An Efficient Method for Implementing Applications of Smart Devices Based on Mobile Fog Processing in a Secure Environment

Huaibao Ding, Xiaomei Ding*, Fang Xia, Fei Zhou
School of Computer Engineering, Anhui Wenda University of Information Engineering, Hefei 231201, China

***Abstract*—Smart technology and the Internet of Things (IoT) are advancing and growing daily in the modern world. The demand for solutions to execute complex applications and protect user security and privacy increases as the number of smart devices in our surroundings increases. Mobile fog processing aids us in this situation by providing a fresh and effective method for running smart device applications in a safe setting. Due to the delay and high volume of requests, the centralized and traditional architecture of cloud processing cannot handle the high user demand and effectively implement delay-sensitive and real-time programs. To address these issues, a virtual mobile fog processing architecture that establishes a layer between mobile apps and the cloud layer was developed in this work. In this layer, storage, processing, and encrypted communication occur on separate nodes not connected to the cloud. These nodes are virtually implemented on a single server. An Android smart system-based augmented reality application that uses a marker to display dynamic 3D objects has been introduced. Its functioning has been assessed in both the cloud-based architecture and the suggested architecture in two 4G and telecom mobile internet networks. The evaluation findings demonstrate the suggested architecture's superior performance in both communication networks. The suggested mobile fog-based architecture makes use of the Internet of Telecommunications to create high-volume 3D models quickly and to the satisfaction of a real-time application. In addition to these accomplishments, the results demonstrate that the suggested architecture outperforms typical cloud-based architecture in terms of lowering overall energy consumption by up to 34%.**

***Keywords*—Cloud environment; IoT; real-time systems; smart devices; mobile fog; energy consumption**

I. INTRODUCTION

The administration of networks, storage, and computation has all faced numerous difficulties with the growth of the mobile Internet and the IoT [1]. High latency, a lack of storage space, scarce resources, round-the-clock services, and security are problems that can't be accurately solved with a cloud computing architecture [2]. A computer is a subset of computer resources, such as servers, storage, and processing space, used to centrally collect and process data from many clients at once [3]. These resources are accessible via customary means and are active in the network [4]. Network delays are caused by the allocation of resources among end users and clouds due to resource concentration in cloud spaces. Solving these issues is crucial for a number of delay-sensitive applications, including transport networks and augmented reality [5].

For this reason, the processing power in various Internet of Things networks has recently increased with the development of cloud and fog computing, and by shifting a portion of the processes to data centers in the fog, the amount of delay and limitations of the Internet of Things have been reduced and improved [6]. Mobile edge computing (MEC) and mobile cloud computing (MCC) are new technologies that have arisen as a result of the substantial role that smart mobile devices play in the expansive realm of the Internet of Things [7]. The use of cloud computing capabilities at the network's edge is a trait shared by all of these systems. When IBM and Siemens' Nokia Networks created an infrastructure that could run applications on a mobile base station in 1923, the term mobile edge computing was first used to describe the implementation of services at the network's edge [8].

European Telecommunications Standards Institute (ETSI) has created reference architecture since 2016 whose functional components support services like program execution at the radio network's edge [9]. Edge computing was developed to address issues with reaction speed, limited battery life, reduced bandwidth costs, and data security and privacy [10]. Additionally, processing and storage facilities at the edge can work together or independently. The control approach in the MEC design is dispersed hierarchically and centrally, in contrast to the centralized control of resources in the MCC [11]. Cloud computing (FC), a different horizontal architecture that distributes resources and services and offers computing, storage, and network control prior to reaching the cloud, was introduced at the system level for networks. When used with mobile devices, cloud computing technology offers quicker and higher-quality services [12]. In order to connect smart devices and provide services in a virtual form (VMFC), mobile processing can be used. All processing and operational services can be accessed through these virtual machines or a network of virtual machines [13]. In addition to providing MEC services, virtualization infrastructure can offer other related services, like the virtualization of software-defined network operations [14]. This article aims to offer a VMFC-based approach that can be utilized to construct a fog layer employing virtual nodes before the cloud layer. This layer's use for applications in smart devices, which have a dedicated node for each network of smart devices, including those for gaming, education, and health, is its unique feature. The author's contribution to this work can be summed up as follows:

- A virtual mobile fog processing architecture is offered to meet the increased user demand and ensure that real-time and delay-sensitive applications are run correctly.
- Two 4G mobile internet and telecommunication networks have been assessed for a pointer-based augmented reality application with dynamic 3D object presentation on Android smart systems.

The various sections of the article are further explained in the sections that follow: In Section II, a summary of the work completed in the MEC, MCC, and FC contexts is provided, and the benefits and drawbacks of each are compared to the proposed work. The suggested methods are provided in Section III. In Section IV, we demonstrate how this strategy is used using an example from an augmented reality application. In Section V, we discuss the suggested method and analyze the outcomes. A summary of this study's findings is provided in Section VI.

II. RELATED WORKS

Mobile phone networks make up a sizable portion of the Internet of Things. Because of their great computational capacity, they need to develop a second network layer to minimize the processing burden on the devices, reduce delay, and conserve bandwidth and energy when connected. The distant cloud server is displayed. Applications for active mobile phones in the Internet of Things networks, such as monitoring stations and smart houses, traffic data transfer, health data transmission in medical networks, augmented reality programs, large data processing in smart cities, and learning through mobile devices The mobile phone and the games utilize less computing and processing power and have shorter battery lives since the computing and processing load is offloaded to the cloud layer. The term "augmented reality" refers to a live physical perspective that instantaneously adds components to the real world of people, either directly or indirectly, and typically in contact with the user [15]. Programs for augmented reality that run on mobile devices need. The proper presentation of information requires high-speed data processing and minimal delay [16]. Cisco first developed cloud computing in 2022 [17], which is an expansion of the cloud computing platform that provides processing, storage, and network services between end devices and traditional cloud servers. In addition to network virtualization and traffic engineering via network performance virtualization, the researchers defined an architecture that supports some edge technologies, including ZigBee, Bluetooth, LoRa, and Wi-Fi [18]. Other investigations have been into edge processing's interoperability with the Internet of Things software. Fog nodes were created in this study as edge devices for various machine-to-machine services and machine-to-machine device management systems, such as road processing units in-vehicle networks. This approach did not take into account other factors like application migration or compatibility and was only effective for a small number of locations. This interpretation states that MEC offers a technological environment with cloud computing capabilities at the mobile network's edge. Utilizing cloud services at the edge of a network of smart devices has several benefits, including low latency, high bandwidth, access to radio network information, and local awareness. An edge scheduler that reduces the device's typical traffic delay is an example. It was demonstrated that the average energy cost in fog computing is 40.48% less than the model in a quantitative examination of energy consumption in a scenario where 25% of Internet of Things applications require real-time services and low latency [19]. Processing in the cloud is usual. In one study, they used the cloud and cloud computing to boost the processing capability of a network of wireless sensors for military applications and real-time execution to get around program limits and damage. A different work sample has offered a collection of applications that enable virtual machines to connect to cloud nodes and offer services. These applications enable virtual computers to access regional data, including sensor data, network statistics, etc. In a project, multiple researchers produced numerous copies of portable programs and several smart gadgets in the cloud [20]. They were moved from the machine to the duplicate. Consequently, a virtual copy of the gadget exists in case the actual one is destroyed or lost. This method also has the benefit of addressing the hardware constraints of smart devices. In a project, they demonstrated the viability of combining the 5G network platform with MEC and the accessibility of underused indoor and external spaces. Another project in the area of big data analysis utilized the convergence of MCC and Hadoop. The traffic congestion and time delay for their task were not resolved by paying attention to the amount of processing data in this strategy. Another piece of work on the subject of smart health involved gathering sensor data and sending it to a mobile device so that it could be processed and used to generate an alarm in an emergency [21]. None of the MCC or MFC services were used in this work. In one project, a mobile cloud server was used to develop an intelligent communication architecture for unmanned air transport networks. Their suggested design may maintain communication over a long period of time, even in the event of network failure, by improving network dependability and stability. Additionally, it can defeat DDoS, Sybil, and Wormhole attacks. In the MEC environment, a location-based augmented reality program has been built. Based on the tracking and identification of the user's position on the edge of an object, it is loaded. Utilizing this architecture significantly reduced mobile phones' computational load and energy usage [22]. Through the use of augmented reality technology, a different program on the topic of digital education was put into place. An image was recognized and exhibited in its static, three-dimensional surroundings. A cloud server was employed in this program to lessen the computational load and energy usage. The more straightforward steps of diagnostics were carried out on the mobile phone, while the more complex programs were moved to the cloud [23]. The program for augmented reality now loads frames more quickly, thanks to the evaluation of this effort. In another application dealing with augmented reality based on cloud processing, a photo of the book columns was taken, the features extracted, and sent to the cloud on the mobile phone [24]. Then, using the information obtained, a search was conducted among the database records in the cloud environment. Information about comparable works is gathered and sent back. In the studies and the use of cloud processing reduced the computational workload and energy consumption of the mobile phone, but the issue of high cloud latency and its distance from availability limited the use of MCC architecture for delay-sensitive applications [25-27]. It was immediately noted that this delay increased the mobile phone's energy use.

The delay issue has been largely resolved in [28], but if base station traffic volume rises, we will need to impose traffic limitations. The study proposed a successful approach to shorten the transfer time from users to distant clouds by utilizing the edge environment [29-30]. In this work, data and objects were entirely moved to the edge environment in order to conserve memory and processing in an electronic learning program based on augmented reality. The effectiveness of the study that was given demonstrated how the suggested strategy could significantly enhance memory storage performance. Edge nodes were created as road units in-vehicle networks in [31] to provide IoT services using OpenM2M in order to reduce network delay and traffic. For the purpose of reducing the typical scheduling and delay time, the authors of [32] investigated the multi-user computing partitioning problem (MCPP). To overcome the issue in delay-sensitive applications, they employed an offline exploratory technique called search in the model (SAPRL). Because of the work done in the MEC environment, all augmented reality applications based on 2D images, 3D edge detection, and sensor detection can be implemented. It is more suited for location-based augmented reality programs and is not extensively employed in health, education, or architecture networks. Due to the limited ability of control units and operator base stations to locate mobile phone devices, this feature is possible. Fog computing is actually a way of processing data from its point of generation to its point of storage. Only processing data close to its original location is attributed to edge computing. In addition to edge computing, fog computing refers to the network connections required to transport data from the edge to its terminus [33]. The use of mobile computing, particularly in mobile and network environments, offers users several advantages while lowering program running costs and conserving computer resources. In general, as mobile phone base stations improve and information technology and telecommunications networks converge, the use of mobile computing is a logical progression. Edge computing, where part of the data processing takes place at the network's edge rather than wholly in the cloud, was made possible by the Internet of Things' congestion and the constraints of cloud services. The problems of latency, mobile device battery life, bandwidth costs, security, and privacy can be resolved via edge computing. The comparison of the three described architectures is shown in Table I. The activities that have been completed have made use of MCC architecture to improve services for mobile devices, which are limited in their ability to run delay-sensitive programs due to the high delay of the cloud layer and the difficulty in accessing it. The fog support service has not been employed in many projects that have been simulated on smart devices. Some of the issues with MCC architecture, such as slowness, have been resolved through the use of MEC architecture. However, there is an issue of delay in the event of crowding when one is in an area of service stations because of the use of a high number of mobile phone devices. The inability to easily access the servers of mobile phone network carriers is another issue with deploying MEC. Applications cannot be uploaded to operators' servers without special authorization. One of our aims in writing this post is to offer programmers and consumers an approach that is simple to use and affordable. As part of the ongoing study, we will show an approach based on MEC and fog processing that is simple to use, and based on this architecture, we will use augmented reality technology to create a portion of a delay-sensitive application. The VMFC approach facilitates the study of local data by building a hierarchical infrastructure and moving global coordination and analysis to the cloud.

TABLE I. COMPARISON OF MCC, MEC, AND VMFC ARCHITECTURES IN TERMS OF FEATURES

Specification	MCC	MEC	VMFC
Expandability	high	Medium and limited	high
location	A higher level than the devices and out of reach	The edge of mobile devices	Between devices and the cloud
Hierarchical architecture	Two levels	Three levels	More than three levels
Server Hardware	Very large data centers	Small data centers	Small data centers
Control systems	Central	Distributed centralized hierarchy	Non-distributed hierarchy
amount of delay	high	medium	Low
They use it	Lots of devices and networks	The average number of devices and networks	The average number of devices and networks
How to connect	All internet networks	Base stations and operators	All internet networks
Interaction and cooperation between nodes	Does not exist	Does not exist	There is
Energy consumed by devices	medium	Low	Low
Implementation cost	medium	medium	low (virtualization capability)
User access	In a remote place and in case of heavy traffic, it isn't easy to access	Close and easy with the condition of being in the service area and having a service	Close and easy access
Access to programmers to develop all kinds of programs	Almost as easy with registration or fees	Easy if you have a license to access the servers located in the license base station	Easy with registration
Implementation of a delay-sensitive application such as augmented reality	Reduced computational load - high latency	Lack of support for the implementation of various augmented reality programs in case of medium delay congestion	Very low latency suitable for most augmented reality applications

III. PROBLEM IDENTIFICATION AND SUGGESTED SOLUTION

The problem's definition and the suggested model are covered in this section. You can upload computational jobs to the fog layer or run them locally on the device. The speed of the processing process can be greatly increased by uploading processing tasks to be completed at the network's edge. Fog nodes execute applications alongside centralized cloud servers, but they are not required to run for services since fog nodes can process applications independently and reply to users. Between the application and the cloud layer, this layer serves as a backup. The internal organization of this layer is utilized to speed up user access to the services they need. ETSI has provided the web foundation for MEC servers, templates, and software. We proposed a hierarchical architecture based on MFC in this article by researching and analyzing this architecture, as seen in Fig. 1. This design uses active networks of smart devices to deliver specific services to networks that are connected to it as well as the larger Internet of Things. The administration of network connectivity, processing, speed, storage, and control are a few examples of tasks that can be carried out at each fog node in this architecture. At a higher level, each mobile network is connected to a set that comprises one or more fog nodes. Nodes can communicate with one another wirelessly or over connected connections.

There are two options for loading programs in the cloud node: loading the complete executable program or just a portion of it. Depending on the application, each VMFC layer of the MFC layer is made up of a number of nodes. Nodes are interconnected for application management, data processing, storage, and security services. Each node is in charge of offering the designated services at both the level of that node and a higher level. Each VMFC layer collaborates with its neighboring layers as necessary. In the MCC layer, big data collection takes place at a higher level. This layer is in charge of overseeing and supporting MFC, which is a lower-level layer. Additionally, this layer is in charge of documenting, confirming, and securing access for VMFC nodes to connect with one another as needed. This layer houses data centers and big data storage.

The MFC layer and the MCC layer communicate with each other over the Internet. Active technologies like WLAN, ZigBee, 4G LoRa, and 5G are used to connect applications on smart devices to the cloud layer so they can receive service and storage in a secure environment. Frameworks like: NET and Java are frequently used to create applications since they make the process of transferring code easier.

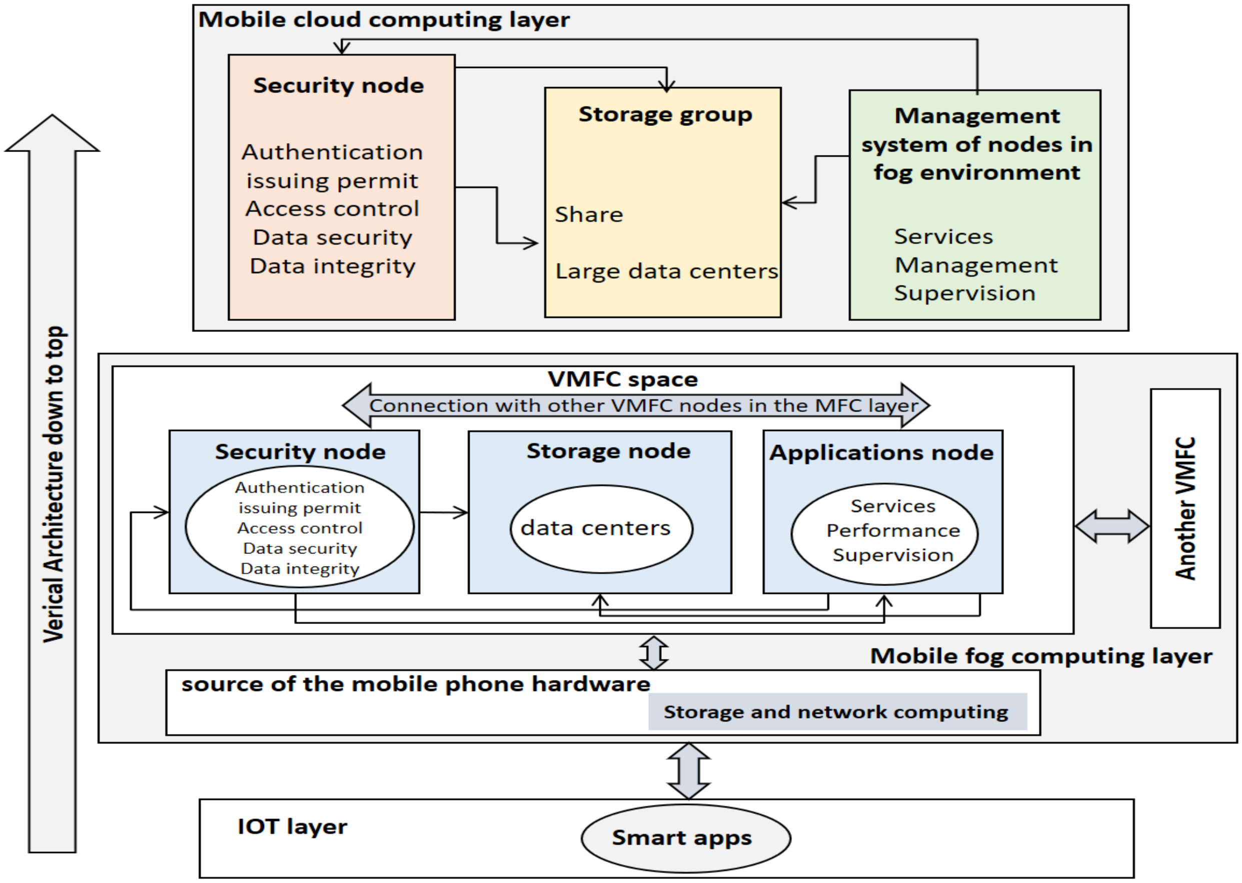


Fig. 1. Block diagram of the proposed model.

IV. IMPLEMENTATION OF THE ARCHITECTURE

This section explains how to put the suggested method into practice. In this system, two fog nodes were employed for program execution, processing, and authentication. The fog layer is made up of two distinct nodes: one is used for security, and the other is used to store and process the application program. The two nodes here are linked. These two nodes are implemented and set up using two virtual computers. The data storage node is not taken into account independently [34]. Every node used for processing has a database for storing the sent and necessary data related to other data. The Java framework has been used to implement the application program. Fig. 2 demonstrates how to use two nodes for the application and security to create the application based on the suggested way described in Section III. Each node contains the node database, which is not a separate entity. On the needed server, each user registers their information and builds a profile. The executable application responds to the user's request for some space. The user is allotted a specific amount of space for each component of their profile, which is separated into many sections. An individual identification number is assigned at random to each segment [35]. The address of the user's profile and the ID of each section written in a piece of code is used to direct the program in order to log in and run the program uploaded in the virtual space. The security node receives the data entered in the processing and storage node and enters it into the profile for registration and verification. The initial step in accessing the cloud server to carry out processing is user authentication. A software ID is generated for the user when they register their information.

Each piece of software is identified by a special key called software ID. With this special key, all requests are signed, and it is also used to verify signatures. The procedure for getting the token in a mobile edge scenario is the same as this one.

The authentication header bears the signature. The time tag and request identification features in this header guard against broadcast and propagation attacks. The signature also changes if the person's ID changes [36]. In order to verify and register the user's codes in his profile database, this signature must be written in a section of the application's Android code that is utilized while loading program data and performing the necessary cloud-based operations. Using a browser to connect to the Internet, the user can access his profile and read and update his data from any location. The application uses an image-based augmented reality program to run dynamic 3D objects. Each 3D object depends on a pointer, which means that it can only be seen by scanning the pointer. The second fog node is utilized in this architecture to store and process the pictures needed to execute the augmented reality software. The outcome of the image processing is then compared to the user's pointer, and the appropriate three-dimensional image is presented. Everywhere the user's marker is, a 3D image of it can be seen. Visual indicators are the type of indicators employed in this project. Between ID-based markers and markerless technology, there is an image marker. With this technique, any image can be used as a target by being printed out; the target should be a colored border against a light background. The image's border need not be black; it might be a dark hue.

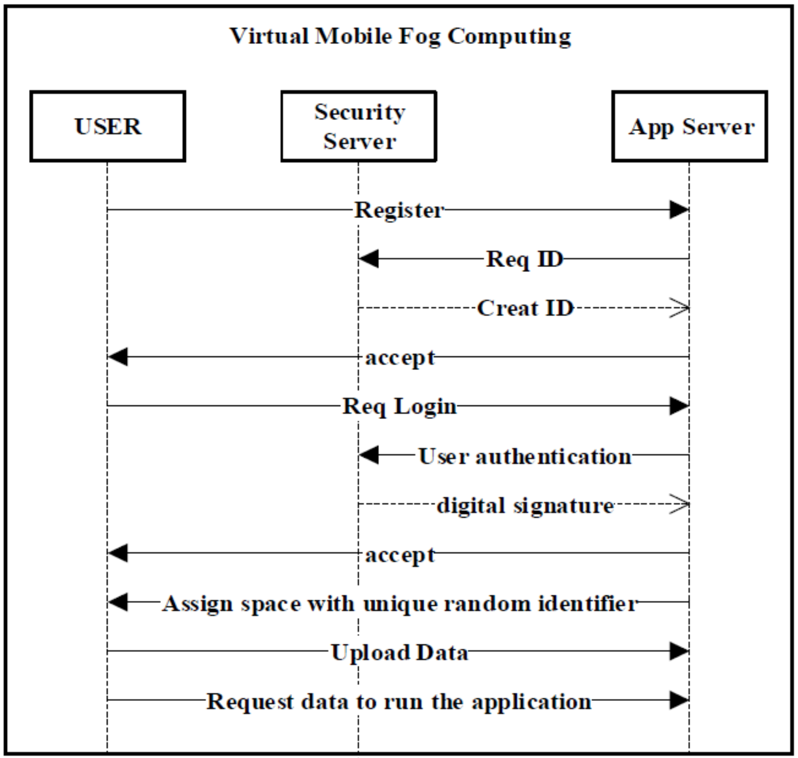


Fig. 2. Data recording and transfer flow in the proposed method.

Developers can use images and track them in pointer-based programs, depending on the requirements of the program. A low-poly model is a low-poly model, and a high-resolution model is a high-poly model. These classifications are based on the tool and how the models are applied. Models of lower resolution, or low-poly models, are frequently employed in real-time systems to boost program efficiency and speed up execution. In a 3D model, a polygon is defined by a number of other polygons. This unit is typically represented by a collection of triangles. The quantity of triangles produced on the 3D model determines how clear the implemented models are. The application node processes a portion of the software in such a way that the user can load a sample of the marker in addition to the desired 3D model. According to each marker with a 3D object and its properties, the.xml file is saved and loaded. The marker that identifies the mobile phone camera and its application is delivered to the fog node each time, the program is run and is compared to the markers in the processing node. If they match, the matching 3D model is presented. When matching and running, 3D models are merely loaded and presented from the node and are not saved on the mobile device. An.xml file with the 3D object's properties is loaded into the user's profile.

V. DISCUSSION

In this section, we describe the assessment of the developed technique in two scenarios: local execution while using the suggested VMFC approaches and using a mobile phone. According to the volume difference, we look at two frame execution rate parameters in each 3D model, and then we compare how long it takes to load data from VMFC and a ready cloud service. To upload data, we took advantage of the Junaio cloud storage service. With the help of the augmented reality browser Junaio, users may build and use augmented reality programs on a variety of platforms [37]. You must first create a channel in this browser in order to upload the augmented reality programs you have generated and assigned to each channel. Dynamic 3D models have restrictions on mobile phones since they require a lot of processing speed to operate and a lot of storage space as the number and volume of data increase. These models, which can move and animate, are made up of a number of layers, each of which mimics a movement. A frame or polygon is the name given to each of these layers. To show a specific movement that has been applied to the object, it is imperative to execute these polygons one after the other. On the other hand, augmented reality programs are real-time programs, and one of the crucial aspects in this sector is the quick execution rate of an event. Six 3D objects with extensions.FBX,.md2,.mm3d, and.ms3d are utilized in this software, ranging in size from 180 to 450 KB. As was already noted, the number of created polygons and the speed at which they are executed determine the resolution of models. In this case, triangles, each 80KB, comprise around 10,000 polygons. A bandwidth of 29 MB/s is required to implement this flow. In frames per second, 3D models are executed. This unit specifies the vertex of the polygons transferred from memory to the screen. If a program loads less than 20 frames per second while running, it indicates that the program is not operating smoothly and rapidly. It is reasonable for programs to transfer at least 20 frames per second while running. Depending on the size of the model, the mobile phone uses a specific amount of device energy to load each of the models and transport them on the screen. Data is loaded into the ES FileExplorer environment and then transferred for presentation to the main program. Eq. (1) is used to compare the difference in program execution time between VMFC and cloud services:

$$T = \frac{R}{S} \times (T_r + T_h + T_s) \quad (1)$$

where, R is the anticipated request ratio, S is the total number of requests that the machine can handle concurrently, T_r is the amount of time it takes for data to travel from the user's mobile device to the virtual machine, T_h is the amount of time it takes for the virtual machine to respond to the request, and T_s is the amount of time it takes for data to travel from the virtual machine to the user's mobile device. We assumed the value for both services to be the same and constant in this R equation. The amount of data being transferred and the network conditions—which are typically poor for mobile networks when handling heavy data loads—determine how long it will take to complete. The user response waiting period has been left out of this calculation because it is not crucial to resource allocation. Six different data sets with varying volumes were examined using the program's implementation on two separate platforms, including WLAN and 4G mobile internet. The software was implemented in three parts based on a cloud server, mobile device, and VMFC and the evaluation metrics were looked into and examined. Internet telephony was utilized to compare the frame transfer rate of 3D models in VMFC. In Fig. 3, the speed and quantity of transitions of the 3D model polygons corresponding to the marker in the minimal normal state are displayed after the mobile phone camera recognizes and processes a marker. For the 3D models incorporated into the program, this value is computed. Fig. 4 illustrates the frame transmission rate to the screen from the mobile phone and the VMFC in contrast to the minimum normal speed. The application takes into account a certain number of polygons for processing on the model and transfers them for display based on the volume of the models being utilized. The 3D model and its dynamic movements must be executed and displayed at a minimum speed of 20 frames per second. For low-volume models, the transfer speed of the frames processed in the application in the mode where the model is loaded from the mobile phone memory is acceptable, and it almost exactly matches the transfer speed from the VMFC to the mobile phone's temporary memory. However, compared to a mobile phone, the transfer speed from a VMFC is faster and offers greater performance when the model volume and count are both raised.

The FBX model has a larger volume than other models, and its memory consumption is also higher. The speed of frame transfer in this model and larger model is higher than loading in a mobile phone. Loading 3D models in VMFC in this program has saved about 1 gigabyte in mobile phone memory. Data transfer from the mobile phone to the Ebro VMFC server was checked in two Internet communication networks. The present research compares the transmission speed values frames in mobile application execution and transmission speed after using VMFC. Paired t-analysis was used after and before using VMFC to determine if there was a change in the transmission speed of frames.

Table II shows the details of this statistical survey. As can be seen in Table II, the rate of frame transfer before using VMFC was 84.250, and after using VMFC, it increased to 83.333, and this increase was statistically significant (p=0 /032) is significant.

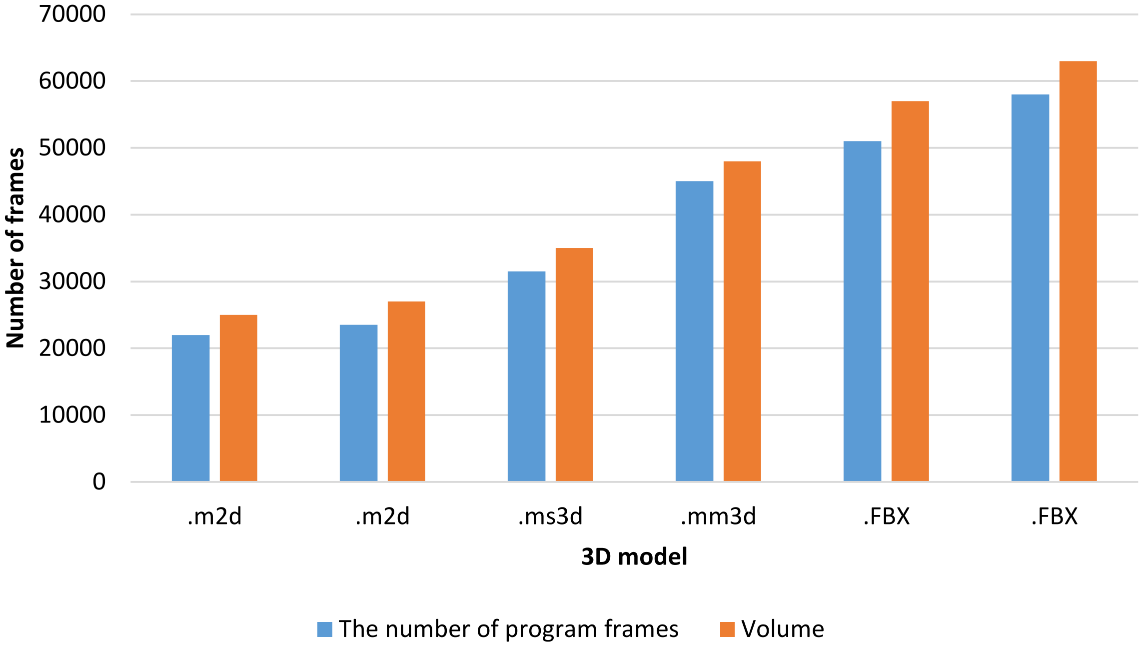


Fig. 3. Volume and number of frames and normal speed of loading seven objects in the program.

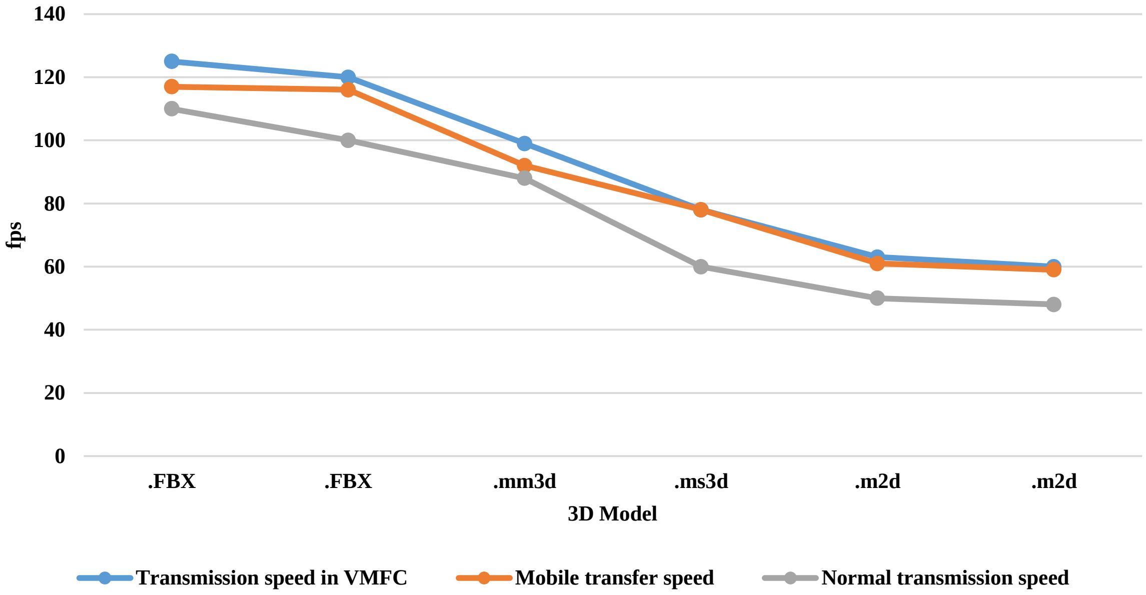
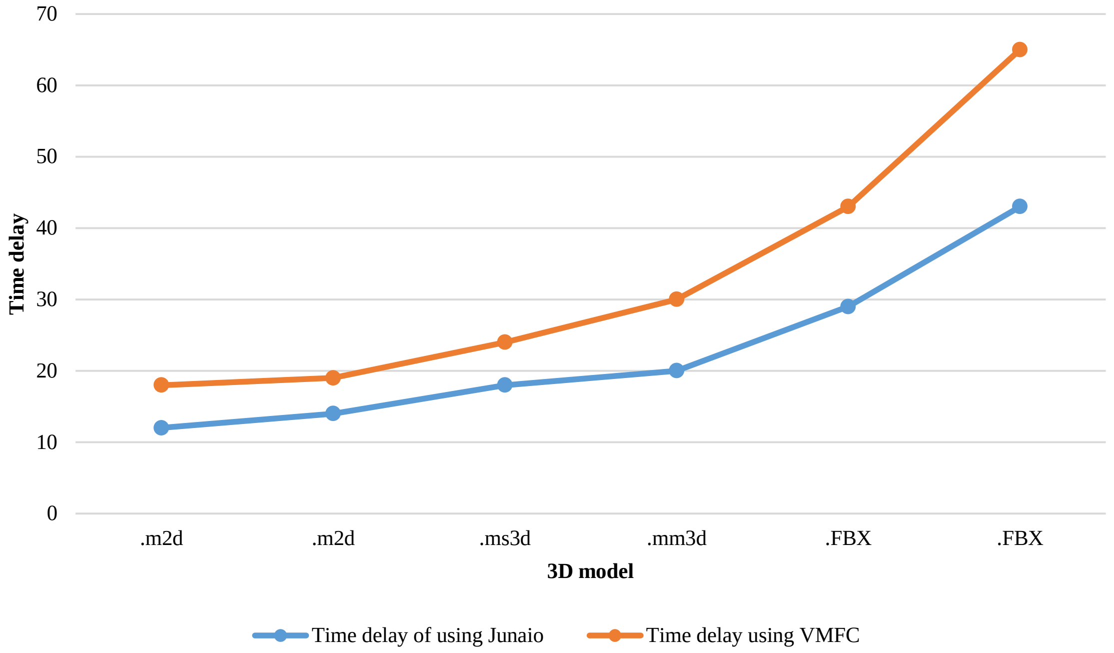


Fig. 4. Comparison of frames transmission speed in two modes of uploading from mobile phone and uploading from VMFC.

TABLE II. PAIRED T-TEST TO CHECK THE DIFFERENCE IN TRANSMISSION SPEED BEFORE AND AFTER USING VMFC

Variable	level	Average	standard deviation	t	p-value
Speed	After using VMFC	89.33	28.73	2.98	0.032
	On mobile and without VMFC	84.25	25.89		

(a) Mobile Internet



(b) Internet communication

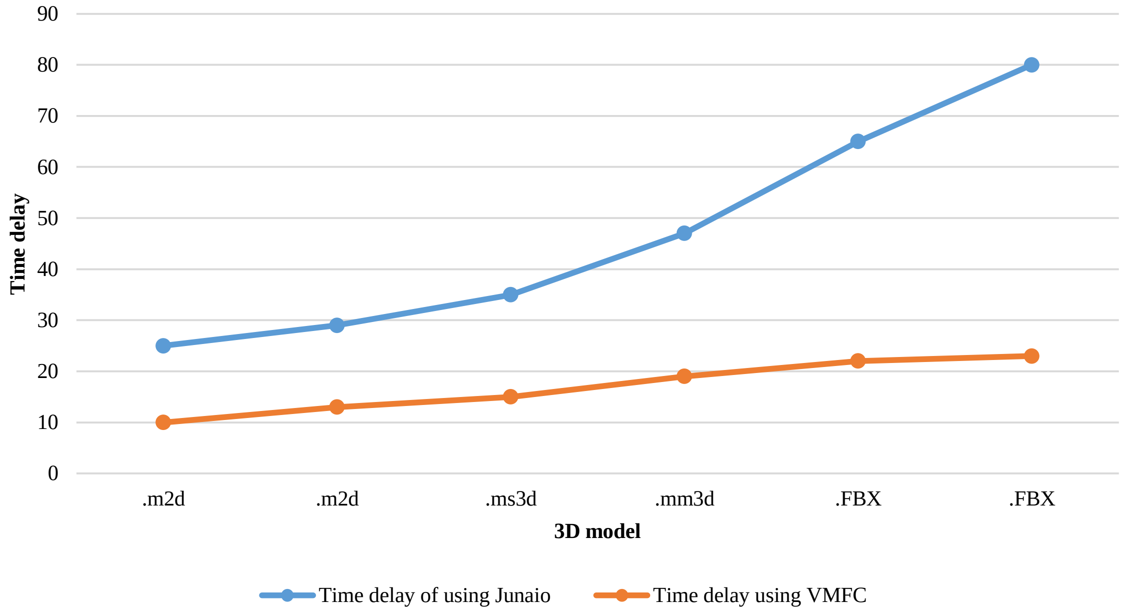


Fig. 5. (a) Data transfer time delay in mobile internet. (b) Time delay of data transfer in Internet telecommunications.

Based on Eq. (1), Fig. 5(a) and 5(b) present the time costs of data transmission in two telecommunication networks and mobile internet, respectively. The transmission of huge amounts of data, including FBX that took place in Junaio had a maximum delay of about 56 seconds, roughly 10 seconds longer than the maximum delay on the mobile Internet network. This quantity becomes apparent when there are more requests made and a lot of data being transferred. The suggested architecture's maximum latency has been achieved utilizing mobile internet, roughly 23 seconds less than the maximum delay in Junaio on the same platform. This number indicates that VMFC offers greater performance than cloud servers. It demonstrates the benefits of a high-speed network for delivering superior service. Additionally, there is a very small- and acceptable-time delay in the transmission of small-volume data, including M2D. In VMFC, as opposed to Junaio, this value is ideal.

Additionally, it takes the Junaio service roughly five minutes for every request to validate, send, and get a response link for a new request from a mobile device. As a result, while employing both platforms, Junaio's average time delay is substantially higher than VMFC's. Paired t-analysis was used after using VMFC and before using it in Junaio to compare the time delay in the transmission of communications with the Internet during the execution of the program on the Junaio server and the time delay after using VMFC to see if there was a change in the amount. Has it just recently been produced? The specifics of this statistical study are shown in Table III and IV. The maximum delay in VMFC varies by roughly 18 seconds, and Table III demonstrates how the maximum delay has decreased after utilizing Junaio to 29.316 seconds and VMFC to 16.770 seconds.

With a value of (p=0.047), this reduction is likewise statistically significant. Table IV shows that there is a decrease in time delay with a value of (P=0.014) between the time delay in 4G Internet transmission while the software is running on the Junaio server and the time delay after utilizing VMFC. Table V shows a comparison of the implemented program with related programs from [29] and [27] in the cloud environment and mobile edge, respectively. The implemented program in Table V was an image-based augmented reality application. Based on the comparison table in [27], the comparison parameters are set. It is evident that the software used in the VMFC environment performs better than its equivalent program.

Additionally, study [29] performs better than study [27] since it was completed in an environment comparable to the fog, the edge environment, which has largely solved the problems associated with delay and waiting in cloud environments. The work [29] makes no mention of the adaptation time. Based on the average data collected in the evaluation section by computing the evaluation circumstances, the parameters in this table were obtained. Due to fewer time delays and faster data transfer speeds throughout, adopting fog processing services is more appropriate for delay-sensitive applications like augmented reality, as shown in Table V. For data in various quantities, this number varies. The benefit of employing fog is that it has decreased the time delay above the cloud in all stages.

Due to the significant delay and large user base, cloud-based program deployment requires a lot of parameters. Of course, the fact that we are unaware of the circumstances under which the programs in [27] and [29] execute leads to significant variation in the numbers, particularly in the value of the model detection time. On the other side, dynamic models are employed because there are more details and animation motions, which result in a larger volume and longer detection times.

A. *Energy Consumption in Mobile Fog Processing*

Due to their extensive coverage and low energy requirements, low-power technologies (LPWA) have opened the way for data storage in virtual mobile fog computing. The long-range system (LORA), which increases its effectiveness because it is a defined protocol for data transport, is one of these technologies. It gives coverage for kilometers or even more and consumes little electricity. In addition to LoRa, a number of wireless technologies, including IoT, WiFi, Zigbee, and 5G, can be employed for coverage areas. Low-cost fog processing has been accomplished using ZigBee technology. However, the wireless data transmission's effectiveness is diminished by the relatively short range of coverage (approximately 25 meters). Table VI provides a thorough comparison of the various IoT wireless network technologies (Zigbee, LoRa, NB-IoT, and 5G).

TABLE III. PAIRED T-TEST TO INVESTIGATE THE DIFFERENCE IN THE USE OF VMFC AND JUNAIO IN INTERNET TELECOMMUNICATIONS

Variable	level	Average	standard deviation	t	p-value
Speed	After using VMFC	16.77	5.57	-2.63	0.047
	After using Junaio	84.25	29.31		

TABLE IV. PAIRED T-TEST TO CHECK THE TIME DELAY DIFFERENCE IN USING VMFC AND JUNAIO IN 4G

Variable	level	Average	standard deviation	t	p-value
Speed	After using VMFC	22.27	11.21	-3.68	0.014
	After using Junaio	32.66	18.04		

TABLE V. COMPARISON OF THE PROGRAM IMPLEMENTED IN [27] AND [29] AND THE PROGRAM BASED ON VMFC (TIMES IN MILLISECONDS)

		Program based:	
parameters	VMFC	Cloud in [27]	Cloud in [29]
Early time	1.8	3.1	2.4
Time of diagnosis	23.47	82.78	40.31
Compliance time	8	10	--
Similar sample	0.7	1.5	0.85
response time	25	-	23
Delay time	18	23.22	18.7
Loading time	10	18	11.3
loading time	0.007	0.1	--

The amount of energy used is examined in this part based on several parameters in this layer. The amount of time that each fog node spends processing data determines how much energy each fog node consumes. The fog node's energy usage rose along with the processing time. Thus, it can be concluded that the processing time at each fog node directly correlates with the energy usage in the fog. Consequently, as indicated by Eq. (2), we will have:

$$\mathrm{Em}_{wi} \propto \frac{1}{\mu wi - \sum_{wh\in f} a_{whwi}} \quad \forall wi \in f \tag{2}$$

The energy relation in each fog node can be expressed as follows, using e^f as the unit of energy consumption per unit of time:

$$\mathrm{Em}_{wi} \propto \frac{1}{\mu wi - \sum_{wh\in f} a_{whwi}} \times e^f \tag{3}$$

The data rate, range, device count, number of gateways or base stations, and energy consumption of both gateways or base stations and sensors are the categories utilized to group all fog-based wireless network technologies in this work, as indicated in Table VI. According to this research, various fog-based wireless network solutions almost all consume the same amount of energy, as illustrated in Fig. 6. Zigbee only has a range of 25 meters, so deploying it in an urban setting is not the ideal option. To transmit and store data over a vast region, hundreds or thousands of gateways are required.

Since using technology to cover a large area while using the least amount of energy is the objective, as illustrated in Fig. 6, LoRa has been chosen as a fog-based wireless communication technology between IoT sensors and gateways that supports remote communication with minimal energy usage. Processing mobile fog is taken into account. In contrast to conventional cloud-based architecture, Fig. 7 compares the energy usage of various tasks in an edge-fog-cloud design. Additionally, it displays the position of each task or application inside the edge-fog-cloud architecture as well as the energy usage figures for each one separately. The findings demonstrated that the edge layer is laden with sensor jobs since it has sufficient capacity. Because the fog layer has enough resources to execute jobs like data transfer and storage, normal processing tasks are sequentially loaded into the fog. The edge and fog layers are unable to fulfill demanding processing duties, so all remaining requests are offloaded to the cloud layer. Fig. 8 compares the energy usage of mobile and fixed Internet using LoRa technology based on fog processing. The energy usage of telecom internet is superior to that of mobile internet, accounting for up to 34% of the overall electricity consumption. However, energy savings based on various wireless solutions based on fog processing range somewhat between 32.4% and 34.3%. Zigbee has higher energy savings since it uses less energy to transmit sensor data.

TABLE VI. COMPARISON OF TECHNOLOGIES BASED ON MOBILE FOG PROCESSING

Technologies based on mobile fog	Data Rate	Range	Device Count	Power consumption(gateway)	Power consumption(sen)
Zigbee	240 Kb-ps	25m	250 [23]]	2W [26]	0.2W[25]
Lora	60 Kb-ps	15km	12000 [28]	35W [21]	0.45W[22]
NBIOT	250 Kb-ps	25km	56 [13]	7421W[39]	0.66W[38]
5G	25 Gb-ps	30km	1 mil. Per 2km [40]	12500W[14]	0.5W[23]

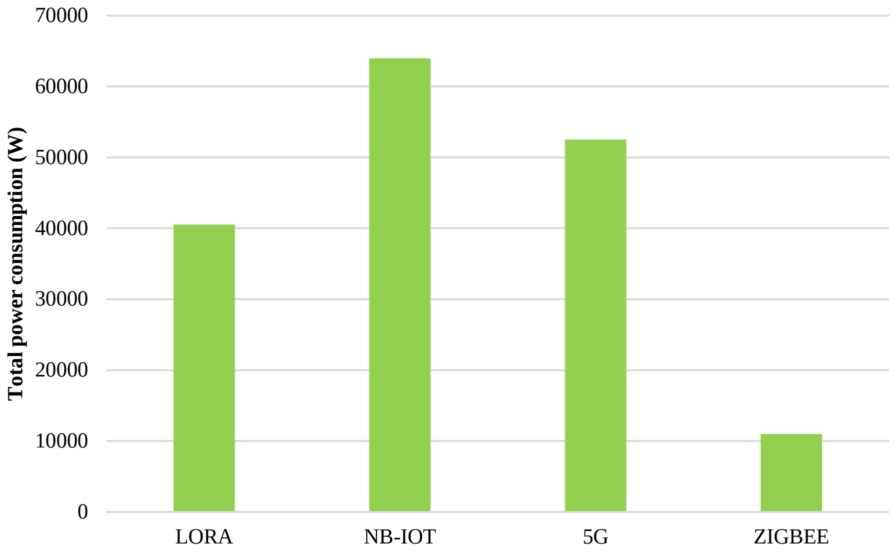


Fig. 6. Compares the energy usage of several wireless network solutions based on mobile fog processing.

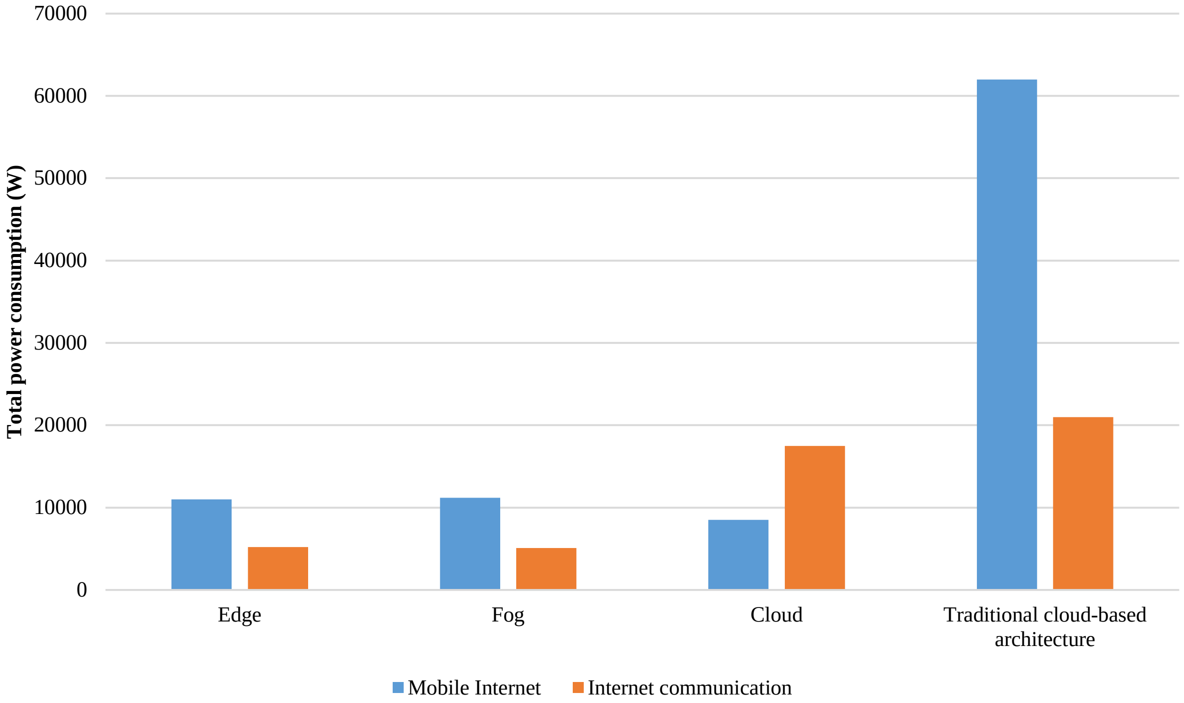


Fig. 7. Energy consumption of telecom Internet versus mobile Internet, considering LoRa technology based on mobile fog processing.

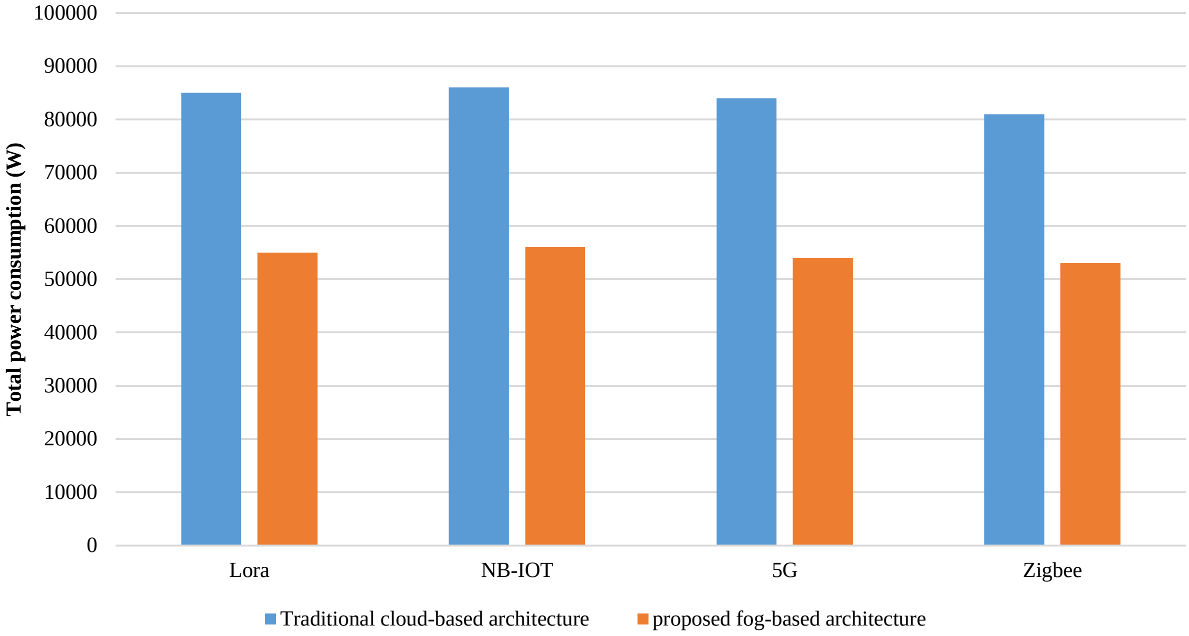


Fig. 8. Energy saving of the proposed architecture versus the traditional cloud-based architecture, using different wireless technologies based on mobile fog processing.

VI. CONCLUSION

The proposed method presented in this article has reduced the limitations of MCC services, including access delay, serial processing and network traffic, as well as the limitations of smart devices in running real-time programs such as augmented reality programs and programs that require. As they have more storage in the environment, it has been solved. Due to user profile registration and encryption with digital signature, providing a unique ID for each user and software ID for each program transferred to the cloud node, using this architecture increases the security of authentication for access to resources and data. And it is not possible to reuse the unique identifier of a user by another person. The image-based augmented reality program was implemented based on the presented method. The transfer of polygons processed on the 3D model to the screen in two modes of loading from a mobile phone and loading from VMFC was compared and it was observed that for models with a larger volume, the use of VMFC technology works faster and this is an advantage in the implementation of sensitive programs. It is delayed and immediate. Data transfer was done in two modes using Junaio and VMFC servers in two platforms, Wi-Fi internet and 4G mobile phone internet. The use of a large number of cloud nodes in mass form is necessary for a wide network of mobile devices, and this method should be able to support a large number of devices, followed by many programs, and run at an acceptable speed.

One of the limitations of this research is the small number of nodes for implementing programs, which should be resolved in future works. For future work, implementation and optimization of architecture layers to provide better and more efficient services, communication in the context of 5G networks should be given more attention, which minimizes delays.

FUNDING

This work was supported by:

1) Natural science foundation of anhui province: Research on Routing Optimization Strategy in Cloud and Fog Computing Network Architecture Based on SDN (No: 2022AH052854);

2) The cooperative education project of production and education of the ministry of education of china: Haiwen Java course school enterprise joint demonstration course construction (No:202101291002).

REFERENCES

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” IEEE Internet Things J, vol. 3, no. 5, pp. 637–646, 2016.

[2] A. A. Khan et al., “A drone-based data management and optimization using metaheuristic algorithms and blockchain smart contracts in a secure fog environment,” Computers and Electrical Engineering, vol. 102, p. 108234, 2022.

[3] S. Subbaraj, R. Thiyagarajan, and M. Rengaraj, “A smart fog computing based real-time secure resource allocation and scheduling strategy using multi-objective crow search algorithm,” J Ambient Intell Humaniz Comput, pp. 1–13, 2021.

[4] R. Roman, J. Lopez, and M. Mambo, “Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges,” Future Generation Computer Systems, vol. 78, pp. 680–698, 2018.

[5] M. Chiang and T. Zhang, “Fog and IoT: An overview of research opportunities,” IEEE Internet Things J, vol. 3, no. 6, pp. 854–864, 2016.

[6] R. Kalaria, A. S. M. Kayes, W. Rahayu, and E. Pardede, “A Secure Mutual authentication approach to fog computing environment,” Comput Secur, vol. 111, p. 102483, 2021.

[7] R. Neware, K. Ulabhaje, G. Karemore, H. Lokhande, and V. Dandige, “Survey on Security Issues in Mobile Cloud Computing and Preventive Measures,” in Smart Computing Paradigms: New Progresses and Challenges: Proceedings of ICACNI 2018, Volume 2, Springer, 2020, pp. 89–100.

[8] 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.

[9] A. Munir, P. Kansakar, and S. U. Khan, “IFCIoT: Integrated Fog Cloud IoT: A novel architectural paradigm for the future Internet of Things.,” IEEE Consumer Electronics Magazine, vol. 6, no. 3, pp. 74–82, 2017.

[10] K. N. Pallavi and V. Ravi Kumar, “Authentication-based access control and data exchanging mechanism of IoT devices in fog computing environment,” Wirel Pers Commun, vol. 116, pp. 3039–3060, 2021.

[11] M. Trik, A. M. N. G. Molk, F. Ghasemi, and P. Pouryeganeh, “A hybrid selection strategy based on traffic analysis for improving performance in networks on chip,” J Sens, vol. 2022, 2022.

[12] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” IEEE Internet Things J, vol. 3, no. 5, pp. 637–646, 2016.

[13] Yahya, R. O., Mahmood, N. H., Kadir, D. H., & Aziz, S. J. (2023). The Use of Factor Analysis and Cluster Analysis Methods to Identify the Most Crucial Key Factors Influencing the Psychological Stability of University Students. Polytechnic Journal of Humanities and Social Sciences, 4(1), 779-789.

[14] D. C. G. Valadares, N. C. Will, J. Caminha, M. B. Perkusich, A. Perkusich, and K. C. Gorgônio, “Systematic literature review on the use of trusted execution environments to protect cloud/fog-based Internet of Things applications,” IEEE Access, vol. 9, pp. 80953–80969, 2021.

[15] S. Shukla, S. Thakur, S. Hussain, J. G. Breslin, and S. M. Jameel, “Identification and authentication in healthcare internet-of-things using integrated fog computing based blockchain model,” Internet of Things, vol. 15, p. 100422, 2021.

[16] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, “A survey on mobile edge networks: Convergence of computing, caching and communications,” Ieee Access, vol. 5, pp. 6757–6779, 2017.

[17] Y. Fan et al., “SBBS: A secure blockchain-based scheme for IoT data credibility in fog environment,” IEEE Internet Things J, vol. 8, no. 11, pp. 9268–9277, 2021.

[18] N. Kumar, S. Zeadally, and J. J. P. C. Rodrigues, “Vehicular delay-tolerant networks for smart grid data management using mobile edge computing,” IEEE Communications Magazine, vol. 54, no. 10, pp. 60–66, 2016.

[19] Trik, M., Mozaffari, S. P., & Bidgoli, A. M. (2021). Providing an adaptive routing along with a hybrid selection strategy to increase efficiency in NoC-based neuromorphic systems. Computational Intelligence and Neuroscience, 2021.

[20] A. Lakhan, Q.-U.-A. Mastoi, M. Elhoseny, M. S. Memon, and M. A. Mohammed, “Deep neural network-based application partitioning and scheduling for hospitals and medical enterprises using IoT assisted mobile fog cloud,” Enterp Inf Syst, vol. 16, no. 7, p. 1883122, 2022.

[21] Hai, T., Alizadeh, A. A., Ali, M. A., Dhahad, H. A., Goyal, V., Metwally, A. S. M., & Ullah, M. (2023). Machine learning-assisted tri-objective optimization inspired by grey wolf behavior of an enhanced SOFC-based system for power and freshwater production. International Journal of Hydrogen Energy.

[22] D. Lanka, C. L. Veenadhari, and D. Suryanarayana, “Application of fog computing in military operations,” Int J Comput Appl, vol. 164, no. 6, pp. 10–15, 2017.

[23] Mokhlesi Ghanevati, D., Khorami, E., Boukani, B., & Trik, M. (2020). Improve replica placement in content distribution networks with hybrid technique. Journal of Advances in Computer Research, 11(1), 87-99.

[24] Khezri, E., Zeinali, E., & Sargolzaey, H. (2023). SGHRP: Secure Greedy Highway Routing Protocol with authentication and increased privacy in vehicular ad hoc networks. Plos one, 18(4), e0282031.

[25] FanghuaTang, Huanqing Wang, Liang Zhang, Ning Xu, Adil M.Ahmad. Adaptive optimized consensus control for a class of nonlinear multi-agent systems with asymmetric input saturation constraints and hybrid faults. Communications in Nonlinear Science and Numerical Simulation, 126: 107446, 2023.

[26] Wenjing Wu, Ning Xu, Ben Niu, Xudong Zhao and Adil M. Ahmad, Low-Computation Adaptive Saturated Self-Triggered Tracking Control of Uncertain Networked Systems, Electronics, 12(13), 2771, 2023.

[27] S. C. Nayak, S. Parida, C. Tripathy, and P. K. Pattnaik, “Task scheduling mechanism using multi-criteria decision-making technique, MACBETH in cloud computing,” in Progress in Computing, Analytics and Networking: Proceedings of ICCAN 2017, Springer, 2018, pp. 381–392.

[28] S. K. Dwivedi, R. Amin, and S. Vollala, “Smart contract and ipfs-based trustworthy secure data storage and device authentication scheme in fog computing environment,” Peer Peer Netw Appl, vol. 16, no. 1, pp. 1–21, 2023.

[29] Khezri, E., Zeinali, E., & Sargolzaey, H. (2022). A novel highway routing protocol in vehicular ad hoc networks using VMaSC-LTE and DBA-MAC protocols. Wireless Communications and Mobile Computing, 2022.

[30] S. Guo, X. Zhao, H. Wang, N. Xu, Distributed consensus of heterogeneous switched nonlinear multiagent systems with input quantization and dos attacks, Applied Mathematics and Computation 456 (2023) 128127.

[31] Fabin Cheng, Ben Niu, Ning Xu, Xudong Zhao, and Adil M. Ahmad. Fault Detection and Performance Recovery Design With Deferred Actuator Replacement Via A Low-Computation Method, IEEE Transactions on Automation Science and Engineering, DOI: 10.1109/TASE.2023.3300723, 2023.

[32] Khezri, E., & Zeinali, E. (2021). A review on highway routing protocols in vehicular ad hoc networks. SN Computer Science, 2, 1-22.

[33] E. Poornima et al., “Fog robotics-based intelligence transportation system using line-of-sight intelligent transportation,” Multimed Tools Appl, pp. 1–29, 2023.

[34] M. Samiei, A. Hassani, S. Sarspy, I. E. Komari, M. Trik, and F. Hassanpour, “Classification of skin cancer stages using a AHP fuzzy technique within the context of big data healthcare,” J Cancer Res Clin Oncol, pp. 1–15, 2023.

[35] J. Sun, Y. Zhang, and M. Trik, “PBPHS: a profile-based predictive handover strategy for 5G networks,” Cybern Syst, pp. 1–22, 2022.

[36] M. Trik, H. Akhavan, A. M. Bidgoli, A. M. N. G. Molk, H. Vashani, and S. P. Mozaffari, “A new adaptive selection strategy for reducing latency in networks on chip,” Integration, vol. 89, pp. 9–24, 2023.

[37] Yanwei Zhao, Ben Niu, Guangdeng Zong, Xudong Zhao, Khalid H. Alharbi. Neural network-based adaptive optimal containment control for non-affine nonlinear multi-agent systems within an identifier-actor-critic framework, Journal of the Franklin Institute. 360 (12), pp.8118-8143, 2023.

[38] Sai Huang；Guangdeng Zong；Huanqing Wang；Xudong Zhao；K. H. Alharbi. Command Filter-Based Adaptive Fuzzy Self-Triggered Control for MIMO Nonlinear Systems With Time-Varying Full-State Constraints. International Journal of Fuzzy Systems, 2023, https://doi.org/10.1007/s40815-023-01560-8.

[39] R. Abeywickrama, M. Haviv, B. Oz, and I. Ziedins, “Strategic bidding in a discrete accumulating priority queue,” Operations Research Letters, vol. 47, no. 3, pp. 162–167, 2019.

[40] S. Guan and A. Boukerche, “Intelligent edge-based service provisioning using smart cloudlets, fog and mobile edges,” IEEE Netw, vol. 36, no. 2, pp. 139–145, 2022.