Journal of Harbin Engineering University Vol 44 No. 8
ISSN: 1006-7043 August 2023

Analyze Combining Iot with Edge Computing to Improve Storage and Transmission

[1]Ihab Khalid Hato Al-Gawwam, [2] Seyed Ebrahim Dashti (Corresponding Author), [3] Ahmed Abed Abbas
[2] Department of Computer Engineering, Jahrom Branch, Islamic Azad University
[1,3] Department of Computer Engineering, Shiraz Branch, Islamic Azad University

Abstract:-The ubiquitous nature of the Internet of Things (IoT) has made it possible to gather and analyze vast amounts of data, which has profound implications for our ability to make sound decisions. Critical smart-world infrastructures are monitored and controlled by millions of sensors and devices that are constantly producing data and exchanging essential information via sophisticated networks. Edge computing has evolved as a new paradigm to address the IoT and localized computing needs as a way to prevent the rise in resource congestion. Edge computing, in contrast to the more widely recognized cloud computing, will move data computation or storage closer to the network's ''edge,'' where it will be closer to the end users. As a result, the latency of message exchange can be greatly reduced by dispersing computing nodes around the network rather than relying solely on a centralized data center. In addition, the decentralized architecture can smooth out traffic spikes on Internet of Things networks, shorten response times for real-time IoT applications, and lower transmission latency between edge/cloudlet servers and end users. Additionally, the system can increase the lifetime of individual nodes by shifting the burden of computation and communication from nodes with limited battery supply to nodes with high power resources. In this research, we analyze the possibility of combining IoT with edge computing and the effect of edge computing on the efficiency of storage and transmission in IoT network. We group edge computing into its architectural components and compare their relative effectiveness in terms of latency, bandwidth utilization, power consumption, and other parameters. Taking into account edge computing's privacy, authenticity, and resilience concerns, we also provide a paradigm for assessing IoT network security using edge computing. We conclude by comparing and contrasting edge computing with traditional cloud computing architectures for various IoT use cases.
Keywords: IoT, Edge computing, Cloud computing, Transmission, Storage.

Introduction:

The Internet of Things (IoT) has become increasingly significant as information technology has advanced. Using the sophisticated communication network infrastructure linked by millions of IoT nodes [1][2], interconnected sensors/devices can gather and share various data among themselves. Then, numerous Internet of Things apps can supply users with more precise and granular network offerings. As more and more sensors and gadgets join the internet in this way, the data they produce will be enormous and will need to be processed further before it can be used to benefit either service providers or end users. Traditional cloud computing necessitates sending data to remote servers for processing and then returning the results to the sensors and devices. The bandwidth and resource requirements for transmitting data become extremely taxed as a result of this process. In addition, the larger the data set, the worse the network's performance will become [3][4].

When it comes to time-sensitive IoT applications like smart transportation[5], smart electricity grid, smart city, etc. [6][7], the situation becomes more dire because conventional cloud computing-based services simply cannot keep up with the demand. This is due to the distance between the cloud and its end users and the fact that compute processes must be uploaded there, using up precious bandwidth and network resources. This, of course, will cause significant network delay, which is intolerable for time-sensitive Internet of Things (IoT) applications[8][9] [10]. The influence of these functions on security and emergency response makes this a crucial issue for the Internet of Things.

Most IoT devices (smart sensors, etc.) also have limited power, therefore it's important to maintain a healthy balance between power consumption

and device longevity by allocating computation to devices with more robust power and computing resources. Additionally, transmission times can be lowered by processing data in the closest computation nodes to the user. Data transmission speeds in cloud computing-based services are impacted by network traffic, with high volumes of traffic resulting in longer transmission times and higher energy costs. As a result, it's important to think about scheduling and processing allocation.

In this paper, we give our perspective on edge computing for the IoT and summarise existing efforts and past work that aim to address the aforementioned concerns and issues[11] [12]. Computing and storage tasks are being carried out at the ''edge'' of the network, close to the user, and this is known as edge computing [13][14]. The edge computing nodes' proximity to the end users will help reduce traffic congestion. Additionally, it greatly minimizes the transmission delay during data computing or storage in IoT, and it mitigates the bandwidth requirements of the centralized network. The response times of IoT applications may be faster than the corresponding cloud computing services if distributed computation nodes are placed at the edge to offload traffic and computational demand from the centralized cloud. In addition, edge computing can shift power-intensive tasks like computation and communication from low-power nodes to high-power ones at the network's periphery. Doing so will prolong the lifespan of the overall IoT network by extending the lifetime of the nodes with limited batteries.

The following are some of our main contributions to this paper:

1-Group various edge computing architectures into categories and discuss the benefits and drawbacks of each. We also evaluate the capabilities of these groups in terms of response time, computing power, and storage space.

2-Take a methodical look at what makes IoT tick, and we go over several common use cases. We use this research to evaluate cloud and edge computing environments from the perspective of Internet of Things devices. The advantages and disadvantages of edge computing for IoT networks are then discussed.

3- using depth knowledge of both the Internet of Things and edge computing, discuss the potential for their integration as edge computing-based IoT. The Internet of Things issue space that makes use of edge computing is then demonstrated. In this article, we take a problem-oriented look at the architectures, operations, task scheduling, security, and privacy of edge computing.

4-Show the pros and cons of edge computing-supported IoT in terms of data transmission, data storage, and data processing. From the vantage points of system integration, resource management, security and privacy, and cutting-edge communication, we explore these new issues. We also provide some examples of IoT smart applications to illustrate the integration of edge computing with the IoT.

Here is how the rest of the paper is structured: In Section II, we provide a high-level overview of the history and fundamentals of IoT, edge computing, and cloud computing. Section III describes the features of the Internet of Things and edge computing, and discusses how the latter can be used to improve the former, showing how they could be combined. Meanwhile, we lay out the framework for edge computing and the Internet of Things. The advantages of combining the Internet of Things with edge computing are discussed in Section IV. We define the domain of interest and develop illustrative viewpoints from the vantage points of transmission, storage, and computing. We examine the difficulties of implementing IoT at the edge in Section V. Section VI is the final section of the paper.

II.The Internet of Things and Peripheral Computing

In this article, we will analyze the possibility of combining IoT with edge computing by reviewing their core principles.

First, let's talk about the "IoT"

The desktop-based model of computing is outdated and will be superseded soon [15]. In particular, being a relatively new form of technology, the Internet of Things is becoming increasingly integrated into everyday life. The Internet of Things (IoT) is a theoretical framework that predicts a future in which virtually all electronic gadgets, including smartphones, automobiles, sensors, actuators, and other embedded devices, are networked to one another

and to central data repositories where they can share data and trigger a dramatic increase in the volume of data generated.

With the broad implementation of smart transportation, smart cities, smart grids, and smart healthcare, people will be unable to function without IoT penetrating their day-to-day lives at home and in the workplace. Therefore, IoT is the next big thing and will change people's lives in significant ways. The corporate world is another one where IoT plays a crucial role. According to a forecast from 2025, the Internet of Things will be one of the most consequential technologies for the United States. Similarly, the number of psychical devices has grown larger than the global human population [16]. The number of connected physical devices reached 9 billion in 2012, and it is predicted to reach 75 billion by the year 2020. Therefore, in the future, IoT devices will be one of the most essential, if not the most important, data sources for big data.

Three distinct IoT communication models will be discussed here.

Communication between machines

Without the need for additional hardware, this communication model depicts a network in which several endpoints can connect and exchange data directly with one another [17][18], The Internet and IP networks are just two of the many types of networks that these gadgets can communicate over with one another. As may be seen in, for instance, a smart switch uses Bluetooth 4.0 to interact with a smart light. These kinds of networks enable hybrid communication protocols, which combine device-to-device with a specialized communication protocol to suit QoS requirements. This paradigm is commonly used by applications that need to exchange data with one another but have minimal data rate constraints. Smart home technology and electrical systems with built-in automation are two such examples. Smart locks, smart switches, and smart lights are all examples of the kind of Internet of Things devices that often only broadcast very brief data packets[19].

Machine-to-Machine connections.

run into trouble for end users when various gadgets from various manufacturers all employ incompatible protocols. As an example, smart home gadgets that use the Z-Wave protocol cannot talk to those that use the ZigBee protocol. Because of compatibility problems, users are restricted in their options [19].

Machine-to-cloud communication.

In a device-to-cloud communication paradigm, IoT devices need to use a cloud application service provider or store data in a cloud storage disc due to limitations in their computing capabilities or storage space [20]. As can be seen, this approach usually needs help from preexisting communication mechanisms like wired or Wi-Fi networks.

The limits of the traditional network, such as bandwidth and available resources, nevertheless limit the effectiveness of Machine-to-Cloud communication despite fixing the problems with the Machine-to-Machine paradigm. Changing the structure of the underlying network can improve the effectiveness of the Machine-to-Cloud communication paradigm [19].

Data transmission from the machine to a center

In the machine-to-gateway strategy, the device-to-application-layer gateway (ALG) paradigm is seen as a proxy or middleware box [21]. The diagram shows how M2G connections are set up. A gateway or other network device is required to perform tasks like data or protocol translation methods and software-based security checks to link IoT devices to cloud-based application services. This improves the safety and adaptability of the IoT network, shifts some processing to the application layer, and lowers the power needs of IoT devices. For example, a smartphone can act as a gateway by hosting apps that facilitate data exchange between Internet of Things gadgets and the cloud. Sensors generate data, sync with a person's smartphone, and then the phone encrypts the data before transmitting it to the cloud in the field of individual healthcare [19][22].

Second, Standard Internet of Things elements

Typically, an IoT network will consist of three types of nodes: sensors/devices, IoT gateways/local networks, and a backhaul network cloud, which represents the data source, data communication networks, and data processing, respectively.

1-Sensors/devices: Millions of sensors are spread out over a large region to create the IoT. These sensors are essential to the Internet of Things because they generate the vast majority of the

networks' measurement data. These sensors can collect a wide range of information that can aid the IoT in its quest for total situational awareness. Furthermore, most resource demands are produced by users' terminal devices. The devices can act as user interfaces, translating user input into data that can be sent to the Internet of Things. These sensors and endpoints will be networked together to share information and facilitate the development of new applications. Each node can obtain the necessary resources for IoT applications via the network that connects them[23][24]25].

2-IoT Gateways: IoT gateways connect the edge devices and backbone networks to the cloud. As and when necessary, IoT application endpoints will delegate data processing and storage duties to external cloud servers. The data collected by sensors and other devices can be transmitted via a network, but not before being preprocessed. IoT gateways will collect information from all of the connected sensors and devices and upload the results to the cloud. In order to save time and money, most IoT gateways perform preliminary processing on collected data. In addition, IoT gateways will relay the outcomes of the cloud servers' data processing to their respective end users [26][27].

3- Core/cloud network: Clients will use backhaul networks to transmit data and requests to cloud servers [28] [29]. Internet of Things applications don't have to worry about storage or processing power because they can use cloud servers. This means that the cloud servers can accommodate a wide range of software by providing the required resources. Once the data has been processed by the cloud servers, the results will be returned to the users. Remember that the majority of your clients will choose to have their IoT data processed via cloud servers.

Third, Edge computing

computing: the front end, the close end, and the far end. Differences between these regions are

Traditional centralized cloud computing is having trouble meeting the QoS requirements of many applications because of the explosion in the number of mobile devices. With the advent of 5G network technology, edge computing will emerge as the primary means of resolving this matter. The Radio-Access Network (RAN) is a significant obstacle to the development of 5G technology. Mobile edge computing is used in RAN to give timely RAN data. Network providers can enhance their customers' Quality-of-Experience (QoE) by capitalizing on the context-aware services made possible by real-time RAN [30][31][32].

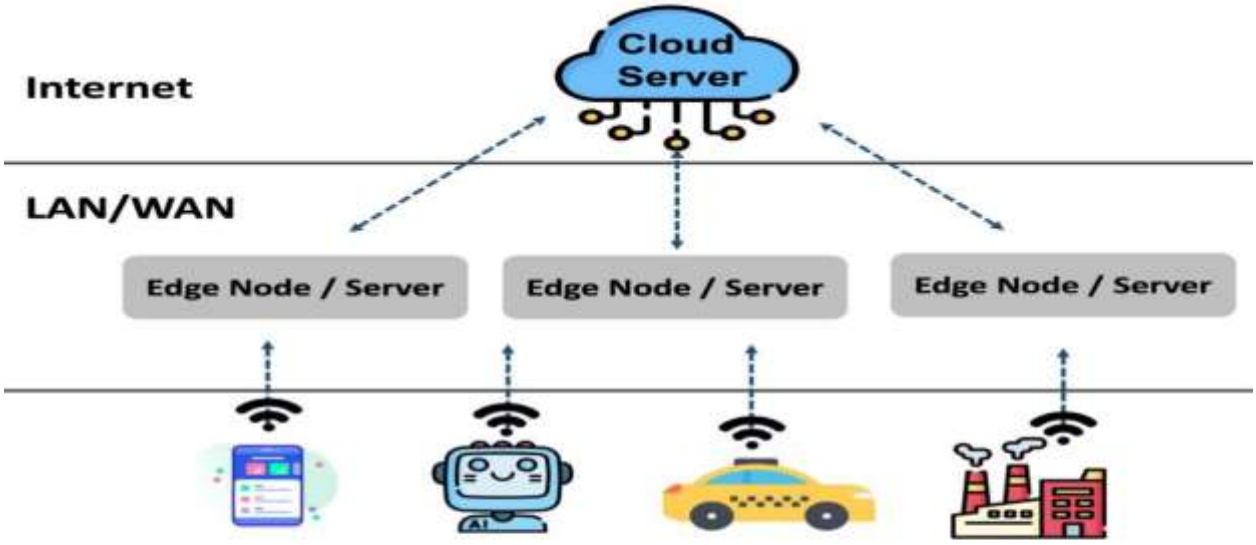
The edge computing platform enables edge nodes to respond to service requests, thereby reducing network latency and bandwidth consumption. By placing RAN in the network's periphery under the control of third-party operators, network operators can expedite the rollout of new applications. However, implementing consistent security measures across the board is difficult because the compute nodes are operated by a large array of different types of independent third-party operators [33][34].

Forth, Developments in computing at the edge

As shown in Fig. 1, the most fundamental components of an edge computing system. When compared to server data centers in the cloud, edge computing servers are positioned closer to the end user. As a result, while having few processing capacities than cloud computing servers, edge computing servers offer higher QoS (Quality of Service) and lower latency to end users. To better understand the benefits and drawbacks of edge computing, we shall compare and contrast the two architectures. Edge computing, in contrast to cloud computing, makes use of on-premises nodes rather than off-site data centers. We refer to these off-site machines as edge/cloudlet servers in this work. As shown in Fig. 2, there are three basic

parts of edge detailed below[19][35].

Fig.1.fundamental framework for edge computing.



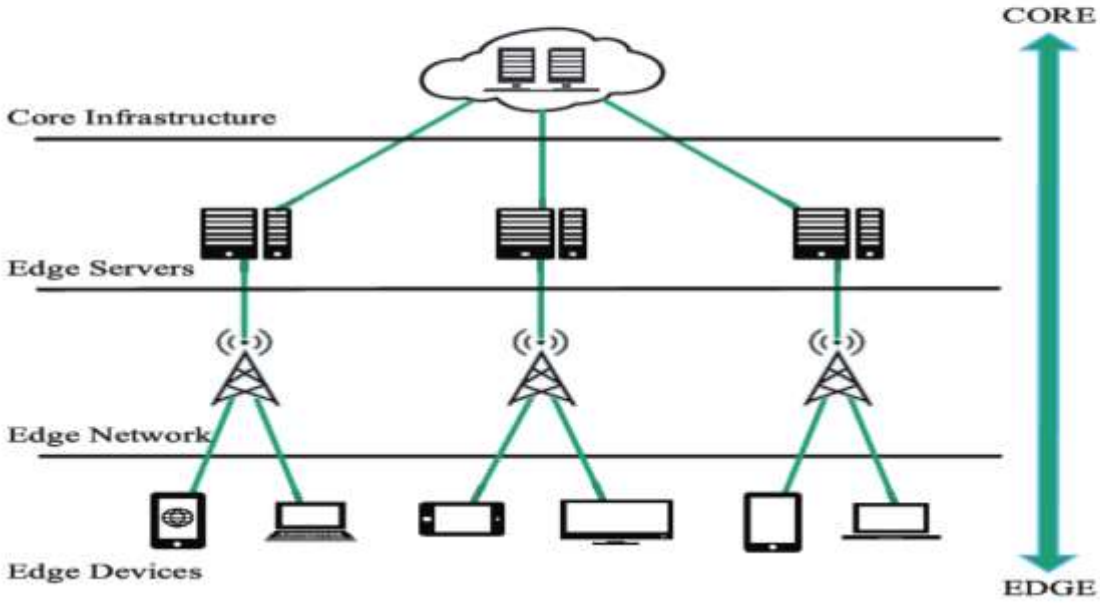


Fig.2. Edge computing network architecture.

FRONT-END: The edge computing architecture's front end is where the actual end devices (such as sensors and actuators) are placed. The front-end environment has room to grow in terms of user interactivity and responsiveness. For some uses, edge computing can provide real-time services thanks to the computing power given by the abundance of adjacent end devices. However, most needs cannot be met in the front-end environment due to the limited capacity of the end devices. Therefore, under such circumstances, client devices must notify backend servers of their need for resources [36].

NEAR-END: Most network traffic will be supported by gateways set up in the near-end environment. Real-time data processing, data caching, and compute offloading are just a few examples of the many resource demands that edge/cloudlet servers may have. Most data processing and storage will be moved to this localized system as part of edge computing. There will be a slight increase in latency, but the benefits to end users in terms of data computing and storage will be substantial [37].

FAR-END: As cloud servers are increasingly located at a greater distance from end devices, there is a

corresponding increase in network latency. However, remote cloud servers can offer greater processing muscle and data storage space. Cloud servers, for instance, can facilitate big data mining, big data management, machine learning, and other such services [38].

Fifth, Embracing Cutting-Edge Computing

Some studies have already centered on the development of edge computing models, which will be necessary to put into practice the aforementioned edge computing architecture. Both the hierarchical approach and the software-defined paradigm are commonplace.

1-Hierarchical Model: The edge architecture is hierarchical because edge/cloudlet servers can be placed at varying distances from end users, with each tier specifying a specific set of responsibilities according to its location and available resources. That's why it makes sense to describe the edge computing network as a hierarchy.

Many studies have been conducted on the hierarchi- cal model. In [39], for instance, Jararweh et al. developed a hierarchical paradigm that connects MEC servers with cloudlets. With MEC, mobile consumers can get what they want because their data storage and processing needs will be met. For handling heavy traffic spikes from mobile users, Tong et al. [40] presented a hierarchical edge cloud approach. The cloudlet servers in this approach are set up at the network's periphery, and the regional edge cloud is set up as a hierarchical tree of edge servers. The computing power of edge servers can be aggregated further to satisfy the need for peak loads by utilizing this well-crafted hierarchical structure [19][41].

2-Model defined by software: Because of the massive number of people and devices involved, edge computing for IoT will be exceptionally challenging to manage. Using software-defined networking (SDN) [39] [42] is one approach to the challenge of managing edge computing.

The SDN model has been the subject of extensive research. For example, Jaraweh et al. [41] offered a software-defined approach to integrating the capabilities of Software Defined Systems and the MEC system. Management and administrative costs can be reduced in this way. Du and Nakao suggested a case-specific MEC model [42]. The software-defined data plane used by MVNOs is taken into consideration in their method. The authors developed algorithms for hop count-based tethering detection and mobile-friendly optimization. TCP concurrent connections can be managed using the established protocols to ensure user equality. Manzalini and Crespi [43] suggest an edge operating system that makes use of open-source software that is already accessible for free to create stable foundations for networks and services. Salman et al. [44] advocated combining software-defined networks (SDNs), network function virtualization (NFV), and multi-access edge computing (MEC). To facilitate wide-scale IoT implementation and boost MEC utilization in mobile networks, this approach can be scaled up. Smart Applications on Virtual Infrastructure Software-Defined Infrastructure (SDI) Smart Edge architecture [45] facilitates the creation of various services and applications across dispersed networks.

III. Integrating the Internet of Things and Edge Computing.

Potential synergies between the Internet of Things and edge computing will be discussed. We compare the features of the Internet of Things, edge computing, and the cloud based on the findings of our research. To further explain how edge computing enhances IoT performance, we zero in on the transmission, storage, and computing features.

1-Overview: To continue, both the Internet of Things and edge computing are undergoing significant development in their own right. Although IoT devices are autonomous, the edge computing platform can assist address key problems and boost efficiency. It has so become obvious in recent years that they should be combined.

The edge computing-based IoT has a three-layer architecture, as shown in Fig. 3. All IoT devices are also end users of edge computing, and its architecture has the same layers. Edge computing and Cloud computing are both useful for the Internet of Things because of their respective strengths (high processing capacity and big storage). Despite its lower computational power and storage space, edge computing offers

additional benefits over cloud computing for the Internet of Things. In particular, low latency responses are more important than massive amounts of storage and processing power for the Internet of Things. Tolerable computational capability, sufficient storage space, and fast response time are all features of edge computing that can meet the needs of IoT applications.

The edge computing framework can be extended by IoT to accommodate the scattered and dynamic nature of the edge computing nodes. Edge nodes can be Internet of Things (IoT) devices or other devices with spare processing power. Many studies have looked into using cloud computing to aid the Internet of Things, but in many cases, edge computing can provide superior performance. As the number of connected devices continues to grow, it's conceivable that IoT and edge computing will merge into a single technology. Most needs in the Internet of Things may be classified into transmission, storage, and computing, as we have already mentioned. Below, we'll break down each group and highlight how Edge Computing has improved the Internet of Things[19][46].

2-Demands On Iot Performance

Transmission: The total time needed to reply is simply the sum of the times it takes to send and process data. Even though they don't require much processing power, IoT gadgets generate copious amounts of data on a regular [46]. A network with unacceptable latency will fail to provide acceptable quality of service. Vehicle-to-vehicle and vehicle-to-infrastructure communications are two instances of this. Concerns for public safety and the needs of first responders need a rapid response time.

Since edge computing may provide numerous distributed processing nodes that are close to the end users, it has an advantage over the standard cloud when it comes to real-time information collecting and analysis services [14]. Meanwhile, edge computing nodes provide enough processing power to suit IoT requirements. Therefore, the needs of IoT applications are not subject to the delay in typical cloud services like Amazon Cloud or Google Cloud, thanks to the quick transmission time of Edge computing.

Storage: In light of what has been said thus far, it should come as no surprise that the Internet of Things (IoT) is the primary driver of big data generation. Therefore, IoT must transfer huge data to either local or remote cloud storage. One obvious advantage of uploading to edge-based storage is the decreased time it takes to do so. However, this has the problem of raising questions about the safety of data stored at the network's periphery [47]. Due to the distributed nature of the edge nodes, it is challenging to guarantee the original data's integrity, information protection, anonymity assessment, non-repudiation, and freshness [48] [49]

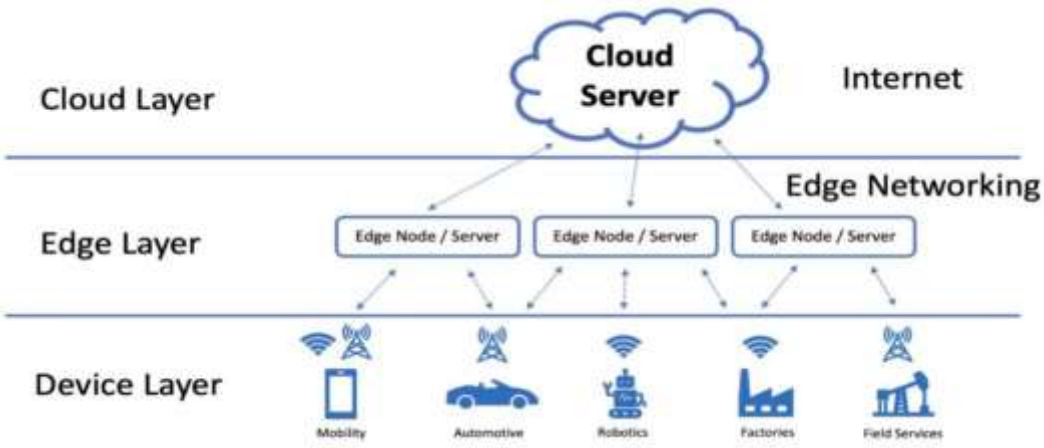


Fig. 3. The Internet of Things' layered design for edge computing.

Edge nodes also have less storage space than cloud computing data centers, and this storage is not as scalable or durable. The complexity of data management will increase further when it

becomes required to upload the data since multiple edge nodes will be used and coordinated to store the data.

Computation: Due to the limited processing power and battery life of most IoT devices, it is not practical to perform complex computations locally. The data is collected by IoT devices and then sent to other, more robust computing nodes for further processing and analysis. However, edge nodes only have so much processing power, making it difficult to scale the amount of processing power available at the edge. However, edge nodes can satisfactorily meet the needs of IoT, especially for real-time services, because IoT devices typically do not require substantial computational capacity. In addition, by outsourcing computational work to other nodes, edge nodes reduce the power consumption of IoT devices[50].

We have developed the issue space for Edge Computing-based IoT using the aforementioned three categories. Here, we'll go deep into how IoT powered by Edge Computing meets data transfer, storage, and processing needs. To help you better understand each trait, we've included some examples.

IV.Benefits of IoT that rely on edge computing

We evaluate the benefits of combining IoT with edge computing in this section.

A. Transmission: The transit time is affected by network performance, which may be measured in many ways. As previously mentioned, one of the key advantages of edge computing is rapid data transfer, which can meet the quality of service requirements of real-time applications such as Microsoft's "Live Video Analytics" [50]. The goal of this project is to construct a low-cost, real-time system for analyzing live films captured by all of the available cameras in a given public space. This architecture will provide a geo-distributed hierarchy of huge clouds and intelligent edges

[51]. Predicting the flow of vehicular traffic is an important part of this project, but it must be done on time. With edge computing's hierarchical design, data can be sent faster than over any other network [52].

The bottleneck issue of network resources in IoT has also inspired the development of edge computing. Response time and traffic flow can be greatly improved by shifting the burden of data calculation and storage onto individual users. ''Live Video Analytics'' [50], ''Human Action Classification'' [53], ''Motion Estimation'' [54], etc. are examples of time-sensitive applications that can benefit from the hierarchically distributed edge nodes.

1. Latency/delay: Computing delay and network latency are the two main contributors to the overall latency of an application. The time it takes to process data, or computer latency, varies from system to system. In contrast to the powerful processing capabilities of the network's servers, sensors are typically low-powered embedded devices. However, transmission latency would significantly rise due to the data transfer between the end devices and the cloud servers.

The difficulty of edge computing stems from the need to strike a balance between the two types of delays in computing and transmission and as a result, an optimal task offloading scheme must be designed to establish where and how each data processing task should be executed[19].

To accomplish this optimal resource allocation, certain mathematical methods have been developed recently. A delay-optimal computing job scheduling technique [55]. With this method, it is possible to decide whether a task will be run locally on the end device, or transferred to the MEC server for processing in the cloud. Multiple factors (task buffer queue length, CPU utilization, etc.) are taken into account by the scheduling technique. When implemented, this scheduling strategy can cut down on end-device power usage and average task delays. A paradigm for offloading multiple users' computations in 5G mobile edge computing [56]. This research formalized the numerous knapsack issue. The problem-solving strategy allows for less overall delay. The mobile cloud computing environment's multi-user computation offloading game challenge [57] presented a distributed computation offloading strategy. By finding a solution to this game challenge, we can drastically cut down on the resources needed to run mobile devices.

Resource allocation is a problem that can be overcome with the help of opportunistic theory. Existing opportunistic techniques have been applied to many areas of edge computing, with

some exhibiting promising performance. One such example is the optimal technique for task offloading in the mobile edge computing environment[58]. Using this method, mobile devices can locate an appropriate virtual computer to perform the work fast and efficiently while conserving power. An opportunistic compute offloading strategy for a mobile edge cloud. The amount of raw data, privacy configurations, context data, and other factors were analyzed to prove that the suggested system is an appropriate execution model for mobile devices. By using this method, we can drastically cut down on both processing time and energy use. Furthermore, Gao provided an analytical framework that makes use of peer mobile devices at the tactical edge (when they move into communication range with one another). This framework can improve task completion ratio and completion time while reducing power consumption by simultaneously addressing the energy consumption and data transmission delay of computational job execution [60].

With the right offloading mechanism in place, shifting compute from the central cloud to the network edge can assist reduce transmission delay. There have been various investigations into the best practices for computing offloading. Using DVS technology in a mobile edge computing setting, investigated the problem of partial computation offloading. Energy consumption minimization (ECM) of smart mobile devices and latency minimization (LM) of application execution are two optimization challenges that are formalized in their research. The developed system allows for improved performance in terms of energy efficiency, latency, and admission probability on mobile devices.

[61] [62] have suggested a multi-user computation offloading technique and a sequential offloading game concept for a multi-cell MEC scenario. In their plan, mobile users account for both ongoing interference and available computational resources while deciding whether or not to do offloading. This allows mobile users to experience less latency and power drain. Clustered network service chaining [63]. Using this method, we can determine how many clusters should be used to get the best possible service throughput. To modify the popular Honeybee work-stealing method [64] presented a work-sharing model for mobile edge-clouds.

However, there are additional programs that aim to maximize anything other than the bottom line. One example is the PRofIt Maximisation Avatar pLacement (PRIMAL) approach for mobile edge computing [65]. This plan allows one to find the best balance between the benefits of relocation and the costs. To lessen the lag in reaction time at the end of the queue for automotive applications[66] suggested a technique that selectively deploys redundancy. To determine which cloud, roadside, or mobile phone platform will provide the quickest reaction time, we can use passive measurement and historical data to estimate network latency and computing times for offloaded sensor processing [67], Rodrigues et al. suggested an analytical approach to reduce service latency in a distributed cloud computing setting at the network's periphery. This paradigm allows for the delay in processing to be managed by the migration of virtual machines and the delay in transmission to be enhanced through the modification of transmission power. This allows for the fastest possible completion of service.

2. Bandwidth: The vast amount of data produced by the IoT is a direct result of the proliferation of sensors it employs. The data must be compressed or otherwise processed before being uploaded to the cloud. The network's resources will be severely taxed by this influx of data, resulting in performance drops and packet losses. Therefore, IoT gateways are required to do preliminary processing and maybe aggregation of data before it is transmitted to remote cloud servers. In order to reduce the bandwidth requirements of end users without sacrificing data quality, it is necessary to optimally move data processing and aggregation operations.

Many investigations have been done to better understand this issue. Abdelwahab et al. [68] presented, for example, an LTE-aware edge cloud architecture and REPLISOM, a mechanism for replicating memory optimized for LTE. The designed protocol may effectively arrange memory replication tasks. This facilitates the resolution of interference caused by the simultaneous use of the same radio resources by different devices. [69]

introduced a method for centralized and distributed stream processing that they called SpanEdge. This approach allows for the efficient distribution of stream processing applications across a globally distributed network, which improves both throughput and latency.

[70] designed a mobile edge computing offloading architecture for cloud-enabled vehicle networks. In this paper, we present a contract-based method for allocating computational resources. This method can be used to satisfy the tasks' offloading requirements, maximizing the value of MEC service providers while decreasing the latency and transmission cost of computing offloading.[71] suggested a real-time context-aware ad hoc collaboration system using revolutionary 5G communication topologies and mobile edge computing principles. Therefore, it can be used in geographically constrained low-latency settings.

[72] offered an adaptation to the stream processing architecture that takes into account topology-external interactions (interactions with databases, users, critical actuators, and more). Reduced latency and bandwidth requirements between the cloud and the edge are possible.

3. Energy: IoT end devices may have drastically varying network capacities, power capacities, and battery lives. This highlights the significance of carefully considering the context in which a given end device will be processing or forwarding data. It is important to get the most out of end devices, especially those with limited battery life. One way that edge computing may aid in this would be through a modifiable workload offloading mechanism that takes into consideration the power resources of each device.

The energy crisis has been the subject of extensive research. [73] suggested utilizing fog computing to host virtual medical device applications within healthcare cyber-physical systems. Considering the association of communication base stations, the allocation of subcarriers, the deployment of virtual machines, and the distribution of tasks, a low-complexity two-phase linear programming-based heuristic algorithm is proposed to solve the mixed-integer linear programming problem. This method boosts application performance while decreasing operating [74]. By using a min-cost mixed-cast flow problem, the method proposed here solves the issue of service distribution in the investigated network. It is demonstrated that after the proposed issue is fixed, smart IoT services may reduce power consumption by more than 80%.

[75] proposed an energy-efficient computation offloading strategy to address the optimization problem. The offloading technique for 5G mobile edge computing (MEC) in such networks consumes less energy as a result. In this analysis, we consider the energy costs of completing tasks and delivering files. According to [76], Mao et al. propose a dynamic computation offloading (LODCO) scheme based on Lyapunov optimization for application in an energy harvesting system for a sustainable MEC. This straightforward online technique improves accuracy while decreasing the algorithm's execution time and cost. A strategy for optimizing both radio and computational resources in a multicell mobile-edge computing environment [77]. This approach allows us to reduce the overall energy usage of our users while still meeting our latency goals.

4. Overhead: Each data packet has a header overhead and a payload when being sent over a network. Although most data packets in the Internet of Things are quite tiny, a large number of IoT devices can nevertheless cause considerable network overhead. One of the main challenges for edge computing is finding ways to reduce network overhead. Edge/cloudlet servers help reduce overhead by grouping and processing low-priority packets in advance [78] presenting a cross-layer approach to reduce unnecessary data transfers and boost 5G mobile networks' transmission efficiency.

B.Storage: Cloud storage often comprises a hierarchical structure of connected, low-cost servers and disc drives shown in Fig.4. It is at the center of the network and connects all the other nodes. Decentralized at the network's edge, edge computing-based storage works similarly to how some edge nodes are responsible for satisfying storage needs. It's similar to grouping hard discs into a cluster, except that the data is stored throughout all of the network's edge nodes.

Using load balancing and failure recovery methods, storage based on edge computing can deliver the performance and availability needed to meet QoS standards. By distributing data storage

to a number of edge nodes, these load-balancing techniques can help ease congestion in the network. Data problems (including software, hardware, packet loss, noise, and power difficulties) can be difficult to spot in a massive data flow from multiple sources, making failure recovery strategies essential to edge computing storage.

IoT devices often have very little onboard storage space. All information gathered or generated by the gadgets must be transmitted to a central server. Furthermore, there are thousands of IoT devices simultaneously creating massive amounts of data. If every connected device tried to access cloud-based storage simultaneously, it would cause a catastrophic overload of the network. For instance, Microsoft's ''Live Video Analytics'' project generates enormous amounts of data that must be immediately transported to storage and integrated into the analysis process.

Cloud-based data sensors or cameras would never be able to achieve such perfection. However, due to the nature of edge computing storage, delivering data to multiple edge storage nodes will decrease network traffic over long distances [79].

The usage of storage balancing solutions is essential for realizing edge computing-based storage for managing geographically dispersed IoT devices with changing data streams, probabilities, and locations. For instance, to solve the issue of IoT storage, [2] proposes a resource allocation technique and a satisfaction function. In this scenario, the satisfaction function can be used to check if there are sufficient resources to fulfill the order. Another method that keeps an eye on storage demand rates and employs data stream replication to keep things even is the MMPacking balance technique given in [80][81]. Specifically, the scheme's ability to discard unnecessary data packets is its most appealing feature. Storage balancing in edge computing-based storage can reduce storage time by giving preference to the closest edge storage nodes or using a rating and weighting method for storage processing.

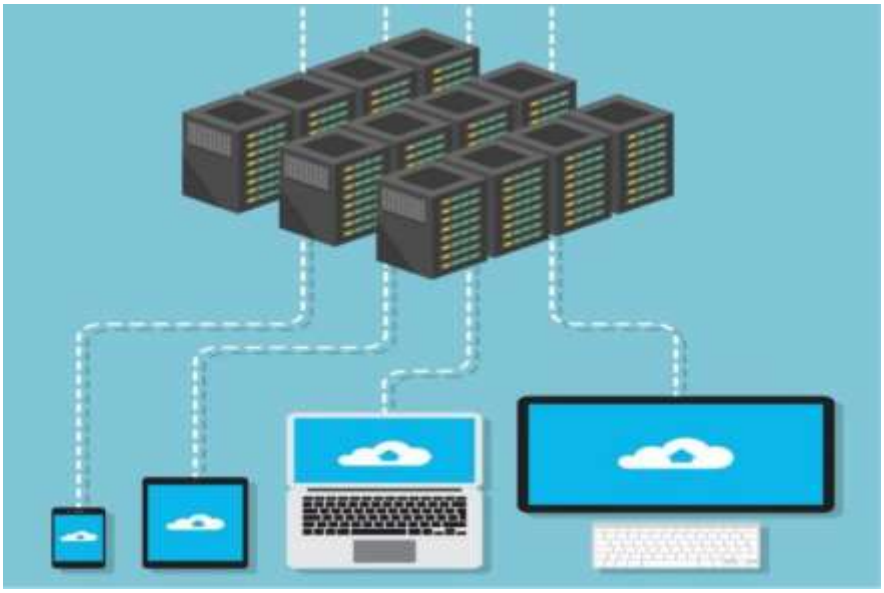


Fig. 4 How does Cloud Storage work

This allows the ''Live Video Analytics'' to fulfill the service's requirements by uploading data to the nearest edge storage nodes using edge computing. In the meantime, if a video packet is identical (for instance, the frame(s) are similar), the system will have a look at it and get rid of any duplicates.

Plan No. 2 for RevivalThe recovery strategy is a vital requirement for edge computing storage systems, as reliability is essential for storing and recovering accurate data representations. In order to ensure the reliability of the system, it will check if the storage nodes are online, replicate the data, and make use of other nodes for redundancy.
Initially, Obtainability

A storage service outage could occur for a number of reasons. Monitoring software will often ping, or "heartbeat," storage systems at predetermined intervals to check on their status and identify online edge nodes. Data centers will inevitably go down at some point. The system may go down for maintenance, a network device may be unavailable, the operating system on the edge storage node may crash or restart, or the discs' authority may be altered or deleted by the automated repair mechanism. According to studies and observations [82], less than 10% of failures last for longer than 15 minutes. Backup storage servers are used by cloud-based systems to address this problem. However, in edge computing storage systems, the other accessible edge nodes will act as backups. The ever-increasing number of linked devices in IoT environments makes persistent data storage a necessity. Therefore, it is essential to select a reliable storage service [83].

Several other methods of measurement [82][84] have been proposed as answers to this issue. They have their pick of many different options for storage facilities.
Second, informational redundancy

A large number of linked devices introduces the persistent demand for data storage in IoT systems. Sensitive data includes things like medical information, power bills, the speeds of smart cars, and traffic conditions. Therefore, Internet of Things (IoT) settings must be integrated into distributed storage systems to assist with the massive demand and ensure the integrity of the stored data.

Replicas are a great way for distributed storage systems to increase their MTTF and dependability [85]. The size and number of code blocks that make up each data chunk in a distributed storage system are both predetermined [19][86]. Furthermore, there is a fixed degree of similarity between all the datasets. Information stored on one component can be used to recreate another component by examining its direct connections [82]. Storage in the network's periphery is, in every sense of the word, a distributed storage system. Edge computing-based storage support makes it possible to make several copies of critical IoT data and store them in separate locations throughout the globe. The potential for data loss has been greatly reduced.
At last, the cloud having

Cloud Storage Types

Cloud services let anyone store and access digital data. Cloud storage is a virtual hard drive. Cloud storage can store vital data like Word documents and videos, handle complex data, and execute programs. User use cases determine the optimum cloud storage. Let's examine cloud storage options in Fig 5 shows Types of Cloud Storage.

private cloud storage

Enterprise or internal cloud storage is private. Data is saved on the company intranet. Company firewalls secure this data. Companies with pricey data centers and in-house data privacy can benefit from private cloud storage. Private cloud storage gives users full control. However, maintaining and updating private cloud storage is costly and time-consuming. Host companies manage private cloud storage.

Public cloud storage

The user and authorized users can access public cloud storage online with little administrative controls. Public cloud storage eliminates system maintenance. Since public cloud storage is hosted by multiple solution providers, security fields are shared by all users and cannot be customized. AWS, IBM Cloud, Google Cloud, and Microsoft Azure are popular public cloud storage providers. Public cloud storage is scalable, inexpensive, reliable, and requires no maintenance.

Hybrid cloud

Hybrid cloud storage combines private and public cloud storage. Hybrid cloud storage gives users the

security and customization of private and public clouds. Cloud computing services can store data in the private cloud and process it in the public cloud in a hybrid cloud. Hybrid cloud storage is cheap, customizable, and user-controlled.

Community cloud storage

Community cloud storage provides private cloud options for organizations and communities. Cloud storage providers address community needs with their cloud architecture, software, and other development tools. For security and compliance, the community's own cloud stores data. Health, financial, and legal firms with tight compliance requirements should use community cloud storage.

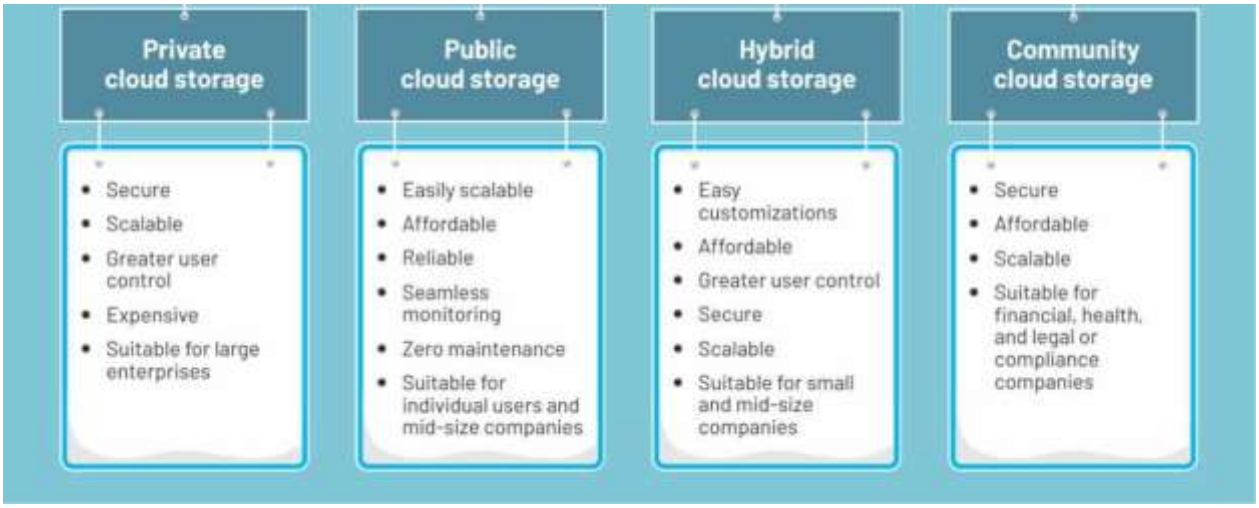


Fig. 5Types of Cloud Storage

C. Computation: When compared to cloud servers, the processing power of an individual edge node is lower in edge computing. Therefore, it is necessary to distribute the compute jobs over multiple edge nodes to satisfy the same requirements. Edge computing, in which processing and data storage is moved to the network's periphery to better serve end users, relies heavily on a well-designed task scheduling scheme. Scheduling methods for tasks can be created in general with a variety of goals in mind. Several strategies for enacting the task schedule in edge computing are discussed below.

1. Computer work sharing: Computing efficiency can be improved using edge computing by moving various calculation jobs to optimal areas.

IN-GROUP: Embedded chips have decreased in price and gained widespread use in today's Internet of Things systems. This means that end devices' processing power has increased dramatically. As a result, users may carry out some computing in the M2M network, which is comprised of a collection of IoT terminals. The quickest reaction time for users is achieved when many devices are located close together.

CLOUDLET, EDGE: Although certain computer resources are made available via the M2M network of end devices, this is not enough to meet the demands of all end users. As a result, the IoT relies heavily on edge/cloudlet servers for network provisioning. The most important challenge is how to schedule tasks on edge/cloudlet servers to make this happen.

Finding the best possible allocation of resources among a set of available servers is the goal of task scheduling for edge/cloudlet servers. The ideal solution to this problem will yield the lowest possible values for the delay of computation and transmission, the energy required for computation and communication, and the bandwidth needed by IoT applications.

CLOUD: Some data processing and storage jobs necessitate more resources than can be reasonably provided by either M2M or Edge/Cloudlet without consuming all of the available resources. In this scenario, the conventional cloud server infrastructure is required for both processing and storage. Since cloud servers boast the most processing power on the network, any work done on them will benefit from the lowest possible computational delay. Since cloud servers are located at a considerable distance from end devices, they also have the

longest transmission latency. As a result, finding a happy medium between computational and transmission delays is a significant obstacle to overcome.

2. Price strategy: Users of edge computing can get the data storage, processing power, and network connectivity for their calculation activities from edge/cloudlet servers or even from other users. So, with the right pricing mechanism in place, we may deduce allocation strategies for network resources.

Single Supplier: Edge/cloudlet servers' computing and communication resources are often handled by a single service provider. In other words, the service provider will determine the rates for the edge/cloudlet servers located at varying distances from the end devices, taking into account their computing and communication capacities. Customers can save money by shifting their workloads to the most cost-effective edge or cloud servers.

To best manage the edge computational resources based on the pricing policy in a delay-aware mobile edge computing environment [87], the edge cloud's revenue can be increased to its full potential. Not only that, but the results of this study show that the cost of the distant cloud can influence the benefit of the edge cloud under certain circumstances. Using the concepts of a "field," "shallow," and "deep" cloudlet, [88] a hierarchical model. Heuristic algorithms and a centralized approach are investigated to achieve time-scale optimization for resource allocation and to solve the convex optimization problem for bandwidth allocation.

Various suppliers of the same service: Because the Internet of Things links together so many distinct devices owned by different people and organizations, its associated processing and storage resources may not be controlled by a single entity. This means that consumers that need data processing activities must contract with various edge computing service providers and pay for the requisite resources. Third parties will be more likely to provide their computer or storage capacity to IoT if they know they will be compensated fairly for their efforts. In addition, the companies offering these edge computing services will both compete and work together. As a result, emerging edge computing networks need to work on the disparate pricing strategies of their constituent service providers. Auctions are examples of economically motivated strategies that could be used in this context for resource management [89].

3. Priority: Edge computing's compute work scheduling also takes priority into account. Prioritization helps maximize the overall benefits of various Internet of Things applications[90]. To improve overall network performance, apps with higher resource requirements, such as multimedia P2P downloading, might be given a lower priority while real-time IoT applications, like monitoring applications, are given a higher priority. For instance, [91] presented a system for monitoring where and how fast users can access web items hosted on edge servers [92], taking into account both local processing energy and channel gains.

V.IoT at the Periphery: The Challenges of Edge Computing

As we've seen, there are a lot of reasons to use edge computing to help the Internet of Things. In this section, we'll talk about some of the difficulties that come with using Edge Computing for IoT.

Integrating Systems: As more and more Internet of Things (IoT) devices and services become available, the edge computing environment is put to the test. Computing at the network's edge makes use of several different kinds of hardware. The system is inherently intricate and multifaceted. Therefore, it will be difficult to write programs and manage data for the many various kinds of applications that run on the many different kinds of heterogeneous platforms in many different places.

All user applications and software are hosted on remote servers and run in the cloud. Cloud service providers like Google and Amazon are responsible for deploying and keeping tabs on all of your software and hardware. The vast majority of people that utilize these tools do not understand how they work or what goes into their creation. This is possible with cloud computing thanks to the centralized nature of the service and the ease with which it can be managed. Since cloud apps are typically delivered on a single cloud provider, developers only need to be fluent in one language

while writing code for several platforms.

Edge computing is very different from cloud computing. Despite the benefits of a dispersed topology, issues may arise due to the varied nature of edge node platforms. For developers, the effort of building an app that can be deployed and run on an edge computing platform is fraught with difficulties. Even while there are existing approaches to addressing the programmability problems of edge computing, such as [36], none of them consider IoT-centric use cases. The first step in the IoT is called edge node discovery [93], and until it is finished, no IoT device will know what kinds of platforms are set up in its near neighborhood. A large amount of server-side software also has to be deployed to the edge nodes. Thus, how edge node providers deploy and manage such server-side apps presents yet another challenging challenge [94] [95].

Several operating systems are employed on numerous varieties of storage servers for data management. This creates difficulties in several areas, including file naming, resource allocation, file reliability management, etc. With so many Internet of Things devices simultaneously creating and sending data, it can be confusing to keep track of everything. Cloud computing and contemporary networks make use of a wide variety of naming systems, some of the most well-known being DNS (Domain Name Service) and URI (Uniform Resource Identifier). Regardless, dynamic edge computing networks and the Internet of Things are not a good fit for these architectures. Furthermore, edge nodes in a multi-source and multi-task environment may not be able to afford an IP-based naming strategy, ruling out its usage as a naming strategy.

Two such examples of innovative naming approaches created for edge computing are Named Data Networking (NDN) [94] and Mobil-first [95].

One such system is the Named Data Network (NDN) naming scheme [94], which provides a hierarchical name for the distributed network and is simple to manage for edge node owners. However, combining different protocols requires the addition of a proxy server to the network. To use the NDN naming scheme, you need to know where the hardware came from, which could compromise security. The MobilityFirst naming method [95] is designed to make it easier to move around by separating a device's name from its IP and MAC address. The MobilityFirst approach is flawed because of the inconvenient requirement for globally unique identification (GUID).

Management of resources: For IoT and edge computing to work together, a thorough and complete understanding of resource management, as well as its optimization, is required. IoT devices will be severely impacted by network congestion and latency due to the increased energy consumption associated with retransmitting data after an interruption. In order to reduce device latency, edge computing, which uses the most locally accessible computation and storage resources, is becoming increasingly important. Decentralized resources will play a significant role in both inspiring and distributing these assets.

In order to be implemented, a strategy for managing these resources must be both computationally efficient and practical. However, there is a considerable lot of complexity brought about by the interdependencies between the numerous service providers, devices, and software. The same factors that drive smart systems also encourage edge/IoT resource management. Allocating, sharing, and pricing the direct service of a system with several resource providers and a wide variety of applications and users can be determined by maximization or optimization of global welfare or other metrics, competitive bidding, or other techniques [96].

First, an auction-based system can be used to efficiently allocate network resources. The broad use of auction methods in computer science research has helped numerous fields, including mobile and cloud computing [97, 99], as well as smart systems [5, 90, 100, 101]. To be effective for peripheral network resource management, auction techniques need to guarantee secure, anonymous bidding on services according to user preferences and the value of their bids. In the context of edge computing and the Internet of Things, it is necessary to consider auction techniques for hiding users from service providers and distributing services equitably. Providers stand to gain financially from optimizing resource utilization. The premise behind this architecture is

that there are two distinct sorts of service providers for cloud and edge computing and that these two groups also host distinct types of edge nodes. Considering the likely existence of complex webs of interconnected networks, subnetworks, ad hoc networks, etc., Quality of Service (QoS) necessitates effective control of the routes taken by vast amounts of network traffic.

As it is currently envisioned, optimization can also be used to control the allocation and division of resources in edge computing. The goal of optimization, like that of auction-based approaches, is to maximize the social or financial benefit of a system by drawing attention to its best features. Local edge systems can't always rely on subscription or patron services since edge infrastructures function as a go-between for end users and cloud services. In the context of cloud and edge computing, optimization is proving to be a viable alternative to auction methods for managing resources [102][103].

Privacy and Security : Privacy and security are always shifting goals that must be taken into account. When building an IoT network using Edge Computing, these are the most pressing issues to address. Due to the complex interdependence of numerous technologies (peer-to-peer systems, wireless networks, virtualization, etc.), edge computing necessitates the adoption of a holistic integrated system to protect and manage each technology platform and the system as a whole.

Although this is edge computing's ultimate goal, it will undoubtedly lead to some novel security issues. In unexplored contexts, such as the cooperation of heterogeneous edge nodes and the movement of services between global and local dimensions, new routes of malevolent behavior may emerge. What security and privacy solutions are feasible may also depend on the core characteristics of edge computing. Edge computing systems have many of the same security concerns and challenges as cloud computing environments.

Potentially useful for the Internet of Things is the distributed architecture. However, keeping separate buildings safe and private is a significant challenge. In the future, the Internet of Things may rely on a secure computer infrastructure that can be provided via edge computing. Data processing at the edge employing edge computing raises concerns about the misuse of personally identifiable information about end users. The edge nodes that store the sensing data from IoT devices are especially at risk [49]. Therefore, edge computing should take into account privacy protections like local differential privacy [104] and differential privacy with high utility [49].

When designing an IoT ecosystem that makes use of edge computing, it is important to keep user privacy in mind [105].

When it comes to security, one of the most typical concerns in edge computing is authenticating gateways on several levels. For instance, each smart metre installed in a house is given its distinct IP address [106].

By providing false data, altering the other users' data, tampering with their smart metre, or spoofing IP addresses, a malicious actor in this situation might wreak havoc on energy management in IoT systems (smart grid, etc.). Since different people are responsible for the upkeep of different edge nodes, it can be difficult to implement a unified security strategy across the board[107][108][109].

1. TRANSMISSION :Some attacks (jamming attacks, sniffer attacks, worm propagation, resource-depletion denial-of-service, and others [110] could be launched during message transmission between end users and servers to disable the links by congesting the network or could monitor network data flow. In a typical network, an administrator's configurations must be reliable and verified [111] [112].

However, Edge Computing-based IoT is installed at the periphery of the network architecture, making it difficult to control diverse networks like Mobile Wireless Networks [113], Ultra-Dense Networks [33] [114], and Wi-Fi.

Therefore, it is extremely difficult to separate management traffic from typical data traffic when tasked with controlling edge networks. In such a scenario, enemies would have a simple time taking command of the network [111].

The implementation of Software-Defined Networking (SDN) [40][115] is essential for resolving this issue. The aforementioned security vulnerabilities can be reduced by SDN in the following ways:

Monitoring data traffic and scanning data packets for applications to detect malicious code is made possible by deploying a network monitoring and intrusion detection system (IDS). Edge Computing-based IoT benefits greatly from SDN's ease of deploying an IDS system and better managing traffic flow.

II.Protection: Traffic isolation and prioritization is the most effective way to safeguard information during transmission.

method. This is where software-defined networking (SDN) comes in handy, as it can simply employ VLAN ID to divide traffic into VLAN groups, thereby further isolating malicious data. Attacks that try to overwhelm a network's shared resources and hardware are typically thwarted by isolating and prioritizing traffic.

II. Reactions: There are continuous efforts to assess and prevent cyber-attacks in edge computing environments [107] [109] [116], building on a long tradition of conventional defenses against network risks in cyber-physical systems [106].

2. STORAGE: Edge Computing-based IoT relies on a wide variety of third-party providers to handle the vast amounts of data produced by the many sensors and devices. Users' data is sent to third-party storage providers, whose equipment is dispersed across a wide geographic area at the network's periphery. There are many reasons why this raises the possibility of attacks. First, it's hard to ensure data accuracy because data is stored in many different places and can easily be misplaced or corrupted if data packets are lost. Second, there is always the risk that enemies or unauthorized users would tamper with or exploit the data you upload to storage, which could result in leaks or other privacy concerns.

Fortunately, there are ways to combat these issues and achieve integrity, secrecy, and verifiability for edge storage systems, such as by employing homomorphic encryption [117] [118]. Additionally, user security is improved to the point that data can be stored on any untrusted servers [112]. In [119], for instance, the authors advocated using a third-party auditor (TPA) to conduct public audits in a way that protects users' privacy while still ensuring the security of their cloud-based data. Even edge storage can benefit from this method.

Cloud file search results can be trusted thanks to the protocols proposed in [120]. In this research, we create two different protocols. In cases where all users have access to the same files, one option is to let them check the accuracy of the search results. The other option is to think about what would happen if users accessed files with varying levels of permissions. A user-trusted TPA in an edge computing setting can boost storage system security while decreasing administrative complexity. TPA employs homomorphic encryption and the random mask approach to safeguard its data.

Maintaining data integrity is another difficult aspect of data storage. Erasure codes and network coding have traditionally been used to discover and repair faulty data in storage systems, but both methods are time-consuming and resource-heavy to implement. For instance, secure coding-based storage [121], this storage method incorporates Luby transform (LT) code into programs, which speeds up data searches while decreasing overhead in terms of both store space and communication time.

From a management standpoint, network Resource Access Control (RAC) is the most efficient solution for securing data in edge storage. A secure network resource access system employs terminals to get entry to network resources behind corporate firewalls. To be more specific, a polling server resides inside the firewall while a proxy server sits outside and collects application data from a terminal. Initiating data transmission from the proxy to the polling server, receiving application data and accompanying network resource data, and routing application data to appropriate network resources based on resource data are just a few of the many tasks performed by the polling server.

3. COMPUTATION :Keeping computational activities private and secure while uploading them to edge computing nodes is a significant difficulty in Edge Computing-based IoT.

Generally speaking, Verifiable Computing allows an untrusted compute node to offload the computation chores, and it was introduced for Edge Computing-based IoT to ensure computation security [122]. In the meantime, this node keeps track of the verifiable results and utilizes them to

check whether or not the computation was finished correctly by comparing them to the results calculated by other reliable computation nodes. Each IoT device, when using an IoT solution based on Edge Computing, should be able to independently validate the accuracy of the results computed by edge nodes [123], a technique called Pinocchio was developed to enable clients to validate computed results using only cryptographic assumptions. In Pinocchio, clients specify the computation task by generating a public evaluation key, and servers verify the correctness of the task by comparing the value of the key with the outcome of the computation. The verifiable computing protocol [122], is conceptually comparable to this. Clients can use this protocol to ensure that the calculated job is computationally sound by checking the returned result from the computation nodes.

Managing and securing networks with such a high number of connected devices is a complex and tough task due to the decentralized nature of edge network management, which cannot deliver suitable security and management features. To protect the network's periphery [124]. The developed system can provide security services by detecting and responding to hostile activity in the system using the security and network management features provided by the proposed system.

Due to the shift of processing power from the cloud to the network's periphery, trust must be established between edge servers and end devices. Solutions to this problem can be used to broaden integrity measurement and attestation systems.

using edge devices' limited resources and restrictive operating environments to gather evidence of data integrity. Identity-based cryptography and secure key exchange in the field as a trusted identity solution in disconnected contexts that do not rely on a trusted third party. Most risks in disconnected environments can be repelled by implementing the necessary controls at the application, OS, network, and site levels.

Several additional preliminary studies attempt to tackle challenges related to security, such as software verification and malicious intrusion detection. An example is the BUFS methodology [123], which is a bottom-up and foundational way to ensure the safety of an IoT system's software stack [122] against hostile device-to-device (D2D) communication. HoneyBot nodes may detect and quarantine direct-to-direct insider attacks using this manner. More- over, it has been demonstrated that the quantity and distribution of HoneyBot nodes in the network can have a major effect on the precision and velocity metrics.

Edge computing infrastructures must be exploited to aid in threat analysis and detection [25] to further improve the efficiency of threat analysis and detection and to lessen the performance effect of threat analysis and detection in edge computing-based IoT systems. Examples include researching how edge computing might be used to speed up the process of detecting threats to IoT and smart devices (by doing things like rapidly learning the profile of threats in parallel) [19][31] [32].

Three generic defense strategies proactive defense, reactive defense, and predictive defense should inform the design of an integrated defense system against cyber threats on edge servers. In particular, it is essential to create approaches at both the data-level and the system-level for proactive defense. The detection should be efficient in terms of overhead (time, code, mem- ory, computation, I/O, storage, architectural heterogeneity, and others), with a focus on how edge resources are used and allocated. It's important to think about data-level processes (such as self-correcting data to find and restore lost or stolen computational data). To effectively and proactively find exploitable vulnerabilities, the edge system's monitoring and detection technologies must be created and incorporated into the edge computing infrastructure. Techniques should be designed at the data and system levels for successful reactive detection. Additionally, at the data level, techniques like low-cost data attestation mechanisms [19] shall be examined, as they can confirm the integrity of data processing results and identify malicious nodes based on inconsistency in results. Machine learning (including deep learning) based anomaly detection approaches are promising and should be evaluated at the system level. In addition, anticipatory defense methods can benefit from

the use of machinesPredicting the actions of novel threats to the system is an additional goal that must be built into the design of learning-based systems .

Advanced means of connectivity : Edge computing is breaking down barriers to the development of low-latency, high-computing applications, marking a departure from the traditional model of faraway computing and storage. Ultra-Dense Networks (UDNs), massive Multiple-Input and Multiple-Output (MIMO), and millimeter-wave are just some of the technologies of future 5G cellular networks that are constantly evolving to lower latency, boost throughput, and support highly interconnected communities of users [33] [115]. Edge computing will continue to develop as a result of the inevitability of integrating these new connectivity technologies [34].

Communication via a 5G network is the cutting edge of modern technology. Its purpose is to ensure that users are always connected to the internet and have access to the data they require

[114]. Therefore, adaptable and effective communication can be achieved by the combination of 5G, IoT, and edge computing. Furthermore, many Internet of Things applications can benefit from 5G technology's ability to increase efficiency [33][35].

For instance, effective subscriber state management in 5G scenarios [19]. For 5G, it will be necessary to combine the fog network and cloud radio access network topologies. A VoWiFi solution incorporating edge computing technology can assist overcome its primary shortcoming (i.e., the absence of user location). VoWiFi user location data can be retrieved with the proposed approach. Proactive content caching in 5G wireless networks, and they suggested doing it with the help of big data. In the context of 5G networks. This infrastructure allows for better authorization controls and better privacy for sensitive records.

An aide for intelligent systems :To achieve system awareness and subsequent remote control, smart systems must inextricably interweave network communication technologies with sensors and actuators [2] [3]. There are countless possibilities for data collection, physical system management, and resource allocation and optimization made possible by the integration of sensing devices.

Smart grids, smart cities, smart transport, and smart health are only a few of the most important applications of smart systems. As more and more systems become intelligent, edge computing can supply computationally inadequate components with the lowest latency computing and storage. In a similar vein, performing data processing at the network's periphery can provide the most resistance against vulnerable systems.

1. SMART GRID :Power grid technology has advanced, but the Smart Grid is seen as the next step. With the goal of

To collect and communicate measurement data in the smart grid [107], a vast number of smart metres, sensors, and actuators are required. Therefore, edge computing may be able to meet the needs of smart grid adoption. However, there are still unanswered questions about how to use a distributed network of edge servers to interpret data streams from metres and sensors covering vast and diverse regions to make timely, optimal judgments about energy management [7].

There have been previous studies that have gone into this route. A good case in point is the RIAPS (Resilient Information Architecture Platform for the Smart Grid).

2. Connected City :The idea of a "Smart City" has been developed and implemented to better manage urban areas' public resources and raise residents' living standards [8][9]. The lack of compatibility between different types of urban technology is a major obstacle. Large numbers of connected devices can degrade network performance, but with user-generated content and the right algorithms, they can also help identify the incidence of anomalous events.

Research in this area has gotten off to a slow start, but examples like real-time video analysis using edge computing are starting to emerge. For instance, a framework for Edge Video Analysis for Public Safety (EVAPS). Then, we may save power on edge devices by eliminating needless data flows, a smart urban surveillance method that makes use of fog computing. The suggested system can monitor traffic in real-time and detect speeding vehicles using data from a case study.

3. Intelligent Transportation :A cloud-based vehicle control system is necessary for safe and effective autonomous driving because it can collect data

from the sensors through a V2V network. That's why it's able to manage and coordinate a fleet of cars. Managing automobiles in real time requires stringent

needs, such as low latency, that can be met by deploying computers near the network's periphery.

Some studies have been conducted in this field already. To facilitate resource sharing, their system architecture takes into account the state of both edge and cloud servers. The proposed solution can stabilize cloud control and drastically cut down on latency. To reduce traffic and increase productivity on the roads,

VI.Conclusions

As the Internet of Things (IoT) matures, edge computing emerges as a viable answer to the complex problems posed by managing millions of sensors/devices and the associated resources. In contrast to the traditional "cloud" model, "edge" computing will move processing and storing data closer to where it will be used. As a result, edge computing can lessen the volume of traffic, which in turn lowers the bandwidth needs in IoT. In addition, unlike conventional cloud services, edge computing can lessen the transmission delay between edge/cloudlet servers and the end users, resulting in faster reaction times for real-time IoT applications. In addition, the lifetime of nodes with limited battery resources can be extended, along with the lifetime of the entire IoT system, by decreasing the transmission cost of the workload and migrating the computational and communication overhead from nodes with limited battery resources to nodes with significant power resources. In conclusion, we have studied typical IoT applications as a means of illustrating the architecture, performance goals, job offloading strategies, security and privacy issues, and countermeasures of edge computing.

In the future, we will attempt to implement a suggested approach to increase security, privacy, and energy control.

References

[1] Pooja Kumari, ‘‘A comprehensive study of DDoS attacks over IoT network and their countermeasures“Computers & Security, Volume 127, April 2023, 103096.

[2] Suiting Ding, ‘‘Opportunities and risks of internet of things (IoT) technologies for circular business models: A literature review,’’ Journal of Environmental Management., Volume 336, 15 June 2023, 117662.

[3] J. A. Stankovic, ‘‘Research directions for the Internet of Things,’’ IEEE Internet Things J., vol. 1, no. 1, pp. 3–9, Feb. 2014.

[4] Juan Pablo García-Martín, ‘‘Energy consumption analytical modeling of NB-IoT devices for diverse IoT applications,’’ Computer Networks,Volume 232, August 2023, 109855.

[5] Mitsuaki Akiyama, ‘‘Seeing is not always believing: Insights on IoT manufacturing from firmware composition analysis and vendor survey,’’ Computers & Security., Volume 133, October 2023, 103389.

[6] Ahmed Rahman Abdulzahra Al-Obaidi, Seyed Ebrahim Dashti "corresponding author", Saba Atiyah Mashaan, Mohammed Ahmed Kadhim Al-khafaji, “PROPOSED NEW SECURE HYBRID MOBILE CLOUD COMPUTING TO IMPROVE POWER AND DELAY,” , Journal of New Zealand Studies NS35,2023 .

[7] Seyed Ebrahim Dashti, Amir Masoud Rahmani,”Dynamic VMs placement for energy efficiency by PSO in cloud computing,”2015.

[8] Seyed Ebrahim Dashti, Ahmed Rahman Abdulzahra Al-Obaidi, Saba Atiyah Mashaan,”An Overview of Deep Learning Methods in the Internet of Things Technology in regular life,” ,Computer Integrated Manufacturing Systems,2023.

[9] Yair Meidan, ‘‘D-Score: An expert-based method for assessing the detectability of IoT-related cyber-attacks,"Computers & Security,Volume 126, March 2023, 103073.

[10] Liting Deng, ‘‘Enimanal: Augmented cross-architecture IoT malware analysis using graph neural networks,’’ Computers & Security, Volume 132, September 2023, 103323 .

[11] Ramyapandian Vijayakanthan et al., ‘‘SWMAT: Mel-frequency cepstral coefficients-based memory fingerprinting for IoT devices,’’Computers & Security,Volume 132, September 2023, 103298

[12] Singam Sai Bala Subrahmanyam, ‘‘A hybrid method for analysis and detection of malicious executables in IoT network,"Computers & Security,Volume 132, September 2023, 103339.

[13] Huned Materwala, ‘‘Energy-SLA-aware genetic algorithm for edge–cloud integrated computationoffloadinginvehicular networks,"Future Generation Computer Systems,Volume 135, October 2022.

[14] D. Georgakopoulos, P. P. Jayaraman, M. Fazia, M. Villari, and R. Ranjan, ‘‘Internet of Things

and edge cloud computing roadmap for manufactur- ing,'' IEEE Cloud Comput., vol. 3, no. 4, pp. 66–73, Jul./Aug. 2016.

[15] Run Yang, ''Dependable federated learning for IoT intrusion detection against poisoning attacks,"Computers & Security,Volume 132, September 2023, 103381.

[16] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, ''Mobile-edge computing architecture: The role of MEC in the Internet of Things,'' IEEE Consum. Electron. Mag., vol. 5, no. 4, pp. 84–91, Oct. 2016.

[17] Fabrizio Carpi, ''Experimental analysis of RSSI-based localization algorithms with NLOS pre-mitigation for IoT applications,"Computer Networks,Volume 225, April 2023, 109663 .

[18] 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, Oct. 2016.

[19] Mostafa Ghobaei-Arani, Ali Asghar Rahmanian, Mohammad Sadegh Aslanpour, Seyed Ebrahim Dashti '"CSA-WSC: cuckoo search algorithm for web service composition in cloud environments,", Soft Computing,2018.

[20] C. Vallati, A. Virdis, E. Mingozzi, and G. Stea, ''Mobile-edge computing come home connecting things in future smart homes using LTE device- to-device communications,'' IEEE Consum. Electron. Mag., vol. 5, no. 4,pp. 77–83, Oct. 2016.

[21] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief. (Jan. 2017). ''A survey on mobile edge computing: The communication perspective.'' [Online]. Available: https://arxiv.org/abs/1701.01090

[22] W. Shi and S. Dustdar, ''The promise of edge computing,'' Computer, vol. 49, no. 5, pp. 78–81, 2016.

[23] Mizna Khalid, ''Towards SDN-based smart contract solution for IoT access control,"Computer Communications,Volume 198, 15 January 2023, Pages 1-31.

[24] P. Mach and Z. Becvar, ''Mobile edge computing: A survey on architec- ture and computation offloading,'' IEEE Commun. Surveys Tuts., vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[25] W. G. Hatcher, J. Booz, J. McGiff, C. Lu, and W. Yu, ''Edge computing based machine learning mobile malware detection,'' in Proc. Nat. Cyber Summit (NCS), Jun. 2017.

[26] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, ''Internet of Things (IoT): A vision, architectural elements, and future directions,'' Future Generat. Comput. Syst., vol. 29, no. 7, pp. 1645–1660, 2013.

[27] The national intelligence council sponsor workshop. (2008). Intelligence,

S. C. B., 2008. Disruptive Civil Technologies. Six Technologies With Potential Impacts on US Interests out to 2025. [Online]. Available: https://fas.org/irp/nic/disruptive.pdf

[28] K. Rose, S. Eldridge, and L. Chapin, ''The Internet of Things: An overview,'' in Proc. Internet Soc. (ISOC), 2015, pp. 1–53.

[29] F. Wortmann and K. Flüchter, ''Internet of Things,'' Bus. Inf. Syst. Eng., vol. 57, no. 3, pp. 221–224, 2015.

[30] Seyed Ebrahim Dashti, Mohammad Zolghadri, Fatemeh Moayedi,"Improving flexibility in cloud computing using optimal multipurpose particle swarm algorithm with auction rules",2022.

[31] W. Yu, G. Xu, Z. Chen, and P. Moulema, ''A cloud computing based architecture for cyber security situation awareness,'' in Proc. IEEE Conf. Commun. Netw. Secur. (CNS), Oct. 2013, pp. 488–492.

[32] Z. Chen et al., ''A cloud computing based network monitoring and threat detection system for critical infrastructures,'' Big Data Res., vol. 3,pp. 10–23, Apr. 2016.

[33] W. Yu, H. Xu, H. Zhang, D. Griffith, and N. Golmie, ''Ultra-dense networks: Survey of state of the art and future directions,'' in Proc. 25th Int. Conf. Comput. Commun. Netw. (ICCCN), Aug. 2016, pp. 1–10.

[34] M. Agiwal, A. Roy, and N. Saxena, ''Next generation 5G wireless net- works: A comprehensive survey,'' IEEE Commun. Surveys Tuts., vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.

[35] P. Demestichas et al., ''5G on the horizon: Key challenges for the radio- access network,'' IEEE Veh. Technol. Mag., vol. 8, no. 3, pp. 47–53, Sep. 2013.

[36] A. Ahmed and E. Ahmed, ''A survey on mobile edge computing,'' in Proc. 10th Int. Conf. Intell. Syst. Control (ISCO), Jan. 2016, pp. 1–8.

[37] Y. Jararweh, A. Doulat, O. AlQudah, E. Ahmed, M. Al-Ayyoub, and E. Benkhelifa, ''The future of mobile cloud computing: Integrating cloudlets and mobile edge computing,'' in Proc. 23rd Int. Conf. Telecom-mun. (ICT), May 2016, pp. 1–5.

[38] L. Tong, Y. Li, and W. Gao, ''A hierarchical edge cloud architecture for mobile computing,'' in Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM), Apr. 2016, pp. 1–9.

[39] G. Wang, Y. Zhao, J. Huang, and W. Wang, ''The controller placement problem in software defined networking: A survey,'' IEEE Netw., vol. 31, no. 5, pp. 21–27, Sep. 2017.

[40] D. Zhu, X. Yang, P. Zhao, and W. Yu, ''Towards effective intra-flow network coding in

software defined wireless mesh networks,'' in Proc. 24th Int. Conf. Comput. Commun. Netw. (ICCCN), Aug. 2015, pp. 1–8.

[41] Y. Jararweh, A. Doulat, A. Darabseh, M. Alsmirat, M. Al-Ayyoub, and E. Benkhelifa,

''SDMEC: Software defined system for mobile edge computing,'' in Proc. IEEE Int. Conf. Cloud Eng. Workshop (IC2EW), Apr. 2016, pp. 88–93.

[42] P. Du and A. Nakao, ''Application specific mobile edge computing through network softwarization,'' in Proc. 5th IEEE Int. Conf. Cloud Netw. (Cloudnet), Oct. 2016, pp. 130–135.

[43] A. Manzalini and N. Crespi, ''An edge operating system enabling anything-as—A-service,'' IEEE Commun. Mag., vol. 54, no. 3, pp. 62–67, Mar. 2016.

[44] O. Salman, I. Elhajj, A. Kayssi, and A.

Chehab, ''Edge computing enabling the Internet of Things,'' in Proc. IEEE 2nd World Forum Internet Things (WF-IoT), Dec. 2015, pp. 603–608.

[45] T. Lin, B. Park, H. Bannazadeh, and A. Leon-Garcia, ''Demo abstract: End-to-end orchestration across SDI smart edges,'' in Proc.

IEEE/ACM Symp. Edge Comput. (SEC), Oct. 2016, pp. 127–128.

[46] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, ''Fog computing: A platform for Internet of Things and analytics,'' in Big Data and Internet of

Things: A Roadmap for Smart Environments (Studies in Computational Intelligence), vol. 546. Cham, Switzerland: Springer, 2014, pp. 169–186.

[47] H. Jiang, F. Shen, S. Chen, K.-C. Li, and Y.-

S. Jeong, ''A secure and scalable storage system for aggregate data in IoT,'' Future Generat. Comput. Syst., vol. 49, pp. 133–141, Aug. 2015.

[48] M. M. Hossain, M. Fotouhi, and R. Hasan,

''Towards an analysis of security issues, challenges, and open problems in the Internet of Things,'' in

Proc. IEEE World Congr. Serv. (SERVICES), Jun. 2015, pp. 21–28.

[49] X. Yang, T. Wang, X. Ren, and W. Yu,

''Survey on improving data utility in differentially private sequential data publishing,'' IEEE Trans. Big Data, to be published.

[50] G. Ananthanarayanan et al., ''Real-time video analytics: The killer app for edge computing,'' Computer, vol. 50, no. 10, pp. 58–67, 2017.

[51] G. Ananthanarayanan, V. Bahl, and P. Bodík. (2017). Microsoft Live Video Analytics.

[Online]. Available:
https://www.microsoft.com/en-us/research/project/live-video-analytics/

[52] Seyed Ebrahim Dashti, Hoasain Zare ," Increase the Efficiency of the Offloading Algorithm in Fog Computing by Particle Swarm Optimization Algorithm,", Journal of Intelligent Procedures in

Electrical Technology,2023.

1[53] J. C. Niebles and L. Fei-Fei, ''A hierarchical model of shape and appear- ance for human action classification,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2007, pp. 1–8.

[54] G. Botella and C. García, ''Real-time motion estimation for image and video processing applications,'' J. Real-Time Image Process., vol. 11, no. 4, pp. 625–631, Apr. 2016. [Online]. Available: http://dx.doi.org/ 10.1007/s11554-014-0478-y

[55] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, ''Delay-optimal computation task scheduling for mobile-edge computing systems,'' in Proc. IEEE Int.

Symp. Inf. Theory (ISIT), Jun. 2016, pp. 1451–1455.

[56] I. Ketykó, L. Kecskés, C. Nemes, and L. Farkas, ''Multi-user computation offloading as multiple knapsack problem for 5G mobile edge computing,'' in Proc. Eur. Conf. Netw. Commun. (EuCNC), 2016, pp. 225–229.

[57] Y. Liu, S. Wang, and F. Yang, ''Poster

abstract: A multi-user computation offloading algorithm based on game theory in mobile cloud computing,'' in Proc. IEEE/ACM Symp. Edge Comput. (SEC), Oct. 2016, pp. 93–94.

[58] Shahrokh Vahabi, Mahmood Lahabi, Mohammadreza Eslaminejad, Seyed Ebrahim Dashti," Geographic and clustering routing for energy saving in wireless sensor network with pair of node groups," Journal of Communication Engineering,2019.

[59] M. H. ur Rehman, C. Sun, T. Y. Wah, A.

Iqbal, and P. P. Jayaraman, ''Opportunistic computation offloading in mobile edge cloud comput- ing environments,'' in Proc. 17th IEEE Int. Conf. Mobile Data Man- age. (MDM), vol. 1. Jun. 2016, pp. 208–213.

[60] W. Gao, ''Opportunistic peer-to-peer mobile cloud computing at the tacti- cal edge,'' in Proc. IEEE Military Commun. Conf. (MILCOM), Oct. 2014, pp. 1614–1620.

[61] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, ''Mobile-edge com- puting: Partial computation offloading using dynamic voltage scaling,'' IEEE Trans. Commun., vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

[62] M. Deng, H. Tian, and X. Lyu, ''Adaptive sequential offloading game for multi-cell mobile edge computing,'' in Proc. 23rd Int. Conf. Telecom-mun. (ICT), May 2016, pp. 1–5.

[63] Y. Nam, S. Song, and J.-M. Chung,

''Clustered NFV service chaining optimization in mobile edge clouds,'' IEEE Commun. Lett., vol. 21, no. 2,pp. 350–353, Feb. 2017.

[64] N. Fernando, S. W. Loke, and W. Rahayu,

''Computing with nearby mobile devices: A work sharing algorithm for mobile edge-clouds,'' IEEE Trans. Cloud Comput., to be published.

[65] X. Sun and N. Ansari, ‘‘PRIMAL: PRofit maximization avatar placement for mobile edge computing,’’ in Proc. IEEE Int. Conf. Commun. (ICC), May 2016, pp. 1–6.

[66] H. Lee and J. Flinn, ‘‘Reducing tail response time of vehicular appli- cations,’’ in Proc. IEEE/ACM Symp. Edge Comput. (SEC), Oct. 2016, pp. 103–104.

[67] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, ‘‘Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control,’’ IEEE Trans. Comput., vol. 66, no. 5, pp. 810–819, May 2017.

[68] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, ‘‘REPLISOM: Disciplined tiny memory replication for massive IoT devices in LTE edge cloud,’’ IEEE Internet Things J., vol. 3, no. 3, pp. 327–338, Jan. 2016.

[69] H. P. Sajjad, K. Danniswara, A. Al-Shishtawy, and V. Vlassov, ‘‘SpanEdge: Towards unifying stream processing over central and near-the-edge data centers,’’ in Proc. IEEE/ACM Symp. Edge Comput. (SEC), Oct. 2016, pp. 168–178.

[70] K. Zhang, Y. Mao, S. Leng, A. Vinel, and Y. Zhang, ‘‘Delay con- strained offloading for mobile edge computing in cloud-enabled vehicular networks,’’ in Proc. 8th Int. Workshop Resilient Netw. Design Model- ing (RNDM), Sep. 2016, pp. 288–294.

[71] S. Nunna et al., ‘‘Enabling real-time context-aware collaboration through 5G and mobile edge computing,’’ in Proc. 12th Int. Conf. Inf. Technol.- New Generat. (ITNG), Apr. 2015, pp. 601–605.

[72] A. Papageorgiou, E. Poormohammady, and B. Cheng, ‘‘Edge-computing- aware deployment of stream processing tasks based on topology-external information: Model, algorithms, and a storm-based prototype,’’ in Proc. IEEE Int. Congr. Big Data (BigData Congress), Jan. 2016, pp. 259–266.

[73] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, ‘‘Cost effi- cient resource management in fog computing supported medical cyber- physical system,’’ IEEE Trans. Emerg. Topics Comput., vol. 5, no. 1, pp. 108–119, Jan./Mar. 2015.

[74] M. Barcelo, A. Correa, J. Llorca, A. M. Tulino, J. L. Vicario, and A. Morell, ‘‘IoT-cloud service optimization in next generation smart envi- ronments,’’ IEEE J. Sel. Areas Commun., vol. 34, no. 12, pp. 4077–4090, Dec. 2016.

[75] K. Zhang et al., ‘‘Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks,’’ IEEE Access, vol. 4, pp. 5896–5907, 2016.

[76] Y. Mao, J. Zhang, and K. B. Letaief, ‘‘Dynamic computation offloading for mobile-edge computing with energy harvesting devices,’’ IEEE J. Sel. Areas Commun., vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[77] S. Sardellitti, G. Scutari, and S. Barbarossa, ‘‘Joint optimization of radio and computational resources for multicell mobile-edge comput- ing,’’ IEEE Trans. Signal Inf. Process. Netw., vol. 1, no. 2, pp. 89–103, Jun. 2015.

[78] J. Plachy, Z. Becvar, and E. C. Strinati, ‘‘Cross-layer approach enabling communication of high number of devices in 5G mobile networks,’’ in Proc. IEEE 11th Int. Conf. Wireless Mobile Comput., Netw. Com- mun. (WiMob), Oct. 2015, pp. 809–816.

[79] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, ‘‘Green cloud com- puting: Balancing energy in processing, storage, and transport,’’ Proc. IEEE, vol. 99, no. 1, pp. 149–167, Jan. 2011.

[80] D. N. Serpanos, L. Georgiadis, and T. Bouloutas, ‘‘MMPacking: A load and storage balancing algorithm for distributed multimedia servers,’’ in Proc. IEEE Int. Conf. Comput. Design, VLSI Comput. Process. (ICCD), Jun. 1996, pp. 170–174.

[81] A. Singh, M. Korupolu, and D. Mohapatra, ‘‘Server-storage virtualiza- tion: Integration and load balancing in data centers,’’ in Proc. ACM/IEEE Conf. Supercomput., Nov. 2008, p. 53.

[82] D. Ford et al., ‘‘Availability in globally distributed storage systems,’’ in Proc. Osdi, vol. 10. 2010, pp. 1–7.

[83] F. Chang et al., ‘‘Bigtable: A distributed storage system for structured data,’’ ACM Trans. Comput. Syst., vol. 26, no. 2, 2008, Art. no. 4.

[84] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, ‘‘A survey on network codes for distributed storage,’’ Proc. IEEE, vol. 99, no. 3, pp. 476–489, Mar. 2011.

[85] E. S. Andreas et al., ‘‘Proactive replication for data durability,’’ in Proc. 5th Int. Workshop Peer-Peer Syst. (IPTPS), 2006, pp. 1–6.

[86] A. Van Kempen, E. Le Merrer, and N. Le Scouarnec, ‘‘Method of data replication in a distributed data storage system and corresponding device,’’ U.S. Patent 8 812 801 B2, Aug. 19, 2014.

[87] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, ‘‘Pricing policy and computational resource provisioning for delay-aware mobile edge com- puting,’’ in Proc. IEEE/CIC Int. Conf. Commun. China (ICCC), Jul. 2016, pp. 1–6.

[88] A. Kiani and N. Ansari. (Dec. 2016). ‘‘Towards hierarchical mobile edge computing: An auction-based profit maximization approach.’’ [Online]. Available: https://arxiv.org/abs/1612.00122

[89] Y. Zhang, C. Lee, D. Niyato, and P. Wang, ''Auction approaches for resource allocation in wireless systems: A survey,'' IEEE Commun. Surveys Tuts., vol. 15, no. 3, pp. 1020–1041, 3rd Quart., 2013.

[90] D. An, Q. Yang, W. Yu, X. Yang, X. Fu, and W. Zhao, ''SODA: Strategy- proof online double auction scheme for multimicrogrids bidding,'' IEEE Trans. Syst., Man, Cybern., Syst., to be published.

[91] N. Kamiyama, Y. Nakano, K. Shiomoto, G. Hasegawa, M. Murata, and H. Miyahara, ''Priority control based on website categories in edge computing,'' in Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS), Apr. 2016, pp. 776–781.

[92] C. You, K. Huang, H. Chae, and B.-H. Kim, ''Energy-efficient resource allocation for mobile-edge computation offloading,'' IEEE Trans. Wireless Commun., vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[93] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos, ''Challenges and opportunities in edge computing,'' in Proc. IEEE Int. Conf. Smart Cloud (SmartCloud), Nov. 2016, pp. 20–26.

[94] L. Zhang et al., ''Named data networking (NDN) project,'' Xerox Palo Alto Res. Center-PARC, Palo Alto, CA, USA, Tech. Rep. NDN-0001, 2010.

[95] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani, ''MobilityFirst: A robust and trustworthy mobility-centric architecture for the future Internet,'' ACM SIGMOBILE Mobile Comput. Commun. Rev., vol. 16, no. 3, pp. 2–13, 2012.

[96] W. Yu, H. Zhang, Y. Wu, D. Griffith, and N. Golmie, ''A framework to enable multiple coexisting Internet of Things applications,'' in Proc. Int. Conf. Comput., Netw. Commun., Mar. 2018.

[97] X. Chen, L. Jiao, W. Li, and X. Fu, ''Efficient multi-user computation offloading for mobile-edge cloud computing,'' IEEE/ACM Trans. Netw., vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[98] A.-L. Jin, W. Song, P. Wang, D. Niyato, and P. Ju, ''Auction mechanisms toward efficient resource sharing for cloudlets in mobile cloud comput- ing,'' IEEE Trans. Serv. Comput., vol. 9, no. 6, pp. 895–909, Nov. 2016.

[99] W. Shi, L. Zhang, C. Wu, Z. Li, and F. Lau, ''An online auction framework for dynamic resource provisioning in cloud computing,'' ACM SIGMET- RICS Perform. Eval. Rev., vol. 42, no. 1, pp. 71–83, 2014.

[100] B. Ramachandran, S. K. Srivastava, C. S. Edrington, and D. A. Cartes, ''An intelligent auction scheme for smart grid market using a hybrid immune algorithm,'' IEEE Trans. Ind. Electron., vol. 58, no. 10, pp. 4603–4612, Oct. 2011.

[101] H. S. V. S. K. Nunna and D. Srinivasan, ''Multiagent-based transactive energy framework for distribution systems with smart microgrids,'' IEEE Trans. Ind. Informat., vol. 13, no. 5, pp. 2241–2250, Oct. 2017.

[102] Q. Yang, D. An, W. Yu, X. Yang, and X. Fu, ''On stochastic optimal bid- ding strategy for microgrids,'' in Proc. IEEE 34th Int. Perform. Comput. Commun. Conf. (IPCCC), Dec. 2015, pp. 1–8.

[103] P. Di Lorenzo, S. Barbarossa, and S. Sardellitti. (Jul. 2013). ''Joint optimization of radio resources and code partitioning in mobile edge computing.'' *Online+. Available: https://arxiv.org/abs/1307.3835

[104] Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren, ''Heavy hit- ter estimation over set-valued data with local differential privacy,'' in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, 2016, pp. 192–203. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978409

[105] X. Yang, X. Ren, J. Lin, and W. Yu, ''On binary decomposition based privacy-preserving aggregation schemes in real-time monitoring systems,'' IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 10, pp. 2967–2983, Oct. 2016.

[106] J. Lin, W. Yu, N. Zhang, X. Yang, and L. Ge, ''On data integrity attacks against route guidance in transportation-based cyber-physical systems,'' in Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC), Jan. 2017, pp. 313–318.

[107] Q. Yang, D. An, R. Min, W. Yu, X. Yang, and W. Zhao, ''On optimal PMU placement-based defense against data integrity attacks in smart grid,'' IEEE Trans. Inf. Forensics Security, vol. 12, no. 7, pp. 1735–1750, Jul. 2017.

[108] Q. Yang, J. Yang, W. Yu, D. An, N. Zhang, and W. Zhao, ''On false data-injection attacks against power system state estimation: Modeling and countermeasures,'' IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 3, pp. 717–729, Mar. 2014.

[109] W. Yu, D. Griffith, L. Ge, S. Bhattarai, and N. Golmie, ''An integrated detection system against false data injection attacks in the smart grid,'' Secur. Commun. Netw., vol. 8, no. 2, pp. 91–109, 2015. [Online]. Avail- able: http://dx.doi.org/10.1002/sec.957

[110] S. Bhattarai, S. Wei, S. Rook, W. Yu, R. F. Erbacher, and H. Cam, ''On simulation studies of jamming threats against LTE networks,'' in Proc. Int. Conf. Comput., Netw. Commun. (ICNC), Feb. 2015, pp. 99–103.

[111] S. Bhattarai, S. Rook, L. Ge, S. Wei, W. Yu, and X. Fu, ''On simulation studies of cyber attacks against LTE networks,'' in Proc. 23rd Int. Conf. Comput. Commun. Netw. (ICCCN), Aug. 2014, pp. 1–8.

[112] S. Yi, Z. Qin, and Q. Li, ''Security and privacy issues of fog computing: A survey,'' in Proc.
Int. Conf. Wireless Algorithms, Syst., Appl., 2015,pp. 685–695.

[113] Y. Huang, X. Yang, S. Yang, W. Yu, and X. Fu, ''A cross-layer approach handling link asymmetry for wireless mesh access networks,''
IEEE Trans. Veh. Technol., vol. 60, no. 3, pp. 1045–1058, Mar. 2011.

[114] W. Yu, H. Xu, A. Hematian, D. Griffith, and
N. Golmie, ''Towards energy efficiency in ultra dense networks,'' in Proc. IEEE 35th Int. Perform.
Comput. Commun. Conf. (IPCCC), Dec. 2016, pp. 1–8.

[115] C.-F. Lai, Y.-C. Chang, H.-C. Chao, M. S.
Hossain, and A. Ghoneim, ''A buffer-aware QoS streaming approach for SDN-enabled 5G vehicular networks,'' IEEE Commun. Mag., vol. 55, no. 8, pp. 68–73, Aug. 2017.

[116] A. Mosenia and N. K. Jha, ''A comprehensive study of security of Internet-of-
Things,'' IEEE Trans. Emerg. Topics Comput., vol. 5, no. 4, pp. 586–602, Oct./Dec. 2017.

[117] M. Van Dijk, C. Gentry, S. Halevi, and V.
Vaikuntanathan, ''Fully homo- morphic encryption over the integers,'' in Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn., 2010, pp. 24–43.

[118] D. Li, Q. Yang, W. Yu, D. An, X. Yang, and W. Zhao, ''A strategy-proof privacy-preserving double auction mechanism for electrical vehicles demand response in microgrids,'' in Proc. IEEE Int. Perform. Comput. Commun. Conf. (IPCCC), Dec. 2017.

[119] B. Pavithra and C. S. Anita, ''Privacy-preserving public auditing for data storage security in cloud computing,'' Adv. Natural Appl. Sci., vol. 10, no. 14, pp. 118–122, 2016.

[120] F. Chen, T. Xiang, X. Fu, and W. Yu, ''User differentiated verifiable file search on the cloud,''
IEEE Trans. Serv. Comput., to be published.

[121] C. Anglano, R. Gaeta, and M. Grangetto, ''Securing coding-based cloud storage against pollution attacks,'' IEEE Trans. Parallel Distrib.
Syst., vol. 28, no. 5, pp. 1457–1469, May 2017.

[122] R. Gennaro, C. Gentry, and B. Parno, ''Non-interactive verifiable com- puting:
Outsourcing computation to untrusted workers,'' in Advances in Cryptology—CRYPTO (Lecture Notes in Computer Science), vol. 6223. Berlin, Germany: Springer, 2010, pp. 465–482.

[123] B. Parno, J. Howell, C. Gentry, and M.
Raykova, ''Pinocchio: Nearly prac- tical verifiable computation,'' in Proc. IEEE Symp. Secur. Privacy (SP), May 2013, pp. 238–252.

[124] I. Hafeez, A. Y. Ding, L. Suomalainen, and
S. Tarkoma, ''Demo abstract: Securebox—A platform to safeguard network edge,'' in Proc. IEEE/ACM Symp. Edge Comput. (SEC), Oct. 2016, pp. 117–118.