CS 236: Project Report
Edge Computing

Anagha Subramanian
University of California Irvine
Irvine, CA, USA
Email: anaghas1@uci.edu

Nishitha Shankar Suvarna
University of California Irvine
Irvine, CA, USA
Email: nssuvarn@uci.edu

***Abstract*—Centralized processing of data has greatly improved the ability of devices to access and share computational resources. However, with the rapid growth in smart applications, more data is generated while processing power has remained the same, leading to higher operational and administrative costs incurred at these data processing units. This has led to the need for fewer large data centers and more computation running closer to end devices, which can be achieved using Edge Computing. In this paper, we first give an overview of Edge Computing and its possible implementations, followed by its applications. In particular, we discuss about Edge Intelligence, as well as Edge Computing in services such as the Automotive Sector and the Healthcare industry.**

1. Introduction

As a large amount of data is generated by mobile and IoT devices, the network bandwidth of cloud computing has been unable to meet the needs of time-sensitive systems. According to a prediction by Ericsson, 45% of the 40ZB global internet data will be generated by IoT devices in 2024 [1]. Transferring this huge amount of data from the edge to cloud can lead to network congestion. Instead we can handle the processing of user data at the edge of the network.

Cloud computing involves transferring and processing large amounts of data in the cloud computing servers in a centralized way. IoT devices primarily use cloud computing for processing data. Edge computing was therefore developed as a means to supplement the existing cloud computing infrastructure. Edge computing is a new computing paradigm in which substantial compute and storage resources are placed at the edge of the Internet, in close proximity to mobile devices, sensors, end users, and IoT devices. [2]

We can leverage the power of edge computing to deploy computing and storage resources closer to the edge of the network and the data can then be summarized in the cloud. It can also be used for analysis of real time data.

2. Edge Computing

Zha et al.[3] has defined edge computing as follows: "Edge computing is a new computing model that unifies resources that are close to the user in geographical distance or network distance to provide computing, storage, and network for applications service."

According to G.Klas [4], some of the reasons for the popularity of edge computing are as follows:

- Edge computing ensures faster response time between the device and application server. At the application level, there is a need for faster response, which cannot be achieved if the data center is distant. Examples of applications that may require low latency request/response loops include autonomous driving, robots in precision farming, smart grid control, and cloud gaming.
- It has more predictable quality of service, which consists of higher reliability and lower delay variation.
- It results in improved battery life, device weight and form factor as the heavy computations are now done at the edge of the network.
- It leads to more computation power and data storage for constrained devices.
- It also saves bandwidth.
- It enhances privacy and security, as data is processed within the scope of the edge node, thus eliminating the need to upload raw sensitive data to the cloud. Even if the attacker gains access to the data, only the local data becomes vulnerable.

2.1. Edge Computing Architecture

Edge computing has a federated network structure - the edge devices lie between the terminal devices and the cloud computing layer. According to K.Cao, et al.[5] the edge computing architecture comprises of three layers, as shown in Figure 1.[6]

- **Terminal layer**
  The devices in this layer are connected to the edge network, and upload raw data to the edge layer. It includes devices like smart cars and cameras.
- **Edge layer**
  It is also known as the boundary layer. It consists of several edge nodes that are present between the

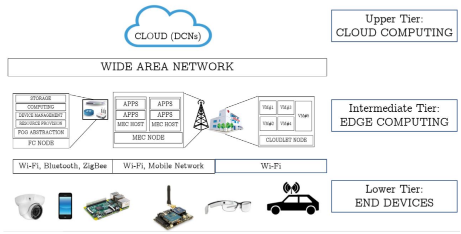

Figure 1: Edge Computing Architecture

terminal device and the cloud computing layer. The devices in this layer include base stations, routers, gateways, access points. The terminal devices upload the data to the edge layer, which then performs computations on this data and then uploads it to the cloud layer.
According to K.Dolui, et al. [6] The implementation of the Edge Layer can further be subdivided as follows:

- **Mobile Edge Computing layer (MEC)**
  It offers cloud computing capabilities inside the Radio Area Network (RAN) by deploying the nodes or server in the base station of the cellular network. Context awareness is improved and latency reduces. The MEC server runs instances of the MEC host which performs computation and stores data on a virtual interface. The MEC Orchestrator handles the MEC hosts and manages the Mobile Edge applications. It also manages the available resources, the network topology and the information on the services offered by the hosts. It offers information on the load and capacity of the network, the location and network information of the end devices connected to the server. The MEC Orchestrator maintains information on the applications running on the host. If an application sends a request and it is running then it is redirected. If it is not running and is supported by the platform then its instantiated. Otherwise, it is passed on to the the cloud.
- **Fog Computing Layer (FC)**
  The Fog Computing layer extends cloud computing properties to the edge. It carries out parallel processing at the core network edge. It consists of Fog Computing Nodes (FCN's) which are placed between the end devices and the cloud in the form of switches, routers, IoT gateways. Due to its heterogeneous nature, it can support devices at different protocol layers and non-IP based access technologies. The Fog abstraction layer hides the heterogeneity of the nodes and performs functions like monitoring and allocation of resources, device management, security, computation and storage. The Service Orchestration layer uses these services to allocate the resources to the end users based on their requirements, which are known as policies. The policies may have parameters like QoS and load balancing. The Fog Orchestrator returns a list of most suitable nodes by matching the policy with the services.
- **Cloudlet Computing (CC)** A Cloudlet is a trusted cluster of computers which are well connected to the Internet, and have resources available for use for nearby mobile devices. It is based on devices that are similar to a data center but is on a lower scale and deployed closer to the user. Cloudlets comprise of three layers: the component layer, the node layer and the Cloudlet layer.[16] The component layer provides interfaces to the higher layers which are managed by the Execution Environment. A node comprises of one or more Execution Environments that run on top of an OS. The nodes are managed by a Node Agent. The Cloudlet layer is a collection of co-located nodes that are managed by the Cloudlet Agent. The Cloudlet Agent communicates with the underlying elements through the Node Agent and the Execution Environments. The components pass the policy violations hierarchically to the Cloudlet Agent. The Cloudlet Agent makes optimal decisions for complex query allocation such that it's allocated to nodes that have higher processing power.

- **Cloud Computing layer**
  The cloud computing layer performs the final processing of the data, and therefore consists of highly efficient, powerful servers and massive storage. As a result, it can perform in depth analysis on the huge amount of data stored in the cloud. In cases where the edge layer is unable to complete the analysis, the cloud computing layer takes over. It also stores the processed data, that it receives from the edge computing layer. Based on the control policy, it adjusts the deployment plan and algorithm of the edge layer.

3. Edge Intelligence

As seen earlier, edge computing can be used to process data at the edge of the network to save bandwidth and to ensure faster computing. Artificial Intelligence (AI) is a growing field and has been incorporated into several applications to ensure optimal performance. With various edge computing applications generating large amounts of data, AI can be used to learn from this massive data and make optimal decisions. Improvements in hardware only further facilitate the deployment of deep learning models,

The research road-map of Edge Intelligence.

Figure 2: Roadmap for Edge Intelligence

which have a high computation overheads. To improve the efficiency and throughput of these models, application-specific accelerators have been designed. Given the potential of AI, a new paradigm know as Edge Intelligence has been developed that integrates edge computing and AI. Edge Intelligence is very vast - Sun et al. [17] has surveyed the application of machine learning techniques in wireless communication, while Mao et al. [18] has studied the applications of deep learning models on the different network layers. AI can supplement edge computing with better algorithms, while edge computing provides AI with data.
According S.Deng et al.[7], Edge Intelligence can be classified as AI for Edge and AI on Edge.

3.1. AI for Edge

Also known as Intelligence-enabled Edge Computing (IEC), this classification uses AI to incorporate intelligence into edge computing and to make it more optimal. It involves solving constrained optimization problems in edge computing. Edge computing faces problems such as resource allocation in the different layers, so it can use AI for coming up with optimal tools thereby supplementing the system efficiency.

3.2. AI on Edge

This classification studies running AI models on the edge. It consists of a framework for running training and inference of AI models in order to extract insight from the distributed edge data. More data is being generated by mobile and IoT devices, and applications such as autonomous driving, smart homes, intelligent networked vehicles rely on AI algorithms to make optimal decisions. Applications that require low computational power and have better communication quality can be moved from the cloud to the edge. With the surge in AI, many mobile devices are integrated with accelerators like Graphics Processing Units (GPUs), Tensor Processing Units (TPUs), and Neural Processing Units (NPUs).

3.3. Roadmap for Edge Intelligence

Following is the roadmap for Edge intelligence which separates the two categories, AI for edge and AI on edge. As shown in Figure 2 [7], research in edge computing is further divided into Topology, Content and Service.

3.3.1. Quality of Experience.

It's the topmost layer in the roadmap. QoS is expected to be application dependent. It is comprised of the following criteria: Performance, Cost, Privacy (Security), Efficiency and Reliability.

- **Performance**
  Performance measures are different for both AI for edge and AI on edge. In case of AI for edge, it is problem-dependent. In computation offloading problems, it could be the ratio of successful offloading. In case of AI on edge, it consists of accuracy and training loss.
- **Cost**
  It comprises of computation and communication cost and energy consumption. Computation cost involves the demand for the resources. Communication cost involves the request for the communication resources such as power and access time. Reducing energy consumption plays an important role, as edge devices have limited battery capacity.
- **Privacy (Security)**
  It is necessary to maintain privacy of the data generated. Techniques such as Federated Learning aggregate the local machine learning models from the distributed devices, thereby preserving privacy.
- **Efficiency**
  It plays an important role in evaluating and improving existing algorithms with the need for systems that need good performance and low overhead.
- **Reliability**
  It ensures that the system will not fail during its operating period. In particular, for AI on edge, the system needs to be reliable as the model training and inference is performed in a distributed and synchronized manner, meaning there is a higher probability of failure.

3.3.2. Roadmap for AI for Edge.

It follows a bottom up approach and consists of the following layers:

- **Topology**
  It consists of Orchestration of Edge Sites (OES) and Wireless Networking (WN). OES involves the deployment and installation of wireless telecom equipment and servers. WN involves Data Acquisition and Network Planning. Data Acquisition focuses on the fast acquisition of distributed data. Network planning involves efficient management with protocols and middleware.
- **Content**
  An emphasis is placed on Data Provisioning, Service

Provisioning, Service Placement, Service Composition and Service Caching. Resources are provided by remote cloud data-centers and edge servers for data and service provisioning. Service placement complements service provisioning, which involves deployment of complex services on edge sites. Service composition involves selecting candidate services for composition based on energy consumption and QoE of mobile end users. Service caching also complements service provisioning. It involves designing a caching pool to store frequently visited data and services.
- **Service**
  It involves Computation Offloading, User Profile Migration, and Mobility Management. Computation Offloading includes load balancing of computational and communication resources based on edge server selection and frequency spectrum allocation. User Profile Migration involves configuration of user profile when users are moving about. It is associated with Mobility Management.

3.3.3. Roadmap for AI on Edge. It follows a top down approach it is further classified as Model Adaptation, Framework Design and Processor Acceleration.

- **Framework Design**
  It involves coming up with better training and inference architectures.
- **Model Training**
  The frameworks for model training are distributed and can be classified into model and data splitting. Model splitting separates the layers of the neural network model and deploys it on different devices. Data splitting involves the following sub division: master-device, helper-device and device-device splitting. Federated Learning is a type of model training framework that trains DNN in a distributed manner.
- **Model Inference**
  A popular approach for Model inference is model splitting, which involves determining where to split the layers constrained on the accuracy.
- **Model Adaptation**
  It improves the model to make it more suitable for the edge. This is done based on the existing training and inference frameworks. Some of the methods that are used are Model Compression, Conditional Computation, Algorithm Asynchronization and Thorough Decentralization. Model Compression uses methods like Singular Value Decomposition (SVD), and Principal Component Analysis (PCA). Conditional Computation is used to reduce the number of calculations performed in a DNN by turning off unimportant ones. Some of the methods are early exit, input filtering. Algorithm Asynchronization involves aggregating the local models in an asynchronous manner. Thorough Decentralization eliminates the need of a central aggregator to avoid leakage.
- **Processor Acceleration** It focuses on optimizing the DNN computation on hardware by designing special instruction sets for DNN training and inference, designing highly paralleled computing paradigms, and moving computation closer to memory.[8]

4. Applications for Edge Computing

Many public-use systems, known commonly as "vertical industry services"[11] commonly use cloud computing for improved results and accuracy. In this section, we discuss two particular applications within the vertical industry, namely automotive applications and e-healthcare, which benefit from cloud computing and, more recently, edge computing.

4.1. Automotive Applications

In recent years, automotive systems have seen a rise in computational requirements due to the growth of modern applications such as smart cars and autonomous vehicles. One solution to keep up with this growing need for processing power is to use better computational infrastructure in the vehicle. While this could be a feasible solution for newer vehicles, it is impractical and expensive to upgrade hardware for older ones, particularly when automotive systems are constantly and rapidly changing. As a result, cloud computing is now widely used for automotive computation.

4.1.1. Classes of Applications. Before we discuss how cloud and edge computing are used, it would be helpful to classify the different applications performed by automotive systems by their priority. The following three categories, adapted from Feng et al.[9], provide a concise description of each application type:

- **Crucial (or Critical) Applications.**
  These have the highest priority over resources in an automotive setting, and cannot afford high delay and uncertainty. This includes applications such as accident prevention, vehicle control, and system monitoring.
- **High-Priority Applications.**
  These also have high priority over resources, but are not considered as important as crucial applications. As a result, they can afford some small amounts of delay and uncertainty, although it is not preferred. Example applications in this category include navigation, information services, and optional safety applications.
- **Low-Priority Applications.**
  These applications have the lowest priority and do not need resources as urgently as the other two categories. Examples include multimedia applications and passenger entertainment.

4.1.2. Automotive and Cloud Computing. In automotive systems, cloud computing helps provide a centralized platform to process vehicular data gathered from connected smart systems. Not only do they make computing resources easy to access and upgrade [9], but they can also provide added functionality with the increased data they receive. Since multiple vehicles tend to share the same cloud computing resources, the aggregate data can be used to produce useful information such as real-time maps and traffic flow prediction [10][12].

While it does provide many computational benefits, a common issue seen in cloud computing is its high data latency, as edge nodes in automotive applications receive volumes of data from multiple vehicles within a short interval of time. Moreover, excess delay is unacceptable for Crucial Applications such as accident prevention, which can be life-threatening.

Another important issue involves unstable network connectivity in road environments. Feng et al. [9] discuss how constant network connection is not guaranteed for vehicles, particularly in rural areas with poor network infrastructure. Even in urban areas, vehicles are still much more mobile with respect to their fixed access points than regular devices connected to the Internet. As a result, vehicles can still occasionally lose connectivity. Cloud computing then becomes less accessible for periods of time, which once again can be problematic when time-sensitive applications require cloud resources during this period.

4.1.3. Automotive and Edge Computing. Edge computing was therefore introduced to mitigate some of these issues. It involves placing the processing unit within closer proximity to the vehicle than a cloud server. Part of the computation can therefore be done at this edge unit before being sent to the cloud server for completion, which greatly reduces data latency. Most of the computation is generally conducted at the edge unit and sent back to the vehicle, forwarding only pooled and partly processed data to the cloud. This is beneficial for both High and Low-Priority Applications. While High-Priority Applications can promptly receive their results from the edge unit, Low-Priority Applications will wait for cloud-generated results, which can also potentially run faster due to less data transferred to the cloud node.

There are also creative solutions that maintain some level of edge processing in unstable network connectivity scenarios. While traditional edge computing with remote edge nodes still struggle with connectivity issues, we can overcome this by redefining what sources can behave as edge nodes. Aissioui et al.[11], for example, discuss the Cellular Vehicle-to-Everything (C-V2X) communication specification for long range network communications. The specification defines three modes (V2X):

- **Vehicle-to-Vehicle (V2V):** Communication between nearby smart vehicles on the road
- **Vehicle-to-Infrastructure (V2I):** Communication between vehicles and network access point infrastructure
- **Vehicle-to-Network (V2N):** Communication between vehicles directly with edge or cloud computing resources/nodes

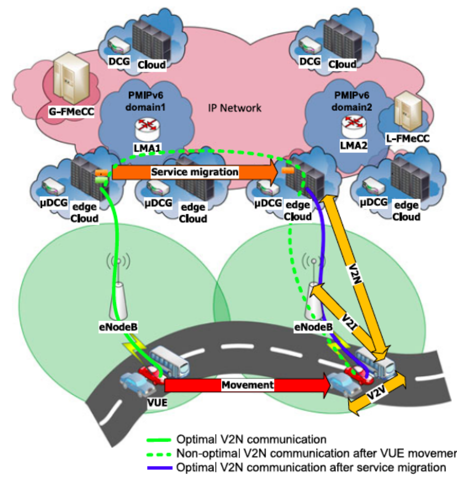

Figure 3: An example of V2X for Edge Communication.

These modes are also depicted in the Figure 3 with the yellow arrows [11].

Traditional edge computing runs via V2I or V2N communication. But in rural areas with poor infrastructure for example, vehicles can engage in V2V communication, for example, to share resources and data, and therefore maintain some additional processing resources until they gain access to an edge again. Moreover, vehicles travelling together on the road tend to move at similar speeds compared to a vehicle with its fixed access point [11]. As a result, data latency between these vehicles may be relatively lower. Ultimately, the motivation is that no vehicle should be without connectivity at any given time, and while we may not always have access to the edge or cloud resource, any small additional amount of computation is good enough.

4.1.4. Uses of Edge Computing. Many applications run on edge nodes in automotive systems. As discussed earlier, traffic flow analysis and route generation are common examples [10]. As depicted in Figure 4 [10], vehicles constantly provide information such as vehicle status, location, and traffic flow to the edge node. The edge node can then pool all of the information to generate the best routes for each vehicle. Similarly, in the event of an accident, the edge also obtains that data, and can inform vehicles and update routes accordingly. Algorithms that run on edge nodes commonly run artificial intelligence or neural networks to perform these functions.

There has also been considerable work in combining automotive systems with modern 5G networks, which is optimized using edge computing [11]. With improved infrastructure, there is anticipation that running automotive systems on 5G networks can further improve data latency.

4.2. Healthcare Applications

Healthcare has also seen immense benefit in using computation to provide improved diagnoses and give feedback

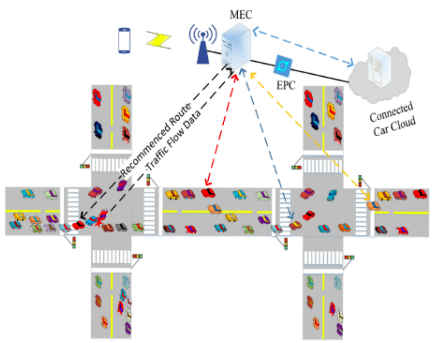

Figure 4: An example of Traffic Flow prediction.

on patient condition. Once again, using improved hardware can be expensive to implement, and so cloud computing is considered a more viable solution.

4.2.1. Healthcare and Cloud Computing. Before we analyze the use of cloud computing in healthcare, let us define the components of the body area network (BAN) where cloud computing is commonly used for processing and analysis. As defined by Chen et al. [13], the BAN consists of:

- **Collection layer.** This layer involves collection of data through sensors placed on the patient's body. These sensors can record data such as vital signs, heart rate, and body temperature.
- **Transmission layer.** This layer involves the infrastructure used to transmit data observed by the collection layer to the cloud or other computing resource located in the analysis layer.
- **Analysis layer.** This layer is where input data from the patient is stored and analyzed to provide feedback or information about the patient's condition.

Cloud computing lies in the analysis layer and provides a more accessible and scalable mode of sharing computational resources in a healthcare environment. They have the benefit of acting as a centralized control and can also concurrently monitor multiple patients. Obtaining data from multiple sources at once also allows support for additional applications that involve bulk data processing, such as disease analysis and patient statistics.

4.2.2. Healthcare and Edge Computing. Once again, time-sensitive healthcare applications struggle with the high data latency from cloud computing, especially since healthcare requires volumes of data transfer over a short period of time. Such high data latency is also unacceptable in scenarios where a patient requires emergency medical assistance. In an attempt to improve these issues, healthcare applications increasingly use edge computing to reduce data latency by having computation nodes in the analysis layer in closer proximity to the collection layer. Only a subset of processed, low-importance data is sent back to the cloud.

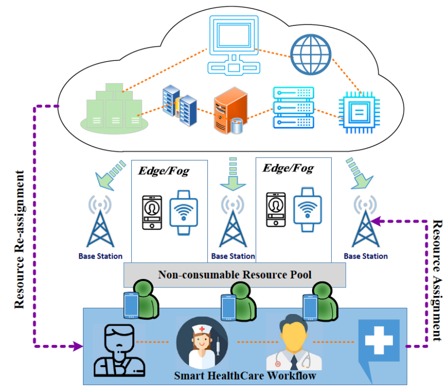

Figure 5: An example of Edge Computing in Healthcare.

4.2.3. Uses of Edge Computing. Perhaps the most common applications that run on edge nodes in a healthcare environment are machine learning and natural language processing (NLP) models that analyze patient symptoms detected from the collection layer and classify disease. Although these models are not advanced enough to be the primary diagnosis source for a patient, they are frequently used to display possible illnesses and provide patient information to medical experts, who use this information for better reasoning [13]. There are also a number of IoT-based applications that aid in telehealth and e-medicine, where edge nodes once again centrally store and process patient records in a way that is accessible by multiple doctors [14][15]. Figure 5 [15] shows a telehealth scenario that uses edge computing.

Another interesting edge computing application that is currently being worked on is edge resource allocation for healthcare. Chen et al.[13], for example, describes an algorithm that reallocates resources based on a patient's requirements. Consider a scenario where many patients are distributed over 2 edge computing nodes. If one of the patients were to experience a sudden life-threatening illness such as a heart attack, then the edge nodes can be reallocated in a way that the patient in need has access to a full resource while the other patients share the remaining resource. Such a reallocation strategy can be implemented at the edge, and can use cognitive computing and deep learning algorithms to optimize reallocation strategies [13].

5. Conclusion

In this paper, we explored the concept of edge computing and its benefits over traditional cloud computing while maintaining the same computational benefits to applications. In particular, we discussed the different implementations and went through various applications that commonly use and actively work on edge computing today. Edge computing has several benefits such as lower latency and better data localization. However, it still has some disadvantages: as edge computing integrates multiple trust domains, traditional data encryption algorithms cannot work, which slightly increases the vulnerability of the system. In order to solve this issue, it needs its own encryption implementation which must be lightweight and not add any processing overhead. User location data may also potentially be exposed, which can be used in malicious attacks. The reliability of data transfer also depends on the connection, the protocol, and also on the proximity of end terminal to the edge layer. Nonetheless, edge computing still outperforms cloud computing, and any drawbacks that the former has happen at a much smaller scale.

Acknowledgments

We would like to specifically thank our CS 236 professor, Marco Levorato, for his useful feedback on our summary presentation, as well as his overall guidance in exploring the Edge Computing topic.

References

[1] Ericsson, *Iot connections outlook: Nb-iot and cat-m technologies will account for close to 45 percent of cellular iot connections in 2024*, 2019. [Online]. Available: https://www.ericsson.com/en/mobility-report/ reports/june- 2019/iot-connections-outlook

[2] Satyanarayanan, Mahadev, *Edge Computing* Computer 50.10 (2017): 36–38. Web.

[3] Z.M.Zha,F.Liu,andZ.P.Cai, *Edgecomputing: Platforms;Applications and challenges* J. Comput. Res. Develop., vol. 55, no. 2, pp. 327–337, 2018.

[4] G. Klas, *Edge Computing and the Role of Cellular Networks* in Computer, vol. 50, no. 10, pp. 40-49, 2017, doi: 10.1109/MC.2017.3641649.

[5] K. Cao, Y. Liu, G. Meng and Q. Sun, *An Overview on Edge Computing Research*, in IEEE Access, vol. 8, pp. 85714-85728, 2020, doi: 10.1109/ACCESS.2020.2991734.

[6] K. Dolui and S. K. Datta, *Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing*, 2017 Global Internet of Things Summit (GIoTS), 2017, pp. 1-6, doi: 10.1109/GIOTS.2017.8016213.

[7] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar and A. Y. Zomaya, *Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence* , in IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7457-7469 Aug. 2020, doi: 10.1109/JIOT.2020.2984887.

[8] V. Sze, Y. Chen, T. Yang and J. S. Emer, *Efficient processing of deep neural networks: A tutorial and survey*, Proceedings of the IEEE, vol. 105, no. 12, pp. 2295–2329, Dec 2017.

[9] J. Feng, Z. Liu, C. Wu and Y. Ji, *AVE: Autonomous Vehicular Edge Computing Framework with ACO-Based Scheduling*, in IEEE Transactions on Vehicular Technology, vol. 66, no. 12, pp. 10660-10675, Dec. 2017, doi: 10.1109/TVT.2017.2714704.

[10] X. Song, Y. Guo, N. Li and L. Zhang, *Online Traffic Flow Prediction for Edge Computing-Enhanced Autonomous and Connected Vehicles*, in IEEE Transactions on Vehicular Technology, vol. 70, no. 3, pp. 2101-2111, March 2021, doi: 10.1109/TVT.2021.3057109.

[11] A. Aissioui, A. Ksentini, A. M. Gueroui and T. Taleb, *Enabling 5G Automotive Systems Using Follow Me Edge-Cloud Concept*, in IEEE Transactions on Vehicular Technology, vol. 67, no. 6, pp. 5302-5316, June 2018, doi: 10.1109/TVT.2018.2805369.

[12] Dennis Grewe, Marco Wagner, Mayutan Arumaithurai, Ioannis Psaras, and Dirk Kutscher. *Information-Centric Mobile Edge Computing for Connected Vehicle Environments: Challenges and Research Directions.*, in Proceedings of the Workshop on Mobile Edge Communications (MECOMM '17). Association for Computing Machinery, New York, NY, USA, 7–12. 2017. DOI:https://doi.org/10.1145/3098208.3098210

[13] Min Chen, Wei Li, Yixue Hao, Yongfeng Qian, Iztok Humar, *Edge cognitive computing based smart healthcare system*, Future Generation Computer Systems, Volume 86, 2018, Pages 403-411, ISSN 0167-739X, https://doi.org/10.1016/j.future.2018.03.054. (https://www.sciencedirect.com/science/article/pii/S0167739X17326766)

[14] Partha Pratim Ray, Dinesh Dash, Debashis De, *Edge computing for Internet of Things: A survey, e-healthcare case study and future direction*, Journal of Network and Computer Applications, Volume 140, 2019, Pages 1-22, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2019.05.005. (https://www.sciencedirect.com/science/article/pii/S1084804519301651)

[15] S. Oueida, Y. Kotb, M. Aloqaily, Y. Jararweh, and T. Baker, *An Edge Computing Based Smart Healthcare Framework for Resource Management*, Sensors, vol. 18, no. 12, p. 4307, Dec. 2018.

[16] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, *Cloudlets: Bringing the cloud to the mobile user*, in Proceedings of the third ACM workshop on Mobile cloud computing and services. ACM, 2012, pp. 29–36.

[17] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, *Application of machine learning in wireless networks: Key techniques and open issues*, CoRR, vol. abs/1809.08707, 2018. [Online]. Available: http://arxiv.org/abs/1809.08707

[18] Q. Mao, F. Hu, and Q. Hao, , *Deep learning for intelligent wireless networks: A comprehensive survey* , IEEE Communications Surveys Tutorials, vol. 20, no. 4, pp. 2595–2621, Fourthquarter 2018.