Assessment of Innovative Architectures, Challenges and Solutions of Edge Intelligence

[1]Heikku Siltanen and [2]Lars Vlrtanen
[1,2] Department of Computer Science, University of Helsinki, Yliopistonkatu 4, 00100 Helsinki, Finland
[1]heikkusinki@yahoo.com

Correspondence should be addressed to Heikku Siltanen : heikkusinki@yahoo.com

Article Info
Journal of Machine and Computing (http://anapub.co.ke/journals/jmc/jmc.html)
Doi: https://doi.org/10.53759/7669/jmc202202020
Received 22 March 2022; Revised form 20 May 2022; Accepted 23 July 2022.
Available online 05 October 2022.

Abstract – Data collecting, caching, analysis, and processing in close proximity to where the data is collected is referred to as "edge intelligence," a group of linked devices and systems. Edge Intelligence aims to improve data processing quality and speed while also safeguarding the data's privacy and security. This area of study, which dates just from 2011, has shown tremendous development in the last five years, despite its relative youth. This paper provides a survey of the architectures of edge intelligence (Data Placement-Based Architectures to Reduce Latency; 2) Orchestration-Based ECAs-IoT. 3) Big Data Analysis-Based Architectures; and 4) Security-Based Architectures) as well as the challenges and solutions for innovative architectures in edge intelligence.

Keywords – Edge Computing, Artificial Intelligence, Edge Intelligence, Software-Defined Network

I. INTRODUCTION

Due to recent advancements in Artificial Intelligence (AI) technology, the number of AI-based applications and services is rapidly increasing. Using AI, it is now feasible to achieve cutting-edge performances in a broad variety of fields, including facial recognition software, natural linguistic processing, machine learning, traffic prediction and anomaly-based. Some deep learning methods are still not enabled by current end devices, despite their increasing capabilities. Many cloud-based programs, such as Siri, Cortana, and Googling Now, become inoperable if the connection goes down. In today's smart applications, centralized information handling is also a popular feature. A solitary cloud-based information center is required for this. The amount of data generated by billions of phone users and IoT devices spread at the edge networks is immense, but it's not all that valuable.

Cisco expects that by 2021, data generated by mobile customers and IoT devices would total 850 ZB. It is impossible to upload such a big volume of information to the cloud at current speeds without causing unacceptable delays for end users. Consumers' worries over their personal data have grown in recent years, though. The European Union verified GDPR (General Data Protection Regulation) to safeguard the users' data. Mobile users who have the capacity to save their data on a cloud system run the risk of privacy leakage, or the unauthorized extraction of such data by hackers or companies using cloud-based intelligent apps.

Edge computing, an expansion of cloud computing, has evolved as a means of bringing web services closer to the user. Edge computing alludes to the virtual computing model, which has the capability to deliver storage, networking and process capabilities at the networking edge. Devices like mobile base stations and vehicles, as well as various IoT gateways, routers and mini data centers are all examples of edge servers, which live up to their name by providing services to end devices. An "edge device" is a piece of hardware that makes service requests to an edge server. This might include anything from mobile phones to Internet of Things (IoT) hardware and embedded systems.

There are three main advantages of the edge technology concept that may be summarized as follows. Due to ultra-low frequency calculations taking place near to the source data, data transmission times are considerably decreased. Edge servers provide near-instantaneous response times to end devices. End devices' energy consumption might be reduced if computational tasks are offloaded to edge servers. Consequently, battery life on end devices will be improved. In case there are scarce resources on the edge servers or devices, cloud computing might still be scaled. As such the cloud services would be put to work. End devices with existing assets may also collaborate on a project with one another. The edge computing model is capable of handling a broad variety of application scenarios due to its versatility.

AI-based applications face a number of challenges that may be addressed by merging edge computing with AI. Intelligence in the "edge," "mobile," or "edge-to-edge" contexts is being discussed. Data collecting, caching, processing, and analysis near to the data source are all examples of "edge intelligence," which is a network of linked devices and systems that increase data quality and speed while simultaneously protecting the safety and confidentiality of that data. In this data. Like cloud-based intelligence, Edge Intelligence analyzes data locally, protecting users' privacy, reducing response time and saving bandwidth resources. User data may also be used to build customized machine learning and deep learning models.

A key component of the 6G system is likely to include edge intelligence. AI has the potential to aid with edge computing as well, which should not be overlooked. In this paradigm, "intelligent edge" is used instead of edge intelligence. Unlike intelligent edge, edge intelligence concentrates on building intelligent programs in the contexts of edge devices and safeguarding the privacy of its users, rather than solving edge computing difficulties with AI solutions. Rather than focusing on the cognitive advantage, we will ignore it for the time being. This paper subdivides the human-created architectures into further classes: 1) Data Placement-Based Architectures to minimize Latency; 2) Orchestration-Based ECAs-IoT. 3) Big-Data-Analysis-Based Architectures; and 4) Security-Based Architectures. The rest of the paper is organized as follows: Section II presents a review of the previous works. Section III provides a survey of innovative edge intelligence architectures, while Section IV analyses the challenges and solutions of edge intelligence architectures. Lastly, Section V provides final remarks to the whole research.

II. LITERATURE REVIEW

A number of studies have defined edge intelligence viability by implementing various concepts to real-world application fields. Foukalas and Tziouvaras [1] have employed smartphones and edge servers to build an application for face recognition. From 900ms to 169ms, the latency has been lowered. Cloudlets may cut energy usage by 30 to 40 percent, for example, in cognitive assistive devices such as smartwatches. Some academics are particularly interested in the performance of edge computing and AI. For activity recognition, Tang, Liu, Xiao and Sebe [2] constructed a limited deep learning model. The example shows that simple DL models may be deployed to smart devices, which outperform shallow models.

Additionally, wearable and integrated gadgets are subjected to the same tests. G-Board, Google's smartphone-based prediction model, is an instance of edge intelligence. G-board picks up on the unique typing styles of those who use it throughout the training process. As a result, the trained G-board could be used to power experiences that were specifically suited to the application's usage by the user.

Researchers studied human-created architectures for edge intelligence. Nonetheless, there were a few flaws in this study. This study breaks down architecture into many different types. However, since architectural search requires hardware, most researchers are unable to use this search approach. Human-made architecture is the subject of the majority of the literature now available. A deep neural network for mobile and embedded devices was created using depth-wise separate convolutions by Georgiev, Bhattacharya, Lane and Mascolo [3]. Using MobileNets, a convolutional filter may be divided into two types: a depth-wise filter and a point-wise filter. Convolution just filters the input channels, which is a drawback. The combining of depth-wise and 1-to-1 inversion with separate convolution may be utilized to overcome this limitation. It uses 33 depth-wise separate convolutions that need 8–9 percent less computation than standard convolutions to execute effectively. The deployment of KWS algorithms and depth estimations on edge devices, on the other hand, may also profit from the usage of convolutions, both in terms of points and depths.

Another approach is to use group convolution to reduce the processing expenses associated with the model construction process. It is unable to use certain basic designs like Xception and ResNeXt because of the resource-intensive deep 1-to-1 convolutions. For 1-to-1 convolutions, Zhang, Lo and Lu [4] suggest using pointwise group convolutions, which reduce the computational burden. Because the outcomes of one band are formed from a small percentage of the input networks, this has an unanticipated effect. Information transmission between groups may be hampered by "scarcely-connected" convolutions, which are often utilized in depth- and organization convolutions. When it comes to dealing with this problem, Qin et al. suggest a merge and develop approach. It is possible to generate a major feature map by integrating information about the same location gathered from several sources. Data from the newly added features are collected and added to the network in this way. As a consequence, the problem of intergroup data loss is effectively addressed since information is distributed across all channels.

This paper subdivides the human-created architectures into further classes. Orchestration-based ECAs-IoT for reducing latency via data placement. In addition, there are architectures based on big data analysis and security. Edge computing architectures are also discussed in this study, along with their obstacles and possible solutions.

III. SURVEY OF INNOVATIVE EDGE INTELLIGENCE ARCHITECTURES

Data Placement-Based Architectures

For reduced maintenance costs and trustworthy SLAs (Service Level Agreements) compared to the conventional data storage method, more and more enterprises are moving their data to the cloud as cloud computing grows. Cloud Service Providers (CSPs) in the mainstream provide a number of data storage options to meet the needs of various customers. CSPs provide a wide range of price options for the same functionality. In addition, different locations have varied pricing policies for the same CSP. Data migration between datacentres of the same CSP is less expensive than migration between data centers of distinct CSPs.

Additionally, a single cloud is subject to risks of vendor lock-in, i.e., main concerns include the pricing of cloud computing and the disruption of Service Level Agreements (SLA). Users may be forced to pay hefty relocation expenses in these cases. An ant colony algorithm-based method was proposed in our past efforts for cost-effective information hosting in multicloud locations with high accessibility. Because of this, we're able to split up our data and store it in numerous CSPs instead of just one. Many researchers have worked to provide a data input to the system in multicloud settings that is both cost-effective and high-availability based on user requirements.

Choosing a data placement strategy is influenced by the data object's workload. If you have a certain amount of time, then the amount of data you have to deal with (DAF, or Get access rate) is directly connected to the amount of data you have to retrieve (DAF). The workload fluctuates during the course of a data placement's lifecycle. It is more probable to be hosted in CSPs with reduced out-bandwidth charges if the data item is read-intensive and in hot-spot state. In contrast, data that has a low DAF is more likely to be kept in CSPs with smaller capacity costs since it is storage-intensive and has a cold-spot status to it. Because of this, as DAF grows, it is possible that a user's data placement technique may result in higher out-bandwidth charges if they stick with it over the storage lifecycle. It is possible to incur high storage costs if a user utilizes tactics that are better suited for hot-spot status across the whole life-cycle of data storage.

It is vital to build a technique for dynamically adjusting the data processing scheme depending on abstract data workload in order to decrease overall costs and boost reliability over the data object's lifespan. The total cost cannot get the optimal outcome because of the unpredictability of future data workload. As a result, the data stream placement technique relies heavily on forecasting future workloads. It's important to think about how to build a dynamic placement system that takes future accessibility frequency into consideration. Data generated by IoT systems is enormous. E-health and other critical IoT applications demand low latency data retrieval. ECAs-IoT have difficulty distributing IoT data to the correct edge nodes. So far, the following designs have been proposed:

IFogStor and IFogStorZ

There are a few ideas in [5] that use fog-node distributions and variations to reduce the overall latency of fog-node storage and retrieval of IoT data. A collection of IoT-enabled devices, fog computing devices, data facilities, IoT services comprise the systems infrastructure. Data created by the IFogStor system is stored and retrieved from efficient fog nodes to decrease overall network latency. Run-time execution of the data takes place on a node that is resilient. Details on information flow, network delay, application location and storage capacity are all available to this node. **Fig. 1** depicts the architecture of the IFogStor system, which comprises three basic types of actors: special nodes that preserve IoT data, such as fog nodes or data centers. Any layer, save for layer 0, may include them. Data producers are nodes that create information. Nodes of this sort may be found in a variety of different tiers. Nodes that analyze or interpret IoT data are known as "data consumers. They might exist in multiple tiers. As a result of their capabilities, fog nodes might serve as both a data host and a data producer and consumer all at once.

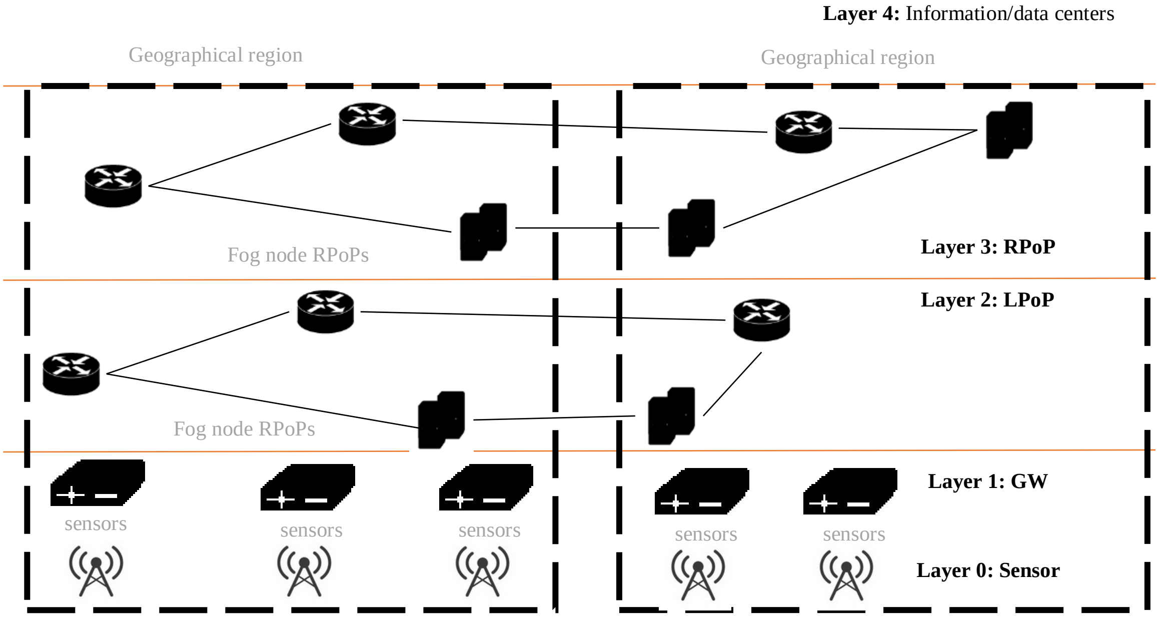

Fig 1. The architecture of the IFogStor system

Two remedies were suggested to mitigate the issue:

- IFogStor: a single integer program-like method to solve the issue of data placement. For small-scale applications, it identifies the best location; but, for a wide range of applications, its efficiency is unacceptably slow.
- A technique that uses regional points of presence (RPoPs) as partitioning sites to separate geographical places as part of a divide-and-conquer strategy. A global solution is found by solving the specific problems in each site.

However, this method does not locate the best location, but it significantly reduces the amount of time it takes to get data.

IFogStorG

Even tthough IFogStorZ is easy to build, it loses a significant optimality amount when the generators of data are far away from information consumers. The amount of fog nodes and Internet of Things (IoT) services may also differ across subregions. As a result, subproblems with imbalanced causes are discovered. Improved runtime speed and reduced intricacy of the data placement technique are the goals of IFogStorG. A more advanced network topology-adapting technique is therefore possible. Following are the main considerations in the partitioning step: Data users and data providers are kept as close to one another as feasible in order to maximize the strategy's efficiency. Data consumers and producers were represented by matrices, while fog nodes were represented by the adjacency matrix, which mapped the value of latency in the architecture. For each subgraph, they used the IFogStor technique to find a solution. After that, the findings of each sub-section are added together to arrive at the ultimate global answer. An example of a real-world smart city was used to gauge the effectiveness of their approach. The trade-off in number of data copies and latency reduction should be made as the number of information subscribers grows.

Multireplica Data-Placement Approach

Baranwal and Vidyarthi [6] dealt with the problem of delay that arises whenever information consumers from various geographic regions subscribe to the same information or data, but only a single copy of the data resides at the precise fog node. A greedy technique called IFogStorM was devised to reduce latency as a result of this problem. Overall latency was lowered with 10 percent over IFogStorG, and with 6 percent over IFogStorZ, according to the results.

Orchestration-Based ECAs-IoT

Because IoT networks improve system and security stability, and make network maintainability easier, they are considered a significant problem. As a central solution, some ECAs-IoT use software-defined networks, whereas other ECAs-IoT utilize other methodologies.

Services and Tasks Allocation-Based Architectures

By distributing critical delivery of services and task allocation to the most efficient edge nodes, cloud infrastructure improves Internet of Things (IoT) systems. This section focuses on ECAs-IoTs, which oversee the distribution of activities and resources in IoT systems.

Mobile Fog Services' Allotment (MFSA)

IoT system may benefit from edge computing, which manages service processing and job distribution at the edge node. In MFSA, Ibaraki [7] used an Integer-Programming (IP) formulation to minimize the overall costs of delivering service while assigning requests to the available resources. The approach assigns a "probability of availability" to each server. Users and fog nodes are connected by a middleware controller, which has access to accurate information about the complete architecture. A nondeterministic component of each server's availability is also known to the program. Each user's service request is broadcast to various servers in order to handle this component.

However, the user's ability to submit queries to an unlimited number of servers is constrained. It was also suggested that each user would have a budget for making requests to every server. The server resources might be shared by various users only if the total number of services supplied by the server does not exceed the capacity of each server. Servers' availability probabilities are unrelated to one another. There is a fee for every service. The Quality of Service (QoS) level was also governed by a set of limitations. Constraints include the likelihood that the user-specified server will be available. For the purpose of reducing the overall cost of assigning services, this challenge has to be solved.

Multiagent-Based Flexible ECA-IoT (MAFECA)

MAFECA is a multi-agent-based, adaptable version of the Internet of Things. Pirbhulal et al. [8] proposes a modular architecture that addresses the usual IoT network issues while optimizing tasks distribution between edge machines and the cloud. Two system abilities are employed in this architecture: user-oriented and environment-adaptation. The capacity to cater services to individual users in real time is enabled by the usage of data received by IoT devices, such as user behavior. When a job has a high volume and high quality, the processing site is determined by the environment in which it is being performed.

Hierarchical Architecture to Place Mobile Workloads (HAM)

HAM was designed by Lee and Lee [9] by presenting an algorithm that puts mobile workloads across various levels and determines how much processing capacity each task needs.

Scalable IoTs Architectures-Centered on Transparent Computation (SAT)

When it comes to the allocation of services, Das, Santra, Bodra and Chakravarthy [10] presented an infrastructure, which utilizes transparent computing to maximize scalability and minimize response time. An end-user layer is made up of IoT devices; an end-layer disburses solutions to end-users; a foundational internet protocol connects edge computing systems and the virtualized system; a cloud layer incorporates high-performance computer technology and stashing resources to negotiate with huge datasets; and a layer for managing the entire infrastructure, which encompasses a managerial and functionality layer.

Edge-centered Aided Living Platforms for Home Automations (E-ALPHA)

Device handler, which separates the technology-based operations from communication-based services; embedded system, which dynamically loads specific protocols; and database, which is responsible for storing and retrieving data from the e-health applications were some of the components proposed by Kopmaz and Arslanoğlu [11] in their architecture for enhancing e-health applications. The EdgeCloudSim simulator was used to model this design.

SDN-Based Fog Architectures

We need to manage resources and data in IoT networks. ECAs-IoT might benefit from Software-Defined Networks (SDNs). ECAs-IoTs, which utilize SDN technologies to potentially manage the networks are covered in this section.

Multi-level SDN-centered 5G Vehicular Architecture (VISAGE)

Every year, the number of automobiles on our roads increases. Two sub-frameworks of the 5G-VANETS scheme proposed by Das and Gurusamy [12] are Local SDN Controller (LSDC), and Central SDN Controller (CSDNC). Fog nodes, such as moving automobiles or even stationary cars, may be used in this way. **Fig. 2** depicts the components of the architecture in [13]: The CSDNC is a permanent portion of the system that is hosted in the cloud and reflects global intelligence. Intelligence is centralized and represented by the DNC. In this case, it is a fog cell that is governed by the CSDNC. The LANC controls the fog cell nodes. Customers are unable to do their own computations, and fog nodes step in to help. People, vehicles, or organizations all fall under this broad category. In order to keep clouds and fog nodes connected, base stations must be used. Fog-SDN capabilities are broadcast by the LSDNC in VISAGE. Alternatively, each vehicle might use the fog cell's services e.g., fog nodes. The LSNDC might be used to link the fog cells to the Internet. As a result, the LSD interacts with the CSDNC, which in turn coordinates the resources.

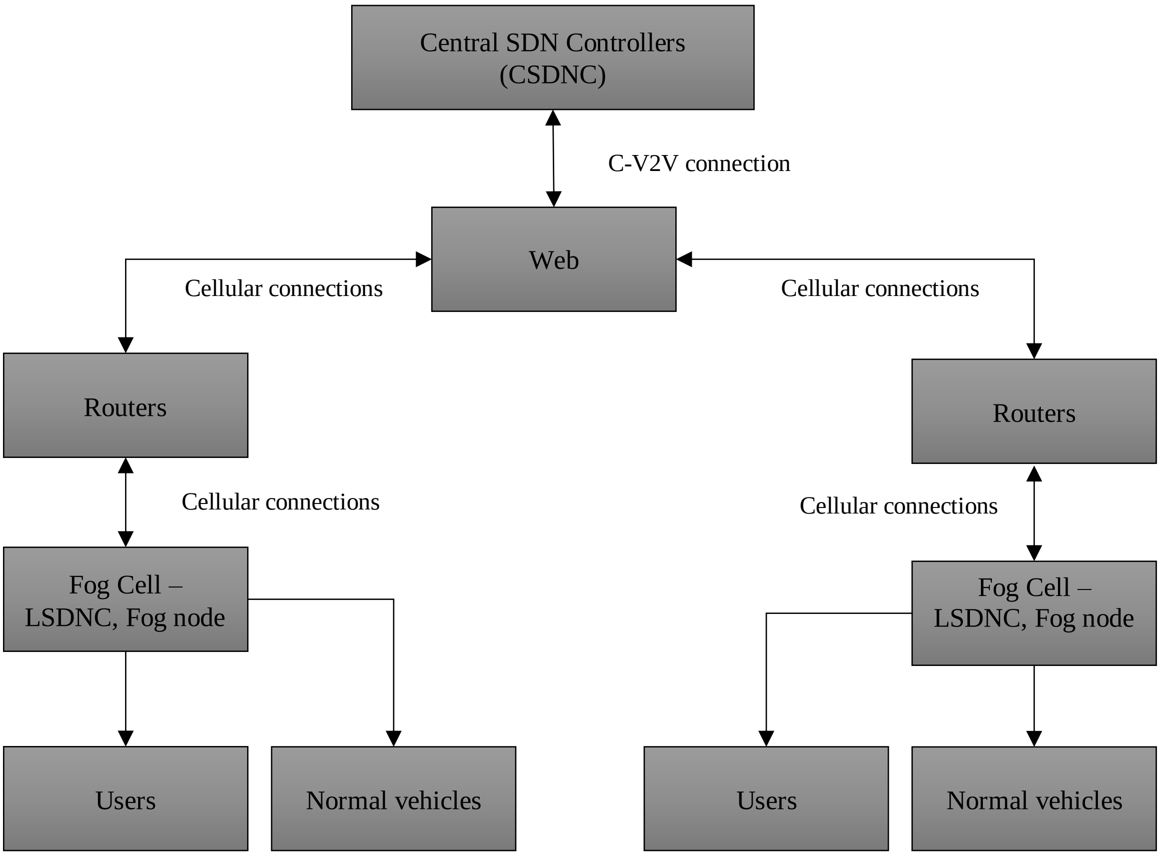

Fig 2. Central SDN Controllers (CSDNC) components

SDN-Based VANET Architecture (FSDN)

VANET's resources are better managed thanks to the design in [14]. Components of this architecture include the following elements: To control a group of Roadside Units (RSUs), forward data, store data on specific road systems, and provide timely services, the SDN controller uses a Road-Side Unit Controller (RSUC), which is located on the roadside and is accountable for global intelligence. The RSU is also obliged for communicating data and it typically controlled by SDN controllers. Cellular base stations are in charge of local intellect, data forwarding, and conveying fog warnings. However, there is no resource management or network orchestration in this design, thus there is no way to evaluate how well it performs.

Software-Defined Fog Computing Networks Architectures for IoTs (SDFN)

Sham and Vidyarthi [15] developed an integrated SDN and fog processing system, which differs from earlier systems by being generic. For the architecture, there are three main components: end-devices, SDN controllers (which are responsible for picking the best access points for the IoT nodes with the skillset about systems e.g., fog-device capacity to delegate works to them) and fog architecture. It is the center of the network where cloud computing takes place, and fog devices utilize APIs to give their services. Using a hierarchical deployment, the same application may execute on numerous fog devices at the same time A job is assigned to each fog device depending on its individual capabilities. Inter-transportation systems, video monitoring, and precision agriculture might all benefit from this architecture's flexibility and scalability in the Internet of Things. For the evaluation of the architecture, there is no simulation and no central control of the networks.

SDN-Based Cloudlet Architectures

In this part, we show how an SDN and a cloudlet may be used to administer an IoT networks.

Dynamic Distribution of IoT Analysis (DDA)

Multilayer architecture built on SDN that keeps track of Internet of Things (IoT) traffic, uses congestion prevention methods, and disperses analysis of IoT data among a Data Center (DC) and the network's edge has been suggested by Lozano-Rizk et al. [16]. The structure of this design is as follows: Connections to DCs at the network's edge is given, and the median bandwidths of IoTs data flows is tracked at the architecture tier of the networks. A specialized DC controller exists for each DC in the infrastructure layer. Cloud orchestrator is also deployed at upper layer of the DC controllers, providing federated cloud services. An IoTs-aware Transportation SDN Orchestrators (TSDNO) work as a controller of controllers and sits atop each domain's SDN controllers. TSDNO is also in charge of keeping IoT traffic flowing smoothly. IoT-aware GSOs are at the top of cloud orchestrators, orchestrating global end-to-end operations from the cloud to the edge.

Big Data Analysis Architectures

Using sensors, massive volumes of data are generated every second. Fog computing architectures for large data processing are discussed in this section.

Hierarchical Dicentralized Fog Computing Platforms for the Smart City (HDF)

Birkholzer, Cihan and Bandilla [17] presented a four-tiered hierarchical framework. The suggested architecture's tiers are shown in **Fig. 3.** Various sensors are spread around the network in order to gather and create data. There are several kinds of sensors in Layer 4 that are deployed across the environment to gather and produce data. Subsequently, Layer 3 receives the unprocessed sensor data. In Layer 3, the edge devices manage a layer 4 sensor network that covers a local area, such a neighborhood. In this layer, edge devices perform real-time data analysis. For example, reports are generated by evaluating the data, and the infrastructure is alerted to dangers that were detected by sensors on the edge devices. Edges are then pooled and linked to one of intermediary compute nodes in Layer two when this step is completed. Using temporal and geographical data, this node can identify and respond to potentially risky situations. When it comes to total infrastructure analysis, monitoring and management, a cloud computing data center is the last layer. They created a pipeline system prototype and ran simulations of 12 distinct events using the sensors as part of their evaluation process. A hidden Markov structure was used to train the model to identify the events. Fog computing using cloud resources reduced latency in big-data processing, according to the findings.

According to Pang, Wang and Fang [18], a comparable architecture was proposed for collecting data from a variety of sensors by employing cloud computing and hierarchical edge strategy. Many sensors provide information to the first layer of collectors, which is called edge level, before it is sent to a generic cloud service provider. In the hands of the main service provider, all information is centralized. Once fused data has been obtained, it may subsequently be utilized for big-data analysis by tailored service providers.

Security-Based Architectures

There are several security issues that arise as a result of the structure of IoT networks, including data confidentiality and authentication. This section focuses on ECAs-IoT, which tackle security issues on IoT networks, and explores edge-computing technologies. Following are some examples of IoT network topologies that address security concerns while avoiding the use of Software-Defined Networks (SDNs).

Privacy Preservation While Aggregating the Information/Data (P2A)

Hu, Dong and Wang [19] suggested an infrastructure for protecting sensor data privacy, which typically manages multi-functional aggregation, computing overhead, and linking overhead, among other things. Devices, fog networks, fog centres, and a cloud provider are all part of this system, as depicted in **Fig. 4**. Smart gadgets use sensors to gather information. Two separate fog nodes receive the acquired data in order to maintain the privacy of the user. When fog centers issue aggregation queries, the fog nodes act as store nodes to aid in the aggregate of data. As a consequence of this, fog centers are able to gather the findings of queries made by fog nodes. Sending the primary query results to the internet center is the next step in the process. The cloud center is a service provider-managed aggregation application. Fog centers and cloud centers are built to be untrustworthy since they attempt to acquire secret original data. In order to avoid collusion, fog nodes have an interest in the original data since they can't trust each other.

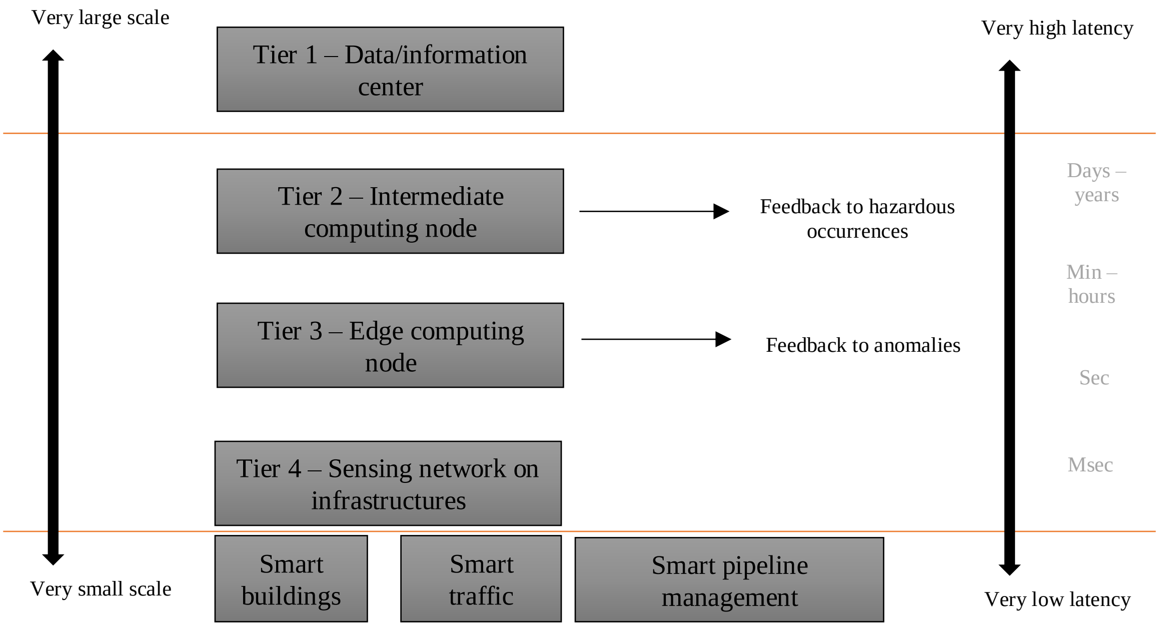
Very large scale
Very small scale
Tier 1 – Data/information center
Tier 2 – Intermediate computing node
Tier 3 – Edge computing node
Tier 4 – Sensing network on infrastructures
Smart buildings
Smart traffic
Smart pipeline management
Feedback to hazardous occurrences
Feedback to anomalies
Very high latency
Very low latency
Days – years
Min – hours
Sec
Msec

Fig 3. Suggested architecture's tiers

It was suggested by Singh [20] to aggregate data while preserving confidentiality at the same time using a machine learning-based approach. Rather than sending the real data, the model delivers a projected value that it has learned via training. Each region's training data is included in the dataset. It's explained here how the procedure works. For example, the cloud center may send questions to fog centers such as "average," "q percentile," "min," "max," and "summation aggregation," among others. All of these inquiries are sent to the fog centre via the cloud center. Due to its inability to respond to cloud center inquiries, the fog center produces its own queries from the ones that were originally sent. Sensors provide fog-node sensory data after separating sensory input into two parts. The fresh set of queries created by the fog centre are used to train and forecast the incoming data. Finally, the cloud center gets the expected values from the fog center and retransmits them to it.

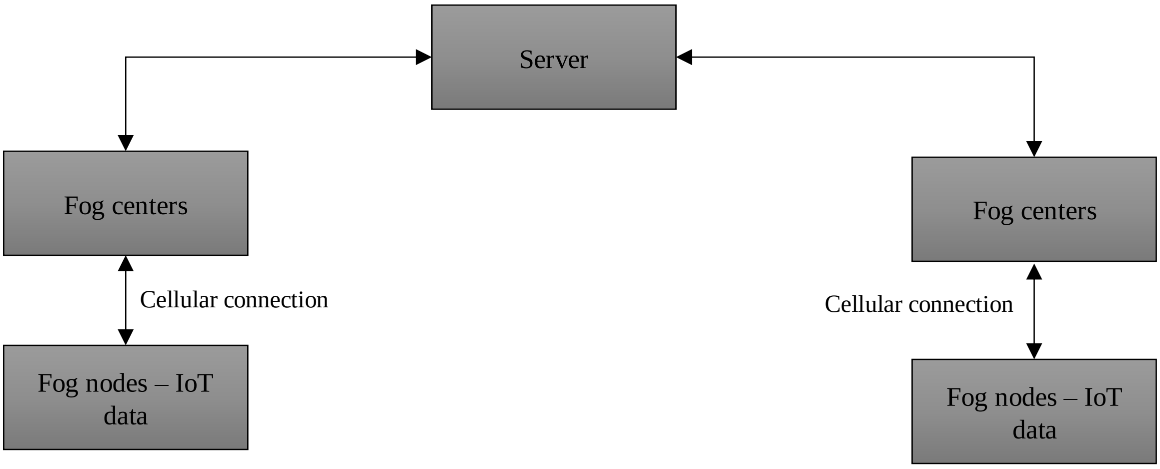

Fig 4. Parts of the server system

Lightweight Security based on Virtualization (LSV) Mechanisms

According to Tiburski et al. [21], embedded virtualization and trust mechanisms may be used to protect edge systems with no necessity to re-engineer the programs placed in edge devices. Certain security needs are met by the proposed architecture: the secrecy of permanently stored components, the authenticity of executed codes, and run-time state integrity. Securing the boot, key storage, and cross domain communications are all part of the security architecture, which is composed of four techniques. By using Root of Trust (RoT), an algorithmically secure foundation, and the Chain of Trust (CoT) that would be built to start it up only after cryptographically secured technologies by a legal source is first installed employing public-key authentication, edge endpoints are protected. Keys are also kept in specialized hardware, which is also accountable for verifying and executing the RoT process. Thus, in order to accomplish a second degree of safe boot verification, this embedded virtualization design uses several Virtual Machines (VMs) from various manufacturers.

Running time assaults are still possible even after the CoT has been formed and hardware assisted virtual maintains a TEE; hence, the system architecture must be secured via run-time mechanisms. It was decided to test the system's design based on three metrics: storage footprint, speed, and latency between VMs. Edge-device protection might be provided with no need to re-engineer edge application.

Service Architectures with Balanced Dynamic centered on the Cloud (SBDC)

Since IoT devices have inherent limitations, traditional security measures are rendered ineffective. It was reported that Xu, Hang, Jin and Kim [22] used distributed edge devices to create a secure architecture built on trust methods and service templates that could withstand assaults and comparable service demands. It is made up of two themes: the service and parsing templates.

The edge system, the edge system, and the cloud system make up this architecture's three main components. The data collecting, processing, and app-service levels are all separated into three tiers. It is on the app-service layer where the cloud is located. The edge platform and edge network are all placed at this level. A trust condition for IoT devices is established, and the trusted IoT devices are chosen to potentially execute services. This design dynamically changes the IoT load, and it satisfies end-users' needs, such as authenticity and accuracy, using this architecture. Virtualization processes are performed by converting physical components into virtual devices on this edge platform. Cloud load is dynamically adjusted through the use of edge layer services. IoT reliability is ensured through the use of the edge node at the data levels as well as the foundation of service-parsing templates.

Services, which need additional resources than those available at the edge computing processing layer are processed in the cloud. Old data is logged and utilized for future analysis and data mining is handled by the cloud, which creates and stores service parameter templates and stores information matching them. The MATLAB platform was used extensively to conduct extensive tests to assess its architecture. Four Interconnects and a cloud make up the system's architecture. There is only one edge platform per IoT network. The findings suggest that this design has the potential to improve service efficiency and data integrity.

SIOTOME: Edge-ISP Collaborative Architectures for IoTs Security

When it comes to IoT devices, Dina Merlinda Izzah [23] collaborated with the Internet service provider (ISP) to discover risks and vulnerabilities early on. When compared to typical networks, SIOTOME's intrusion-detection system can learn from several domains to recognize different types of attacks. As an example, one domain may represent an individual Internet Service Provider's (ISP), as well as an individual building network. There are two high-level domains in the SIOTOME system architecture: SIOTOME/edge and SIOTOME/cloud. As part of the system design, the following components may be found in the smart house: IoT data is collected by the edge data collector, which is then processed by the edge analyzer, which then reports back to the edge controller via SDN for further analysis. The edge controller, based on SDN, is then used to configure the gateway, ensuring that all IoT devices on the home network are managed by the gateway. Cloud collector, cloud analyzer, cloud controller, and cross-layer controller make up the SIOTOME/cloud component.

Edge-Computing Architectures for Mobile Crowd Sensing (MCS)

In order to support mobile crowd sensing, Hamdan, Ayyash and Almajali [24] suggested a four architecture: the user-equipment layer, which includes IoTs devices like wearable sensor devices; the edge computing layer, which manages workers in specific geographic areas; the cloud computing layer, which processes complex data; and the application layer, which analyzes the data. When a wireless crowd-sensing situation happens, this architecture delivers an alert to mobile devices, ensuring data privacy and reducing latency, while it distributes data across servers.

ECAs-IoTs Integrating Virtualized IoTs Devices (ECV)

Ullah et al. [25] presented an ECAs-IoTs to develop the intelligent cities. This design acts as the intermediary layer for potentially processing IoTs data. Data validation, metadata annotation, and security are all part of this architecture's six components, which include collection proxies, which connect every IoTs system to other components in the design; information affirmation that potentially maintains the integrity of gathered information; and safety, which accomplishes symmetric information encryption for Cloud computing before conveying them to virtual IoT devices.

IV. CHALLENGES AND SOLUTIONS TO EGDE INTELLIGENCE ARCHITECTURE

As sensors become more widely used in the real world, more physical objects are being linked to the Internet of Things (IoT) to share data. Wearable medical devices, smart cities, smart homes, and environmental perception are just a few examples of where Internet of Things (IoT) technology is now being used. Traditional IoT services need data to be uploaded to cloud servers by sensors and devices that are linked through IoT. The IoT devices will get the processed data when the tasks have been finished. Sensors and gadgets may benefit from cloud computing, but the significant data transmission overhead cannot be overlooked. In 2018, the total number of IoT-enabled devices throughout the globe surpassed 11.2 billion, and this number is expected to expand to 30 billion by 2025. However, network capacity expansion is now lagging significantly behind the growth rate of data, and the complexity of the network environment makes it difficult to reduce latency. Traditional IoT services have a bandwidth crunch, which must be addressed if they are to be successful.

A new computing concept known as "Edge Computing" (EC) has recently been suggested to alleviate the aforementioned bottleneck. EC is a term used to describe the technology that moves computing workloads to the periphery of the network. Comparing EC to cloud computing, there are several benefits: end-users' confidentiality is protected, data transmission is more efficient, network bandwidth is less burdened, and data centers' energy consumption is lessened. To reduce latency, Edge Nodes (ENs) may process, store, and send raw data produced by IoT devices rather than relying on centralized cloud platforms [26]. This eliminates duplicate data transfer. EC may better serve IoT and mobile computing applications that have tight reaction time requirements.

There is no guarantee that EC will solve all of your problems. It is true that IoT systems under the EC have substantially increased their potential in many domains, including computation offloading, accurate location and real-time processing. However, it is also true that low-latency data manufacturing near end-users has been given credit for this expansion. EC, on the other hand, raises additional security concerns and expands the system's attack surfaces in three ways: As a result of the ENs being scattered over the network, it is impossible to centrally supervise all of the equipment. The attacker may target vulnerable ENs and utilize the nodes it has taken control of as a launching pad for an assault on the whole system. Limitation in processing power: unlike cloud computing, the physical construction of ENs limits the computing power available, making them vulnerable to large-scale centralized assaults like Distributed Denial of Service (DDoS), which may inflict significant damage to the ENs in question. A broad variety of technologies, including wireless sensor systems, mobile data collecting, grid computing, and mobility data gathering, are used in EC. It is challenging to build a single security mechanism and ensure consistency across multiple security domains in this diverse environment.

Due to the inherent dangers of edge computing, several security strategies and algorithms have been developed. Algorithms and models for intrusion prevention, privacy preservation, and access control all follow a consistent pattern. Traditional defenses are often rendered obsolete by the constant improvement of assault tactics and approaches. Artificial Intelligence (AI) is fascinating because it can help solve some of the most pressing security and privacy challenges. DDoS assaults and Distributed Denial of Service (DDoS) attacks are prevalent forms of intrusion. DDoS refers to the use of several hacked ENs to assault the server, increasing the strain on the website and affecting the server's responsiveness to routine requests. Attacks from the hijacked ENs are detected by the network's intrusion detection system (IDS), which blocks their access by looking for unusual network traffic. ML may assist IDS detect intrusions more quickly and correctly than classic identification approaches by extracting harmful access patterns from earlier data sets and training on that data.

We need to keep our privacy protected since IoT devices are present in every area of our life, which holds a lot of sensitive information. In order to assure data security and privacy protection, the majority of currently used technologies encrypt the transferred data. However, the following approaches often have a substantial computational burden, rendering them inaccessible to resource-constrained ENs. Distributed Machine Learning (DML) reduces the danger of data leakage and network stress during transmission by making the ENs only need to communicate the variables to other ENs for collaborative learning after each training instead of directly transferring the actual data. Access control represents a critical problem when several Internet of Things (IoT) devices are working together in that environment. In other words, only the networks and data under their authority may be accessed by each authorized node. The classification technique under ML corresponds to the necessity to classify ENs into distinct groups based on permissions. Low-privilege IoT applications and high-privilege IoT devices are categorized by the algorithm. There will be rigorous controls on who has access to these high-privilege gadgets.

Artificial Intelligence (AI) is increasingly being used in a wide range of edge security applications as research into the topic progresses. However, the implementation of EN-related ideas faces several obstacles. Large volumes of unambiguous data are critical to the efficacy of ML training; yet, the assumption of adequate information is that the computer has suffered mass assaults and can properly recognize these hostile actions. The model's performance will suffer if the training set is tampered with, thus it's important to keep an eye out for assaults on the training set. However, since ENs have a limited computation and storage capacity, a lightweight AI method is also required.

V. CONCLUSION

The term "edge intelligence" refers to a network of interconnected devices and systems \used for artificial intelligence-based data gathering, caching, processing, and evaluation near to the point of data acquisition. Data processing quality and speed may be improved while maintaining data privacy and security via edge intelligence. Because of AI's recent advancements, the number of AI-based applications and services is on the rise. Face recognition, natural language generation, computer vision, traffic predictions, and anomaly-based may now be achieved utilizing AI technology. In this study, researchers looked at designs for human-created edge intelligence. However, there were several flaws in this study. This study breaks down architecture into many different types. However, since it requires specialized hardware, architectural search is out of reach for most researchers. Human-made architecture is the subject of the majority of the literature now available. This paper subdivides the human-created architectures into further classes: 1) Data-Placement-Based Architectures to minimize Latency; 2) Orchestration-Based ECAs-IoT. 3) Big-Data-Analysis-Based Architectures; and 4) Security-Based Architectures. In most cases, existing security measures are based on the same algorithms and models for penetration detection, privacy retention or access control. Traditional defenses are often rendered obsolete by the constant improvement of assault tactics and approaches. However, the growth of artificial intelligence (AI) offers new answers to privacy and security challenges, including penetration detection, privacy retention, and access controls.

Data Availability

No data were used to support this study.

Conflicts of Interest

The author(s) declare(s) that they have no conflicts of interest

References

[1]. F. Foukalas and A. Tziouvaras, "Edge Artificial Intelligence for Industrial Internet of Things Applications: An Industrial Edge Intelligence Solution", IEEE Industrial Electronics Magazine, vol. 15, no. 2, pp. 28-36, 2021. Doi : 10.1109/mie.2020.3026837.

[2]. H. Tang, H. Liu, W. Xiao and N. Sebe, "When Dictionary Learning Meets Deep Learning: Deep Dictionary Learning and Coding Network for Image Recognition With Limited Data", IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 5, pp. 2129-2141, 2021. Doi : 10.1109/tnnls.2020.2997289.

[3]. P. Georgiev, S. Bhattacharya, N. Lane and C. Mascolo, "Low-resource Multi-task Audio Sensing for Mobile and Embedded Devices via Shared Deep Neural Network Representations", Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 1, no. 3, pp. 1-19, 2017. Doi : 10.1145/3131895.

[4]. P. Zhang, E. Lo and B. Lu, "High Performance Depthwise and Pointwise Convolutions on Mobile Devices", Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 04, pp. 6795-6802, 2020. Doi : 10.1609/aaai.v34i04.6159.

[5]. C. García-Pérez and P. Merino, "Experimental evaluation of fog computing techniques to reduce latency in LTE networks", Transactions on Emerging Telecommunications Technologies, vol. 29, no. 4, p. e3201, 2017. Doi : 10.1002/ett.3201.

[6]. G. Baranwal and D. Vidyarthi, "FONS: a fog orchestrator node selection model to improve application placement in fog computing", The Journal of Supercomputing, vol. 77, no. 9, pp. 10562-10589, 2021. Doi : 10.1007/s11227-021-03702-x.

[7]. T. Ibaraki, "Integer programming formulation of combinatorial optimization problems", Discrete Mathematics, vol. 16, no. 1, pp. 39-52, 1976. Doi : 10.1016/0012-365x(76)90091-1.

[8]. S. Pirbhulal, W. Wu, K. Muhammad, I. Mehmood, G. Li and V. de Albuquerque, "Mobility Enabled Security for Optimizing IoT based Intelligent Applications", IEEE Network, vol. 34, no. 2, pp. 72-77, 2020. Doi : 10.1109/mnet.001.1800547.

[9]. J. Lee and J. Lee, "Hierarchical Mobile Edge Computing Architecture Based on Context Awareness", Applied Sciences, vol. 8, no. 7, p. 1160, 2018. Doi : 10.3390/app8071160.

[10]. S. Das, A. Santra, J. Bodra and S. Chakravarthy, "Query processing on large graphs: Approaches to scalability and response time trade offs", Data &amp; Knowledge Engineering, vol. 126, p. 101736, 2020. Doi : 10.1016/j.datak.2019.101736.

[11]. B. Kopmaz and A. Arslanoğlu, "Mobile health and smart health applications", Health Care Academician Journal, vol. 5, no. 4, p. 251, 2018. Doi : 10.5455/sad.13-1543239549.

[12]. T. Das and M. Gurusamy, "Controller Placement for Resilient Network State Synchronization in Multi-Controller SDN", IEEE Communications Letters, vol. 24, no. 6, pp. 1299-1303, 2020. Doi : 10.1109/lcomm.2020.2979072.

[13]. N. Radam, S. Al-Janabi and K. Shaker, "Optimisation Methods for the Controller Placement Problem in SDN: A Survey", Webology, vol. 19, no. 1, pp. 3130-3149, 2022. Doi : 10.14704/web/v19i1/web19207.

[14]. S. Park, M. Hong, T. Shon and J. Kwak, "VANET Privacy Assurance Architecture Design", Journal of Internet Computing and Services, vol. 17, no. 6, pp. 81-91, 2016. Doi : 10.7472/jksii.2016.17.6.81.

[15]. E. Sham and D. Vidyarthi, "CoFA for QoS based secure communication using adaptive chaos dynamical system in fog-integrated cloud", Digital Signal Processing, vol. 126, p. 103523, 2022. Doi : 10.1016/j.dsp.2022.103523.

[16]. J. Lozano-Rizk, J. Nieto-Hipolito, R. Rivera-Rodriguez, M. Cosio-Leon, M. Vazquez-Briseño and J. Chimal-Eguia, "QoSComm: A Data Flow Allocation Strategy among SDN-Based Data Centers for IoT Big Data Analytics", Applied Sciences, vol. 10, no. 21, p. 7586, 2020. Doi : 10.3390/app10217586.

[17]. J. Birkholzer, A. Cihan and K. Bandilla, "A tiered area-of-review framework for geologic carbon sequestration", Greenhouse Gases: Science and Technology, vol. 4, no. 1, pp. 20-35, 2013. Doi : 10.1002/ghg.1393.

[18]. M. Pang, L. Wang and N. Fang, "A collaborative scheduling strategy for IoV computing resources considering location privacy protection in mobile edge computing environment", Journal of Cloud Computing, vol. 9, no. 1, 2020. Doi : 10.1186/s13677-020-00201-x.

[19]. R. Hu, X. Dong and D. Wang, "Protecting Data Source Location Privacy in Wireless Sensor Networks against a Global Eavesdropper", International Journal of Distributed Sensor Networks, vol. 10, no. 8, p. 492802, 2014. Doi : 10.1155/2014/492802.

[20]. A. Singh, "Maintaining Analytic Utility while Protecting Confidentiality of Survey and Nonsurvey Data", Journal of Privacy and Confidentiality, vol. 1, no. 2, 2010. Doi : 10.29012/jpc.v1i2.571.

[21]. R. Tiburski et al., "Lightweight Security Architecture Based on Embedded Virtualization and Trust Mechanisms for IoT Edge Devices", IEEE Communications Magazine, vol. 57, no. 2, pp. 67-73, 2019. Doi : 10.1109/mcom.2018.1701047.

[22]. R. Xu, L. Hang, W. Jin and D. Kim, "Distributed Secure Edge Computing Architecture Based on Blockchain for Real-Time Data Integrity in IoT Environments", Actuators, vol. 10, no. 8, p. 197, 2021. Doi : 10.3390/act10080197.

[23]. Dina Merlinda Izzah, "Kompetensi Keamanan Jaringan Sesuai Kebutuhan Industri ISP (Internet Service Provider)", Jurnal Teknologi dan Bisnis, vol. 3, no. 1, pp. 33-42, 2021. Doi : 10.37087/jtb.v3i1.43.

[24]. S. Hamdan, M. Ayyash and S. Almajali, "Edge-Computing Architectures for Internet of Things Applications: A Survey", Sensors, vol. 20, no. 22, p. 6441, 2020. Doi : 10.3390/s20226441.

[25]. F. Ullah, I. Ullah, A. Khan, M. Uddin, H. Alyami and W. Alosaimi, "Enabling Clustering for Privacy-Aware Data Dissemination Based on Medical Healthcare-IoTs (MH-IoTs) for Wireless Body Area Network", Journal of Healthcare Engineering, vol. 2020, pp. 1-10, 2020. Doi : 10.1155/2020/8824907.

[26]. Y. Xu, "Resource Management of Maritime Edge Nodes for Collected Data Feedback", IEEE Access, vol. 8, pp. 131511-131521, 2020. Doi : 10.1109/access.2020.3007669.