An IoT-Fog-Cloud Framework for Demand Side Management in Smart Grid

Yasin Boloorchi, Sadoon Azizi
Department of Computer Engineering and IT
University of Kurdistan, Sanandaj, Iran
Yasin.b76@gmail.com
s.azizi@uok.ac.ir

***Abstract*— As technology and the human population growth, the urge for power consumption grows with it. From daily and normal technologies to the latest and most advance of them, need electrical power; in such terms, there will be a need to manage this power distribution in an optimal way. To keep this need satisfy, a reliable framework to monitor and analyze the grid behavior required. The purpose of this paper is to present a framework architecture, its implementation, and evaluate its performance over a real-life dataset. This framework will use the latest technologies such as the Internet of Things (IoT), big data, fog and cloud computing field to decrease network traffic and also increase data analysis speed and illustrate real-time data analysis on a web-based dashboard about the current state of the power grid. These features also give administrators the ability to control the grid in crisis situation and clients to consume electric power more efficiently.**

Keywords—Internet of Things (IoT), fog computing, cloud computing, demand side management, smart grid.

I. Introduction

The Internet of Things (IoT) is one of the newest and most popular technologies in IT. This popularity is mostly because of its productivity and portability in so many aspects which made it viral around the globe. IoT is also distributed in a huge diversity, from wearable technology to smart houses and industrial sensors. This large variety opens a lot of opportunities for other applications and technology to co-work with each other. There are so many applications that with help of IoT, can make life today so much easier and also can make it more efficient like smart grid applications that are discussed in detail in [1].

As of today with the development of IoT devices, sensors, and smart metering of power consumption IoT made its way into smart grid architecture as argued deeply in [2], [3], [4]. Smart grid technology has begun to emerge, and governments start to notice its potential to manage the power grid properly. Dissimilar to the traditional power grid that only delivers electrical power to the consumers and makes it difficult for the grid to respond to changes in power demand, the smart grid presents a two-way transactional architecture that gives the grid the ability to respond to a different level of power demand.

There have been so many attempts to implement a smart grid but each of them had weaknesses in network traffic management and low speed in processing and analyzing as the heavy workload is on the cloud server. One of the newest cloud computing architecture is fog-computing, which allow smaller computers that are closer to clients, do some preprocessing and decrease workload from the cloud server. Fog computing architecture is introduced by Cisco corporation which is a networking hardware company that supplies most of the networking hardware in the world. This model uses edge devices to execute a valid amount of computation and carry enough storage for proceeding to requested process.

As of today many kinds of research have been done around smart grids and their functionality, but most of them are depending on cloud computing and held all of the process and data storage to cloud servers which made the process flow slow and non-responsive to fast changes in grid power demands. In this paper, we present a new aspect of the smart grid by using fog-cloud computation, big data, and Internet of Things (IoT) technologies in its architecture and implementation. With these changes in the smart grid, we can witness a great improvement in network traffic reduction and data process speed.

By merging these technologies that are available today, a framework can be created to manage the smart grid and to analyze its behavior in the most optimized way possible to not only prevent power loss in the entire system but to give consumers this option to consume electrical power efficiently. To achieve this goal different variety of technology has been used, such as IoT devices for measuring power consumption and socket programming for transferring gathered data to configured fog-node; in fog-node for preprocessing, *PySpark* has been used which is a big data processing technology; And lastly in cloud server other than *PySpark*, *Dash* has been used that is a web-based live dashboard to visualize analyzed data, collected from the smart grid. Also for keeping analyzed and raw data in a cloud server an HDFS (Hadoop Distributed File System) [5] used to save and protect data from system fault and missing data.

By implementing this framework in an urban area, government of that city can visibly see the benefits of its in economy and also in the environment as it is nowadays one of the biggest challenges that humanity is facing. From melting ice in the pole to acidic raining all over the world is a big alarm that reminds us to change our traditional fossil fuel that till now is our top resource for generating electrical power. With the enhancement in technology, we should consider solving this old problem that we caused to our environment.

This paper is structured as follows: Section II will review recent related researches, implementations, and publications on this topic. In Section III the proposed framework will be explained and how the framework implemented while Section IV will evaluate this framework performance with a real-world dataset that gathered from over five thousand smart meters in London. Finally, Section V entails the conclusion about this framework.

II. Related work

Throughout recent years, there have been several pieces of research and publications working on smart grids and their demand side management systems. For instance, in [6], a smart home system based on IoT with the help of web application has been introduced which will give real-time information data monitoring of energy consumption to its user; and also [7] which can be a good introduction to the smart grid, its demand side system, and its potentials and benefits.

Further, in a more detailed work like [8], a demand side management framework for smart cities has been discussed and presented which is the next step towards a reliable smart grid; in this framework, developers tried to visualize information from the measurement of several points of consumption that can help consumer reduce energy waste. On the other side, electric utilities can use a real-time measurement system to infer the energy consumption behavior, to find metering inconsistencies, optimize the system, or maintain the reliability of the supply system based on data information. However, there are some problems and defects in these works which others have worked on and have tried to discuss and solve, such as [9] and [10] which present practices to be used in the future of smart grid and Internet of Things with the help of smart sensor network, big data technology and the techniques used to manage big data generated by sensors and meters for application processing; which is a good idea except for the fact that smart grid is still incomplete in many ways. For instance, its demand side management security as argued in [11]. Smart grid also has so many more issues that should be discussed as it is cited in [12] and [13].

As conferred above, using bid-data technology is a good strategy and there are projects that do their work based on that, like the project discussed in [14]. But it would not be much of a help if we can't manage this huge amount of received data. As said in [15], Single cloud computing can hardly meet the requirements of large-scale real-time data processing. One of the best strategies for transferring data from houses to cloud servers in a smart grid is to use fog-cloud computing paradigm as argued in [16] so that we can transfer data more reliably and also use the edge nodes to reduce the amount of data which is being sent to cloud server as examined in [17] that worked on this challenge to handle the data flow that does not provide much information so that we can remain it unprocessed.

There are also other publications like [18] and [19] which discussed other dimensions of using the IoT and fog-cloud architecture to determine a dynamic price so that it benefits both the consumers and the producers of the energy. In [20], a particular topic has been worked on as peak load shifting. This idea will be discussed more extensively in this paper with the help of other technologies that are cited here.

In all above-mentioned studies, a multitude of researches looked into the smart grid and its demand side management in various viewpoints but none of them consider the responsibility of the system to fast changes in power demand of the grid that may happen in a fraction of minutes that would cause a critical situation in the grid. Besides that none of them worked on decreasing data transferring or data processing that rush to the central processor unit of the smart grid. This study considers all the issues above and present a framework and its architecture as a solution for the mentioned problems that smart grids are dealing with.

III. Proposed Framework

This framework's architecture generally is construct with three main layer; **client**, **fog-node** and **cloud server**. Each layer is built on top of the others and will receive data, do a process, send data, or all of them at one layer. Each layer will be explained one by one in the coming sections.

The suggested framework is capable of monitoring and analyzing customer's power usage, gather useful information, and save them in a safe environment. This framework is also able to set dynamic power prices for each district or even each client by analyzing the power usage of all clients and more consumption users. In more emergent events when the grid is on edge of consumption, the system can also cut the power of those users that consume more than their limit.

A. Client

The first layer that will describe in our framework is the client layer. Client is the lowest layer which its only job is to send measured power usage of the power counter to the fog-node which is fixed in the configuration file of the program. The client program can be written in any programming language that can send data through the network but in this implementation, we used python programming language for its simplicity and its usage in other places of this project. The client program can be run on any computer device that can connect to a network and communicate through it like raspberry pi or other small devices that can be used in houses with the lowest space and energy possible.

These small devices can be apart from the main counter or that the counter itself has the ability to do some small computation and sending data, which in that case it can be called smart counter. One of the most important and more difficult parts of this project is to set up these devices on each consumer house, office, market, etc. But hopefully, this job may be done by the power department of that city.

Data that will be sent from each smart counter from the consumer location will be only a few kilobytes that contain average or an instant of power consumption of that user with users identification, date, and time. This data will send to the nearest fog-node possible for more calculation and preprocessing.

B. Fog-node

One of the most complex and key layers is the fog-node, because of its important rule in this architecture. The fog-node job shown in Fig. 1, is separated into three different parts. The first part is to gather data from different clients that are around the fog-node with a small program that communicates by socket programming. The second part's job is to process this received data; this process is being done with *PySpark* python library and finally give this useful information to the last part which its job is to send this information to the cloud server.

As said in the previous paragraph, fog-node collects data from client programs running all around the chosen area by socket programming. For this, to work we used socket library which is a built-in library of python programming language. By using the pickle library, each data has converted to binary data that can send through the network to the server address which has been set in the configuration file of the program.

After gathering all data from the client's program, the next step would be preprocessing and filtering all of that collected data to be as abstract as possible so that network traffic and computation needed in the cloud server would be minimum. This job is done by *PySpark* library which does the processing by the architecture of *Apache Spark*. After processing has been done, a big amount of data has been embedded in a few kilobyte data. The Data which will be sent to a cloud server contains information about total usage of power in that specific area, number of active consumers, most and least consumers, and finally data and time of collected data.

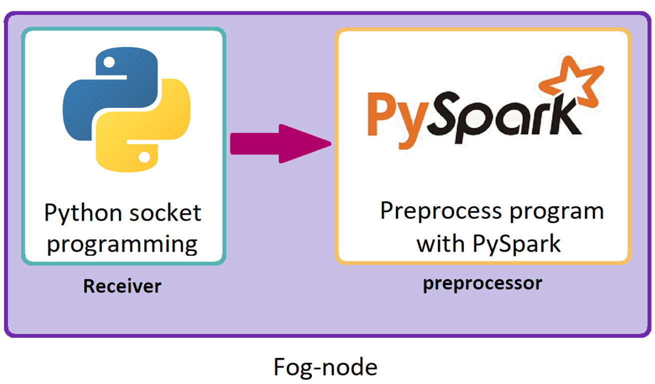


Fig. 1: Fog-node architecture.

C. Cloud

The last layer and the end of data flow is the cloud server which is responsible for three main tasks. The first is to gather information from all fog-nodes in the network then in the second step, it will analyze gathered information and transfer them into more abstract information and pass it to the third step which is showing the analyzed information in graphs for administrators. Parallel with step three, there is step four which its task is to save gathered information in a cloud server using Hadoop distributed file system (HDFS).

At first look, the cloud server program maybe seems the same as a fog-node program, but it is just the basis of these two programs that are similar. The receiver program task as shown in Fig. 2, is to collect data from fog-nodes all around the grid, but an Analysis program that uses *PySpark* is responsible for analyzing received data and it is completely different from the fog-node preprocessing unit.

Analyzed data will be sent to two different places; one copy will be sent to the dashboard program so that the administrator of the system can monitor the grid status and another copy will be saved in a Hadoop distributed file system (HDFS). The reason for using the Hadoop cluster is that the analyzed data will be immune against data loss besides other benefits of HDFS for future programs. In Fig. 3 a big picture of framework's architecture and co-operation between different layers has been illustrated

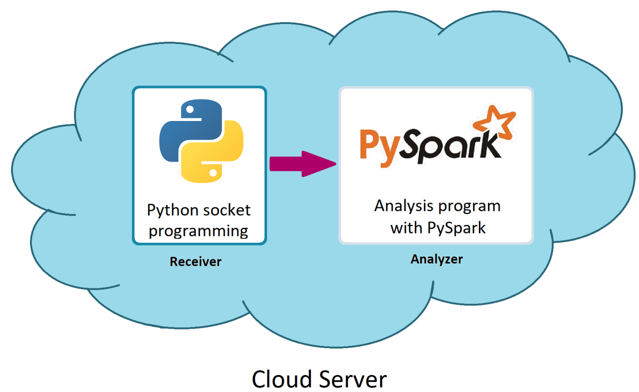


Fig. 2: Cloud server architecture.

In the end, what administrators will see is the framework's dashboard as an instant of it in Fig. 4 which will show them the useful information that they need. This information is divided into four different segments. The total consumption graph will show the total amount of power used in the grid by summation of all received power usage data in a specified time interval from all fog-nodes. The region consumption is a pie chart that illustrates the consumption of each area by comparing the power consumption of gathered data from fog-nodes.

Finally, there are two tables that list most and least consumption users; this list will be helpful for dynamic pricing for least consumption users or managing power grid in consumption piking by cutting power of most consumption users or area and bring balance to the power grid.

IV. Performance Evaluation

To evaluate the framework explained in the previous section, we used a dataset to simulate a scenario and inspect the framework's workflow. In this section, we explain this scenario and the dataset that we used in it.

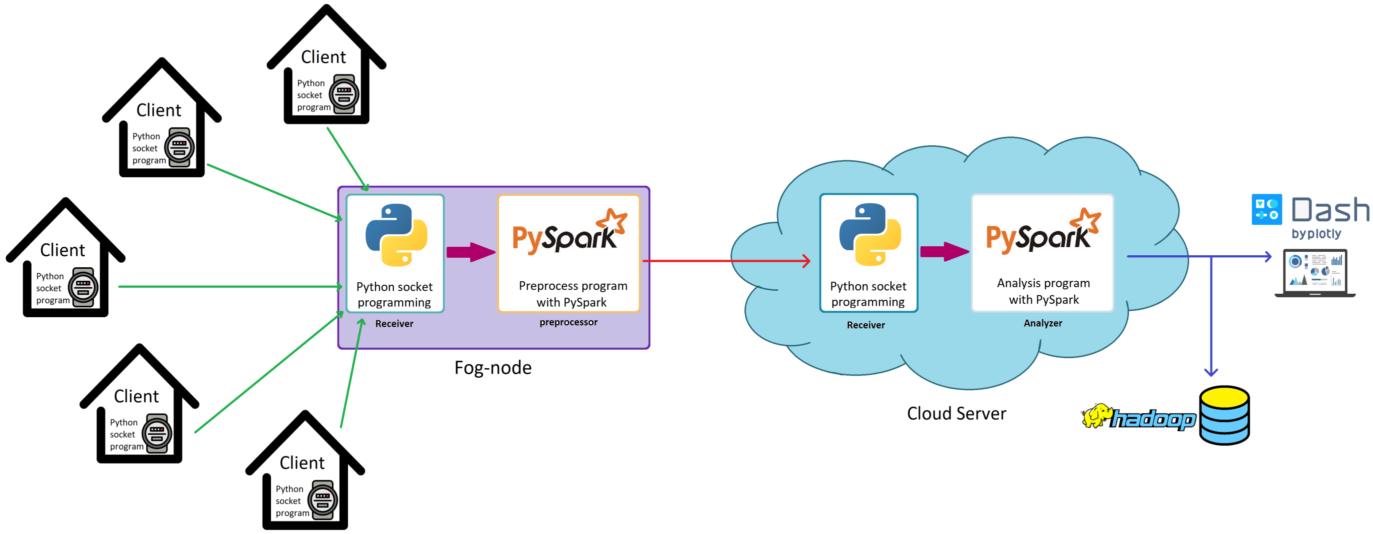
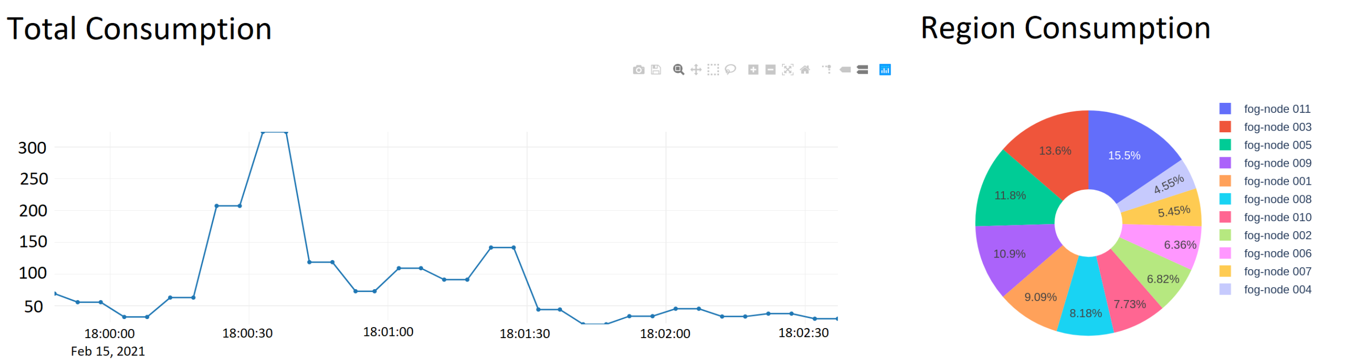


Fig. 3: Framework architecture.



Least Consumption Users

User ID	Usage	Time
39	0.009	2021 – 02 -15 18:02:21
204	0.003	2021 – 02 -15 18:02:03
741	0.001	2021 – 02 -15 18:02:14
3901	0.002	2021 – 02 -15 18:02:29

Most Consumption Users

User ID	Usage	Time
16	1.2	2021 – 02 -15 18:02:20
107	1.1	2021 – 02 -15 18:02:27
305	1.0	2021 – 02 -15 18:02:12

Fig. 4: Dashboard interface.

A. Dataset

As the British government decides to update the old energy system, they adopt smart meters as part of their plan. These changes made as the European union asked all of its members to upgrade their energy supply and tackle climate change. This dataset, which you will find in the London data store, contains the energy consumption readings for a sample of 5,567 London households that took part in the UK power networks led low carbon London project between November 2011 and February 2014. The data from the smart meters seem associated only with electrical consumption. The dataset contains energy consumption, in kWh (per half hour), unique household identifier, date and time, and CACI Acorn group.

B. Scenario

In this scenario we consider a city with 11 urban areas, each area has a fog-node that can cover approximately 500 smart meters. To reduce this huge amount of data that rush into the cloud server, we set a threshold for each fog-node to filter the received data. To find this threshold we analyze the consumption of a month of our users; after finding the threshold, we put it in use to see how much of use it will be for the rest of the data. Down below is the result of data reduction after setting a threshold to filter received data in fog-nodes.

To reduce data transmission to cloud server we fixed two thresholds; the first threshold is to reduct most consumption users who consume more than 1 kWh/hh (per half hour) and the second one is to filter least consumption users who consume less than 0.01 kWh/hh. With these thresholds, as shown in Fig. 5, only 5% of data will get through and the rest of them can be used to calculate the average consumption of that area. Also in the future, each user can have its own threshold for consumption to even reduce network traffic and transmitting data to fog-nodes. In the end, in Table I we compared data count and size before and after filtering them for a day, week, month, and finally for a year to show how much of difference this filtering will make.

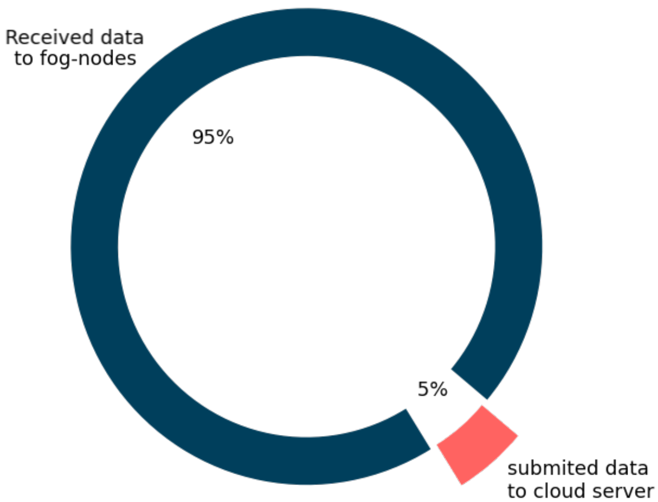


Fig. 5: Received data count to fog-node vs Submitted data to cloud server for the first month.

Transmitted data from clients to fog-nodes are clearly different from fog-nodes to the cloud server and the data format that used in this simulation to transmit data from clients to fog-nodes is as same as it is in the dataset; data are separated with a comma to six sections that each one contains information about that specific client such as client MAC address, date time, power usage and urban area name.

This format contains the necessary information for this framework to work. In the fog-nodes, this data will be pre-processed and most of it will filter out; remained data which consist of most and least consumption users in addition to the sum of total power consumption and time interval of that data will be sent to the cloud servers for more analysis.

95% data reduction will greatly affect network traffic and also cloud server processing. For instants, data analysis speed with spark will be increased due to its in-memory processing. Also, this data filtering would not affect system functionality or efficiency because each fog-node can save all received data from clients for future analysis.

TABLE I: Data reduction comparison.

	One Day	One Week	One Month	One Year
Received (Count)	266K	2.1M	7.4	21M
Received (Size)	17MB	139MB	493MB	1.4GB
submitted (Count)	12K	110K	379K	1M
submitted (Size)	792KB	7.26MB	25MB	6.6MB

V. Conclusion

In this article, we presented a framework architecture and its implementation for monitoring and analyzing electric power smart grid with help of big data frameworks and fog-node architecture to increase the performance and functionality of the smart grid.

In the end, we evaluate this framework performance by simulating a scenario with the use of a dataset with more than five thousand client data and shown that this system can reduce enough data transferring so that cloud server can process all gathered data sufficiently to visualize analyzed data on a real-time dashboard. With help of this framework, power consumption can become more efficient for clients, and also smart grid can be manageable for power grid administrators. With analyzed data and a list of least and most consumption users that will be accessible in a real-time web-based dashboard, in the pick of power consumption administrator can cut the power of those with the most power consumption, and as a result, we will witness less vast power cut.

References

[1] J. Liu, et al., "Applications of Internet of Things on smart grid in China," 13th International Conference on Advanced Communication Technology (ICACT2011), Gangwon, Korea (South), pp. 13-17, 2011.

[2] Z. M. Fadlullah, AS. K. Pathan, and K. Singh, "Smart Grid Internet of Things", Mobile Netw Appl, vol. 23, pp. 879-880, 2018.

[3] A. Ghasempour, "Internet of Things in Smart Grid: Architecture, Applications, Services, Key Technologies, and Challenges", Inventions, vol. 4, no. 1, p. 22, 2019.

[4] A. R. Al-Ali, R. Aburukba, "Role of Internet of Things in the Smart Grid Technology", Journal of Computer and Communications, vol. 03, pp. 229-233, 2015.

[5] K. Shvachko, et al., "The Hadoop Distributed File System," 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), Incline Village, NV, USA, pp. 1-10, 2010.

[6] R. Bhilare, S. Mali, "IoT based smart home with real time E-metering using E-controller," *2015 Annual IEEE India Conference (INDICON)*, New Delhi, India, pp. 1-6, 2015.

[7] P. Siano, "Demand response and smart grids—A survey", Renewable and Sustainable Energy Reviews, vol. 30, pp. 461-478, 2014.

[8] F. Mentzingen, et al., "Demand-Side Management Framework for Smart Cities," *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, Niteroi, Brazil, pp. 435-440, 2020.

[9] J. Manar, et al., "The Internet of Energy: Smart Sensor Networks and Big Data Management for Smart Grid", Procedia Computer Science, pp. 592-597, 2015.

[10] S. Chatterjee, "Demand Side Management in Smart Grid using Big Data Analytics", Utah State University, 2017.

[11] M. Babar, et al, "A Secured Demand Side Management Engine for Smart Societies using Industrial IoT and Big Data Analytics", IEEE Access, 2018.

[12] S. Oprea, et al., "Internet of Things, Challenges for Demand Side Management", Informatica Economica, Academy of Economic Studies - Bucharest, Romania, vol. 21(4), pp. 59-72, 2017.

[13] S. Ahmad, et al., “A Compendium of Performance Metrics, Pricing Schemes, Optimization Objectives, and Solution Methodologies of Demand Side Management for the Smart Grid,” *Energies*, vol. 11, no. 10, pp. 2801, 2018.
[14] S. A. Hashmi, C. F. Ali, and S. Zafar, ”Internet of things and cloud computing-based energy management system for demand side”, International Journal of Energy Research, vol. 45, Issue 1, pp. 100-1022, 2020.
[15] X. Pan, A. Jianga, and H. Wang, ”Edge-cloud computing application, architecture, and challenges in ubiquitous power Internet of Things demand response”, Journal of Renewable and Sustainable Energy, vol. 12, issue. 6, 2020.
[16] K. Shahryari, A. Anvari-Moghaddam, ”Demand Side Management Using the Internet of Energy Based on Fog and Cloud Computing,” 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Exeter, UK, 2017, pp. 931-936, 2017.
[17] S. V. Oprea, et al., ”Insights into demand-side management with big data analytics in electricity consumers’ behaviour”, Elsevier, Computers & Electrical Engineering, vol. 89, 2021.
[18] H. A. Khattak, et al., ”Dynamic pricing in industrial internet of things: Blockchain application for energy management in smart cities”, Journal of Information Security and Applications, vol 55, 2020.
[19] T. Chiu, et al., ”Optimized Day-Ahead Pricing With Renewable Energy Demand-Side Management for Smart Grids”, IEEE Internet of Things Journal, vol. 4, no. 2, pp. 374-383, 2017.
[20] C. Lin, et al., ”Peak Load Shifting in the Internet of Energy With Energy Trading Among End-Users”, IEEE Access, vol. 5, pp. 1967-1976, 2017.