Journal of AI-Powered Medical Innovations
ISSN: 3078-1930 DOI: 10.60087
2024, Vol. 2, No. 1, pp. 64–92
Home page https://japmi.org/
DOI: https://doi.org/10.60087/vol2iisue1.p006

AI-Driven Data Processing and Decision Optimization in IoT through Edge Computing and Cloud Architecture

Shiji Zhou[1]*, Jun Sun[1.2], Kangming Xu[2] Gaike Wang[3]

1. Computer Science, University of Southern California, CA, USA
1.2 Business Analytics and Project Management, University of Connecticut, CT, USA
2. Computer Science and Engineering, Santa Clara University, CA, USA
3. Computer Engineering, New York University, NY, USA

Abstract.

This study discussion point of this paper is to make an in-depth analysis of the development impact of the Internet of Things combined with edge computing and artificial intelligence. In the analysis process, the importance and criticality of data processing and decision making of edge computing as well as the challenges faced should be elaborated respectively. With the rapid popularization and development of Internet of Things devices, edge computing has brought more innovative solutions for different application scenarios such as intelligent furniture industrialization, automatic driving and intelligent transportation by reducing the delay of processing data and improving the characteristics of security data, film and television. Resource and energy efficiency have certain limitations, so it is necessary to combine artificial intelligence to enhance edge computing devices, hardware accelerators, and its utility and federated learning technologies, which can effectively improve the performance and scalability of edge computing and promote the development of more self-service network systems for smart devices. The core of this study is how to promote the Internet through AI-driven edge computing to further develop and provide insights for research priorities and suggest related future research directions.

Keywords: Edge Computing；Artificial Intelligence；Internet of Things；Data Privacy；Energy Efficiency

* **Corresponding author:** Shiji Zhou **(E-mail:** rexcarry036@gmail.com**)**

Received: 01-11-2024; **Accepted:** 01-12-2024; **Published:** 09-12-2024

1. Introduction

With the rapid adoption of Internet of Things (IoT) devices and the continuous advancement of artificial intelligence (AI) technology, Edge computing is becoming a trend that cannot be ignored[1]. The combination of Edge computing and AI not only changes the way data is processed, but also provides unprecedented opportunities for many application scenarios. The core content of this paper is to discuss the integration of Edge computing and artificial intelligence background as the theme of the current application challenges, as well as some practical cases and future directions as the outline of the paper[2]. Edge computing as a distribution is one of the paradigms of cloud computing, which refers to the transfer of centralized data and the boundary points of data generators in the process of data processing and data storage[3]. These disadvantages, including some Internet of things devices, smart furniture devices, industrial sensors, smart phones, etc. In the traditional Edge computing model combined with cloud computing, all data will first be transmitted to the cloud for processing, so such a traditional processing method will not only lead to data delay, but also bring certain privacy and security risks[4][5]. That is, Edge computing can significantly reduce data latency and save broadband, thereby enhancing data privacy and secure handling.

With the more and more extensive and application of artificial intelligence technology, data processing and data decision-making have also become one of the deployments of artificial intelligence algorithms, which includes the independent data analysis and decision-making of Edge devices, thereby reducing the dependence on Yulin, therefore, this combination also makes the data processing of many traditional applications that are difficult to achieve relatively simple. In daily life, the biggest example is the real-time perception and analysis of automatic driving[6][7]. Through the artificial intelligence model on the vehicle equipment, driving decisions can be made through the data of camera radar and lidar light sensor through Edge computing technology, which can improve safety and improve the efficiency of data analysis[8][9]. Secondly, many smart furniture, including sound cameras, home robots, etc., are provided with more personalized services and security through the combination of artificial intelligence technology and Edge computing. The main advantage of Edge computing on these devices is that it can reduce the dependence on cloud processing while processing language recognition and image analysis locally, improve user data privacy and process data more quickly; In addition, the combination of Edge computing and artificial intelligence in the industrial iot environment can play a great role in the production line, including monitoring and predicting equipment failures, primary prevention of new maintenance, and so on. From these everyday applications, we can see that the mutual assistance of Edge computing and artificial intelligence not only improves industrial productivity but also reduces maintenance costs. At the same time, rapid data processing can quickly identify risks and make relevant decisions. Although the current Edge computing, artificial intelligence shows a wide range of applications, in our view are comparative advantages, but there are still many challenges in the actual deployment.

At present, there are four major challenges: First, limited computing resources, and the computing power of Edge computing devices is generally lower than that of cloud servers. Therefore, in combination with the complexity and scale of artificial intelligence models, Edge computing has a certain phenomenon of hardware accelerator weakening[10]. Therefore, in the process of improvement, artificial intelligence algorithms can be implemented on Edge devices. Larger and more efficient models must be developed to achieve more efficient data processing. The second is the energy efficiency of Edge computing. Most Edge computing devices will rely on battery power, so these devices will generate a certain amount of energy in the process of being highly optimized by artificial intelligence algorithms, which involves hardware design, algorithms, optimization, and energy efficiency management[11]. The third challenge is the security and privacy issues of Edge computing technology. Due to the wide application range of Edge devices themselves, they are exposed to many open environments in the same city, so they are easy to become the target of some security risks. It can be seen that ensuring the security of Edge artificial systems and preventing data leakage is one of the biggest research issues of Edge computing[12]. The last challenge is model update and maintenance. Due to the relatively wide distribution of Edge devices, effective device maintenance is a relatively big challenge. Therefore, distributed mechanism, federated learning and other technologies need to be combined in daily maintenance. Figure 1 below is an architecture diagram of the current challenges of Edge computing devices.

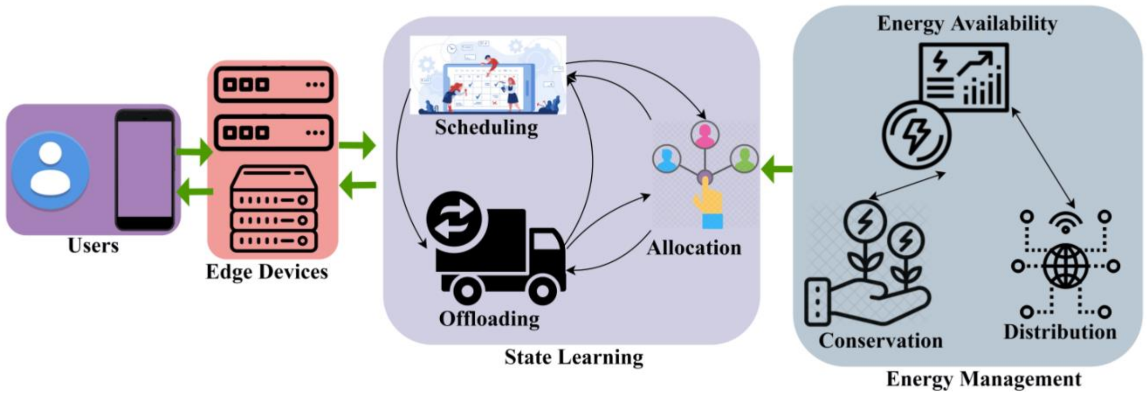


Figure 1. Architecture diagram of the challenges of Edge computing devices.

However, from the current development direction of the combination of artificial intelligence and Edge computing, the trend can be seen that Edge computing has the following important developments, among which the first is the development of hardware accelerators for Edge devices, combined with the hardware of artificial intelligence technology, to accelerate the emergence of some technologies that enhance the computing power of Edge devices, such as TPU and NPU, and so on. These technologies can not only reduce energy consumption, but also make AI more efficient and widespread. Secondly, the future Edge computing system will have a certain adaptive learning ability, that is, it can be adjusted and optimized according to the surrounding environment and the performance and state of the device data[13][14]. The third is the combination of federated learning applications. Federated learning generally allows multiple Edge devices to train the model together. In this case, while protecting privacy, the model can be trained on a larger scale through distributed computing resources. The last advantage is the development of AIoT, the deep combination of artificial intelligence and the Internet of Things, will promote the development of the next generation of new intelligent devices, the formation of a certain highly interconnected and autonomous intelligent network system, and promote the accelerated development of intelligence. The specific implementation architecture is shown in Figure 2 below.

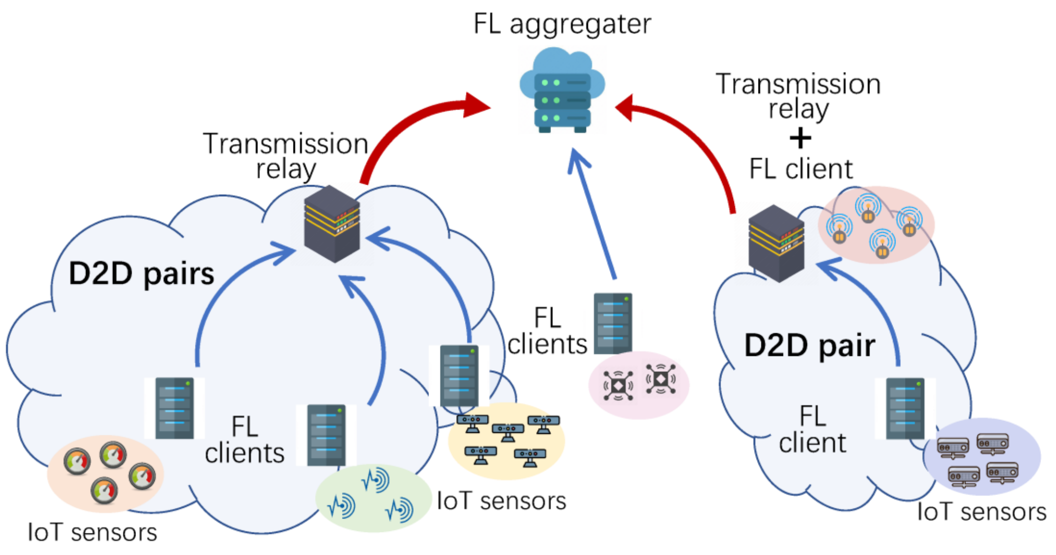


Figure 2. Trends in the combined development direction of artificial intelligence and Edge computing

It can be seen from the above that the combination of Edge computing and artificial intelligence is changing the traditional application of data processing and decision-making. Through the deployment of artificial intelligence algorithms on Edge devices for data processing, three aspects of low latency, high performance and high privacy protection can be greatly achieved[15]. However, there are still great challenges in the issue of computing resources energy efficiency efficiency and security of Edge computing. The combination of Edge computing devices and artificial intelligence can achieve a wider range of applications in more fields and promote the further development of more social intelligence in the future. This combination will not only improve the efficiency of devices and reduce latency, but also greatly improve the data privacy and security issues that people are most concerned about, as the application of a wide range of fields, including autonomous driving smart furniture and industrial automation, optimized AI models and strong security measures, the need will become increasingly important[16][17][18]. Therefore, this paper analyzes the future prospects by exploring in depth the specific impact of AI-driven Edge computing on the development of the Internet of Things, as well as solving existing applications and challenges.

2. Related Work

2.1. Edge computing on the Internet of Things

With the continuous development of the global Internet of Things, speed, countless devices and sensors will be interwoven, so intelligent decision-making and intelligent optimization management will become one of the core concerns of the Internet of Things. As the number of iot devices continues to grow, the demand for data processing and transmission will continue to increase, such accounting not only brings new challenges, but also gives rise to the rise of Edge computing, Edge computing plays a crucial role in the iot architecture, he calculates and optimizes the relevant response time through the execution of the data of the iot device itself[19][20]. In this way, the complex burden of the server is reduced, and the data people in the task data storage and service area are brought to the Edge while ensuring the timeliness and security of data processing[21]. Therefore, it can also be understood as Edge computing, which is to reduce latency and save broadband to provide faster and more efficient data processing and solutions. In the Internet of Things, Edge computing can solve the problem that central servers can't handle the massive amount of data in a timely manner. For example, self-driving cars need to process data generated by onboard sensors in real time, and delays or interruptions can be disastrous[22]. Here, Edge computing ensures that data is processed quickly at the point of generation to drive real-time decisions.

The implementation of Edge computing in iot systems can bring the following significant advantages: 1. Low latency - By processing data at the Edge nodes, response times are greatly reduced, which is critical for applications that require real-time feedback. 2. Bandwidth optimization - Pre-processing data at the Edge, sending only the necessary information to the cloud or central server, can significantly reduce data traffic. 3. Fault tolerance for interruptions - Processing data locally guarantees a degree of autonomy for the system, even when cloud services are not available. 4. Privacy and security - Edge computing can realize localized data processing, reduce sensitive data transmission, and thus improve data security.

2.2. IoT Edge Platform: Specific requirements for iot Edge computing

In general, operators want to roll out Edge computing sites in their networks using COTS (plain off-the-shelf) servers that don't have application-specific hardware. While today, onpremises Edge deployments are often very customized and deployed and integrated for a specific customer, in the future, these deployments will also have to be more standardized[23][24].

Edge providers must manage some iot specific requirements. For example, some iot applications that require heavy AI or data analytics at the Edge may have specific hardware requirements, such as servers with GPU capacity[25]. An Edge gateway that supports iot devices must also support connectivity for multiple device communications such as ZigBee and Bluetooth, as well as connectivity via cellular and Wi-Fi technologies.

In addition, operators need to consider the type of iot platform they are using and how it supports and integrates their Edge infrastructure. Many operators with iot businesses rely on third-party vendor platforms such as PTC ThingWorx and Software AG's Cumulocity to manage their iot networks[26]. As the demand for Edge applications continues to grow, platform vendors are increasingly investing in introducing Edge-specific functionality to their platforms[27].

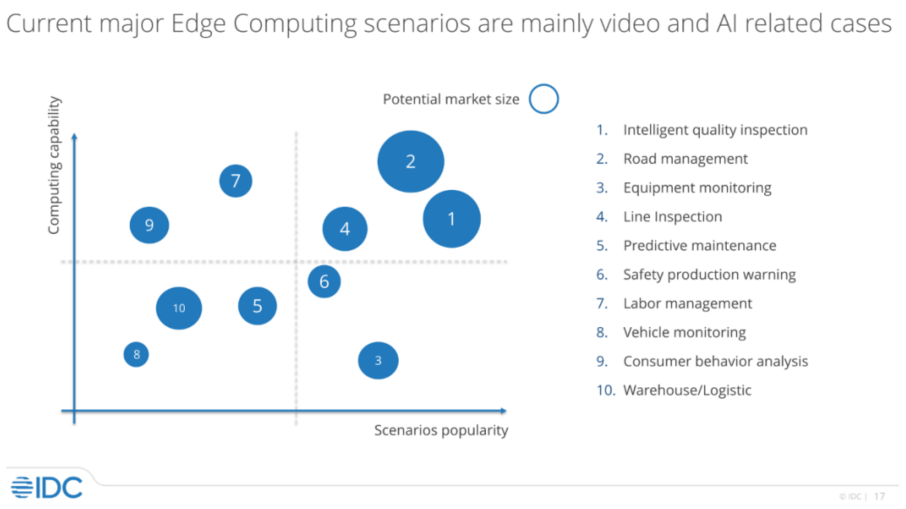
Many IoT use cases have already been commercialized and offer benefits to various industries today. Some of these may be enhanced by Edge computing[28]. This includes use cases that require low latency, high bandwidth, or local data storage. An example of this is condition-based monitoring. Today, remote assets are monitored, but this is often difficult and time consuming because they can span vast areas and failures can be catastrophic (e.g., oil pipeline explosions, water pipe leaks). Iot sensors can be used to monitor the condition of an asset (e.g. temperature, pressure, vibration, etc.) These sensors can alert the owner to potential problems and ensure that the asset is repaired before it fails.

Today, this happens through a connection to a centralized cloud. However, Edge computing can provide some real benefits. Due to the potential risk to critical systems and the safety of nearby people, an alarm needs to be triggered immediately if an abnormal reading is reported[29]. Edge computing means that data processing can take place closer to the detection location and does not need to depend on the quality of the network connection.

2.3. AI-Assisted Edge Computing – Tesla

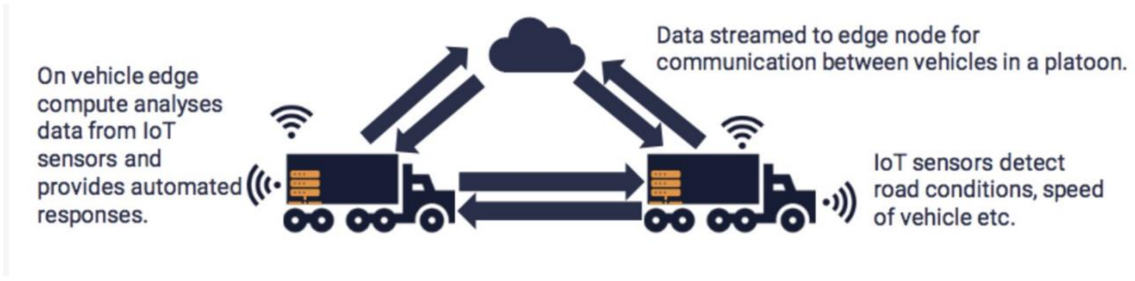
At Tesla's Shanghai Gigafactory, Edge computing is enabled by AI (artificial intelligence) to play a big role in the production process. In fact, the integration of the two is also the general trend[30][31]. Last year, International Data Corporation (IDC) released the "China Semi-annual Edge

Computing Server Market (the first half of 2021) Tracking Report", summarizing the top ten scenarios of Edge computing[32]. These include intelligent quality inspection, road management, equipment monitoring, line inspection, preventive maintenance, safety production early warning, workforce management, consumer behavior analysis, warehouse/logistics management, most of which are closely related to video and AI[33].



In addition to this, when the device is operating as expected, the raw monitoring data indicates normal behavior and is not useful. Edge computing means there is no need to unnecessarily send large amounts of data to the cloud. For condition-based monitoring, Edge computing is not required to implement use cases, but it can provide some substantial benefits. Stratus Technologies is an example of how this works in practice today[34][35][36].

Edge computing can unlock some new IoT use cases that do not exist on a commercial scale today. One example of this is automated platooning for truck convoys which is one of the first likely use cases for autonomous vehicles. Here a group of trucks travel close behind one another in a convoy, saving fuel costs and decreasing congestion. In the future, it may be possible to leverage Edge computing to remove the need for drivers in all trucks apart from the front one, further reducing costs for transport and logistics companies[37]. The low latency, high bandwidth requirements that come with autonomous control of vehicles mean that adoption of this at scale will likely only be possible with a combination of 5G and Edge computing.


On vehicle edge compute analyses data from IoT sensors and provides automated responses.
Data streamed to edge node for communication between vehicles in a platoon.
IoT sensors detect road conditions, speed of vehicle etc.

2.4. The improvement of Edge computing data processing and decision-making

Compared with traditional computing, edge computing technology is one of the newer paradigms at present, and in recent years, with the rise of the Internet of Things (IoT) and the expanding demand for real-time data processing and analysis combined with artificial intelligence, edge computing technology has gained unprecedented appeal. However, it is important to note that edge computing capabilities can be hosted at the edge of infrastructure (such as micromodule data centers or carrier cellular sites) or at the edge of devices (such as smartphones, drones, smart cameras or connected cars), running in applications or workflows as extensions to the centralized cloud[38]. These applications or workflows are centrally managed in the cloud and run at the edge.

This approach reduces latency and improves performance, which is critical for IoT-related industries in time-sensitive applications. There are several types of edge computing, including fog computing, mobile edge computing, and cloud computing. Fog computing is a type of edge computing that puts computing resources between edge devices and the cloud, bringing data processing closer to the source[39]. Edge computing can not only reduce bandwidth costs by processing data locally, but also improve agility through real-time, on-site decision making. With intermittent cloud connectivity, edge computing enables processes, including the Industrial Internet of Things (IoT), to run autonomously. Finally, combined with the edge computing involved in cloud computing, computing resources can be placed at the edge of the cloud, which can be implemented and processed closer to the end device and reduce latency[40].

Network Topology refers to the physical layout of various devices that are interconnected using transport media. A specific physical (i.e., real) or logical (i.e., virtual) arrangement of network members. If two networks have the same connection structure, they have the same network topology, although their internal physical connections and distance between nodes may be different. Compared with traditional computing, edge computing also has many advantages, such as the security and management of distributed computing resources.In the processing work of Edge computing, organizations should carefully consider their needs, and distinguish and optimize data, so as to make certain adjustments to the importance of Edge computing in future computing[41]. The key to Edge computing is to be able to get closer to the data source more quickly and to process and analyze the data quickly. Edge computing is the ability to place resources at the Edge of the network to reduce transmission distances, thereby reducing latency and speeding up data processing events to calculate the content of the data more sensitively.

Secondly, reliability, Edge computing can reduce the use of broadband by processing data locally. For example, in some remote or rural areas, broadband itself is limited and environmentally effective, so it takes a lot of time for delayed data to be transmitted from one point to another through the network, which will also bring a certain decline in performance and user satisfaction. Therefore, cloud computing can reduce this delay in a variety of ways and improve user satisfaction on network facilities for a certain low latency, efficient processing. The last point is Edge computing, which can also improve network security by reducing the amount of data sent by the network. That is to say, when Edge computing deals with some local sensitive data, it will reduce certain risk of leakage and security threats, so that placing resources closer to the data source can release and innovate more security possibilities while improving user experience.

2.5. Edge computing architecture model for edge devices

Edge computing architecture refers to the structural computation of frameworks and components at the edge of the network. That said, the core of the edge computing architecture is to allocate edge computing resources closer to the data source or processing power, enabling more efficient edge computing data processing and decision making. The components of the architecture typically include edge devices, edge cloud servers, and edge devices, which are small and low-power devices. The core components have these distributions. First, the Kubernetes control node adopts the original data model of the cloud part and keeps the original control and data flow unchanged, that is, the node run by KubeEdge appears as an ordinary node on Kubernetes. Kubernetes can manage the nodes that KubeEdge runs on just like normal nodes.

The reason why KubeEdge can run on edge nodes with limited resources and uncontrollable network quality, On the basis of Kubernetes control nodes, KubeEdge realizes the sinking of

Kubernetes cloud computing orchestration containerization applications through CloudCore in the cloud part and EdgeCore in the edge part.

CloudCore in the cloud part is responsible for monitoring the instructions and events of the Kubernetes control node to the EdgeCore in the edge part and submitting the status information and event information reported by the EdgeCore in the edge part to the Kubernetes control node. EdgeCore at the edge is responsible for receiving commands and event information from CloudCore at the cloud, executing related commands and maintaining edge loads, and reporting status information and event information at the edge to CloudCore at the cloud.

In addition, EdgeCore is tailored and customized on the basis of Kubelet components, which cuts the rich functions that Kubelet cannot use on the edge, and adds offline computing functions on the basis of Kubelet in view of the status quo of limited edge resources and poor network quality. EdgeCore is well adapted to the marginal environment. Generally speaking, the architecture of Edge computing and network is very complex and multifaceted. In the process of data processing and architecture processing, it is necessary to carefully plan and manage complex data of multiple parties to ensure effective and reliable data processing. At the same time, the Edge platform is also increased, and other components are constantly innovating and developing. These innovations not only help drive data processing, but also allow for faster progress in reanalysis and real-time data decision-making.

2.6. Edge computing equipment

Edge computing devices are a general class of small, low-consumption devices used to collect, process, and analyze data at the network edge. These edge devices are often placed very close to data sources in the ocean and can be deployed in different environments, including remote engineering, automated vehicles, industrial intelligent control, and more. Many edge computing devices are used in our daily life, such as smart sensors, this small edge device can collect certain data from the environment. Such as the temperature and humidity of the environment and air quality, so as to apply intelligent data in environmental detection and building design. The edge server and edge gateway both belong to the intermediary role combined with the cloud server. They can obtain certain effective data from various applications, such as intelligent transportation of industrial automation, intelligent cameras, Internet camera lights and other different environmental changes, and deploy these data to the relevant application content delivery edge devices, deploy edge computing, and deploy edge computing.

The architectural aspect plays an important role and also provides facts. As the use of edge computing continues to grow, our development and innovation of edge computing devices and related components will need to continue to improve and expand. As the use of Edge computing continues to grow, expect continued innovation in the development of Edge devices and other components to help drive new advances in data processing and real-time decision making.

3. Proposed Streaming ETL Framework

In order to solve the challenge of real-time data processing in current iot applications, we propose a comprehensive streaming ETL framework. The design takes a partially decentralized approach to communication, using a ZMQ event bus similar to an MQTT broker, to facilitate communication between streaming ETL services and iot devices. Thanks to a partially decentralized architecture, modular systems can be easily scaled. The ZMQ Event bus is a messaging mechanism based on the ZeroMQ (ZMQ) library for efficient asynchronous communication. It allows messages to be sent and received between different applications or services, and supports a variety of communication modes, such as publish-subscribe, request-response, etc. This bus structure can enhance the scalability and flexibility of the system, and is suitable for scenarios that require real-time data transmission, especially in iot applications. With ZMQ, communication between services can be faster and more reliable.

Containerization technology plays a key role in providing a consistent environment for applications and their dependencies, ensuring the modularity and portability of the framework. This feature enables the framework to be smoothly integrated into the infrastructure while simplifying deployment and management. As a result, the proposed framework can be easily adapted to different use cases and requirements, demonstrating flexibility in the face of various iot application requirements. In addition, the combination of ZMQ and MQTT communication technologies enables efficient data transmission and processing even in large or restricted network environments.

Both communication technologies (ZMQ and MQTT) adopt a publish-subscribe architecture that further enhances the modularity of the system. In the publish-subscribe model, there is no direct interdependence between the sender (publisher) and the receiver (subscriber) of a message.

Publishers post messages to specific topics, and subscribers can choose to subscribe to topics that interest them.

This approach brings several benefits: first, system components are less coupled to each other, allowing them to be developed and deployed independently; Second, subscribers can be easily added or removed without affecting the operation of other components. Finally, this flexibility allows the system to scale rapidly according to demand, such as when adding new features or handling higher data volumes, without the need to redesign the entire system.

By using publish-subscribe architecture, ZMQ and MQTT are able to support dynamic message flow, ensuring efficient communication and data processing in different network environments to meet the needs of various iot applications. The architectural design of a bus and event platform is shown in Figure 5.

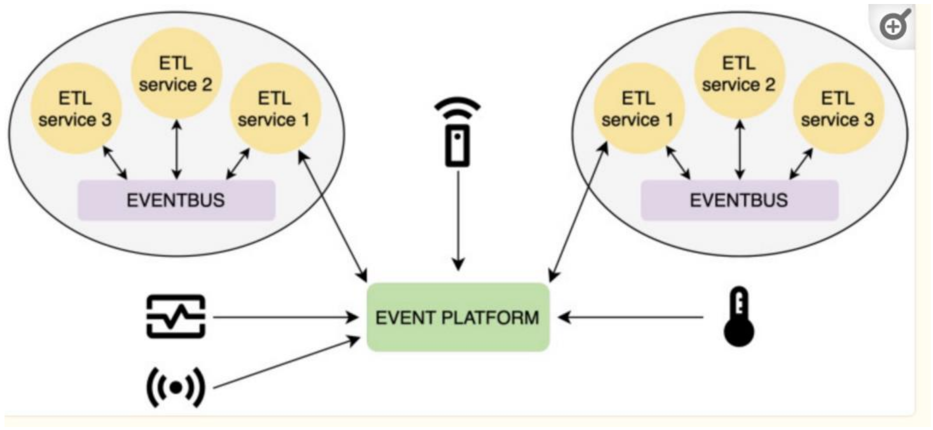


Figure 5. Architecture of the proposed framework.

By combining Kubernetes and Terraform, Preferred needs to build a robust and flexible infrastructure to adapt to the dynamic demands of iot applications. Terraform takes a declarative approach to seamless infrastructure management and versioning to accelerate solution development and deployment. Using Terraform in combination with Kubernetes ensures that infrastructure changes are applied consistently and reliably across different environments, simplifying the transition from development to production. Kubernetes is an open source container orchestration platform for automating the management, deployment, and scaling of containerized applications. It provides an efficient way to manage microservices architecture, ensuring the availability and scalability of applications.

Terraform is an infrastructure-as-a-Code (IaC) tool that allows users to define and manage cloud infrastructure through declarative profiles. It automates the creation, updating, and versioning of infrastructure resources to support multiple cloud platforms and services, simplifying the infrastructure management process.

Combining Kubernetes and Terraform allows for more efficient infrastructure management and container scheduling to meet the needs of complex applications.)

In this experiment, we chose Python as the development language for the streaming ETL service. However, using the ZMQ event bus in our framework adds an extra layer of modularity, making it possible to write components in C, C++, Java, JavaScript, Go, C#, and many other programming languages. Adopting a language-independent approach will allow future researchers and developers to take advantage of different programming languages to build independent components, thereby increasing the flexibility and adaptability of the overall system.

Therefore, the system can be extended by easily adding more services (either subscribers or publishers) on the ETL ZMQ event bus, as shown in Figure 6. In addition, the system can be customized to the unique needs of different iot applications and environments. By introducing such modularity and versatility, our framework becomes more robust and able to effectively address the complex and ever-changing challenges in iot data processing.

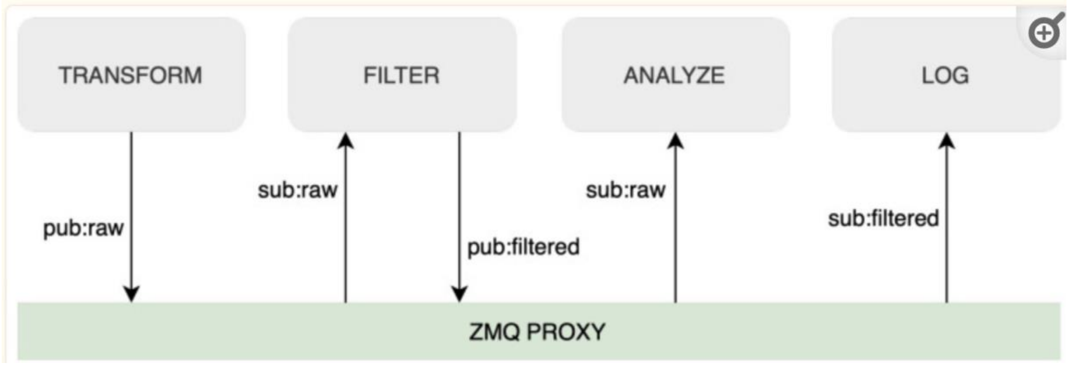


Figure 6. Streaming ETL ZMQ eventbus.

The next section covers the details of each service embodied as part of the streaming ETL framework. We will discuss the capabilities of each service, the technologies used, and how they can work together to provide effective real-time data processing tools for IoT applications. This project has developed a standardized framework for real-time data processing that can be adapted for a variety of applications. According to this framework, users can use containerization, edge computing, Kubernetes automation infrastructure and efficient communication protocols to generate data processing pipelines that meet specific requirements.

3.1. Input Data Transformation Service

Input data conversion service is one of the important components of edge framework in the process of data processing by edge devices. The most commonly used data formats in iot solutions are JSON, numbers or strings that change raw quality into binary data and analyze it. In this process, in order to ensure that all types of data can be used normally in the edge computing framework, the first step must be to convert them into a unified format, followed by the processing of transformers. Transformers collect raw data from various iot devices, and introduce services into the data, using conversion rules. All data will be converted into standardized JSON format, which is convenient for the subsequent data analysis, and the subsequent ideas can be easily used. Finally, the conversion process will be standardized into copper in the conversion data field, and the measured value will be 45 into the last two decimal places after the name is used. After the data conversion is completed, the data will be sent to the ZMQ event bus. Standardizing this data is the first step toward achieving autonomous AI algorithms that can select and manage applications and data processing tasks and facilitate more complex processing later on.

The service can receive data from MQTT devices through MQTT mediations or from REST API clients. For the REST API client, we implemented a modular client that gets data from the endpoint every n seconds. The n here is made up of the service's environment variable. Through data standardization, input data conversion services can effectively improve the overall efficiency and interoperability of the system, ensuring smooth integration with other back-end services.

3.2. Filtering Mechanism

In the process of database processing, it is necessary to filter and reduce the amount of data in the database. Thus, the converted data is predefined and filtered to the implementation process, ensuring that only relevant and necessary data information is forwarded to the edge device system to form subsequent components. By filtering data, optimizing equipment storage and processing related needs, you can reduce the consumption of a certain broadband network. This efficiency is very important in the Internet of Things, because devices generate a lot of data, not all of the data is relevant or useful, so filtering data is an essential part.

In addition, the service integrates modular filtering algorithms. The implemented filters include a value change filter that only sends data to the ZMQ event bus for further processing if the new value differs from the previous one. Another filter handles numerical values with a certain degree of precision. In this case, the data is passed to the ZMQ event bus for further processing only if the change in the new value exceeds the set precision.These filters can be easily extended to suit different data processing requirements and scenarios.

3.3. Edge device computing analysis services

Edge computing analysis service is an important part of Streaming ETL system, which is responsible for analyzing the transformed data to extract valuable insights. The service consists of three main components: Redis Logger, Redis Consumer, and Redis Streams. These components and their data flow relationships are shown in Figure 7.

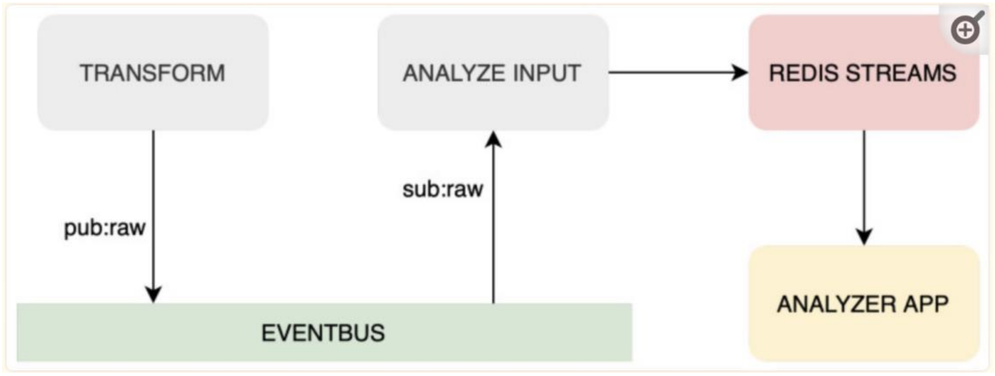


Figure 7. Analysis services for data flows in edge devices.

Redis Logger reads converted data from the ZMQ event bus and sends it to Redis Streams, a storage method for managing and processing data streams and in-memory structured data. In other words, Redis Streams can provide efficient processing of data and can efficiently distribute the data flow to the various ideal states of the Internet of Things. Redis Consumer reads the data from Redis Streams and analyzes the last n records.

The analysis process includes the description of changes in the temperature environment, such as humidity, and the analysis is carried out to support the so-called final decision. It should be noted that edge computing devices can combine any number and every specific data of the application in the analysis service to carry out real-time data analysis of the service and get more comprehensive and broader decisions based on Redis Streams analysis data.

3.4. MongoDB Time Series and Logging Service

MongoDB time series analysis and log input is one of the final components that can represent the data remaining in a 1TB system, and this process is an iot device, which has certain advantages for the storage of future data streams. Therefore, using MongoDB for time series database analysis is one of the solutions for specialized data storage and resolution as well as high-volume data and time streaming for high intake. By using the time series database edge device system, the data can be stored and indexed more efficiently, and a large number of time-stamped data systems can be queried, and the analysis of historical data and the trend of time passage can be performed more effectively.

In addition, MongoDB time series analysis can also provide flexibility for different data types for iot applications, such as sensitive reading of environmental and other sensor data. The log service can store the filtered data in the MongoDB time series database during the reading process of the ZML event bus. In addition to transforming and filtering the related data, the log service can also list the location device ID of the relevant edge device and the raw data in the execution process of the received data. This additional information can be understood as one of the valuable pieces of information provided by Internet data that can lead to more informed decisions and a better understanding of overall system performance, leading to more informed judgments.

3.5. Deploying Framework to Kubernetes Cluster

The deployment of melt frameworks on Proxmox edge servers via the Debian GNU/ Linux-based open source virtualization platform is closely related to edge computing. Edge computing emphasizes data processing near the source of data generation to reduce latency and bandwidth consumption. Using Proxmox's virtualization technology, we are able to efficiently manage virtual machines and containers on edge devices to quickly respond to the data requests and processing needs of iot devices. This deployment enhances the flexibility and scalability of the system to adapt to dynamic edge computing environments.

In this experiment, we designed and created two virtual edge devices to form a distributed system cluster management node, which is responsible for managing and coordinating cluster activities and making necessary adjustments. The worker node runs the containerized ETL service, the MongoDB time series database, and the MQTT agent. We chose to use CRI-O as the container runtime for Kubernetes instead of Docker shim to better meet Kubernetes' needs. The complete architectural framework is shown in Figure 8.

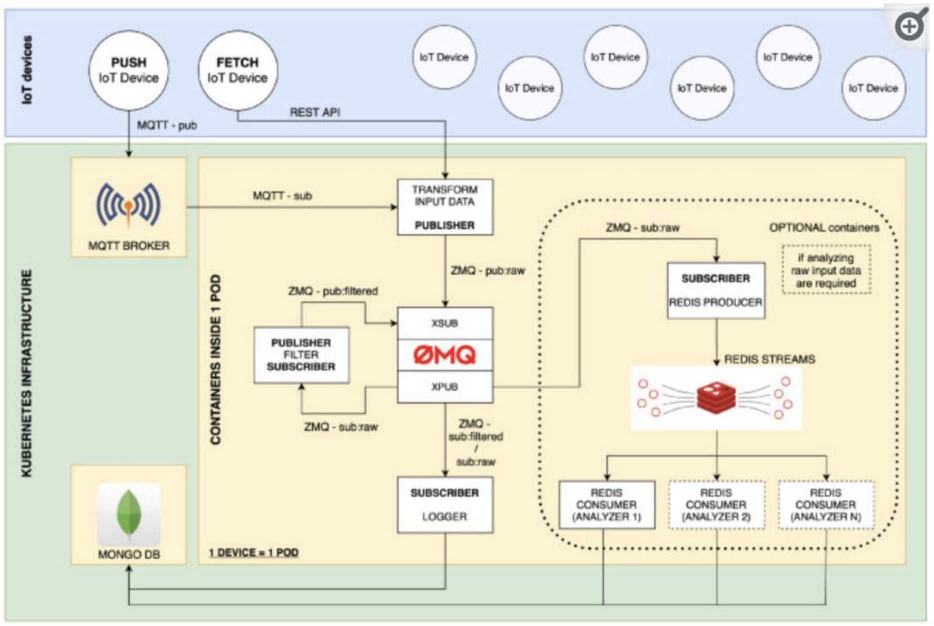


Figure 8. The complete framework of the frame of the edge device in the experiment

4. Testing of the Framework

In this experimental evaluation process, we conducted a comprehensive analysis of the edge device framework, focusing on the round-trip time (RTT) metric in ETL. We tested multiple data flows (such as 100 and 1000) and different latencies (1 ms, 10 ms and 100 ms) to simulate different scenarios. For example, we use a short burst of 10 values with a 1-millisecond delay and a long burst of 1000 values with a 1-millisecond delay, which can represent an industrial sensor that requires an immediate response. In contrast, longer latency is suitable for commercial iot devices such as temperature sensors.

Table 1. Testing Parameters for Edge Device Framework Evaluation

Test Scenario	Data Volume	Latency	Hardware Platform	CPU Cores	RAM

Short Burst	10	1 ms	64-bit ARM	8	8192 MB
Long Burst	1000	1 ms	64-bit ARM	8	8192 MB
Short Burst	10	1 ms	64-bit x86	8	8192 MB
Long Burst	1000	1 ms	64-bit x86	8	8192 MB
Long Delay	1000	10 ms	64-bit ARM	8	8192 MB
Long Delay	1000	100 ms	64-bit x86	8	8192 MB

To verify the efficiency and performance of the framework on different modern hardware platforms, we chose two test methods: first on a 64-bit ARM processor, and then on a 64-bit x86 processor. Eight CPU cores and 8192 MB of RAM were each allocated in the test environment. In this way, we can compare the performance of the framework under different platforms. Since most of the experiment was focused on testing ETL services within our proposed edge device framework, issues such as lack of computing resources and limited control over data transfer rates would arise, which would require ignoring actual iot devices from our test environment. A test container was designed to simulate iot devices, allowing easy adjustment of data transfer speed and capacity.

4.1. The Test Environment

To fully evaluate our framework, we took two approaches to compare the effectiveness and performance of the two hardware platforms, 64-bit ARM cpus and 64-bit x86 cpus. The reason for choosing these two platforms is that they represent the two most common CPU architectures in edge computing, ensuring that our framework has good compatibility. In addition, we wanted to test whether there were significant performance differences between frameworks running on different hardware platforms.

During testing, we designed several scenarios to see how these platforms performed with different workloads. This includes short burst data streams and long delay scenarios to simulate the variety of situations that may be encountered in real-world applications. In this way, we are able to gain insight into how the framework performs under different conditions.

Table 2. Hardware used in testing.

Device	CPU	CPU Frequency	Number of Cores	RAM	Used Cores	Used RAM

ARM64	Macbook Pro	Firestorm: 600-3228 MHz	4x Firestorm (performance)	16 GB	4	8 GB
		Icestorm: 600-2064 MHz	4x Icestorm (efficient)			
AMD64	HP Server	2400-3200 MHz	12	32 GB	12	8 GB

Table 2 shows the different platforms used in our tests. The Apple M1 CPU is based on the ARM architecture, using a combination of a big processor and LITTLE architecture, with a high-performance core (CPU-P) Firestorm and an energy-efficient core (CPU-E) Icestorm. This design allows tasks to switch seamlessly between different cores to optimize performance and energy consumption. In addition, we also considered other hardware platforms to ensure broad applicability of the test results. Through such testing, we hope to be able to determine how the framework performs under various workloads, which can provide a basis for future optimizations and improvements.

4.2. Tests on ARM64 with 100 ms Delay

During the first test, 10 RTT measurements were passed across ETL servers, where the delay between 10 servers was 100 ms and the center bit value was 5.392ms, with an average of 5.729ms distributed symmetrically. The average deviation on the deviation paper is 1.265ms at one point, and the specific results can be shown in Figure 8. It can be seen that in the testing process of the whole edge device, the core of the data does not increase more activities, but there is a large change in the core of performance at high frequency.

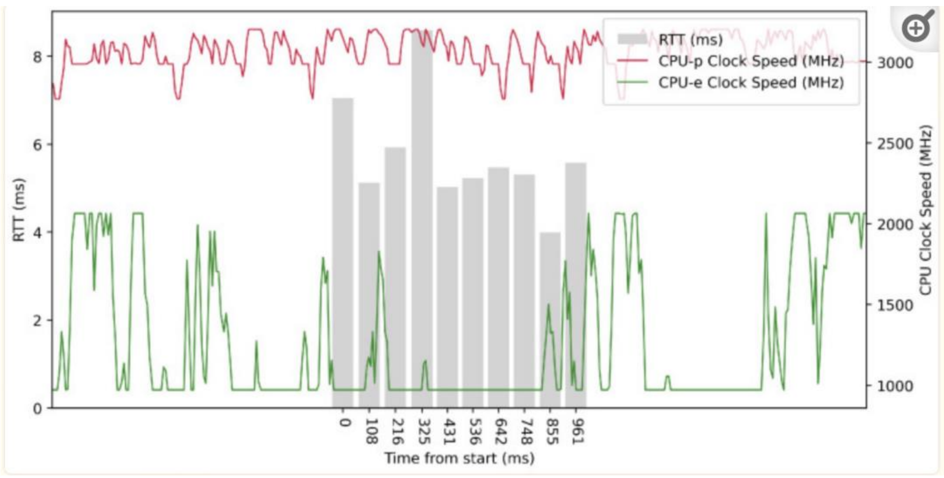
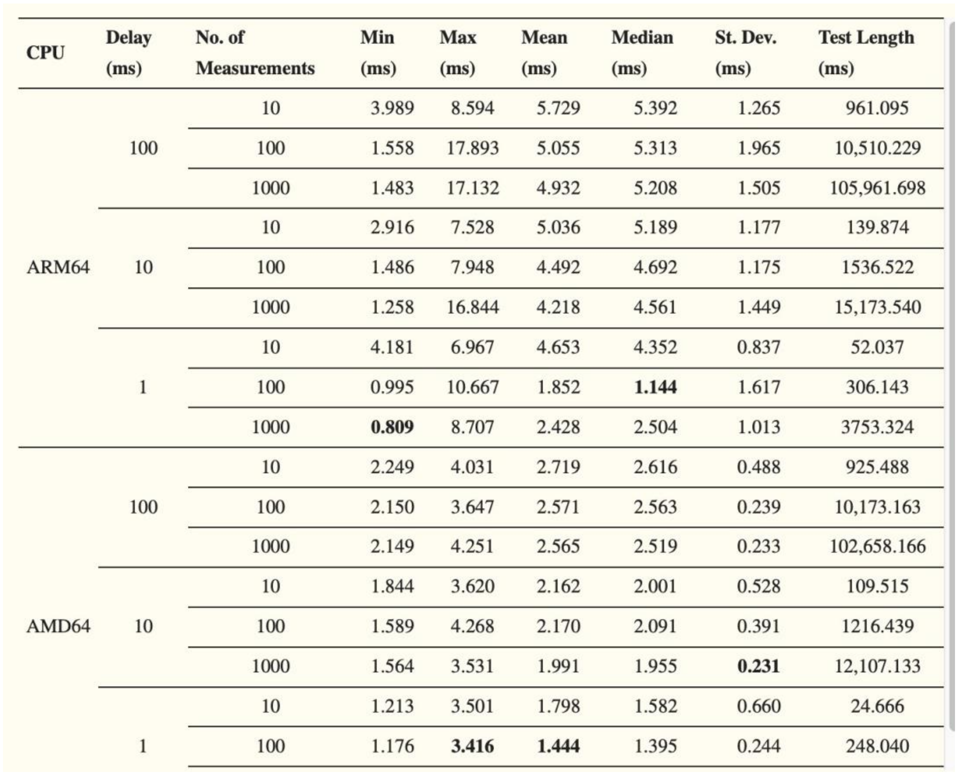


Figure 8. Efficiency results for specific data in test 1

Table 3. Data device edge computing measurement results.



CPU	Delay (ms)	No. of Measurements	Min (ms)	Max (ms)	Mean (ms)	Median (ms)	St. Dev. (ms)	Test Length (ms)
ARM64	100	10	3.989	8.594	5.729	5.392	1.265	961.095
		100	1.558	17.893	5.055	5.313	1.965	10,510.229
		1000	1.483	17.132	4.932	5.208	1.505	105,961.698
	10	10	2.916	7.528	5.036	5.189	1.177	139.874
		100	1.486	7.948	4.492	4.692	1.175	1536.522
		1000	1.258	16.844	4.218	4.561	1.449	15,173.540
	1	10	4.181	6.967	4.653	4.352	0.837	52.037
		100	0.995	10.667	1.852	**1.144**	1.617	306.143
		1000	**0.809**	8.707	2.428	2.504	1.013	3753.324
AMD64	100	10	2.249	4.031	2.719	2.616	0.488	925.488
		100	2.150	3.647	2.571	2.563	0.239	10,173.163
		1000	2.149	4.251	2.565	2.519	0.233	102,658.166
	10	10	1.844	3.620	2.162	2.001	0.528	109.515
		100	1.589	4.268	2.170	2.091	0.391	1216.439
		1000	1.564	3.531	1.991	1.955	**0.231**	12,107.133
	1	10	1.213	3.501	1.798	1.582	0.660	24.666
		100	1.176	**3.416**	**1.444**	1.395	0.244	248.040

In the second test, mainly by measuring the data of 100 RTT fingers of the leap-over ETL server, the latency is still 100 ms, the center bit is 5.313ms, and the average is only 5.055. The average index indicates that the average values of this measurement are mostly close to the norm. The standard deviation of the data is 1.965ms, which is also in line distribution, as can be seen in Table 3. From the results, it can be seen that the RTT decreases significantly when the data analysis volume with an efficient core reaches the maximum speed, but the performance and the head always maintain a high frequency.

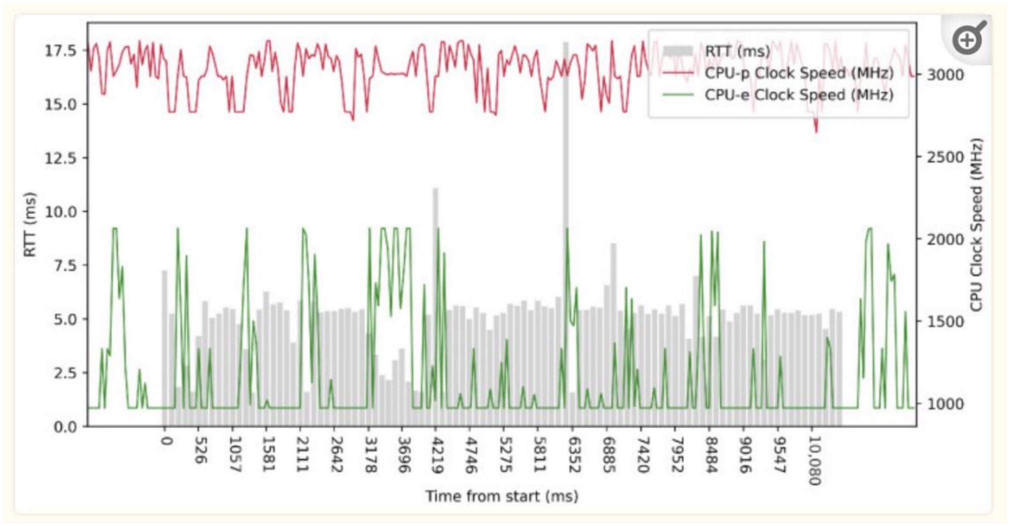


Figure 9. Efficiency results for specific data in test 2

4.3. Tests on ARM64 with 10 ms Delay

In the remaining measurement tests of this experiment, we performed multiple RTT (round trip time) measurements on the ARM64 platform with a 10ms delay. Compared to the first test, the median was reduced by 0.203 ms, indicating an improvement in the RTT measure. The results of the fifth and sixth tests showed a similar trend, with a median of 4.692 ms and 4.561 ms observed, respectively, indicating a further tendency for RTT to decrease as the number of measurements increased.

These test results have important implications for edge computing and iot applications. As data transfers between edge devices and the cloud become more efficient, low-latency RTT metrics can support real-time data processing to meet the needs of smart devices for rapid response. For example, in industrial iot scenarios, the ability to monitor and analyze data in real time can significantly improve operational efficiency and response speed. As a result, these tests provide data support for future optimization of the edge computing framework, ensuring that it can effectively handle the diverse needs of iot applications.

4.4. Tests on AMD64 with 10 ms Delay

In this experiment, we made several measurements of RTT (round trip time), focusing specifically on the performance at a 10-millisecond delay. In the fourth test, 10 measurements were taken, and the results showed a mean RTT of 2.162 ms, a median of 2.001 ms, and a standard deviation of 0.538 ms. This shows that the shorter delay significantly reduces the RTT value compared to the 100 ms delay of the first test.

These results have important implications for edge computing and iot applications. With reduced RTT, the system is able to process data from various iot devices more quickly, which is critical for application scenarios that require real-time reactions. For example, in intelligent manufacturing and automated monitoring, fast data transmission and processing can significantly improve the response speed and efficiency of the system. Therefore, this experiment provides empirical support for the optimization of the edge computing framework to ensure that it can maintain efficient and stable performance in the face of diverse iot applications.

4.5. Discussion

Our test results provide important insights into ETL service performance and efficiency across different hardware platforms and configurations. By using analog iot devices as test containers, we were able to focus on data processing speed and communication between ETL services, while controlling transfer speed and other test parameters. In addition, testing the framework on ARM64 and x86 64-bit processors allowed us to evaluate its performance and compatibility in modern hardware architectures.

The results show that the proposed framework performs well under different conditions (as can be seen in Figure 10), and the RTT index always remains within the acceptable range during the test. Tests also showed that the framework is capable of handling multiple data transfer speeds and measurements, demonstrating its scalability and adaptability in various iot deployment scenarios. This is especially important for edge computing, as devices in these environments often need to process large amounts of data in real time for fast response and decision making.

However, it is important to note that while our test environment is relatively comprehensive, it does not cover all possible scenarios. Further testing with real-world networking devices and different network conditions may reveal more challenges and potential directions for optimization. Comparing the RTT performance of ARM64 and AMD64 processors, we found that ARM64 experienced a greater decrease in median RTT as the number of measurements increased, while AMD64 maintained consistently low latency across all test scenarios, underscoring its suitability for real-time computing applications that require low latency. These results provide an important reference for selecting the hardware required for specific use cases, ensuring optimal performance in edge computing and iot.

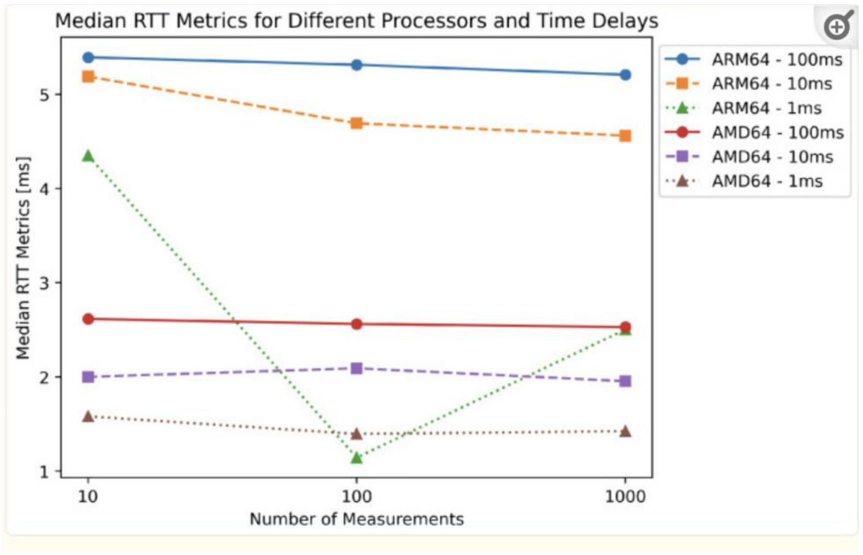


Figure 10. Edge computing high frequency test results

The final test results showed that even at the highest CPU frequency, the processing speed was not enough to process the existing data before the new data arrived, resulting in an increase in the mean and median, and an increase in the processing queue and processing time. While the proposed framework shows good efficiency and performance in data processing and communication between ETL services, further testing with real iot devices, networks, and analytics services is needed to better understand its potential in different iot scenarios. In addition, our tests on ARM64 and AMD64 cpus show that although single-board computers (SBC) do not use a uniform CPU architecture, it is critical to ensure that the framework works on both architectures. Future research is needed to explore whether the differences between architectures apply to smaller devices.

There are still some research challenges in edge computing. The use of multiple devices complicates task offloading and load balancing. For latency-sensitive tasks, the offload mechanism must consider the distance of available nodes, the current load, and potential performance. In addition, device performance differences make load balancing more difficult, and compatibility also needs to be considered, because not all devices can perform the required tasks, and the system must be able to adapt to the capabilities of other nodes. In mobile edge computing, the movement of devices or users needs to anticipate and ensure that services are deployed in a timely manner, especially in places where users gather, and services need to be dynamically scaled and scaled down. At the same time, the security of edge computing also needs to be paid attention to, devices may not be able to support the security tools of other devices, and diverse devices require specific protection measures. Any one vulnerable device can compromise the entire network, so data must be protected throughout its journey from the sensor to the cloud to prevent interception at any step.

5. Conclusion

Edge computing technology belongs to an innovative artificial intelligence computing model, the core of which lies in the decentralization of data processing functions and data storage capabilities to the edge of the network where data is generated. This mode can not only significantly reduce the delay caused by data transmission distance, but also improve the timeliness and efficiency of data processing. In traditional cloud computing architectures, data often needs to be transferred to remote data centers for processing, a process that is inevitably delayed due to physical distance. Near edge computing effectively avoids this problem by performing the calculation on the data source. The rise of the computing edge stems from the urgent need for real-time data processing and low-latency operations. With the proliferation of edge computing and iot devices and the proliferation of sensors, traditional cloud computing infrastructure is struggling to cope with the sheer volume of edge data.

Edge computing provides an efficient solution that enables data to be processed immediately where it is generated, speeding up the decision-making process and optimizing the use of network resources. With the advancement of technology, edge computing has evolved into a variety of forms to adapt to different application scenarios and industry needs. It plays a vital role in smart cities, autonomous driving, industrial automation, telemedicine and other fields. The architecture of edge computing is also evolving, including edge servers, fog computing, and the integration of distributed computing models.

Since edge computing is a rapidly growing field, the lack of standardization in practical applications has hindered its development. So, the diversity of edge devices that collect and generate data is proving to be a barrier. In this experiment, we first looked at tools that can help develop and deploy edge computing solutions, such as Docker, Kubernetes, and Terraform. We then described the architecture of our data processing pipeline and our framework to bring the pipeline together. Containerization makes it easy to deploy, scale, and monitor tasks using a coordinator. Some parts of the pipeline can be in different languages or different versions of the same language. We have also created a unified test platform that can be used to evaluate and compare the performance of Edge devices.

The flexibility of Edge computing devices and AI model architectures enables organizations to leverage the centralized benefits of cloud computing and the decentralized nature of edge computing to build more efficient and agile computing environments. The rise and development of Edge computing reflects a central requirement of our highly connected, data-driven era - the constant pursuit of efficient, real-time data processing capabilities. Continued innovation in technology, combined with the rapid growth of iot devices and the continued emergence of new applications, has driven the continued advancement and development of edge computing as a key element of modern computing architectures.

6. References

[1] Huh, Jun-Ho, and Yeong-Seok Seo. "Understanding edge computing: Engineering evolution with artificial intelligence." IEEE Access 7 (2019): 164229-164245.

[2] Fragkos, Georgios, Sean Lebien, and Eirini Eleni Tsiropoulou. "Artificial intelligent multi-access edge computing servers management." IEEE Access 8 (2020): 171292-171304.

[3] Slama, Sami Ben. "Prosumer in smart grids based on intelligent edge computing: A review on Artificial Intelligence Scheduling Techniques." Ain Shams Engineering Journal 13.1 (2022): 101504.

[4] Gong, Chao, et al. "Intelligent cooperative edge computing in internet of things." IEEE Internet of Things Journal 7.10 (2020): 9372-9382.

[5] Wen, X., Shen, Q., Zheng, W., & Zhang, H. (2024). AI-Driven Solar Energy Generation and Smart Grid Integration A Holistic Approach to Enhancing Renewable Energy Efficiency. International Journal of Innovative Research in Engineering and Management, 11(4), 55-55.

[6] Lou, Q. (2024). New Development of Administrative Prosecutorial Supervision with Chinese Characteristics in the New Era. Journal of Economic Theory and Business Management, 1(4), 79-88.

[7] Liu, Y., Tan, H., Cao, G., & Xu, Y. (2024). Enhancing User Engagement through Adaptive UI/UX Design: A Study on Personalized Mobile App Interfaces.

[8] Xu, H., Li, S., Niu, K., & Ping, G. (2024). Utilizing Deep Learning to Detect Fraud in Financial Transactions and Tax Reporting. Journal of Economic Theory and Business Management, 1(4), 61-71.

[9] Carvalho, G., Cabral, B., Pereira, V., & Bernardino, J. (2020). Computation offloading in edge computing environments using artificial intelligence techniques. Engineering Applications of Artificial Intelligence, 95, 103840.

[10] Liu, Y., Tan, H., Cao, G., & Xu, Y. (2024). Enhancing User Engagement through Adaptive UI/UX Design: A Study on Personalized Mobile App Interfaces.

[11] Huang, D., Yang, M., Wen, X., Xia, S., & Yuan, B. (2024). AI-Driven Drug Discovery: Accelerating the Development of Novel Therapeutics in Biopharmaceuticals. Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online), 3(3), 206-224.

[12] Li, S., Xu, H., Lu, T., Cao, G., & Zhang, X. (2024). Emerging Technologies in Finance: Revolutionizing Investment Strategies and Tax Management in the Digital Era. Management Journal for Advanced Research, 4(4), 35-49.

[13] Shi J, Shang F, Zhou S, et al. Applications of Quantum Machine Learning in Large-Scale E-commerce Recommendation Systems: Enhancing Efficiency and Accuracy[J]. Journal of Industrial Engineering and Applied Science, 2024, 2(4): 90-103.

[14] Ji, H., Alfarraj, O., & Tolba, A. (2020). Artificial intelligence-empowered edge of vehicles: architecture, enabling technologies, and applications. IEEE Access, 8, 61020-61034.

[15] Yang, M., Huang, D., Zhang, H., & Zheng, W. (2024). AI-Enabled Precision Medicine: Optimizing Treatment Strategies Through Genomic Data Analysis. Journal of Computer Technology and Applied Mathematics, 1(3), 73-84.

[16] Wang, S., Zheng, H., Wen, X., & Fu, S. (2024). DISTRIBUTED HIGH-PERFORMANCE COMPUTING METHODS FOR ACCELERATING DEEP LEARNING TRAINING. Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online), 3(3), 108-126.

[17] Lei, H., Wang, B., Shui, Z., Yang, P., & Liang, P. (2024). Automated Lane Change Behavior Prediction and Environmental Perception Based on SLAM Technology. arXiv preprint arXiv:2404.04492.

[18] Singh, S. K., Rathore, S., & Park, J. H. (2020). Blockiotintelligence: A blockchain-enabled intelligent IoT architecture with artificial intelligence. Future Generation Computer Systems, 110, 721-743.

[19] Xie, H., Zhang, Y., Zhongwen, Z., & Zhou, H. (2024). Privacy-Preserving Medical Data Collaborative Modeling: A Differential Privacy Enhanced Federated Learning Framework. Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online), 3(4), 340-350.

[20] Real-time Anomaly Detection in Dark Pool Trading Using Enhanced Transformer NetworksGuanghe, C., Zheng, S., & Liu, Y. (2024). Real-time Anomaly Detection in Dark Pool Trading Using Enhanced Transformer Networks. Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online), 3(4), 320-329.

[21] Guanghe, C., Zheng, S., & Liu, Y. (2024). Real-time Anomaly Detection in Dark Pool Trading Using Enhanced Transformer Networks. Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online), 3(4), 320-329.

[22] Chen, J., Yan, L., Wang, S., & Zheng, W. (2024). Deep Reinforcement Learning-Based Automatic Test Case Generation for Hardware Verification. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 6(1), 409-429.

[23] Zhang, Haodong, et al. "Enhancing facial micro-expression recognition in low-light conditions using attention-guided deep learning." Journal of Economic Theory and Business Management 1.5 (2024): 12-22.

[24] Wang, J., Lu, T., Li, L., & Huang, D. (2024). Enhancing personalized search with ai: a hybrid approach integrating deep learning and cloud computing. International Journal of Innovative Research in Computer Science & Technology, 12(5), 127-138.

[25] Zhou, S., Zheng, W., Xu, Y., & Liu, Y. (2024). Enhancing user experience in VR environments through AI-driven adaptive UI design. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 6(1), 59-82.

[26] Yang, M., Huang, D., Zhang, H., & Zheng, W. (2024). AI-enabled precision medicine: Optimizing treatment strategies through genomic data analysis. Journal of Computer Technology and Applied Mathematics, 1(3), 73-84.

[27] Wen, X., Shen, Q., Zheng, W., & Zhang, H. (2024). AI-driven solar energy generation and smart grid integration a holistic approach to enhancing renewable energy efficiency. International Journal of Innovative Research in Engineering and Management, 11(4), 55-66.

[28] Wang, Y., Zhou, Y., Ji, H., He, Z., & Shen, X. (2024, March). Construction and application of artificial intelligence crowdsourcing map based on multi-track GPS data. In 2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE) (pp. 1425-1429). IEEE.

[29] Lu, T., Jin, M., Yang, M., & Huang, D. (2024). Deep Learning-Based Prediction of Critical Parameters in CHO Cell Culture Process and Its Application in Monoclonal Antibody Production. International Journal of Advance in Applied Science Research, 3, 108-123.

[30] Zheng, W., Yang, M., Huang, D., & Jin, M. (2024). A Deep Learning Approach for Optimizing Monoclonal Antibody Production Process Parameters. International Journal of Innovative Research in Computer Science & Technology, 12(6), 18-29.

[31] Bi, Wenyu, et al. "A Dual Ensemble Learning Framework for Real-time Credit Card Transaction Risk Scoring and Anomaly Detection." Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online) 3.4 (2024): 330-339.

[32] Ju, Chengru, Yibang Liu, and Mengying Shu. "Performance Evaluation of Supply Chain Disruption Risk Prediction Models in Healthcare: A Multi-Source Data Analysis."

[33] Zheng, H., Xu, K., Zhang, M., Tan, H., & Li, H. (2024). Efficient resource allocation in cloud computing environments using AI-driven predictive analytics. Applied and Computational Engineering, 82, 6-12.

[34] Ma, X., Lu, T., & Jin, G. AI-Driven Optimization of Rare Disease Drug Supply Chains: Enhancing Efficiency and Accessibility in the US Healthcare System.

[35] Ma, D., Jin, M., Zhou, Z., & Wu, J. Deep Learning-Based ADLAssessment and Personalized Care Planning Optimization in Adult Day Health Centers.

[36] Ju, C., Liu, Y., & Shu, M. Performance Evaluation of Supply Chain Disruption Risk Prediction Models in Healthcare: A Multi-Source Data Analysis.

[37] Lu, T., Zhou, Z., Wang, J., & Wang, Y. (2024). A Large Language Model-based Approach for Personalized Search Results Re-ranking in Professional Domains. The International Journal of Language Studies (ISSN: 3078-2244), 1(2), 1-6.

[38] Ni, X., Yan, L., Ke, X., & Liu, Y. (2024). A Hierarchical Bayesian Market Mix Model with Causal Inference for Personalized Marketing Optimization. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 6(1), 378-396.

[39] Zhang, H., Pu, Y., Zheng, S., & Li, L. (2024). AI-Driven M&A Target Selection and Synergy Prediction: A Machine Learning-Based Approach.Zhang, H., Pu, Y., Zheng, S., & Li, L. (2024). AI-Driven M&A Target Selection and Synergy Prediction: A Machine Learning-Based Approach.

[40] Ahmed, I., Zhang, Y., Jeon, G., Lin, W., Khosravi, M. R., & Qi, L. (2022). A blockchain-and artificial intelligence-enabled smart IoT framework for sustainable city. International Journal of Intelligent Systems, 37(9), 6493-6507.

[41] Ghosh, A., Chakraborty, D., & Law, A. (2018). Artificial intelligence in Internet of things. CAAI Transactions on Intelligence Technology, 3(4), 208-218.