DOI: https://doi.org/10.33103/uot.ijccce.21.4.5

An Enhanced Hybrid Edge-Cloud algorithm For Reducing IoT Service Delay

Rawaa Ammar Razooqi[1], Hassan Jaleel Hassan[2], Ghaida Muttashar Abdel Saheb[3]

[1,2,3]*Computer Engineering Department, University of Technology, Baghdad, Iraq*

[1]*ce.19.11@grad.uotechnology.edu.iq,* [2]*60012@uotechnology.edu.iq,* [3]*30834@uotechnology.edu.iq*

Abstract*—The Savvy Manufacturing plant could be a concept that communicates the conclusion goal of fabricating digitization. A Smart Factory, within the most common sense, profoundly digitized shop floor that collects and offers information persistently through associated computers, gadgets, and generations. In this work, the factory is represented by five types of sensors. The reading of the sensor values is sent to one of the Edge servers and cloud computing. One Edge server is selected based on calculating the time it takes for each server. The highest least time priority is chosen to receive the data coming from the sensors. This paper suggests a way to reduce the delay by using the edge server in addition to cloud computing by using methods that overcome any malfunction in one of the servers via another one that can work without the need to stop the factory systems.*

Index Terms*— Smart Factory, edge computing, delay in IoT.*

I. INTRODUCTION

Generally, Producers arranged in creating were all around making physical models, testing or testing with them, and after that going through an iterative refinement to be prepared until coming to a final thing. But, dynamically, cloud-hosted, computer-aided plan gadgets are streamlining product-development shapes, speeding improvement cycles, and enlivening time to promote. Cloud computing grants computer-aided plan, building, and manufacturing (CAD/CAE/CAM) to be performed on high-powered supercomputers, meaning architects and engineers not got to depend on effective machines in their office or per-seat computer program licenses to run ground-breaking reenactments or make data-intensive plans. Cloud computing gives the immense data capacity and virtualized computing control to engage this lively, carefully based design [1].

By centralizing and bringing all that data into the cloud, engineers can have the same plans accessible to them over any device [2].

The way we make and plan things is quickly changing with the onset of cloud computing. It isn't essentially an unused innovation, or maybe it has as it were as of late come to a level that can be received and utilized on a more stupendous scale. Cloud computing is permitting CAM and CAD to be done on farther supercomputers.

This implies that engineers and architects now not got to depend on a capable machine in their office in case they need to run groundbreaking reenactments or make serious plans. Putting away much of a CAD software's computing control off location permits for a way better productivity in information and control administration. The other advantage cloud computing brings is that all of the information are centralized within the cloud. This implies

Received 8/6/2021; Accepted 28l8l2021

Iraqi Journal of Computers, Communications, Control & Systems Engineering (IJCCCE), Vol. 21, No. 4, December 2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.5

that engineers can have the same plans accessible to them across any gadget. Ablility to make modifies and the records will be upgraded on each other machine you work on.

This can be the advantage of being associated with one central cloud. You'll be able to make alters and the records will be updated on each other machine you work on. In this particular, Enterprises and academics are paying more attention to the issue devoted to developing and extending new computing technologies Industry 4.0, smart factories, and so on are examples of technologies or frameworks and intelligent manufacturing [3].

Conventional generation strategies, on the other hand, are inefficient in terms of moderate computing speed on complex errands. not suitable for complex fabricating forms and analyzing expansive sums of information, particularly for Artificial intelligence (AI) tasks, As a result, an extended design on best of ordinary fabricating systems has been created by consolidating computing assets of different types.levels of computer assets, such as cloud and edge computing stages, which can meet fundamental prerequisites for the point of talking about the over issues. Be that as it may, cloud computing frameworks are ordinarily found distant from mechanical hardware, which increments inactivity and causes problems. There could be slack in information preparation, and we cannot ensure real-time execution. The real-time function of data flow in the entire industrial system, which has a direct impact on production efficiency and system normal operation, is crucial in a system. Industry 4.0 and smart factories [4].

As a result, in computing-intensive assignments, the overhead of information handling and transmission can be decreased (e.g., AI and deep learning tasks), this is another factor that must be considered to ensure real-time results. The latest latency-constraints approaches are primarily concerned with computing-task optimizations, data fusion, and network optimizations Decentralization of computing resources and simplification. Edge/fog computing frameworks, which are near to manufacturing equipment, reduce data transmission latency between servers and clients. Machines, for example, are ideal candidates for sustainable development. manufacturing to solve new problems.

II. RELATED WORKS

Controlling computing assets viably is required to guarantee low latency and nonstop yield in shrewd industrial facilities, in this manner to make strides in productivity, rise generation execution, and come about in budgetary gains. As a result, we are going to briefly summarize current endeavors in this section. real-time approaches, and other angles of edge computing are summarized in the Table 1 and the Table 2.

Received 8/6/2021; Accepted 28l8l2021

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.21.4.5

TABLE 1. LITERATURE REVIEW

Name of paper	Year	*Algorithm*	An explanation of the research
"Software-defined Cloud Manufacturing for Industry 4. 0 Lane Thames, Dirk Schaefer * a Tripwire," [5]	2016	*IoT , CBM, SPD*	centered on cloud-based applications, administrations, and stages that take advantage of the reflection that exists between fabricating equipment and cloud-based applications, administrations, and stages SDCM's objective is to quicken Cloud-Based Fabricating and other Industry 4.0 columns by expanding deftness, adaptability, and flexibility whereas lessening complexity challenges.
"Resource Allocation Strategy in Fog Computing Based on Priced Timed Petri Nets," [6]	2017	*PTPNs*	a fog computing asset assignment strategy based on priced timed Petri nets (PTPNs) that permits the client to select the leading assets on their claim. Since the real-time constraint was not taken under consideration within the calculations, these tests were unacceptable for fabricating. Moreover, the writing at times addresses the issue by consolidating the collaboration of different partners.
"On reducing IoT service delay via fog offloading," [7]	2018		the authors A new algorithm for reducing delay has been proposed. This paper looks at haze computing as a complement to cloud computing and a key component of the Internet of Things. They implemented a system for overseeing IOT demands within the mist layer and an explanation showing to identify benefit delay in IoT-fog-cloud scenarios. They illustrated how our mist offloading procedure, which minimizes delay, would advantage the IoT. Different numerical disclosures are included to back our affirmations by outlining how parameter changes can affect the typical advantage delay and how our information appears can be utilized to clarify the policy's productivity how to disentangle these issues, making a difference for perusers in their shop-floor decision-making. Other fog computing arrangements can be upheld by our explanatory show. When choosing whether or not to offload a mission, for illustration, Pj and Lij can be supplanted with the required conditions based on their arrangement. Extra measurements of IoT demands, such as the amount of information carried within the accommodation, may be considered in future investigations. Moreover, a strategy for powerfully altering mist hubs' edge (j 's) can be proposed. Besides, it would be valuable to see into the tradeoffs between delay, cost, and vitality in mist offloading plans.
"An approach for the secure management of hybrid cloud–edge environments," [8]	2018	*XMPP is used by different CLEVER middleware modules to communicate with one another.*	An instant-message communication solution is a valuable approach to developing an effective Cloud-to-Edge framework. A Message Oriented Middleware (MOM) based on an Instant Message Protocol (IMP) provides good performance in today's cloud environments, but it ignores security requirements... This paper aimed to close the gap using the CSA guidelines. We focus on the problems that need to be addressed to improve a Cloud-to-Edge system's data security, transparency, authenticity, and non-repudiation.
"An edge-fog-cloud architecture of streaming analytics for Internet of Things applications," [9]	2019	*clustering algorithm*	This study proposes a new architecture for analyzing IoT information streams and giving data-driven bits of knowledge in a shrewd stopping situation based on the edge-fog-cloud continuum. This paper presents preparatory discoveries from thinking that looked at IoT engineering that utilized edge, fog, and cloud instruments to back gushing analytics for a keen parking application.

Received 8/6/2021; Accepted 28l8l2021

Iraqi Journal of Computers, Communications, Control & Systems Engineering (IJCCCE), Vol. 21, No. 4, December 2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.5

TABLE 2. OTHER LITERATURE REVIEW

Name of paper	Year	*Algorithm*	An explanation of the research
"A hybrid computing solution and resource scheduling strategy for edge computing in smart manufacturing," [10]	2019	resource scheduling strategy	The proposed methodologies by the authors have demonstrated that computing administrations in smart manufacturing with edge computing provide excellent real-time, fulfillment, and energy consumption performance. Methodology for ensuring that the moo idleness requirement is met... To meet the real-time prerequisites of keen fabricating with edge computing back, make a half breed computing framework and a clever asset planning strategy.
"Implementation of workshop air pollution monitoring system based on wireless sensor network,"[11]	2019	MQTT protocol, gas sensors (CO, H2, NH3, Butane, Propane, Ethanol, and NO2), ARDUINO platform	The proposed system prototype includes a collection of gas sensors (CO, H2, NH3, Butane, Propane, Ethanol, and NO2) that are deployed on a Wireless Sensor Network stack and infrastructure. These sensors are calibrated using the most up-to-date calibration methods. They are controlled using a microcontroller based on the ARDUINO platform. Also, there is the primary server, which is installed on a Raspberry Pi 3 and holds the system's core database, which provides real-time management methods by monitoring air Sensors controlled by the ARDUINO platform communicate with the server using wireless technology (Wi-Fi), with the Message Queuing Telemetry Transport protocol used for communication (MQTT). When the carbon monoxide gas level exceeds the threshold, the system will send an alarm email to the civil defense department .llution in the form of figures and charts via the web interface, from this project, we can take the idea of creating a smart factory using sensors and Arduino to Monitor performance in the event of a fire or any malfunction.
"Cloud and edge computing for developing Smart Factory models using an iFogSim" [12]	2020	I FogSim wrapper	The literature presented focuses on developing a Smart Factory model based on Cloud and Edge computing, which is then used to develop a Transportation Management System (TMS) using an iFogSim wrapper. The use of the iFogSim Simulator to implement TMS is also discussed in the literature.
"Edge-Computing Architectures for Internet of Things Applications: A Survey" [13]	2020	(ECAs-IoT)	This study examines edge-computing architectures for IoT (ECAs-IoT) in-depth and categorizes them based on factors such as data placement, orchestration services, security, and big data. Furthermore, the paper examines each architecture in detail and compares them based on a variety of criteria the authors proposed a way to represent the smart factory.
"IoT Monitoring System Based on MQTT Publisher/Subscriber Protocol," [14]	2020	MQTT protocol	After reviewing previous work and conducting this research, the designers concluded that MQTT is a more suitable and effective protocol for IoT applications. The MQTT protocol can gain popularity in the coming years. On this work platform, they proposed a monitoring framework based on the MQTT internet routing protocol. The Raspberry Pi 3 served as a subscriber, while the ESP8266Node MCU served as a publisher. We can see how sensors can capture data as it is passed from one system to the next. Also, when compared to the temperature readings, the statistics read from the sensor are accurate. There is no difference in the readings, and if there is, it is very minor, ranging from 0.5 to -0.5. This system was a monitoring of the temperature and humidity regularly, and directly show the results on the website.
"IoT-fog-cloud based architecture for smart systems: Prototypes of autism and COVID-19 monitoring systems" [15]	2021	BPMN extension (business process model and notation)	To enable the Internet of Things (IoT)-aware business process (BP) modeling, the authors propose a business process model and notation (BPMN) extension in this article. After addressing state-of-the-art issues related to IoTaware architectures, this paper describes a new architecture for an IoTaware BP deployment into an HFC federation.

Received 8/6/2021; Accepted 28l8l2021

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.21.4.5

III. THE PROPOSED SYSTEM

This system proposes to represent the factory with a set of sensors linked to the Arduino. The Arduino receives the readings of the five sensors via the serial port and sends them to Raspberry Pi via Wi-Fi after they reache the Raspberry. Request the PHP page of three servers is the first, and the second the servers are locally called the edge server, as shown in *Fig.*1.

The third is called cloud computing, which is hosted on the host via the Internet. All servers carry the same information about the sensors to monitor the performance of the factory and this architecture is a solution to the problem of real-time requirements in the traditional way of the Internet, as shown in *Fig.2.* It does not depend entirely on cloud computing, which is prone to internet outages or any failure, but it also depends on two edge servers that are in the event of any malfunction in the first edge server, the second edge server can complete the work and monitor the factory without the need to stop types of sensors to represent the factory, as shown in *Fig.*3.

types of sensors to represent the factory:-

1. Gas Sensor (300 ppm to 10,000 ppm): Gas sensors (also known as gas finders) are electronic gadgets that distinguish and recognize diverse sorts of gasses.
2. Pressure Sensor (300 to 1100hPa): BMP180 is one of the sensors of the BMP XXX series.
3. Ultrasonic Sensor (2cm to 400cm): Ultrasonic Sensor HC-SR04 could be a sensor that can degree separate. It radiates an ultrasound at 40 000 Hz (40kHz) which travels through the discussion and if there's a protest or deterrent on its way It'll bounce back to the module
4. DS18B20 temperature sensor (-55°C ~ 125°C): The DS18B20 is one sort of temperature sensor and it supplies 9-bit to 12-bit readings of temperature
5. Pyroelectric / Passive Infra-Red sensor (up to 7 meters): PIR sensors permit you to sense movement, by and large utilized to identify whether a human has moved in or out of the sensors extend. They are little, reasonable, low-power, simple to utilize, and do not wear out.

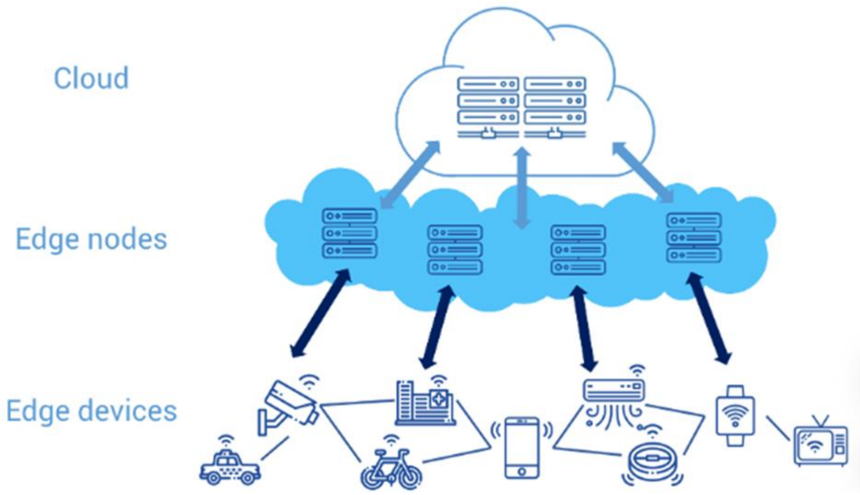

FIG. 1. EDGE SERVER

Received 8/6/2021; Accepted 28l8l2021

Iraqi Journal of Computers, Communications, Control & Systems Engineering (IJCCCE), Vol. 21, No. 4, December 2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.5

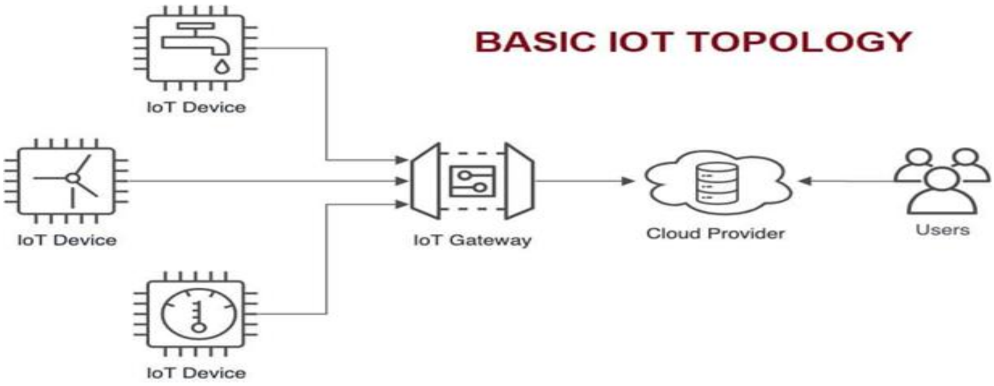

FIG. 2. IOT TOPOLOGY

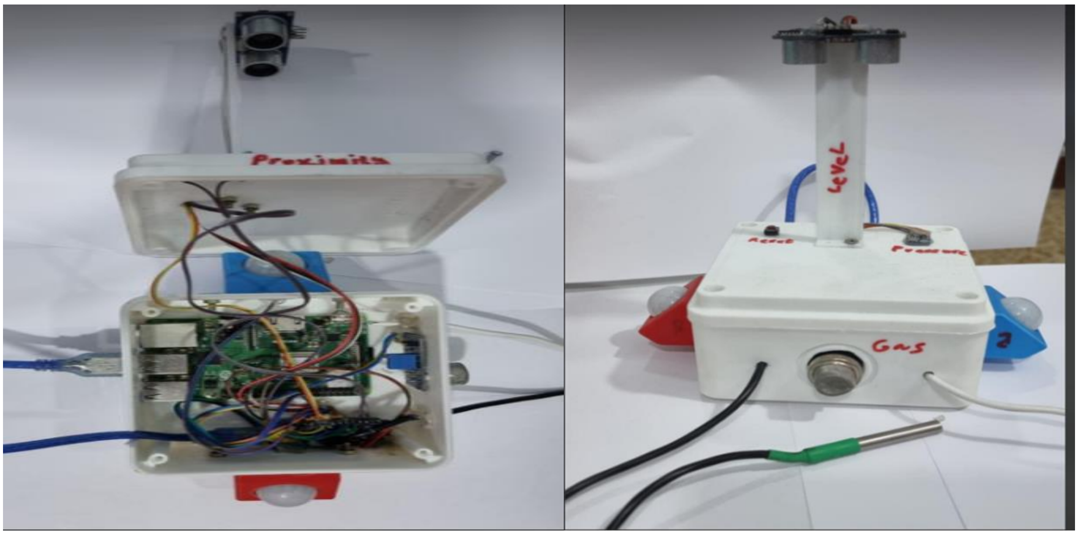

FIG. 3. HARDWARE OF FACTORY WITH THE SENSORS AND THE ARDUINO

The edge server is represented by a computer on which the Xampp program is installed, and it contains PHP and a database to store sensor values coming from the factory.

IV. RESULTS

A. Sensors Reading

Sensors reading values as input in the smart factory are sent from the Arduino to the Raspberry Pi via Wi-Fi and then sent from the Raspberry Pi to the servers by requesting each server PHP page.

A page consists of tables to store the reading of the sensors and statuses after a decision is made ,as shown in *Fig.4*. Also, it consists of a counter in the cloud to represent the graph.

Received 8/6/2021; Accepted 28l8l2021

 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

Iraqi Journal of Computers, Communications, Control & Systems Engineering (IJCCCE), Vol. 21, No. 4, December 2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.5

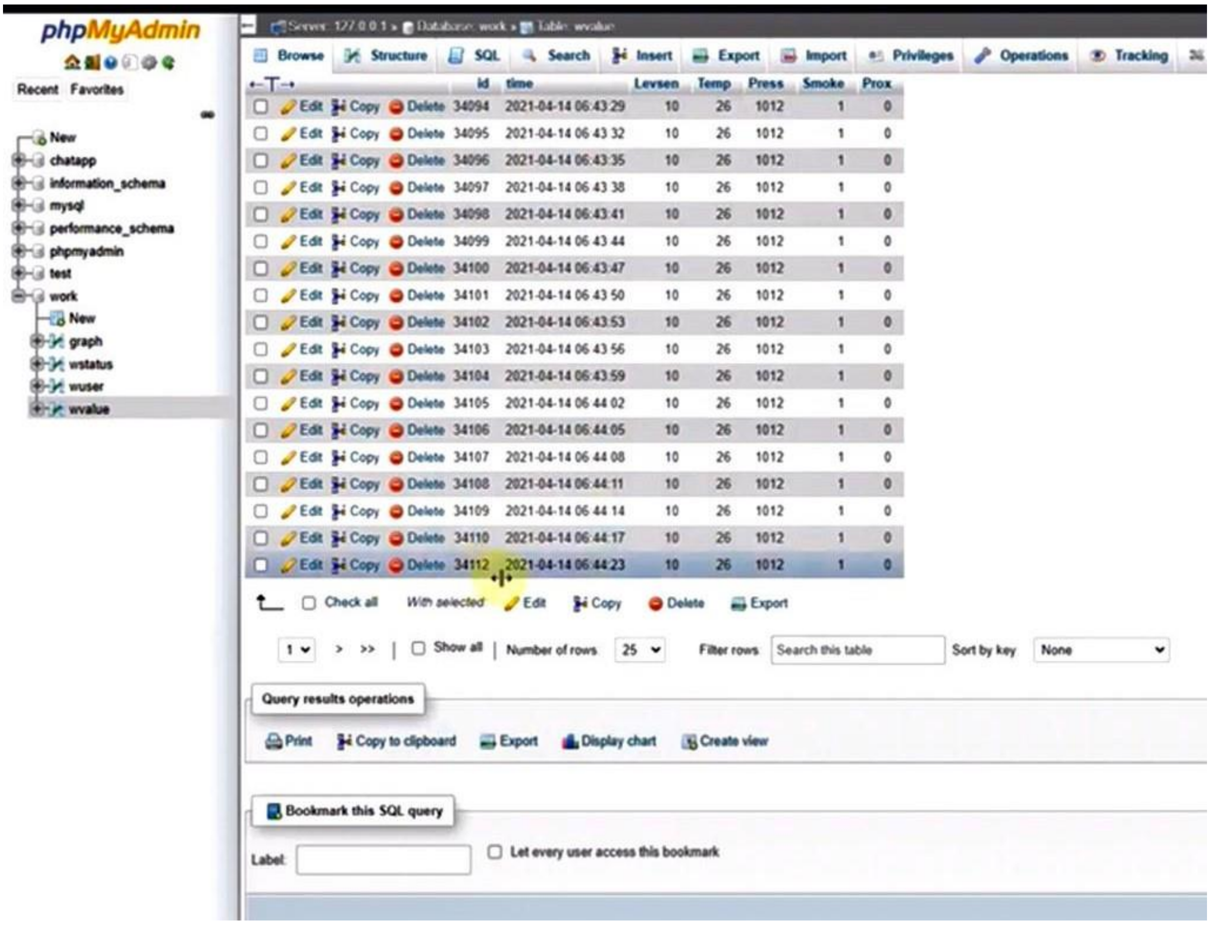

FIG. 4. SENSORS READING VALUES IN PHP

B. Cloud Computing

Steps work on Cloud (PHP, HTML):

1. These PHP program files upload to Cloud Storage.
2. addvalue.php and addstatus.php, this page uses PHP language to connect SQL database on Cloud and upload data (request on Raspberry in step 4).
3. connect.php have the code to connect to SQL.
4. delete.php all data on SQL (using to reset on Html page).
5. graph.php has a javascript graph with some HTML code to plot temp with pressure sensors (load this page in the index to show graph).
6. index.php this page showing to the user to show the value of all sensors on the cloud.
7. value.php connect to SQL database to get value (load by index.php).

Reading sensors appear in cloud computing with time, as shown in *Fig*. 5.

Received 8/6/2021; Accepted 28l8l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.5

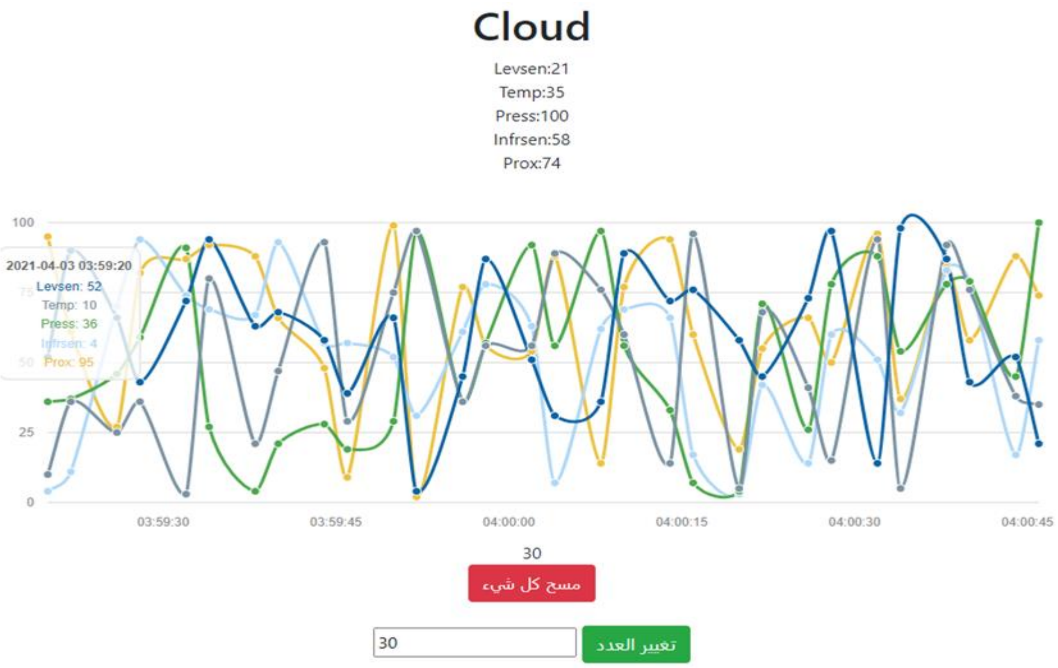

FIG. 5. INDEX PAGE WITH RANDOM VALUE OF SENSOR

C. Delay Reduced

In this work, the ping library in Python is used to calculate the total time required to send, receive and process one of the edge servers and compare it with the time in cloud computing, and it becomes clear the big difference between the two values, and by this, we prove that we have reached the requirements of real-time and we have reduced the delay, as shown in *Fig.* 6.

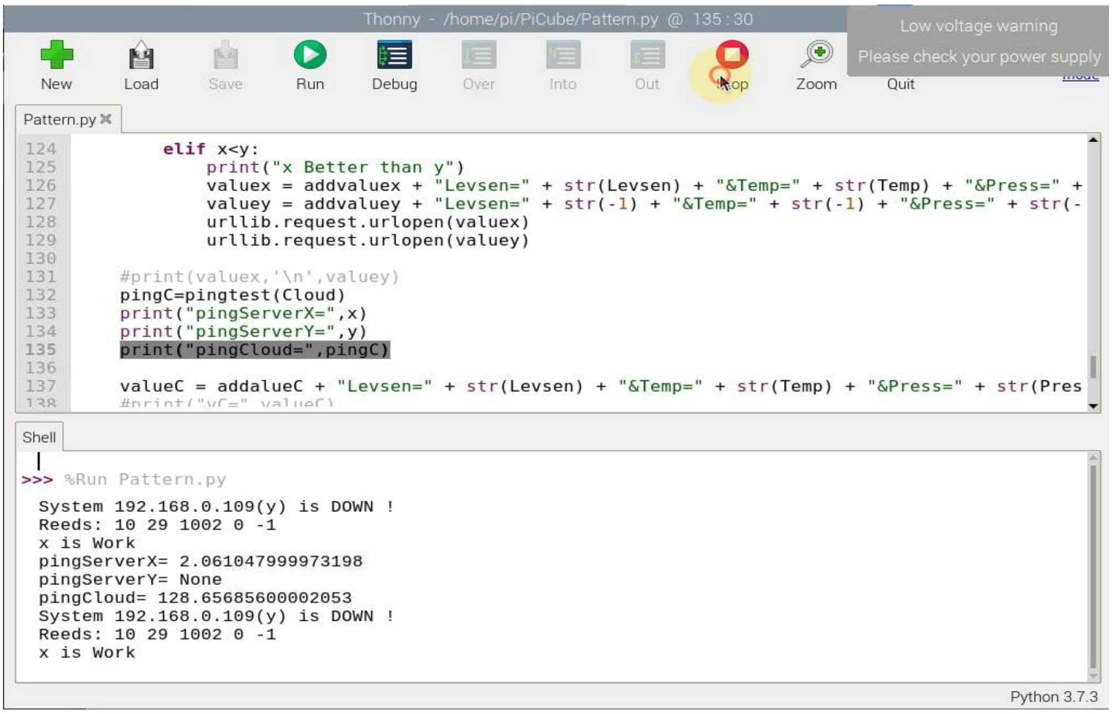

FIG. 6. PING VALUE OF EDGE SERVER AND CLOUD

Received 8/6/2021; Accepted 28l8l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.5

D. Check The Primary Key of The Identifier

Check the primary key of the identifier, (this key is not repeated) if the last data is repeated for a period of 5 minutes (default) do not decide, in the sense that the Raspberry does not send data. Read the decision from the second server.

Below are the steps of checking the primary key of the identifier (Id):-

1- The server reads the sensor values from the Raspberry Pi and checks the Id.

2- The Edge server receives a reading for the second time and checks if the Id finds himself not making any decisions and taking cases from the second server because this reading is considered a previous record and that Raspberry Pi does not send new reading sensors.

3- The process is repeated five times to check the Id and it keeps reading the decision from the second server, which means that the reading of the sensor values is repeated, and there is a problem with Raspberry Pi that does not send new readings.

4- After that, if the reading of the sensor values is sent to the server and it turns out that it has Id in any way different from the reading of the previous sensors, it takes the decision itself and sends an update with a decision to the second server. This means that the Raspberry Pi is working and sends new readings for the sensor values.

5- These steps reduce the delay in finding out if there is a problem with a factory or Raspberry Pi, as shown in *Fig.* 7.

As well as reducing the delay in the event of a failure in one of the servers, the second server can complete the work because it carries the same copy of the decision-making data to read the factory sensors.

```
Python 3.7.9 (bundled)
>>> %Run projslave.py
 id: 34137
 False
 Values: 5 1 34 34 7
 Status: 1 0 1 -1 1
 Second server Not Found! OR Your Connection Error
 id: 34137
 False
 Values: 5 1 34 34 7
 Status: 1 0 1 -1 1
 Second server Not Found! OR Your Connection Error
 id: 34137
 False
 Values: 5 1 34 34 7
 Status: 1 0 1 -1 1
 Second server Not Found! OR Your Connection Error
 id: 34137
 False
 Values: 5 1 34 34 7
 Status: 1 0 1 -1 1
 Second server Not Found! OR Your Connection Error
 id: 34137
 True
 Values: 5 1 34 34 7
 Status: 1 0 1 -1 1
```

FIG. 7. CHECK THE PRIMARY KEY OF THE IDENTIFIER

Received 8/6/2021; Accepted 28l8l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.5

V. CONCLUSION

Time is a fundamental factor, and it is considered an approach to improve the performance of factories. This paper focuses on reducing time and reaching real-time requirements. Edge servers are used to reduce the time because they are closer to the factory than cloud computing and can complete data processing without the need to stop the factory when any failure occurs. The factory is represented by five types of sensors connected to the Arduino, which sends the readings of a sensor to the Raspberry Pi. A better path is chosen from the edge servers that have the least time of the highest priority. It sends to servers by making requests to the PHP of each server to add value. the edge server will make sure that the Raspberry Pi sends new readings and makes a decision, or if it finds that the id values are the same and not changed five times, so it stops making any decision. it waits until receiving new reading sensors from the Raspberry Pi to process them. The decision appears on the Arduino board, for example, when the temperature rises, there is a fire in a factory, the red light turns on.

This work has come to significantly reduce the delay by using the Edge servers. When comparing the total time of the Edge servers with the total time of cloud computing, we notice a significant difference in the total time of sending and receiving reading sensor values. But this work focuses on reducing the delay and does not care about the principles of safety. This point is considered as disadvantage in the work that can be solved in the future by using encryption and security technology for the factory data.

REFERENCES

[1] K. Witkowski, “Internet of Things, Big Data, Industry 4.0 – Innovative Solutions in Logistics and Supply Chains Management,” Procedia Engineering, vol. 182, no.1, pp. 763–769, 2017.

[2] A. Ahmed, and M. Hussan, “Cloud computing study of security issues and research challenges. International Journal of Advanced Research in Computer Engineering & Technology, Vol.7, no.4, pp.2278-1323, 2018.

[3] Y. Bi, G. Han, C. Lin, Q. Deng, L. Guo, and F. Li, "Mobility Support for Fog Computing: An SDN Approach," IEEE Communications Magazine, vol. 56, no. 5, pp. 53-59, May 2018.

[4] E. Shellshear, R. Berlin, and J. S. Carlson, "Maximizing Smart Factory Systems by Incrementally Updating Point Clouds," IEEE Computer Graphics and Applications, vol. 35, no. 2, pp. 62-69, 2015.

[5] L. Thamesa, and D. Schaefer "Software-Defined Cloud Manufacturing for Industry 4. 0" vol. 52, Alpharetta: Inc, pp. 12–17, 2016.

[6] L. Ni, J. Zhang, C. Jiang, C. Yan, and K. Yu, "Resource Allocation Strategy in Fog Computing Based on Priced Timed Petri Nets," IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1216-1228, 2017.

[7] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, “On reducing IoT service delay via fog offloading,” IEEE Internet Things J., vol. 5, no. 2, pp. 998–1010, 2018.

[8] A. Celesti, M. Fazio, A. Galletta, L. Carnevale, J. Wan, and M. Villari, “An approach for the secure management of hybrid cloud–edge environments,” Future Gener. Comput. Syst., vol. 90, pp. 1–19, 2019.

[9] H. Cao and M. Wachowicz, “An edge-fog-cloud architecture of streaming analytics for Internet of Things applications,” Sensors (Basel), vol. 19, no. 16, p. 3594, 2019.

[10] X. Li, J. Wan, H.-N. Dai, M. Imran, M. Xia, and A. Celesti, “A hybrid computing solution and resource scheduling strategy for edge computing in smart manufacturing,” IEEE Trans. Industr. Inform., vol. 15, no. 7, pp. 4225–4234, 2019.

[11] H. Jaleel, A. Addul-Amer, and H. Addulkareem, “Implementation of workshop air pollution monitoring system based on wireless sensor network,” Iraqi Journal of Computer, Communication, Control and System Engineering, vol. 19, no. 4, pp. 61–71, 2019.

[12] D. Patel and S. Mohammed, "Cloud and edge computing for developing Smart Factory models using an iFogSim wrapper: Transportation Management System (TMS) case study,” 2020.

[13] S. Hamdan, M. Ayyash, and S. Almajali, “Edge-computing architectures for Internet of Things applications: A survey,” Sensors (Basel), vol. 20, no. 22, p. 6441, 2020.

[14] N. A-hussein, and A. Salman, “IoT Monitoring System Based on MQTT Publisher/Subscriber Protocol,” Iraqi Journal of Computer, Communication, Control and System Engineering, vol. 20, no. 3, pp. 75–83, Jul. 2020.

[15] A. Kallel, M. Rekik, and M. Khemakhem, “IoT- fog- cloud based architecture for smart systems: Prototypes of autism and COVID- 19 monitoring systems,” Softw. Pract. Exp., vol. 51, no. 1, pp. 91–116, 2021.

Received 8/6/2021; Accepted 28l8l2021