Analytics (BDA) is used to process huge amount of data. With the growth of BDA, problem of analysing huge data is resolved. The structure of MapReducepresented a novel methodology to process the data. This process is similar to divide and conquer approach. The complete data is divided into partitions and is distributed among various nodes for processing. Once the processing is completed, then the results are communicated back and final results are determined [10-11].

Commercialized file systems like Hadoop File System (HDFS), CPU memory, etc are being used to implement MapReduce procedure by many viable providers. Bandwidth requirement is more and the procedure is also more expensive in the case of these Hadoop File System, Google File System, etc. Hadoop structure is used in the case of embedded computers in [12]. As HDFS doesn't suits servers with more computational capacity, the exemplary do not function properly.

MapReduce Structure is explored by several investigators, scholars, etc. Shared storage is utilized to explore MapReduce structure in [13, 14].MapReduce procedure is iteratively performed in [8, 15, 16]. Transitory files are used store the information as an alternative to Key Value Stowage. Hyper Text Transfer Protocolis used to develop MISCO architecture to process data in mobiles [17]. MapReduce procedure is proposed in [18] for devices that are heterogeneous. The focus of this procedure is to perform computations on data in the mobiles. Further research in this area is carried out in [19-24].

Learning Automata and Agent based Architecture for Processing Data Locally in IoT

The proposed architecture helps in processing the data in the nodes of the network itself, i.e., locally only. The nodes in the network are presumed to be independent of each other and can compute or process the data individually. Such that the data can be dealt simultaneously by every node involved in the process. Also, in the proposed architecture, the nodes that are generating the data are responsible for processing also. Hence, the data need not be transmitted to external system like cloud or servers that have more processingpower. In the proposed architecture, the data need not be exchanged among nodes also for processing.

M-Agent

M-agent is hosted in the proposed architecture. M-agent can migrate from one node to another in the network. The role of M-agent is to process the data at nodes in the network, migrate to other nodes during processing when required and attain the output.The proposed architecture have separate procedures for performing mapping and reducing as similar to the traditional MapReduce structure in order to give provision to process the data.There are several benefits for M-Agents in comparison to the earlier. The M-Agent migrates from one to another node based on the result obtained at a particular node. The migration can be done to more nodes simultaneously. The network which is not stable requires this kind of procedure. M-Agent is essential to function in decentralized way for processing the data such that the scalability can be preserved. The M-Agents will be able to save and collect the information from the network nodes storage. Hence, there is no requirement of the knowledge about any file system for designing this system.

Architecture

There are 2 modules in the architecture. They are map and reduce.There are two M-agents in every node in the network. They are master M-Agent and slave M-Agent. Master M-agent is used to process the data in the node itself. Slave M-agent is used to process the data of other nodes. Hence, the slave agents will be able to migrate from node to another.

Map Module:

The data generated by a sensor node, N_S is divided into partitions. These partitions need to be processed by the master M-agent in the corresponding node. As the data generated is huge, this master M-agent will not be able to process the complete data. So, the node, N_S sends a request message to other nodes, N_{i-1}, where i is the number of nodes in the network for help. In return, the some nodes, $N_j$in the network, where $i \leq j$, which are ready to help the node for processing the data will allow its slave M-agent to migrate to the source node. The data partitions are processed by one slave M-agent. The data processing in this parallel way reduces the time taken to complete data process.

Reduce Module:

Once a particular slave M-agent completes it processing then it re-migrates to its home sensor node. The results obtained after processing of all data partitions, they can be aggregated to obtain the final result.

The number of slave M-agents that need to be a part of data processing is computed using learning automata. This is because, when more number of slave M-agents are involved in the data processing, then the cost and complexity is increased. When less number of slave M-agents are involved in the data processing, then time is increased. When more number of slave M-agents are involved in the data processing, then the cost is increased. Hence, the optimal number of slave M-agents are to be chosen. This optimal number is determined using learning automata concept.

Algorithm:

Input:

Number of Nodes – i

Source Node – N_S

Number of data partitions – D_p

Begin

N_S sends request to N_{i-1}

N_S receives response from N_j, where $j \leq i$

N_S selects N_k slave M-agents

For data partitions, D_p

Each S_{MNk} selects one D_{pi}

S_{MNk} starts data processing

If S_{MNk} completes data processing then

If any data partitions remaining then

Select another data partition

Else

Re-migrate to home node

End

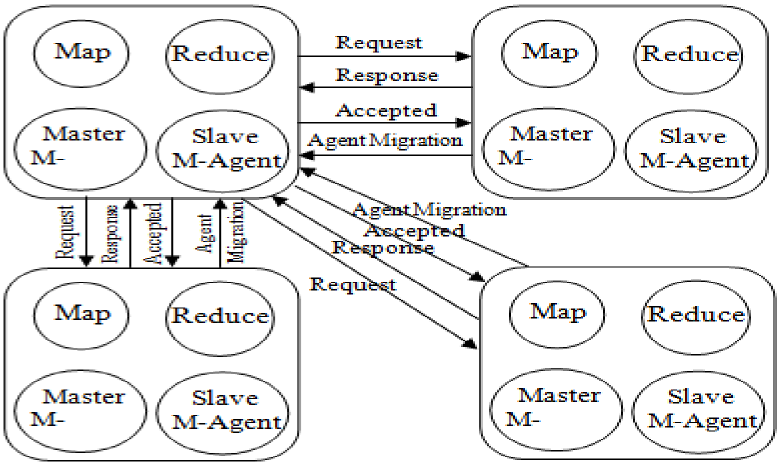

Figure 2. Agent based Architecture for Processing Data Locally in Internet of Things

The usage of M-agents is the key innovation in the proposed agent based architecture to process the data locally in the node itself in IoT. The M-agents selects another data partition once it completes the processing of one partition.

Determination of number of slave agents using Learning Automata

Assume that the possible values of number of copies of slave agents are considered as input actions to the learning automaton. They are represented as SA_1, SA_2, SA_3, …SA_n. The probability of all input actions is equal. Initially,one of the actions is selected randomly and later the probabilities of actions are modified according to the performance of the system in the environment. The probability of the selected action is increased (rewarded) or decreased (penalized) based on the performance of the system. In this paper, the cost function is considered to assess the behaviour of the system. So, if the cost is reduced for the selected action, then its corresponding probability is increased and the probability of the remaining actions is decreased accordingly. Similarly, if the cost is increased then the probability of the selected action is decreased and the probability of the remaining actions is increased accordingly. The probabilities of the input actions are represented as P_{SA1}, P_{SA2}, P_{SA3}, …P_{SAn}.

The sum of all probabilities is equal to 1 as shown in Eq. (1).

$$\sum_{i=1}^{n} P_{SAi} = 1 \quad (1)$$

The reward equations:

$$P_{SAi}(t+1) = (1-r)P_{SAi}(t) + r \quad (2)$$

$$P_{SAj}(t+1) = (1-r)P_{SAj}(t), \text{ where } i \neq j \quad (3)$$

The penalty equations:

$$P_{SAi}(t+1) = (1+p)P_{SAi}(t) - p \quad (4)$$

$$P_{SAj}(t+1) = (1+p)P_{SAj}(t), \text{ where } i \neq j \quad (5)$$

Where, r is the reward constant and p is the penalty constant

Learning Automata Algorithm:

Input:

Number of actions - n

Probability of Actions - P_{SA1}, P_{SA2}, P_{SA3}, …P_{SAn}

Output:

Optimal Action – SA_i

Begin

{

For each action

P_{SAi}= 1/n

i = random (n);

do

{

If ((ET <ET_{Thre}) and ($Cost_{dup}$<$Cost_{dup\_Thres}$) and ($Cost_{migr}$<$Cost_{migr\_Thres}$))

{

//reward the selected action

$P_{SAi}(t+1) = (1-r)P_{SAi}(t) + r$ $P_{SAj}(t+1) = (1-r)P_{SAj}(t)$, where $i \neq j$

}

Else

{

//penalize the selected action

$P_{SAi}(t+1) = (1+p)P_{SAi}(t) - p$ $P_{SAj}(t+1) = (1+p)P_{SAj}(t)$, where $i \neq j$

}

If $P_{SAi} \approx 1$ then

Optimal action = SAi

Else

Select an action with highest probability

} while (optimal action is not selected);

}

3. Results and Discussion

The simulation of the proposed system is done using java. The cost is the parameter used to evaluate the performance of the proposed system. The cost is based on the energy and time required for processing data. The number of nodes considered are 8, processor used are dual core.

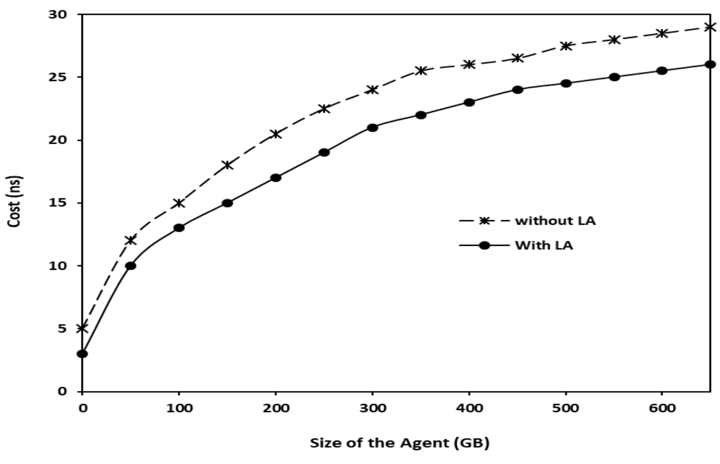

Figure 3: Cost function for Control Messages vs Size of Agent

The performance of the proposed system in terms of cost for control messages for varying size of agent is shown in Figure. 3. The performance of the proposed system, learning automata and agent based architecture for processing data locally in IoT is compared with the system without learning automata. It can be observed from the Figure. 3 that the system with LA performs better than the system without LA.

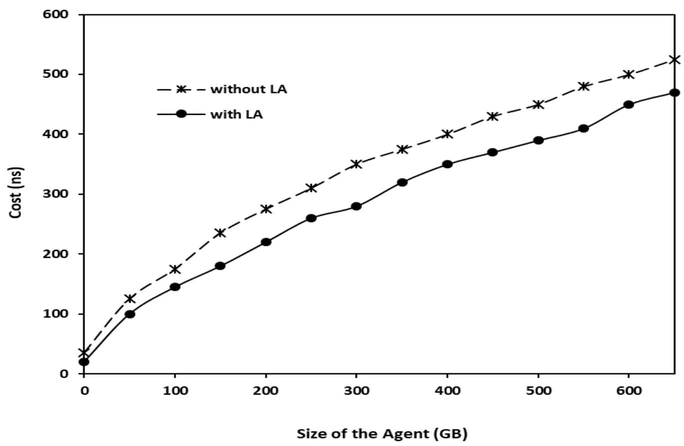

Figure 4: Cost Function for Agent migration

The performance of the proposed architecture in terms of cost for slave m-agent migrating from one node to another with varying size of the agent is shown in Figure 4. It can noticed that the performance of the system with LA is better than without LA. The cost for slave m-agent migrations includes the establishment of TCP connection, migration cost, and authorization validation. The proposed system evade unwanted data processing. Hence, it is more suitable for IoT applications.

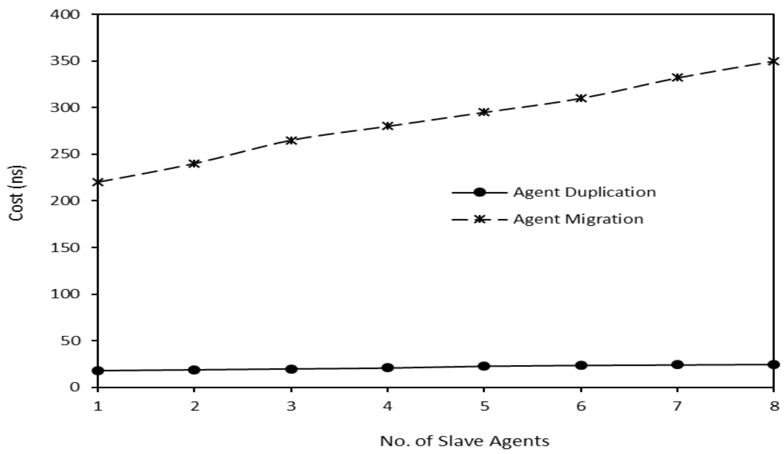

Figure 5: Cost vs No. of Slave Agents

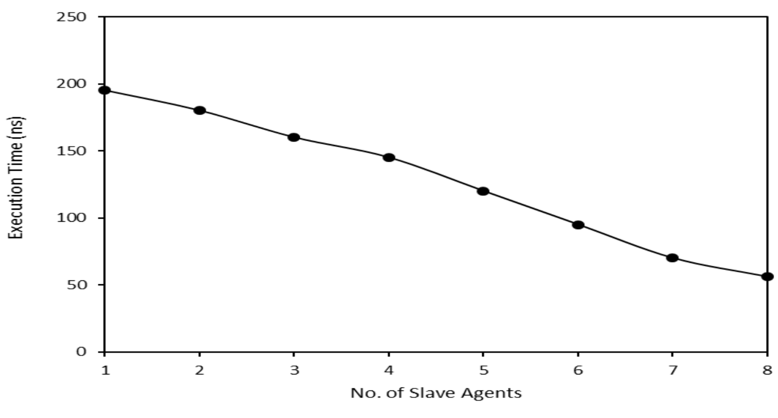

Figure 6: Execution Time vs No. of Slave Agents

Figure 5 depicts the behaviour of the proposed system with respect to the time taken to make number of slave agents and time taken to migrate to sensor nodes. It can be observed that the time taken to make copies and migration increases as the number of slave agents are increased. Figure 6 shows the time taken to execute a given task with varying number of slave agents. It is obvious that the time taken will be less when there are a greater number of slave agents but at the cost of time taken to make agent duplication and agent migration. Hence, it can be observed that there is trade-off between these two parameters which is considered using learning automata concept in this paper. The number of slave agents are chosen such that the time taken for duplication and migration is less with satisfactory execution time.

4. Conclusion

Learning Automata (LA) and agent based architecture for processing data locally in IoT environment is proposed in this paper. Local processing of data is done according to the proposed system without the requirement of offloading the data to cloud or servers with high processing power. There are Map and Reduce functions that are executed in every node where data processing need to be carried out. There are two M-agents: Master and slave. Slave agent can migrate from one node to another to perform data processing in other nodes. The number of slave agents to be utilized is determined using learning automata. The cost function for control messages, agent migration are used to evaluate the performance of the proposed system. The comparison of the system with LA and without LA is projected and proven that the system with LA performs well.

References

1. Atzori, L., Iera, A., &Morabito, G. (2010). The Internet of Things: A survey Computing. Networks.vol. 54, no. 15,pp. 2787–2805, Oct. 2010.
2. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A. H. (2011). Big data: The next frontier for innovation, competition (Vol. 5, No. 6). and productivity. Technical report, McKinsey Global Institute.
3. Alessandrelli, D., Petraccay, M., & Pagano, P. (2013, May). T-res: Enabling reconfigurable in-network processing in iot-based wsns. In 2013 IEEE International conference on distributed computing in sensor systems (pp. 337-344). IEEE.
4. Zhu, Q., Wang, R., Chen, Q., Liu, Y., & Qin, W. (2010, December). Iot gateway: Bridgingwireless sensor networks into internet of things. In 2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (pp. 347-352). IEEE.
5. Nagaraju, D., &Saritha, V. (2017). An evolutionary multi-objective approach for resource scheduling in mobile cloud computing. Int J IntellEngSyst, 10(1), 12-21.
6. Cai, H., Xu, B., Jiang, L., &Vasilakos, A. V. (2016). IoT-based big data storage systems in cloud computing: perspectives and challenges. IEEE Internet of Things Journal, 4(1), 75-87.
7. Bu, Y., Howe, B., Balazinska, M., & Ernst, M. D. (2010). HaLoop: Efficient iterative data processing on large clusters. Proceedings of the VLDB Endowment, 3(1-2), 285-296.
8. Bonomi, F., Milito, R., Zhu, J., &Addepalli, S. (2012, August). Fog computing and its role in the internet of things. In Proceedings of the first edition of the MCC workshop on Mobile cloud computing (pp. 13-16).
9. Dean, J., &Ghemawat, S. (2004). MapReduce: Simplified data processing on large clusters.in Proceedings of the 6th conferenceon Symposium on Opearting Systems Design and Implementation(OSDI'04), 2004.
10. Krishna, P. V., Misra, S., Nagaraju, D., Saritha, V., &Obaidat, M. S. (2016, July). Learning automata based decision making algorithm for task offloading in mobile cloud. In 2016 International Conference on Computer, Information and Telecommunication Systems (CITS) (pp. 1-6). IEEE.
11. Montori, F., Bedogni, L., &Bononi, L. (2016, September). On the integration of heterogeneous data sources for the collaborative Internet of Things. In 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI) (pp. 1-6). IEEE.
12. Jung, Y., Neill, R., &Carloni, L. P. (2012, December). A broadband embedded computing system for MapReduce utilizing Hadoop. In 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings (pp. 1-9). IEEE.
13. Jiang, W., Ravi, V. T., & Agrawal, G. (2010, May). A map-reduce system with an alternate api for multi-core environments. In 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (pp. 84-93). IEEE.
14. Talbot, J., Yoo, R. M., &Kozyrakis, C. (2011, June). Phoenix++ modular mapreduce for shared-memory systems. In Proceedings of the second international workshop on MapReduce and its applications (pp. 9-16).
15. Sehrish, S., Mackey, G., Wang, J., & Bent, J. (2010, June). Mrap: A novel mapreduce-based framework to support hpc analytics applications with access patterns. In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing (pp. 107-118).
16. Ekanayake, J., Li, H., Zhang, B., Gunarathne, T., Bae, S. H., Qiu, J., & Fox, G. (2010, June). Twister: a runtime for iterative mapreduce. In Proceedings of the 19th ACM international symposium on high performance distributed computing (pp. 810-818).
17. Dou, A., Kalogeraki, V., Gunopulos, D., Mielikainen, T., &Tuulos, V. H. (2010, June). Misco: a mapreduce framework for mobile systems. In Proceedings of the 3rd international conference on pervasive technologies related to assistive environments (pp. 1-8).

18. Elespuru, P. R., Shakya, S., & Mishra, S. (2009, November). MapReduce system over heterogeneous mobile devices. In IFIP International Workshop on Software Technolgies for Embedded and Ubiquitous Systems (pp. 168-179). Springer, Berlin, Heidelberg.
19. Lindqvist, J. (2018). New challenges to personal data processing agreements: is the GDPR fit to deal with contract, accountability and liability in a world of the Internet of Things?. International journal of law and information technology, 26(1), 45-63.
20. Kotenko, I. V., Saenko, I., &Branitskiy, A. (2018). Applying Big Data Processing and Machine Learning Methods for Mobile Internet of Things Security Monitoring. J. Internet Serv. Inf. Secur., 8(3), 54-63.
21. Babar, M., &Arif, F. (2019). Real-time data processing scheme using big data analytics in internet of things based smart transportation environment. Journal of Ambient Intelligence and Humanized Computing, 10(10), 4167-4177.
22. Babar, M., Arif, F., Jan, M. A., Tan, Z., & Khan, F. (2019). Urban data management system: Towards Big Data analytics for Internet of Things based smart urban environment using customized Hadoop. Future Generation Computer Systems, 96, 398-409.
23. Rashid, M., Singh, H., Goyal, V., Ahmad, N., &Mogla, N. (2020). Efficient Big Data-Based Storage and Processing Model in Internet of Things for Improving Accuracy Fault Detection in Industrial Processes. In Security and Privacy Issues in Sensor Networks and IoT (pp. 215-230). IGI Global.
24. Huang, M., Liu, A., Xiong, N. N., Wang, T., &Vasilakos, A. V. (2020). An effective service-oriented networking management architecture for 5G-enabled internet of things. Computer Networks, 107208.
25. Mahalakshmi, J., & Krishna, P. V. (2020). Localized Mobile Agent Framework for data processing on Internet of Things. arXiv preprint arXiv:2004.01477.